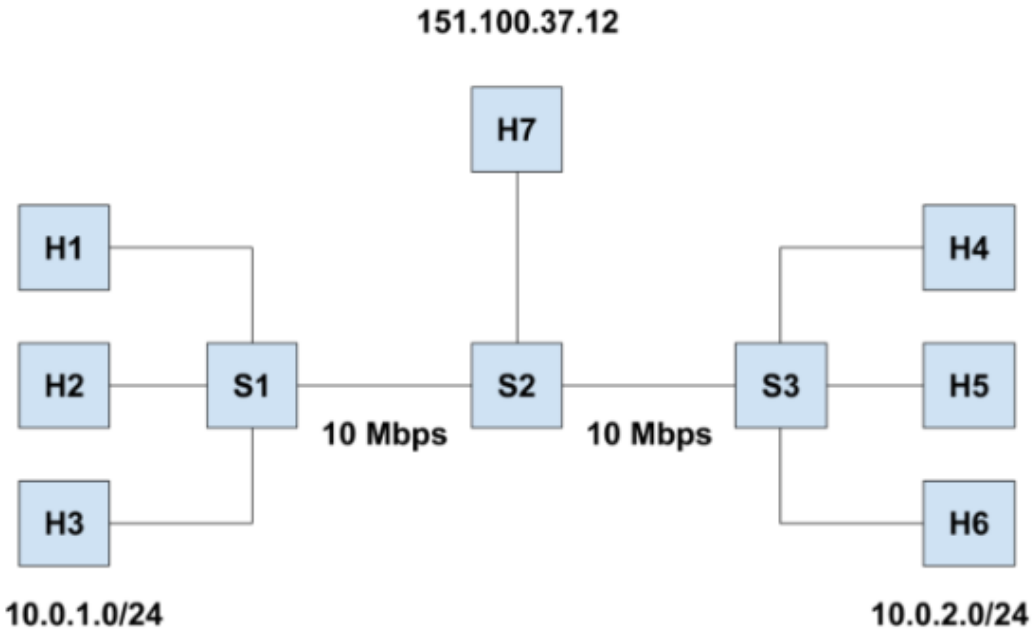
PRACTICA 3. MININET

PEDRO MIGUEL CARMONA

ÍNDICE

Trabajo a realizar

Dado el siguiente escenario de red, compuesta por 3 switches SDN y 7 hosts, crear la topología en *Mininet* y configurar las tablas de flujo de los switches SDN.



La red no tiene un controlador definido. Conceptualmente, existen dos redes privadas: la red 10.0.1.0/24 a la que pertenecen los hosts H1, H2 y H3; y la red 10.0.2.0/24 a la que pertenecen los hosts H4, H5 y H6. Ambas redes privadas están conectadas a través de una red pública que está representada por los switches S1, S2, S3 y el host H7.

La capacidad de los enlaces que conectan los switches es de 10 Mbps cada uno. Además, en el host H7 existe un servidor web en ejecución. Se pide:

1. Asignar las direcciones IP a los hosts de manera coherente según el esquema descrito.
2. Crear un script Python que genere la topología de red descrita.
3. Utilizar el pipeline normal para cada tipo de tráfico en el switch S2.
4. Configurar los switches S1 y S3 de modo que se satisfagan los siguientes puntos:
    a. En el interior de cada red privada, los hosts se deben comunicar entre sí.
    b. El tráfico que va desde la red 10.0.1.0/24 hacia la red 10.0.2.0/24 debe pasar por la red pública utilizando direcciones IP públicas (función NAT) 87.13.148.68 para la dirección IP origen y 87.100.12.18 para la dirección IP destino y viceversa.
    c. Los hosts de la red 10.0.2.0/24 no se deben poder comunicar con el servidor web.
    d. El host H1 es el único de la red 10.0.1.0/24 que puede abrir una conexión SSH con el host H4 (todo el tráfico SSH entre el resto de hosts debe descartarse)

Para la configuración de los tres switches se deben crear tres archivos de texto que contengan las reglas de flujo. Se deben describir cada uno de los pasos realizados en una memoria técnica, además de entregar todos los archivos necesarios.

Topología creada

Para la creación de la topología *Mininet* permite definirla mediante el lenguaje de programación *Python*.

tcp (Workspace) - mininet_topology.py

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-

from mininet.net import Mininet
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink


def myTopology():

    net = Mininet(topo=None, build=False, link=TCLink, controller=None)

    h1 = net.addHost('h1', ip='10.0.1.1/24', mac='00:00:00:00:00:01')
    h2 = net.addHost('h2', ip='10.0.1.2/24', mac='00:00:00:00:00:02')
    h3 = net.addHost('h3', ip='10.0.1.3/24', mac='00:00:00:00:00:03')
    h4 = net.addHost('h4', ip='10.0.2.1/24', mac='00:00:00:00:00:04')
    h5 = net.addHost('h5', ip='10.0.2.2/24', mac='00:00:00:00:00:05')
    h6 = net.addHost('h6', ip='10.0.2.3/24', mac='00:00:00:00:00:06')
    h7 = net.addHost('h7', ip='151.100.37.12/24', mac='00:00:00:00:00:07')

    s1 = net.addSwitch('s1')
    s2 = net.addSwitch('s2')
    s3 = net.addSwitch('s3')

    net.addLink(s1, s2, cls=TCLink, bw=10)
    net.addLink(s2, s3, cls=TCLink, bw=10)

    net.addLink(h1, s1)
    net.addLink(h2, s1)
    net.addLink(h3, s1)
    net.addLink(h7, s2)
    net.addLink(h4, s3)
    net.addLink(h5, s3)
    net.addLink(h6, s3)

    net.start()

    h1.cmdPrint('route add default gw 10.0.1.254 h1-eth0')
    h2.cmdPrint('route add default gw 10.0.1.254 h2-eth0')
    h3.cmdPrint('route add default gw 10.0.1.254 h3-eth0')
    h4.cmdPrint('route add default gw 10.0.2.254 h4-eth0')
    h5.cmdPrint('route add default gw 10.0.2.254 h5-eth0')
    h6.cmdPrint('route add default gw 10.0.2.254 h6-eth0')
    h7.cmdPrint('route add default gw 151.100.37.254 h7-eth0')

    h1.cmdPrint('arp -s 10.0.1.254 00:00:00:00:11:11')
    h2.cmdPrint('arp -s 10.0.1.254 00:00:00:00:11:11')
    h3.cmdPrint('arp -s 10.0.1.254 00:00:00:00:11:11')
    h4.cmdPrint('arp -s 10.0.2.254 00:00:00:00:22:22')
    h5.cmdPrint('arp -s 10.0.2.254 00:00:00:00:22:22')
    h6.cmdPrint('arp -s 10.0.2.254 00:00:00:00:22:22')
    h7.cmdPrint('arp -s 151.100.37.254 00:00:00:00:33:33')

    h7.cmdPrint('sudo python3 -m http.server 80 &')

    CLI(net)

    net.stop()


if __name__ == '__main__':
    setLogLevel('info')
    myTopology()
```

Configuración de Switches

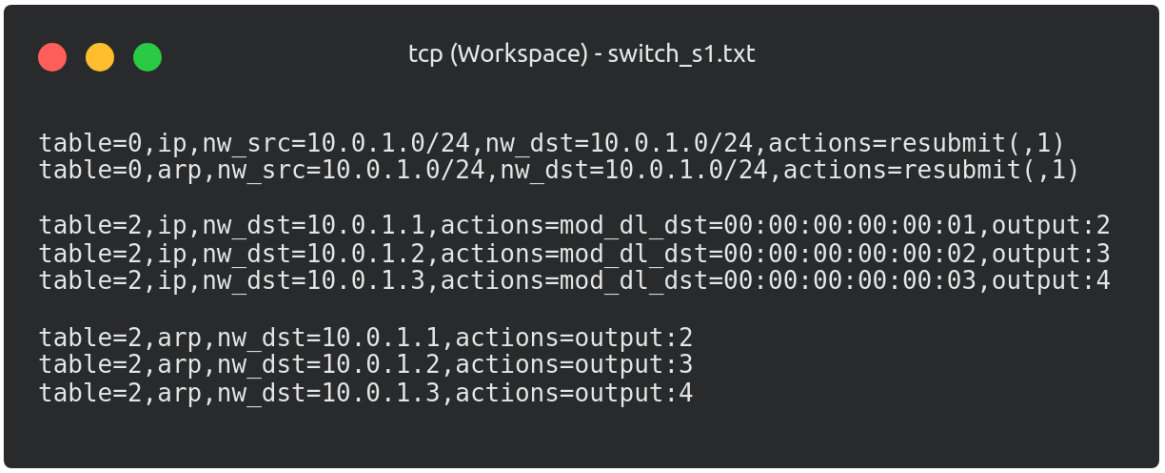
Switch S1

tcp (Workspace) - switch_s1.txt

```
table=0,ip,nw_src=10.0.1.0/24,nw_dst=10.0.1.0/24,actions=resubmit(,1)
table=0,arp,nw_src=10.0.1.0/24,nw_dst=10.0.1.0/24,actions=resubmit(,1)

table=0,ip,nw_dst=87.100.12.18/24,actions=resubmit(,1)
table=0,arp,nw_dst=87.100.12.18/24,actions=resubmit(,1)

table=0,ip,nw_dst=87.13.148.68/24,actions=resubmit(,1)
table=0,arp,nw_dst=87.13.148.68/24,actions=resubmit(,1)

table=0,ip,nw_src=10.0.1.0/24,nw_dst=151.100.37.12/24,tp_dst=80,actions=resubmit(,1)
table=0,arp,nw_src=10.0.1.0/24,nw_dst=151.100.37.12/24,tp_dst=80,actions=resubmit(,1)

table=0,ip,nw_src=151.100.37.12/24,nw_dst=10.0.1.0/24,tp_src=80,actions=resubmit(,1)
table=0,arp,nw_src=151.100.37.12/24,nw_dst=10.0.1.0/24,tp_src=80,actions=resubmit(,1)

table=0,ip,nw_src=10.0.1.1/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=10.0.1.1/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)

table=0,ip,nw_src=87.100.12.18/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=87.100.12.18/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)

table=0,priority=0,actions=drop

table=1,ip,nw_dst=87.13.148.68,actions=mod_nw_dst=10.0.1.1,resubmit(,2)
table=1,ip,nw_dst=87.100.12.18,actions=mod_nw_src=87.13.148.68,resubmit(,2)

table=1,priority=0,actions=resubmit(,2)

table=2,ip,nw_dst=10.0.1.1,actions=mod_dl_dst=00:00:00:00:00:01,output:2
table=2,ip,nw_dst=10.0.1.2,actions=mod_dl_dst=00:00:00:00:00:02,output:3
table=2,ip,nw_dst=10.0.1.3,actions=mod_dl_dst=00:00:00:00:00:03,output:4
table=2,ip,nw_dst=151.100.37.12,actions=mod_dl_dst=00:00:00:00:00:07,output:1
table=2,ip,nw_dst=87.100.12.18,actions=output:1

table=2,arp,nw_dst=10.0.1.1,actions=output:2
table=2,arp,nw_dst=10.0.1.2,actions=output:3
table=2,arp,nw_dst=10.0.1.3,actions=output:4
table=2,arp,nw_dst=87.100.12.18,actions=output:1
table=2,arp,nw_dst=151.100.37.12,actions=output:1
```

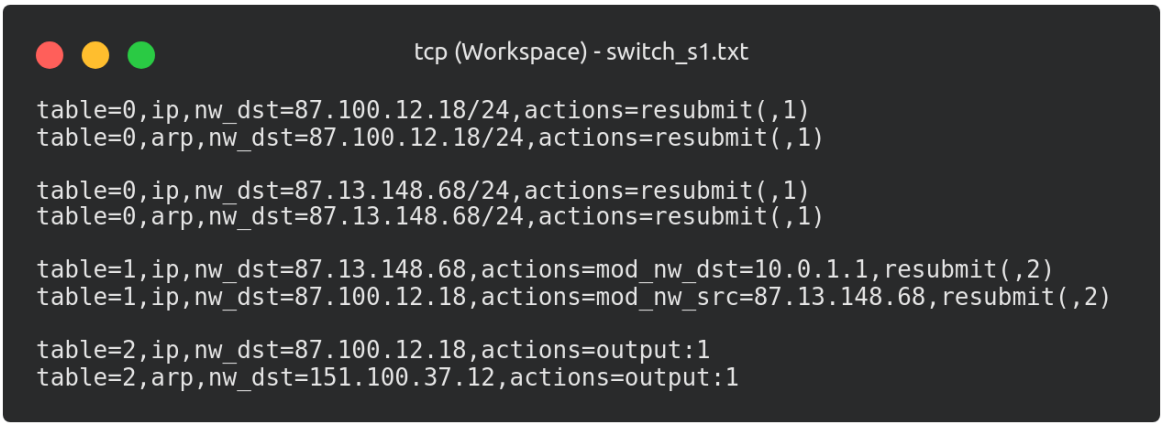
Para cumplir con los requisitos de configuración del primer switch (S1) se han definido las reglas que se pueden apreciar en la imagen anterior. A continuación se detallan las reglas que se han definido para cumplir los puntos detallados en el trabajo a realizar.

Para el punto A, para satisfacer la conexión entre los host de la misma red se ha habilitado las normas que satisfagan que el origen y el destinos sean de la misma red y para cada uno de los host, se ha habilitado la salida correspondiente del *Switch S1* que está conectado ese host.

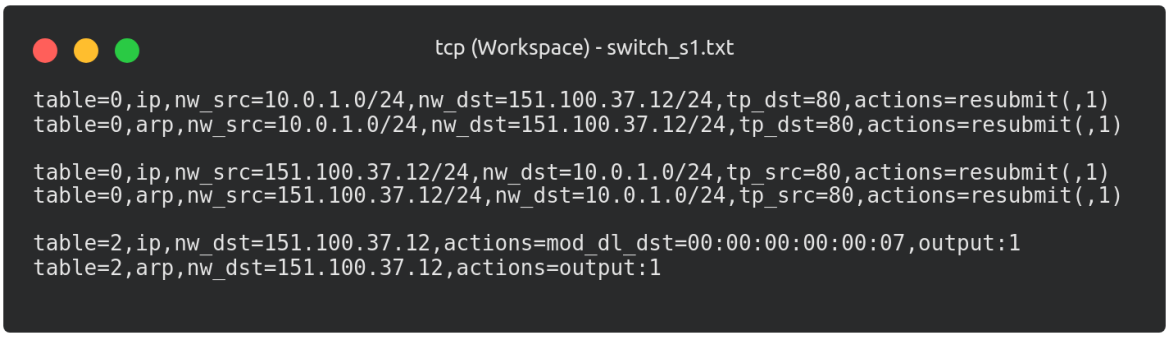


Para el punto B, se ha establecido el host *H1* como único host que puede hacer el *NAT*. Hay que establecer las normas en este *Switch S1* de entrada de paquetes que tengan la procedencia de la otra red pública establecida en esta topología, todo lo que procede del host *H1 (10.0.1.1)* y que se vaya a mandar a la red pública.

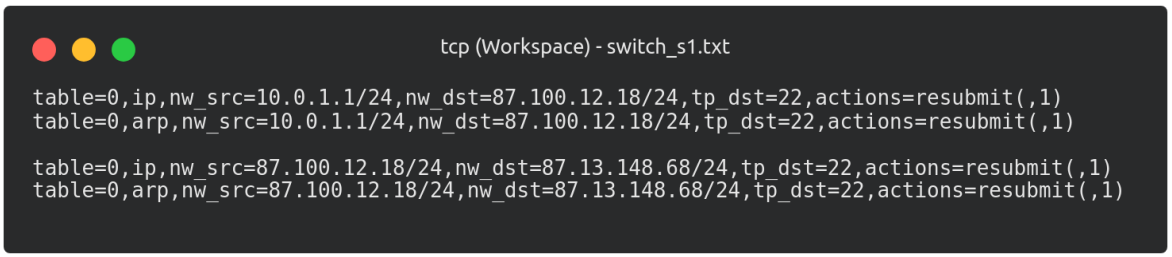
En la tabla 1 se hace la conversión de la dirección ip privada a la dirección ip pública de la red, en caso de que salga el paquete de la red y a la inversa si entra el paquete de la otra red pública.



Para el punto C, se establece el flujo de los datos para establecer la conexión con el servidor web, estableciendo por qué puerto del *Switch S1* tiene que salir.

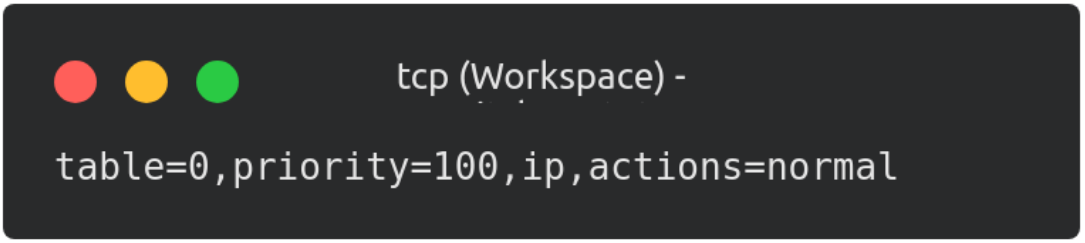


Para el punto D, para establecer esa conexión por ssh , se establecen las normas que nos van a permitir el flujo de los paquetes que tengan como destino el puerto 22, que es precisamente el que utiliza SSH. Se establece tanto la salida del red como la entrada a la misma.



Switch S2

Para el *Switch S2* el enunciado tan solo establece que se haga la acción normal.



Switch S3

tcp (Workspace) - switch_s3.txt

```
table=0,ip,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24,actions=resubmit(,1)
table=0,arp,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24,actions=resubmit(,1)

table=0,ip,nw_src=87.13.148.68/24,actions=resubmit(,1)
table=0,arp,nw_src=87.13.148.68/24,actions=resubmit(,1)

table=0,ip,nw_dst=87.13.148.68/24,actions=resubmit(,1)
table=0,arp,nw_dst=87.13.148.68/24,actions=resubmit(,1)

table=0,ip,nw_src=10.0.2.1/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=10.0.2.1/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)

table=0,ip,nw_src=87.13.148.68/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=87.13.148.68/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)

table=0,priority=0,actions=drop

table=1,ip,nw_src=87.13.148.68,actions=mod_nw_dst=10.0.2.1,resubmit(,2)
table=1,ip,nw_dst=87.13.148.68,actions=mod_nw_src=87.100.12.18,resubmit(,2)

table=1,priority=0,actions=resubmit(,2)

table=2,ip,nw_dst=10.0.2.1,actions=mod_dl_dst=00:00:00:00:00:04,output:2
table=2,ip,nw_dst=10.0.2.2,actions=mod_dl_dst=00:00:00:00:00:05,output:3
table=2,ip,nw_dst=10.0.2.3,actions=mod_dl_dst=00:00:00:00:00:06,output:4
table=2,ip,nw_dst=87.13.148.68,actions=output:1

table=2,arp,nw_dst=10.0.2.1,actions=output:2
table=2,arp,nw_dst=10.0.2.2,actions=output:3
table=2,arp,nw_dst=10.0.2.3,actions=output:4
table=2,arp,nw_dst=87.13.148.68,actions=output:1
```

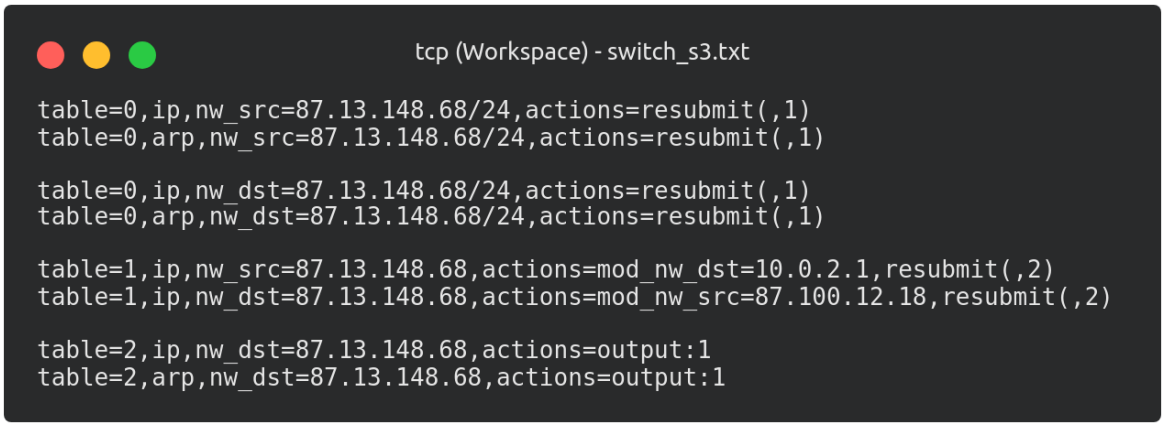
Para el punto A, para satisfacer la conexión entre los host de la misma red se ha habilitado las normas que satisfagan que el origen y el destinos sean de la misma red y para cada uno de los host, se ha habilitado la salida correspondiente del *Switch S3* que está conectado ese host.

tcp (Workspace) - switch_s3.txt

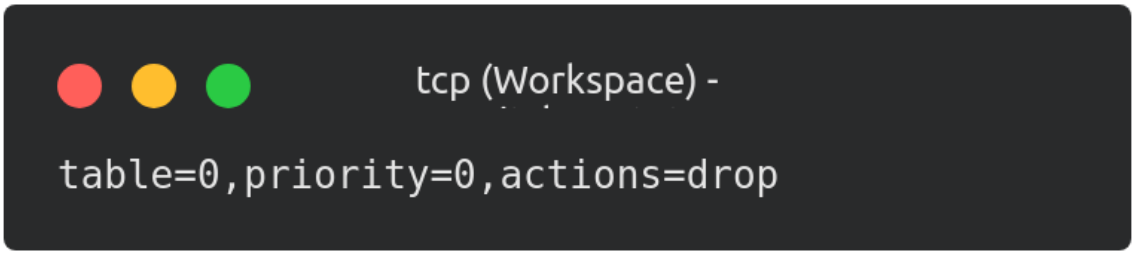
```
table=0,ip,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24,actions=resubmit(,1)
table=0,arp,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24,actions=resubmit(,1)

table=2,ip,nw_dst=10.0.2.1,actions=mod_dl_dst=00:00:00:00:00:04,output:2
table=2,ip,nw_dst=10.0.2.2,actions=mod_dl_dst=00:00:00:00:00:05,output:3
table=2,ip,nw_dst=10.0.2.3,actions=mod_dl_dst=00:00:00:00:00:06,output:4

table=2,arp,nw_dst=10.0.2.1,actions=output:2
table=2,arp,nw_dst=10.0.2.2,actions=output:3
table=2,arp,nw_dst=10.0.2.3,actions=output:4
```

Para el punto B, se ha establecido el host *H4* como único host que puede hacer el *NAT*. Hay que establecer las normas en este *Switch S3* de entrada de paquetes que tengan la procedencia de la otra red pública establecida en esta topología, todo lo que procede del host *H4 (10.0.2.1)* y que se vaya a mandar a la red pública.

En la tabla 1 se hace la conversión de la dirección ip privada a la dirección ip pública de la red, en caso de que salga el paquete de la red y a la inversa si entra el paquete de la otra red pública.



Para el punto C, se establece la norma por defecto, con prioridad 0, para que se dropeen los paquetes que no coincidan con ninguna norma y como no se han establecido ninguna norma con respecto al servidor web, estos paquetes se descartan



Para el punto D, para establecer esa conexión por ssh , se establecen las normas que nos van a permitir el flujo de los paquetes que tengan como destino el puerto 22, que es precisamente el que utiliza SSH. Se establece tanto la salida de la red como la entrada a la misma.

tcp (Workspace) - switch_s3.txt

```
table=0,ip,nw_src=10.0.2.1/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=10.0.2.1/24,nw_dst=87.13.148.68/24,tp_dst=22,actions=resubmit(,1)

table=0,ip,nw_src=87.13.148.68/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)
table=0,arp,nw_src=87.13.148.68/24,nw_dst=87.100.12.18/24,tp_dst=22,actions=resubmit(,1)
```

Pruebas realizadas

En este apartado se muestran las pruebas realizadas para ver que todo funciona correctamente en la topología, tal y como se ha diseñado.

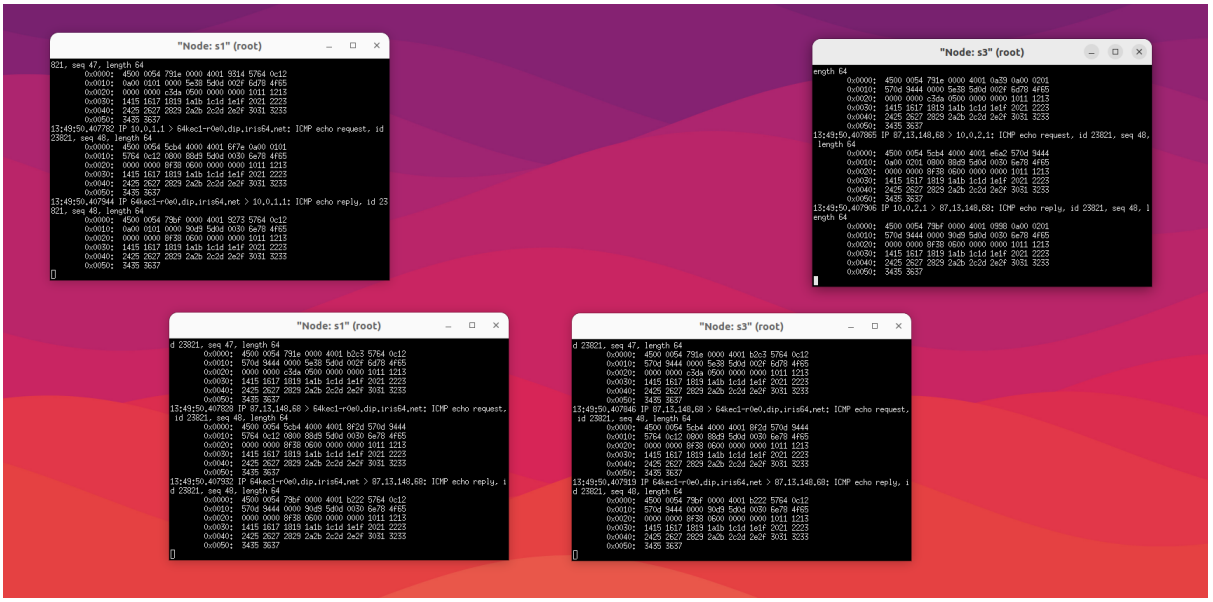
Redes internas

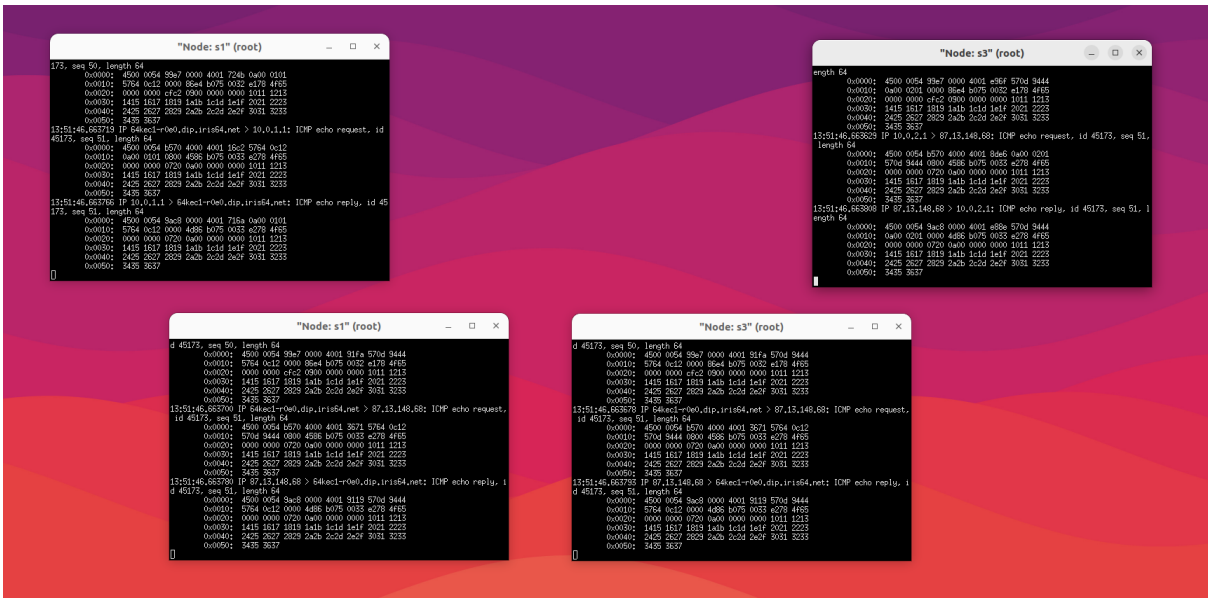
Haciendo un pingall para ver que host se pueden comunicar, nos sale que las redes internas funcionan perfectamente. Además también se puede apreciar que la red privada 10.0.1.0 tiene conexión con el servidor web, mientras que la otra, la red 10.0.2.0 no la tiene.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X X X h7
h2 -> h1 h3 X X X h7
h3 -> h1 h2 X X X h7
h4 -> X X X h5 h6 X
h5 -> X X X h4 h6 X
h6 -> X X X h4 h5 X
h7 -> h1 h2 h3 X X X
*** Results: 57% dropped (18/42 received)
mininet>
```

Redes públicas y privadas

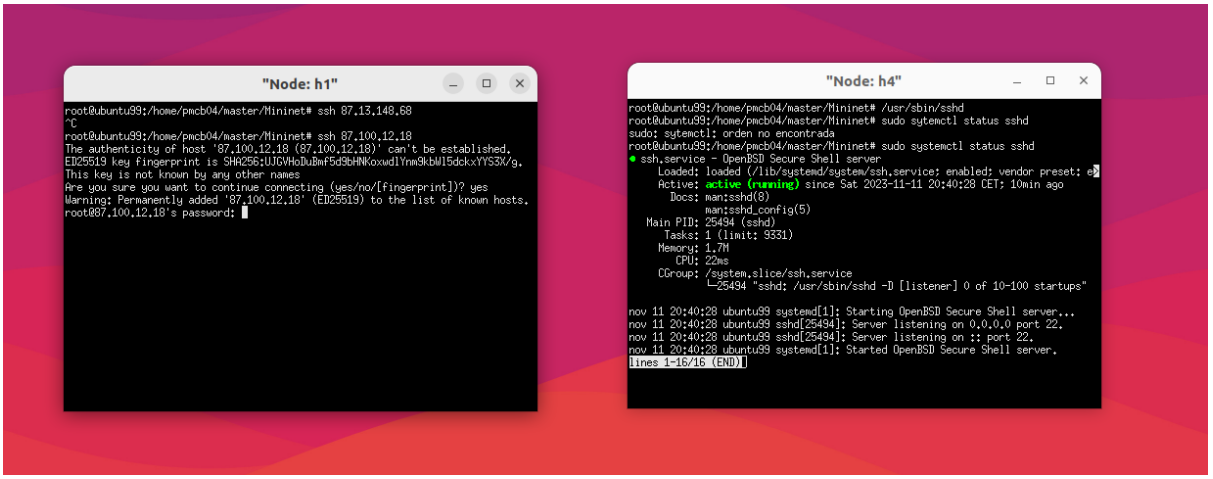
En las siguientes imágenes se muestran las pruebas realizadas para ver el funcionamiento del NAT diseñado. En la primera imagen se muestra el ping desde el host *H1* hacia la red pública 87.100.12.18, mientras que en la segunda imagen se hace la prueba desde el host *H4* hacia la red pública 87.13.148.68.





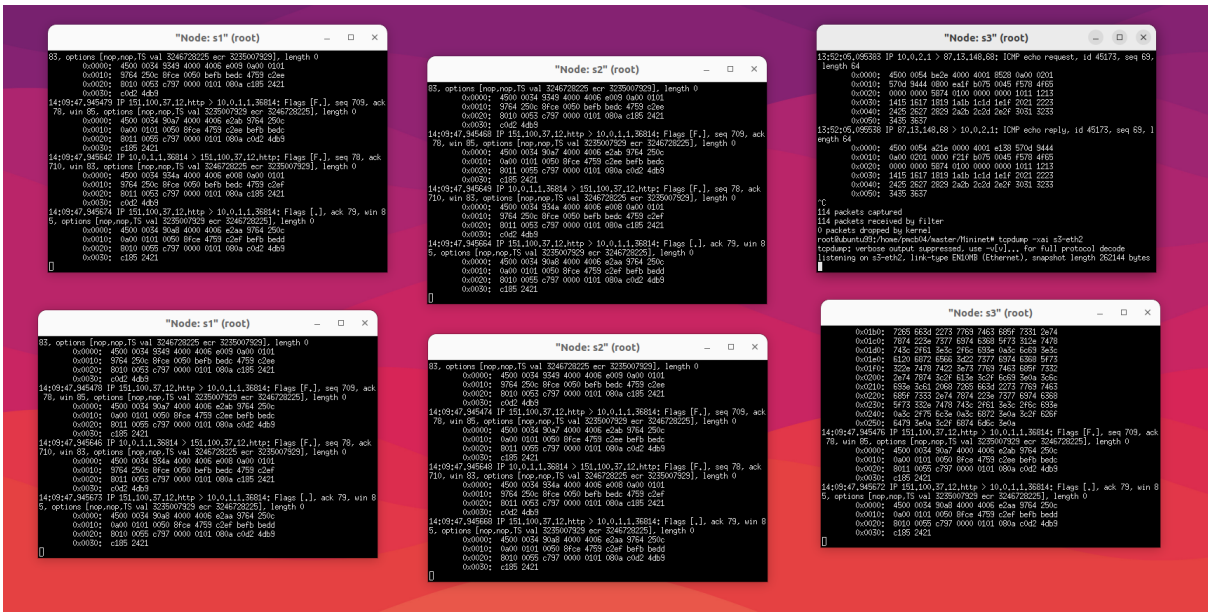
Conexión por SSH

Para este caso, se ha abierto el servidor en el host contrario y se ha establecido la conexión por *SSH*. En la primera imagen se hace la conexión desde el host *H1* al servidor *H4*, mientras que en el segundo se hace desde el host *H4* al servidor *H1*.



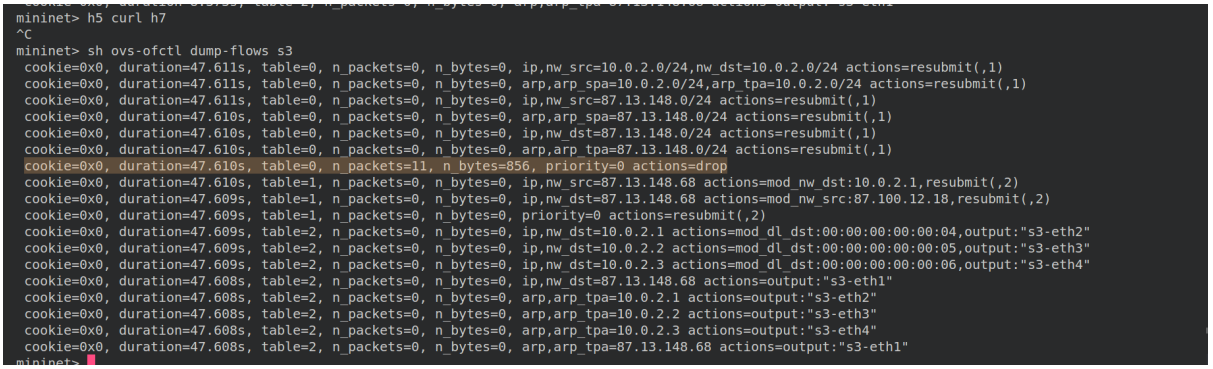
Conexión con el servidor web

En la siguiente imagen se puede ver el camino que recorre un paquete desde el host *H1* hacia el servidor web establecido en el host *H7*.



Mientras que en la siguientes imágenes se muestra como se hace efectiva la norma de dropeo que se ha establecido en el *Switch S3* para que los hosts de la red 10.0.2.0 no puedan establecer la conexión con el servidor web. En la primera imagen se puede ver el estado de la norma antes de ejecutar la conexión y en la segunda se aprecia como se ha ejecutado debido a que el *n_bytes* captados por la norma ha incrementado.

```
mininet> sh ovs-ofctl add-flows s3 switch_s3.txt
mininet> sh ovs-ofctl dump-flows s3
 cookie=0x0, duration=8.376s, table=0, n_packets=0, n_bytes=0, ip,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.376s, table=0, n_packets=0, n_bytes=0, arp,arp_spa=10.0.2.0/24,arp_tpa=10.0.2.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.376s, table=0, n_packets=0, n_bytes=0, ip,nw_src=87.13.148.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.375s, table=0, n_packets=0, n_bytes=0, arp,arp_spa=87.13.148.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.375s, table=0, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.375s, table=0, n_packets=0, n_bytes=0, arp,arp_tpa=87.13.148.0/24 actions=resubmit(,1)
 cookie=0x0, duration=8.375s, table=0, n_packets=3, n_bytes=247, priority=0 actions=drop
 cookie=0x0, duration=8.375s, table=1, n_packets=0, n_bytes=0, ip,nw_src=87.13.148.68 actions=mod_nw_dst:10.0.2.1,resubmit(,2)
 cookie=0x0, duration=8.374s, table=1, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.68 actions=mod_nw_src:87.100.12.18,resubmit(,2)
 cookie=0x0, duration=8.374s, table=1, n_packets=0, n_bytes=0, priority=0 actions=resubmit(,2)
 cookie=0x0, duration=8.374s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.1 actions=mod_dl_dst:00:00:00:00:00:04,output:"s3-eth2"
 cookie=0x0, duration=8.374s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.2 actions=mod_dl_dst:00:00:00:00:00:05,output:"s3-eth3"
 cookie=0x0, duration=8.374s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.3 actions=mod_dl_dst:00:00:00:00:00:06,output:"s3-eth4"
 cookie=0x0, duration=8.373s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.68 actions=output:"s3-eth1"
 cookie=0x0, duration=8.373s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.1 actions=output:"s3-eth2"
 cookie=0x0, duration=8.373s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.2 actions=output:"s3-eth3"
 cookie=0x0, duration=8.373s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.3 actions=output:"s3-eth4"
 cookie=0x0, duration=8.373s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=87.13.148.68 actions=output:"s3-eth1"
```


mininet> h5 curl h7
^C
mininet> sh ovs-ofctl dump-flows s3
cookie=0x0, duration=47.611s, table=0, n_packets=0, n_bytes=0, ip,nw_src=10.0.2.0/24,nw_dst=10.0.2.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.611s, table=0, n_packets=0, n_bytes=0, arp,arp_spa=10.0.2.0/24,arp_tpa=10.0.2.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.611s, table=0, n_packets=0, n_bytes=0, ip,nw_src=87.13.148.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.610s, table=0, n_packets=0, n_bytes=0, arp,arp_spa=87.13.148.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.610s, table=0, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.610s, table=0, n_packets=0, n_bytes=0, arp,arp_tpa=87.13.148.0/24 actions=resubmit(,1)
cookie=0x0, duration=47.610s, table=0, n_packets=11, n_bytes=856, priority=0 actions=drop
cookie=0x0, duration=47.610s, table=1, n_packets=0, n_bytes=0, ip,nw_src=87.13.148.68 actions=mod_nw_dst:10.0.2.1,resubmit(,2)
cookie=0x0, duration=47.609s, table=1, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.68 actions=mod_nw_src:87.100.12.18,resubmit(,2)
cookie=0x0, duration=47.609s, table=1, n_packets=0, n_bytes=0, priority=0 actions=resubmit(,2)
cookie=0x0, duration=47.609s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.1 actions=mod_dl_dst:00:00:00:00:00:04,output:"s3-eth2"
cookie=0x0, duration=47.609s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.2 actions=mod_dl_dst:00:00:00:00:00:05,output:"s3-eth3"
cookie=0x0, duration=47.609s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=10.0.2.3 actions=mod_dl_dst:00:00:00:00:00:06,output:"s3-eth4"
cookie=0x0, duration=47.608s, table=2, n_packets=0, n_bytes=0, ip,nw_dst=87.13.148.68 actions=output:"s3-eth1"
cookie=0x0, duration=47.608s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.1 actions=output:"s3-eth2"
cookie=0x0, duration=47.608s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.2 actions=output:"s3-eth3"
cookie=0x0, duration=47.608s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=10.0.2.3 actions=output:"s3-eth4"
cookie=0x0, duration=47.608s, table=2, n_packets=0, n_bytes=0, arp,arp_tpa=87.13.148.68 actions=output:"s3-eth1"
mininet>