

# MININET 2: TRABAJO DE ADMINISTRACIÓN REDES

**Autor: Alejandro Rodríguez Rojas**

## Indice

1 Introducción.....	3
2 Conexión a Mininet.....	3
3 Conexión al entorno.....	4
4 Conexión mediante Xterm.....	5
5 Cambio de IP de los Host y del Router.....	6
6 Modificación del Enrutamiento.....	8
7 Ping hacia los Hosts y Router.....	9
8 Esquema del problema.....	11
9 Captura Tráfico.....	12
10 Conclusión.....	12

## 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).

```
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Mon Nov  5 14:01:54 PST 2018 from 172.22.3.18 on pts/1
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

mininet@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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d5:85:32 brd ff:ff:ff:ff:ff:ff
    inet 172.22.1.85/16 brd 172.22.255.255 scope global eth0
        valid_lft forever preferred_lft forever
mininet@mininet-vm:~$
```

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.

```
mininet@mininet-vm: ~  
Archivo  Editar  Ver  Buscar  Terminar  Ayuda  
usuario@debian:~$ ssh -X mininet@172.22.1.85  
mininet@172.22.1.85's password:  
Warning: No xauth data; using fake authentication data for X11 forwarding.  
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
New release '16.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Nov  6 09:17:04 2018 from 172.22.3.18  
mininet@mininet-vm:~$
```

### 3 Conexión al entorno

Ya estamos conectados mediante SSH a la máquina Virtual.

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

```
sudo python {nombre}
```

```
mininet@mininet-vm:~$ sudo python view.php?id=9935  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2 h3 r0  
*** Adding switches:  
s1 s2 s3  
*** Adding links:  
(h1, s1) (h2, s2) (h3, s3) (s1, r0) (s2, r0) (s3, r0)  
*** Configuring hosts  
h1 h2 h3 r0  
*** Starting controller  
c0  
*** Starting 3 switches  
s1 s2 s3 ...  
*** Routing Table on Router:  
Kernel IP routing table  
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface  
  
*** Starting CLI:  
mininet>
```

## 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:

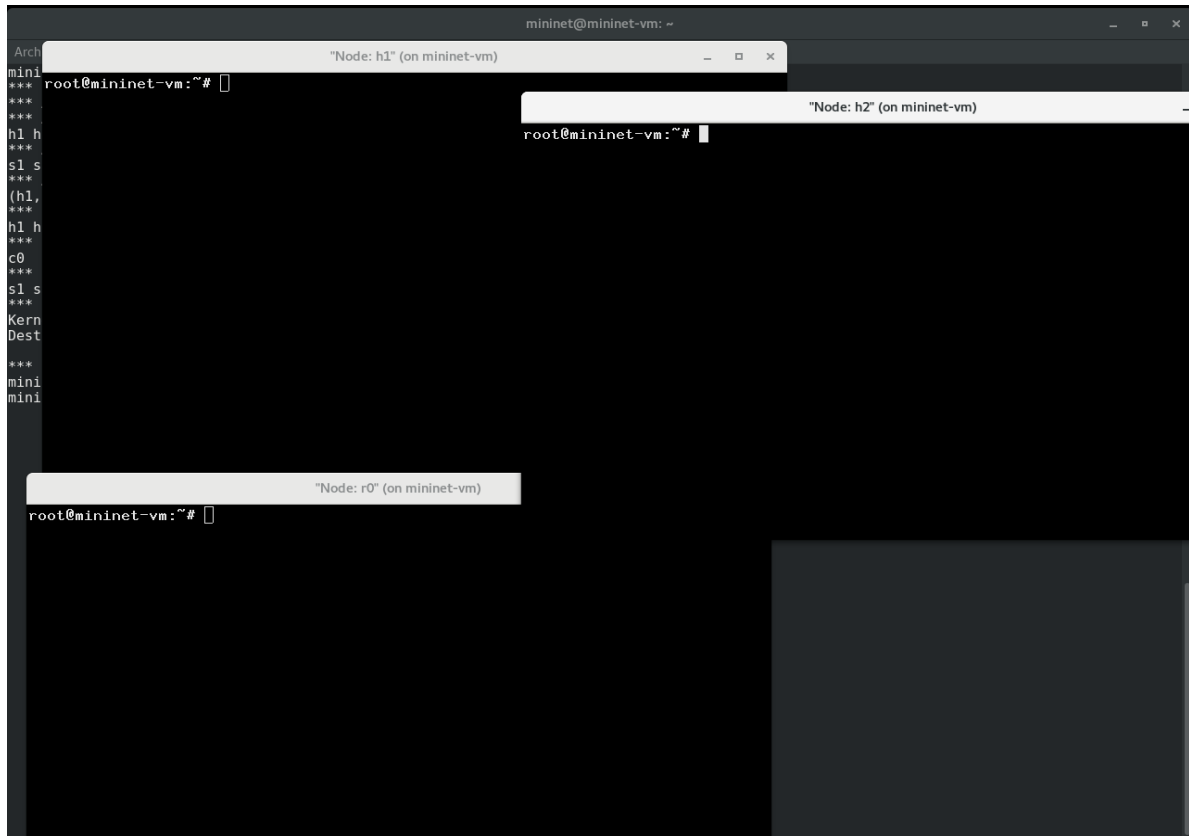
h1 → Host 1

h2 → Host 2

h3 → Host 3

r0 → Router

```
*** Starting CLI:
mininet> xterm h1 h2 h3 r0
mininet> █
```



## 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 192.168.0.1 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@if22: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 2c:0e:78:b3:7d:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/32 scope global h1-eth0
        valid_lft forever preferred_lft forever
root@mininet-vm:~#

```

H2 →

```
root@mininet-vm:~# ip addr add 172.32.0.1/12 dev h2-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: h2-eth0@if23: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 88:ca:2b:54:c8:28 brd ff:ff:ff:ff:ff:ff
    inet 172.32.0.1/12 scope global h2-eth0
        valid_lft forever preferred_lft forever
root@mininet-vm:~#
```

H3 →

```
root@mininet-vm:~# ip addr add 10.1.0.1/7 dev h3-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: h3-eth0@if24: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 9e:86:d2:2b:16:dd brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.1/7 scope global h3-eth0
        valid_lft forever preferred_lft forever
root@mininet-vm:~#
```

R0 →

```

root@mininet-vm:~# ip addr add 192.168.0.2/24 dev r0-eth1
root@mininet-vm:~# ip addr add 172.32.0.2/12 dev r0-eth2
root@mininet-vm:~# ip addr add 10.1.0.2/7 dev r0-eth3
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: r0-eth1@if25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 96:93:71:09:00:e9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.2/24 scope global r0-eth1
        valid_lft forever preferred_lft forever
3: r0-eth2@if26: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether f6:a7:d3:fa:98:cd brd ff:ff:ff:ff:ff:ff
    inet 172.32.0.2/12 scope global r0-eth2
        valid_lft forever preferred_lft forever
4: r0-eth3@if27: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 62:1d:1a:9d:16:65 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.2/7 scope global r0-eth3
        valid_lft forever preferred_lft forever
root@mininet-vm:~#

```

## 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 192.168.0.2
root@mininet-vm:~# route -e
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	192.168.0.2	0.0.0.0	UG	0	0	0	h1-eth0
192.168.0.0	*	255.255.255.0	U	0	0	0	h1-eth0

```
root@mininet-vm:~#
```

h2 →

```
root@mininet-vm:~# ip r add default via 172.32.0.2
root@mininet-vm:~# route -e
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	172.32.0.2	0.0.0.0	UG	0	0	0	h2-eth0
172.32.0.0	*	255.240.0.0	U	0	0	0	h2-eth0

```
root@mininet-vm:~#
```

h3 →

```
root@mininet-vm:~# ip r add default via 10.1.0.2
root@mininet-vm:~# route -e
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	10.1.0.2	0.0.0.0	UG	0	0	0	h3-eth0
10.0.0.0	*	254.0.0.0	U	0	0	0	h3-eth0

```
root@mininet-vm:~#
```

## 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 →

```
root@mininet-vm:~# ping 172.32.0.1
PING 172.32.0.1 (172.32.0.1) 56(84) bytes of data.
64 bytes from 172.32.0.1: icmp_seq=1 ttl=63 time=2.08 ms
64 bytes from 172.32.0.1: icmp_seq=2 ttl=63 time=1.11 ms
64 bytes from 172.32.0.1: icmp_seq=3 ttl=63 time=0.239 ms
64 bytes from 172.32.0.1: icmp_seq=4 ttl=63 time=0.043 ms
■
```

H2 a H1 →

```
root@mininet-vm:~# ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=63 time=3.71 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=63 time=0.383 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=63 time=0.088 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=63 time=0.052 ms
■
```

H1 a H3 →

```
root@mininet-vm:~# ping 10.1.0.1
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
64 bytes from 10.1.0.1: icmp_seq=1 ttl=63 time=4.10 ms
64 bytes from 10.1.0.1: icmp_seq=2 ttl=63 time=0.795 ms
64 bytes from 10.1.0.1: icmp_seq=3 ttl=63 time=0.361 ms
■
```

H2 a H3 →

```
root@mininet-vm:~# ping 10.1.0.1
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
64 bytes from 10.1.0.1: icmp_seq=1 ttl=63 time=2.97 ms
64 bytes from 10.1.0.1: icmp_seq=2 ttl=63 time=1.02 ms
64 bytes from 10.1.0.1: icmp_seq=3 ttl=63 time=0.308 ms
■
```

H1 a Router →

```
root@mininet-vm:~# ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=1.92 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.624 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.267 ms
■
```

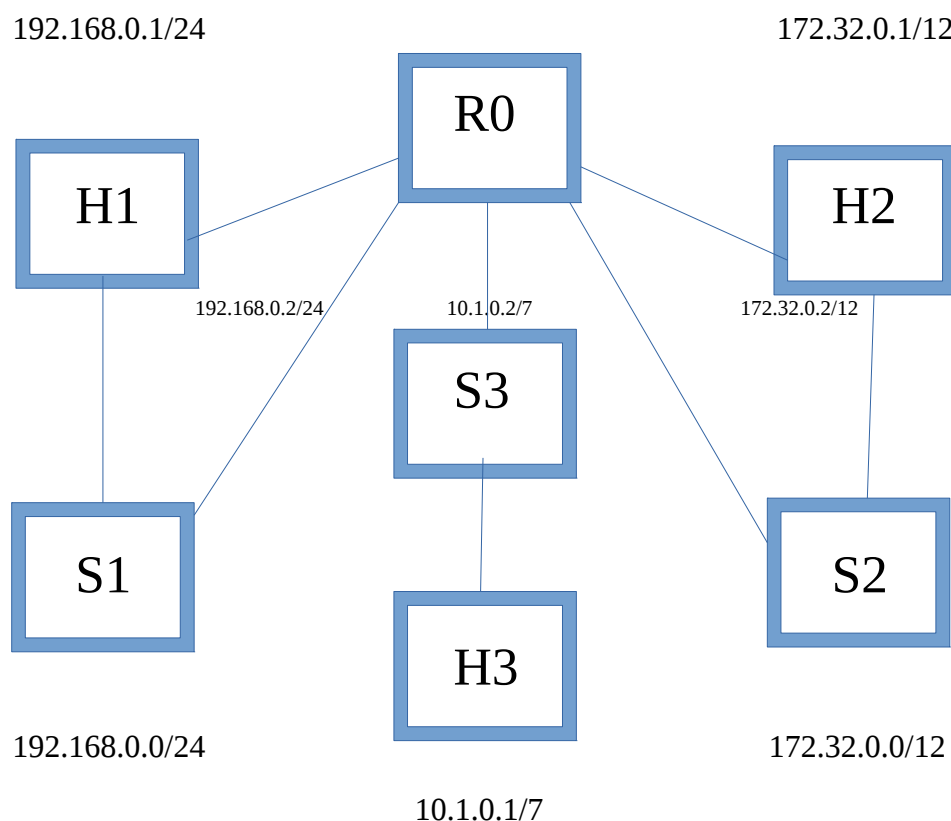
H2 a Router →

```
root@mininet-vm:~# ping 172.32.0.2
PING 172.32.0.2 (172.32.0.2) 56(84) bytes of data.
64 bytes from 172.32.0.2: icmp_seq=1 ttl=64 time=2.76 ms
64 bytes from 172.32.0.2: icmp_seq=2 ttl=64 time=0.737 ms
64 bytes from 172.32.0.2: icmp_seq=3 ttl=64 time=0.167 ms
█
```

H3 a Router →

```
root@mininet-vm:~# ping 10.1.0.2
PING 10.1.0.2 (10.1.0.2) 56(84) bytes of data.
64 bytes from 10.1.0.2: icmp_seq=1 ttl=64 time=1.84 ms
64 bytes from 10.1.0.2: icmp_seq=2 ttl=64 time=0.709 ms
64 bytes from 10.1.0.2: icmp_seq=3 ttl=64 time=0.163 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 -vi r0-eth3
tcpdump: listening on r0-eth3, link-type EN10MB (Ethernet), capture size 262144
bytes
12:07:34.872949 IP (tos 0x0, ttl 63, id 24226, offset 0, flags [DF], proto ICMP
(1), length 84)
    192.168.100.1 > 10.1.0.1: ICMP echo request, id 2870, seq 16, length 64
12:07:34.872975 IP (tos 0x0, ttl 64, id 28019, offset 0, flags [none], proto ICM
P (1), length 84)
    10.1.0.1 > 192.168.100.1: ICMP echo reply, id 2870, seq 16, length 64
12:07:35.873095 IP (tos 0x0, ttl 63, id 24412, offset 0, flags [DF], proto ICMP
(1), length 84)
    192.168.100.1 > 10.1.0.1: ICMP echo request, id 2870, seq 17, length 64
12:07:35.873113 IP (tos 0x0, ttl 64, id 28052, offset 0, flags [none], proto ICM
P (1), length 84)
    10.1.0.1 > 192.168.100.1: ICMP echo reply, id 2870, seq 17, length 64
12:07:36.872979 IP (tos 0x0, ttl 63, id 24558, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.100.1 > 10.1.0.1: ICMP echo request, id 2870, seq 18, length 64
12:07:36.873010 IP (tos 0x0, ttl 64, id 28247, offset 0, flags [none], proto ICMP (1), length 84)
    10.1.0.1 > 192.168.100.1: ICMP echo reply, id 2870, seq 18, length 64
```

## 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.