

MININET: 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.....	4
5 Cambio de IP de los Host y del Router.....	5
6 Modificación en Enrutamiento.....	7
7 Ping hacia los Hosts y Router.....	7
8 Esquema del problema.....	8
9 Captura Tráfico.....	8
10 Conclusión.....	9

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  Terminal  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}`

```
Last login: Tue Nov  6 08:46:54 2018 from 172.22.3.18
mininet@mininet-vm:~$ sudo python view.php?id=9915
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 r0
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s1, r0) (s2, r0)
*** Configuring hosts
h1 h2 r0
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** 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 r0`

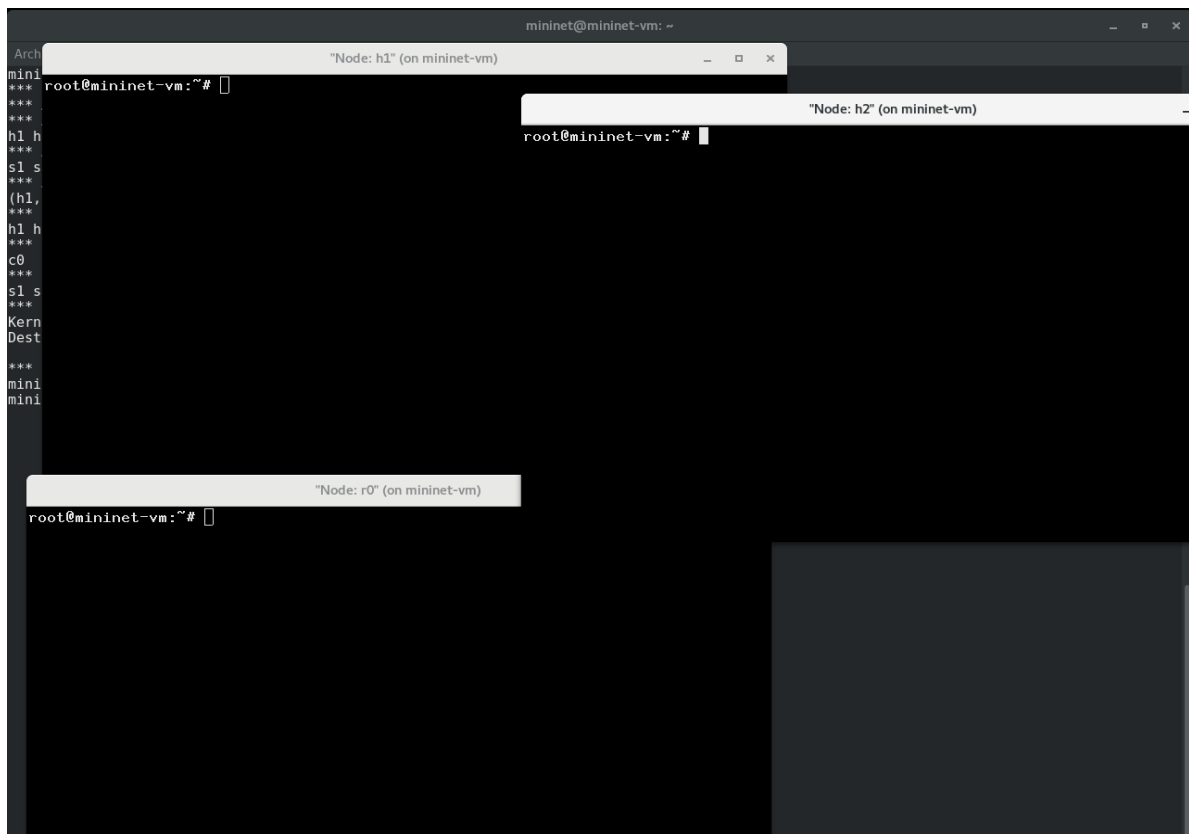
Siendo:

h1 → Host 1

h2 → Host 2

r0 → Router

```
*** Starting CLI:
mininet> xterm h1 h2 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 172.16.0.0/16 para host2)

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

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

H1 →

```

"Node: h1" (on mininet-vm)
root@mininet-vm:~# ip addr add 192.168.0.1/24 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@if28: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether ca:54:00:55:44:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 scope global h1-eth0
        valid_lft forever preferred_lft forever
root@mininet-vm:~#

```

H2→

```

"Node: h2" (on mininet-vm)
root@mininet-vm:~# ip addr add 172.16.0.1/16 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@if29: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 56:e3:ac:10:4e:95 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.1/16 scope global h2-eth0
        valid_lft forever preferred_lft forever
root@mininet-vm:~#

```

R0→

```

"Node: r0" (on mininet-vm)
root@mininet-vm:~# ip addr add 192.168.0.2/24 dev r0-eth1
root@mininet-vm:~# ip addr add 172.16.0.2/16 dev r0-eth2
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@if30: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 12:b2:8f:ce:ba:6f 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@if31: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 3a:9f:1a:30:26:1c brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.2/16 scope global r0-eth2
        valid_lft forever preferred_lft forever
root@mininet-vm:~#

```

6 Modificación en 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	hss	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 172.16.0.2
root@mininet-vm:~# route -e
Kernel IP routing table
```

destination	gateway	genmask	flags	hss	window	irtt	iface
default	172.16.0.2	0.0.0.0	UG	0	0	0	h2-eth0
172.16.0.0	*	255.255.0.0	U	0	0	0	h2-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 usando el comando:

```
ping {IP}
```

H1 a H2 →

```
root@mininet-vm:~# ping 172.16.0.1
PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.
64 bytes from 172.16.0.1: icmp_seq=1 ttl=63 time=1.61 ms
64 bytes from 172.16.0.1: icmp_seq=2 ttl=63 time=1.06 ms
64 bytes from 172.16.0.1: icmp_seq=3 ttl=63 time=0.071 ms
64 bytes from 172.16.0.1: icmp_seq=4 ttl=63 time=0.040 ms
64 bytes from 172.16.0.1: icmp_seq=5 ttl=63 time=0.046 ms
64 bytes from 172.16.0.1: icmp_seq=6 ttl=63 time=0.038 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=2.24 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=63 time=0.401 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=63 time=0.047 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=63 time=0.042 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=63 time=0.037 ms
```

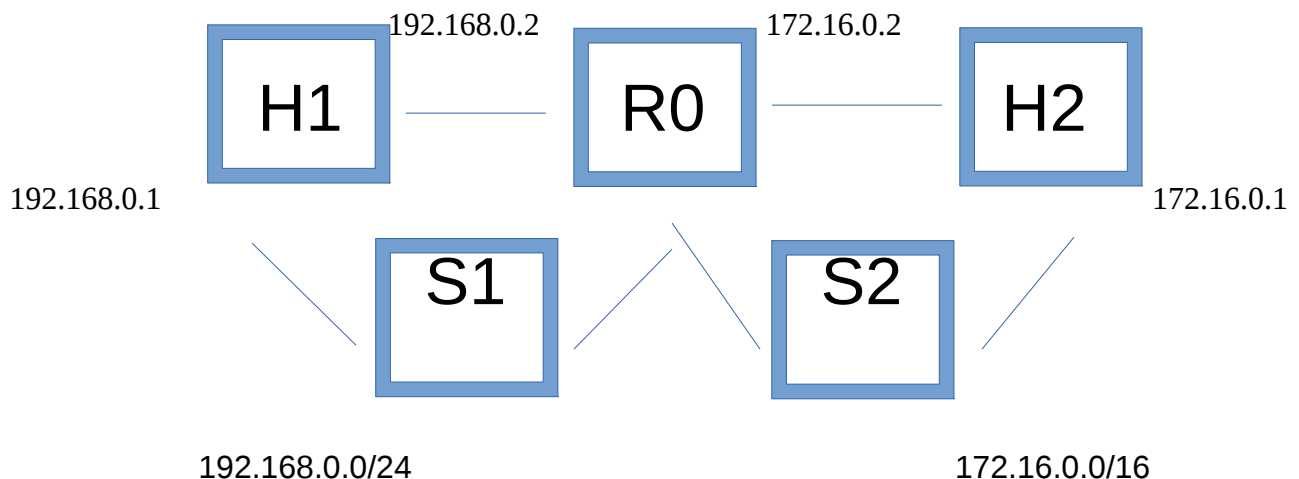
H1 a Router →

```
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=1.19 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.172 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.043 ms
^[[3~64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from 192.168.0.2: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 192.168.0.2: icmp_seq=7 ttl=64 time=0.523 ms
```

H2 a Router →

```
root@mininet-vm:~# ping 172.16.0.2
PING 172.16.0.2 (172.16.0.2) 56(84) bytes of data.
64 bytes from 172.16.0.2: icmp_seq=1 ttl=64 time=5.83 ms
64 bytes from 172.16.0.2: icmp_seq=2 ttl=64 time=1.17 ms
64 bytes from 172.16.0.2: icmp_seq=3 ttl=64 time=0.199 ms
64 bytes from 172.16.0.2: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 172.16.0.2: icmp_seq=5 ttl=64 time=0.037 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-eth1
tcpdump: listening on r0-eth1, link-type EN10MB (Ethernet), capture size 262144
bytes
12:01:10.460883 IP (tos 0x0, ttl 64, id 5121, offset 0, flags [DF], proto ICMP (
1), length 84)
    192.168.0.1 > 172.16.0.1: ICMP echo request, id 2371, seq 21, length 64
12:01:10.460906 IP (tos 0x0, ttl 63, id 11757, offset 0, flags [none], proto ICM
P (1), length 84)
    172.16.0.1 > 192.168.0.1: ICMP echo reply, id 2371, seq 21, length 64
12:01:10.462796 IP (tos 0x0, ttl 64, id 39970, offset 0, flags [DF], proto UDP (
17), length 69)
    192.168.0.2.34069 > 192.168.0.1.domain: 52762+ PTR? 1.0.16.172.in-addr.arpa.
(41)
12:01:10.463895 IP (tos 0xc0, ttl 64, id 3922, offset 0, flags [none], proto ICM
P (1), length 97)
    192.168.0.1 > 192.168.0.2: ICMP 192.168.0.1 udp port domain unreachable, len
gth 77
    IP (tos 0x0, ttl 64, id 39970, offset 0, flags [DF], proto UDP (17), len
gth 69)
    192.168.0.2.34069 > 192.168.0.1.domain: 52762+ PTR? 1.0.16.172.in-addr.arpa.
(41)
12:01:10.464105 IP (tos 0x0, ttl 64, id 39971, offset 0, flags [DF], proto UDP (
17), length 69)
    192.168.0.2.44724 > 192.168.0.1.domain: 52762+ PTR? 1.0.16.172.in-addr.arpa.
(41)
12:01:10.464875 IP (tos 0xc0, ttl 64, id 3923, offset 0, flags [none], proto ICM
```

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.