

Laboratorio 3

Especificaciones

Para el desarrollo de esta tarea se utilizó una maquina virtual con Linux 22.04, instalando Mininet 2.3.0.

1. Anillo interconectado

Creación de la red

Se debe crear una red con 4 hosts y 4 switches, estableciendo las conexiones entre hosts y switches correspondientes, lo cual fue realizado a través del siguiente Script de Python.

```
1 from mininet.topo import Topo
2
3 class MyTopo(Topo):
4     def __init__(self):
5         Topo.__init__(self)
6
7         switches = []
8         hosts = []
9
10        # Creacion de switches y hosts
11        for i in range(4):
12            switches.append(self.addSwitch(f's{i + 1}', dpid = f'00:00:00:00:00:00:{i + 1:02x}'))
13            hosts.append(self.addHost(f'h{i + 1}', mac = f'10:00:00:00:00:00:{i + 1:02x}'))
14
15        #Conexion entre el host y el switch
16        self.addLink(self.addLink(hosts[i], switches[i], 1, 1))
17
18        # Conexiones entre los switches
19        for i in range(4):
20            self.addLink(switches[i], switches[i - 1], 2, 3)
21        # Conexion adicional
22        self.addLink(switches[0], switches[2], 4, 5)
23
24        topos = {'MyTopo': (lambda: MyTopo())}
```

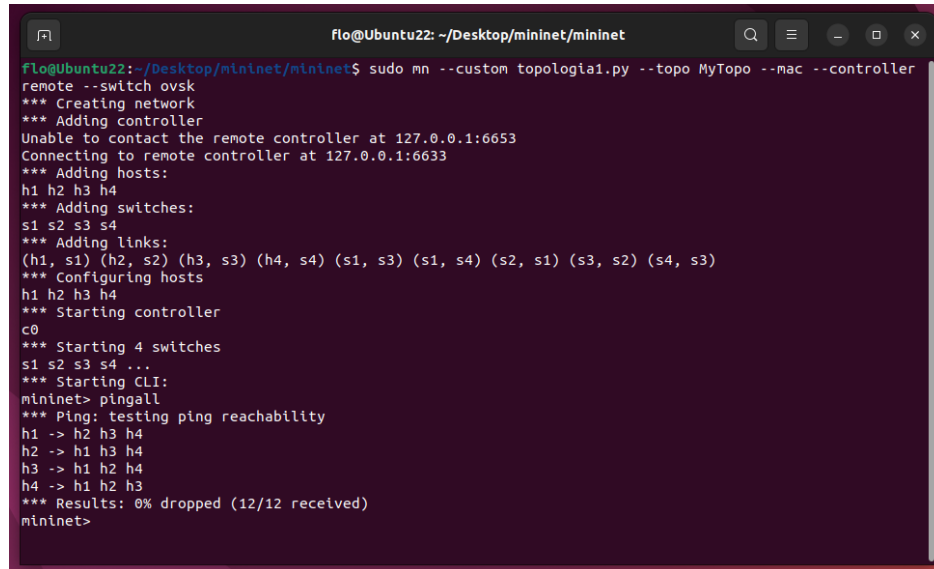
A medida que se van creando los switches, también se crea el host respectivo y se añade la conexión entre ellos usando el puerto 1. Luego, se establecen las conexiones entre los switches, se consideró la conexión extra entre los switches 1 y 3 como se mostraba en la imagen, y no entre el 2 y 4 como decía el enunciado.

Controlador

Se utilizó el controlador de ejemplo que fue proporcionado al momento de instalar POX, llamado `l2_learning.py` dentro de la carpeta *forwarding*.

Verificación de las conexiones

Para verificar que todos los hosts se pueden comunicar entre sí, se probó utilizando pingall.



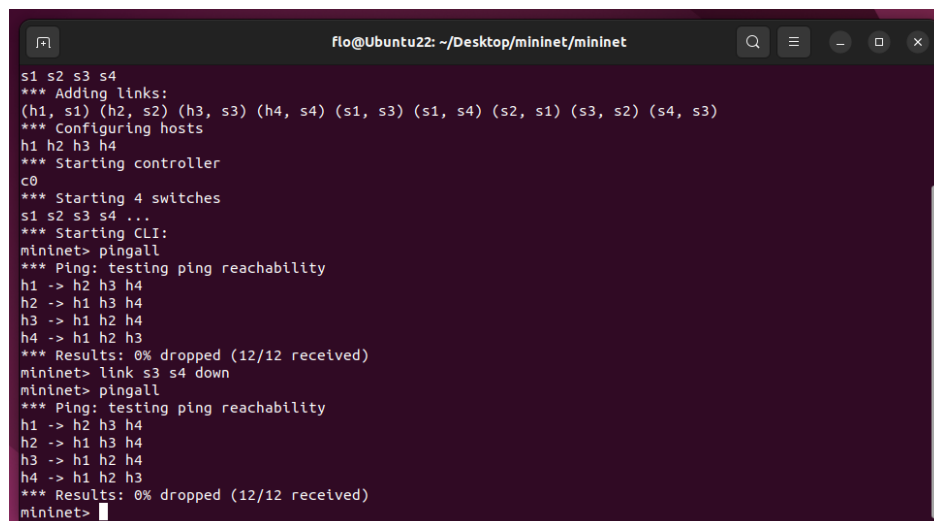
```
flo@Ubuntu22: ~/Desktop/mininet/mininet
flo@Ubuntu22:~/Desktop/mininet/mininet$ sudo mn --custom topologia1.py --topo MyTopo --mac --controller
remote --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s1, s3) (s1, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

Figura 1: Verificación utilizando pingall.

Como se puede observar, se estableció de manera correcta la conexión entre los hosts y los switches.

Eliminar una conexión

Al eliminar una de las conexiones entre los switches, en este caso se eliminó la conexión entre 3 y 4, se puede observar como los hosts también son capaces de comunicarse entre ellos a través de las conexiones existentes anteriores.



```
flo@Ubuntu22: ~/Desktop/mininet/mininet
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s1, s3) (s1, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> link s3 s4 down
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

Figura 2: Verificación luego de eliminar una conexión.