

I Ciclo, 2022.

Proyecto parte 2 - Mininet y SDN.

Fabián Orozco Chaves - B95690.

Pruebas que validan la realización - Tutorial 1. (<https://hackmd.io/@pmanzoni/BklqpKddS>)

1 Test connectivity between hosts

The image shows two terminal windows side-by-side. The left window is titled 'Capturing from Loopback: lo' and displays Wireshark output for interface 'lo'. It shows several OpenFlow frames being transmitted and received. The right window is titled 'mininet>' and shows host status (CPU usage, memory, disk, network interfaces) and ping statistics between hosts h1 and h2.

```
[fabian:Labi]
$ sudo wireshark
[sudo] contraseña para fabian:
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

openflow_v1

No. Time Source Destination Protocol Length Info
31 15.10999842 36:b9:a4:af:f1:c4 02:a6:57:2b:d... OpenFlow 128 Type: OFPT_PACKET_IN
32 15.110976123 127.0.0.1 127.0.0.1 OpenFlow 145 Type: OFPT_FLOW_MOD
34 305.9110496308 36:b9:a4:af:f1:c4 02:a6:57:2b:d... OpenFlow 128 Type: OFPT_PACKET_OUT
36 35.9110936318 02:a6:57:2b:d... OpenFlow 128 Type: OFPT_PACKET_IN
37 15.111152852 127.0.0.1 127.0.0.1 OpenFlow 145 Type: OFPT_FLOW_MOD
39 15.111238532 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 132 Type: OFPT_PACKET_IN
41 29.110971754 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
42 30.110971754 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
44 25.116967806 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
45 25.116967806 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
47 30.117724124 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
48 30.117724124 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: h1 (00:0c:5a:f6:49:40) (00:0c:5a:f6:49:40), Dst: h2 (00:0c:5a:f6:49:3f) (00:0c:5a:f6:49:3f)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 33648, Dst Port: 6653, Seq: 1, Ack: 1, Len: 8
OpenFlow 1.0

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 c0 ... E
0010 00 3c 5a f6 49 40 00 40 06 e1 03 7f 00 00 01 7f 00 -> Z 0 0 ...
0020 00 01 83 70 19 fd 42 7d 44 fb 23 d3 eb 64 80 18 ... p B D# d-
0030 00 56 fe 30 00 00 01 01 08 08 38 00 0b 19 37 ff -V 0 ... 8 - 7-
0040 f7 91 01 02 00 08 00 00 00 00 00 00 00 00 00 00

Loopback: lo:<live capture in progress>
Packets: 49 - Displayed: 31 (63.3%) Profile: Default
```

```
:/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1
ether 8a:63:95:49:dd:42 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 14 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::e465:90ff:fea7:bbf5 prefixlen 64 scopidel 0x20<link>
ether e6:65:90:a7:bb:f5 txqueuelen 1000 (Ethernet)
RX packets 8 bytes 656 (656.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 3259 (3.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::8bb:18ff:fe9a:cf0c prefixlen 64 scopidel 0x20<link>
ether 0a:bb:18:9a:cf:0c txqueuelen 1000 (Ethernet)
RX packets 8 bytes 656 (656.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 3259 (3.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet> h1 ps -a
PID TTY TIME CMD
1092 ttv2 00:00:38 Xorg
1193 ttv2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4368 pts/2 00:00:00 ps
mininet> s1 ps -a
PID TTY TIME CMD
1092 ttv2 00:00:39 Xorg
1193 ttv2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4371 pts/4 00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.31 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.308/3.308/3.308/0.000 ms
mininet>
```

2 Test connectivity between hosts

The image shows two terminal windows side-by-side. The left window is titled 'Capturing from Loopback: lo' and displays Wireshark output for interface 'lo'. It shows a large volume of OpenFlow frames being transmitted and received. The right window is titled 'mininet>' and shows host status (CPU usage, memory, disk, network interfaces) and ping statistics between hosts h1 and h2.

```
[fabian:Labi]
$ sudo wireshark
[sudo] contraseña para fabian:
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

openflow_v1

No. Time Source Destination Protocol Length Info
442 00.00.91056863 10.0.0.1 10.0.0.1 OpenFlow 182 Type: OFPT_PACKET_IN
443 305.9110496308 10.0.0.1 10.0.0.1 OpenFlow 145 Type: OFPT_PACKET_OUT
445 305.911121577 10.0.0.2 10.0.0.1 OpenFlow 182 Type: OFPT_PACKET_IN
446 305.911238784 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
448 305.911255252 10.0.0.2 10.0.0.2 OpenFlow 188 Type: OFPT_PACKET_OUT
449 310.911255252 10.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
450 310.911255252 127.0.0.1 10.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
451 311.044581531 36:b9:a4:af:f1:c4 02:a6:57:2b:d... OpenFlow 126 Type: OFPT_PACKET_IN
451 311.044581531 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 126 Type: OFPT_PACKET_IN
451 311.044581531 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
452 311.044756702 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_PACKET_OUT
453 311.044756702 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
454 311.044766269 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 132 Type: OFPT_PACKET_OUT
455 311.044766269 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 126 Type: OFPT_PACKET_IN
456 311.045099771 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 126 Type: OFPT_PACKET_IN
464 311.045099771 02:a6:57:2b:d:6d 36:b9:a4:af:f... OpenFlow 126 Type: OFPT_PACKET_IN
465 311.046096707 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
467 311.046096707 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_PACKET_OUT
469 311.046096707 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_PACKET_OUT
471 311.046096707 36:b9:a4:af:f1:c4 02:a6:57:2b:d... OpenFlow 132 Type: OFPT_PACKET_OUT
473 316.051490909 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
474 316.051490909 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
476 321.059394731 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
477 321.059394731 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: h1 (00:0c:5a:f6:49:40) (00:0c:5a:f6:49:40), Dst: h2 (00:0c:5a:f6:49:3f) (00:0c:5a:f6:49:3f)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 33648, Dst Port: 6653, Seq: 1, Ack: 1, Len: 8
OpenFlow 1.0

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 c0 ... E
0010 00 3c 5a f6 49 40 00 40 06 e1 03 7f 00 00 01 7f 00 -> Z 0 0 ...
0020 00 01 83 70 19 fd 42 7d 44 fb 23 d3 eb 64 80 18 ... p B D# d-
0030 00 56 fe 30 00 00 01 01 08 08 38 00 0b 19 37 ff -V 0 ... 8 - 7-
0040 f7 91 01 02 00 08 00 00 00 00 00 00 00 00 00 00

Loopback: lo:<live capture in progress>
Packets: 478 - Displayed: 166 (34.7%) Profile: Default
```

```
:/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1
inet6 fe80::e465:90ff:fea7:bbf5 prefixlen 64 scopidel 0x20<link>
ether e6:65:90:a7:bb:f5 txqueuelen 1000 (Ethernet)
RX packets 8 bytes 656 (656.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 3259 (3.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::8bb:18ff:fe9a:cf0c prefixlen 64 scopidel 0x20<link>
ether 0a:bb:18:9a:cf:0c txqueuelen 1000 (Ethernet)
RX packets 8 bytes 656 (656.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 3259 (3.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet> h1 ps -a
PID TTY TIME CMD
1092 ttv2 00:00:38 Xorg
1193 ttv2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4368 pts/2 00:00:00 ps
mininet> s1 ps -a
PID TTY TIME CMD
1092 ttv2 00:00:39 Xorg
1193 ttv2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4371 pts/4 00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.31 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.308/3.308/3.308/0.000 ms
mininet>
```

3 Test connectivity between hosts

The image shows two terminal windows side-by-side. The left window is titled 'Capturing from Loopback:lo' and displays Wireshark output. It lists several OpenFlow messages, including Flow Mod, Packet In, and Echo Request/Reply, all originating from or destined to 127.0.0.1. The right window is titled 'mininet>' and shows host configuration commands. It includes setting flags for interfaces, listing ports, and running ping tests between hosts h1 and h2.

```
[fabian:Lab1]
$ sudo wireshark
[sudo] contraseña para fabian:
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

openflow.v1

No. Time Source Destination Protocol Length Info
581 399.988889912 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
583 399.98933666 10.0.0.2 OpenFlow 188 Type: OFPT_PACKET_IN
585 399.989632837 10.0.0.1 OpenFlow 182 Type: OFPT_PACKET_IN
586 399.989778238 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
588 399.989835297 10.0.0.1 OpenFlow 182 Type: OFPT_PACKET_IN
589 399.99000000 127.0.0.1 OpenFlow 127 Type: OFPT_ECHO_REQUEST
591 399.991247325 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
593 396.036898218 36:09:a4:af:f1:c4 02:46:57:2b:0d:0c OpenFlow 126 Type: OFPT_PACKET_IN
594 396.036953624 02:46:57:2b:d5:6d 36:b9:a4:af:f1:c4 OpenFlow 126 Type: OFPT_PACKET_IN
595 396.037073798 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
597 396.037077201 02:46:57:2b:d5:6d 02:46:57:2b:0d:0c OpenFlow 132 Type: OFPT_PACKET_OUT
599 396.0370982239 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
601 396.037120676 02:46:57:2b:d5:6d 36:b9:a4:af:f1:c4 OpenFlow 132 Type: OFPT_PACKET_OUT
604 396.037819347 02:46:57:2b:d5:6d 36:b9:a4:af:f1:c4 OpenFlow 126 Type: OFPT_PACKET_IN
605 396.037819411 02:46:57:2b:0d:0c 02:46:57:2b:0d:0c OpenFlow 146 Type: OFPT_PACKET_IN
607 396.038108329 127.0.0.1 02:46:57:2b:d5:6d 36:b9:a4:af:f1:c4 OpenFlow 148 Type: OFPT_FLOW_MOD
609 396.038145329 127.0.0.1 127.0.0.1 OpenFlow 146 Type: OFPT_FLOW_MOD
611 396.038162362 36:b9:a4:af:f1:c4 02:46:57:2b:0d:0c OpenFlow 132 Type: OFPT_PACKET_OUT
613 401.040397456 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
614 401.040397456 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY
616 406.040668263 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REQUEST
617 406.040668263 127.0.0.1 127.0.0.1 OpenFlow 74 Type: OFPT_ECHO_REPLY

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 33648, Dst Port: 6653, Seq: 1, Ack: 1, Len: 8
OpenFlow 1.0

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 cd ..E...
0010 00 3c 5a 76 49 00 00 06 e1 03 7f 00 00 01 7f 00 <Z @.-
0020 00 01 83 7d 19 fd 42 7d 44 fb 23 d3 eb 64 80 18 ..p B) D# d-
0030 00 56 fe 38 00 00 01 01 08 0a 38 00 00 19 37 ff V 0... 8-7.
0040 f7 91 01 02 00 08 00 00 00 00 00 00 00 00 00 00

Loopback: lo: <live capture in progress>
Packets: 618 · Displayed: 221 (35.8%)
Profile: Default

/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1

s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inetd fe80::8bb:18ff:fe9a:cf0c brd ff:ff:ff:ff:ff:ff scopeid 0x20<link>
ether 0a:bb:18:9a:cf:0c txqueuelen 1000 (Ethernet)
RX packets 8 bytes 656 (656.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 28 bytes 3259 (3.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet> h1 ps -a
PID TTY TIME CMD
1092 tty2 00:00:38 Xorg
1193 tty2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4368 pts/2 00:00:00 ps
mininet> s1 ps -a
PID TTY TIME CMD
1092 tty2 00:00:39 Xorg
1193 tty2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4371 pts/4 00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.31 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.308/3.308/3.308/0.000 ms
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.56 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.562/1.562/1.562/0.000 ms
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1370.502 seconds

[fabian:Lab1]
$
```

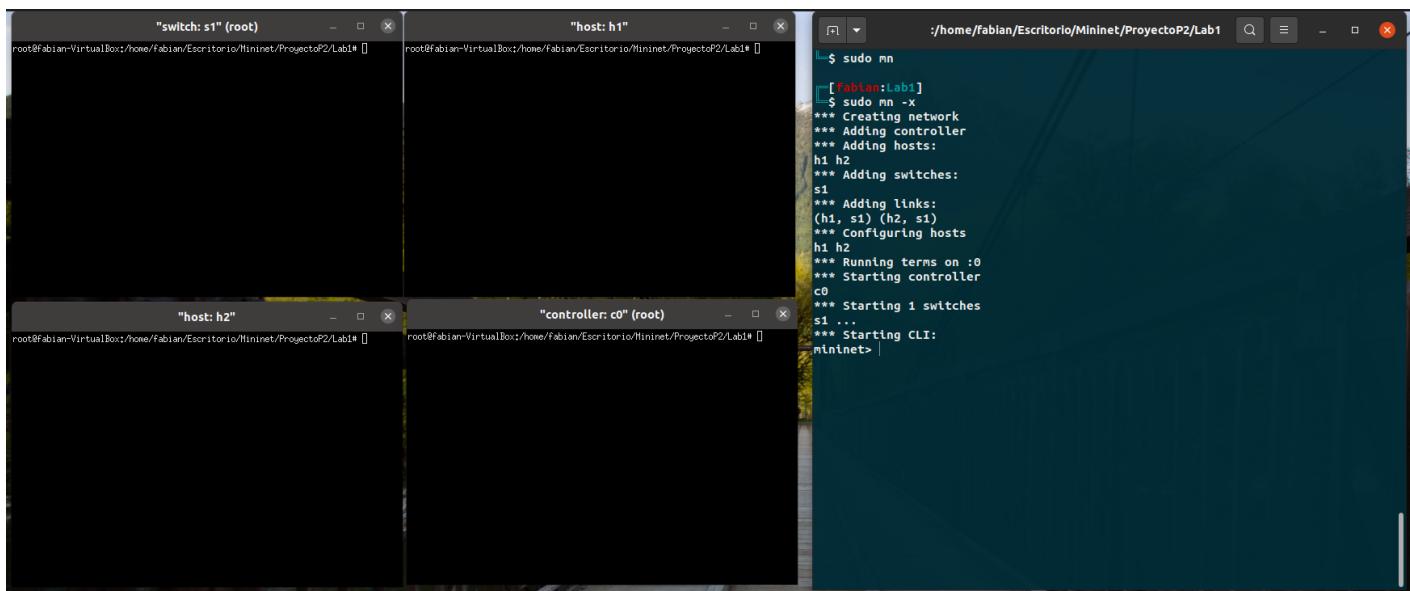
4 h1 ping -c 1 h2

The image shows a single terminal window titled 'mininet>'. It displays the execution of a ping command from host h1 to host h2. The ping command is run three times, resulting in three successful replies with round-trip times of 3.31 ms, 1.56 ms, and 3.31 ms respectively. After the pings, the user exits the mininet session.

```
1193 tty2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4368 pts/2 00:00:00 ps
mininet> s1 ps -a
PID TTY TIME CMD
1092 tty2 00:00:39 Xorg
1193 tty2 00:00:00 gnome-session-b
3980 pts/0 00:00:00 sudo
4006 pts/0 00:00:00 mn
4070 pts/1 00:00:00 controller
4371 pts/4 00:00:00 ps
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.31 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.308/3.308/3.308/0.000 ms
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.56 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.562/1.562/1.562/0.000 ms
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1370.502 seconds

[fabian:Lab1]
$
```

5 XTerm Display



6 Changing Topology Size and Type

```
[fabian:Lab1]
$ sudo mn --test pingall --topo single,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 5.625 seconds

[fabian:Lab1]
$ |
```

7 Changing Topology Size and Type

```
[fabian:Lab1]
$ sudo mn --test pingall --topo linear,4
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Waiting for switches to connect
s1 s2 s3 s4
*** 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)
*** Stopping 1 controllers
c0
*** Stopping 7 links
.....
*** Stopping 4 switches
s1 s2 s3 s4
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 5.975 seconds

[fabian:Lab1]
$
```

8 Namespaces and xterms

```

"controller: c0 (root)"                                     "switch: s1 (root)"                                     "host: h1"                                         "host: h2"
root@fabian-VirtualBox:~/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1# ./c0
root@fabian-VirtualBox:~/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1# ./s1
root@fabian-VirtualBox:~/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1# ./h1
root@fabian-VirtualBox:~/home/fabian/Escritorio/Mininet/ProyectoP2/Lab1# ./h2

*** Configuring hosts
h1 *** h1 : ('l3config', 'hi-eth0', '10.0.0.1/8', 'up')
*** h1 : ('l3config', 'hi-eth0', '10.0.0.1/8', 'up')
h2 *** h2 : ('l3config', 'h2-eth0', '10.0.0.2/8', 'up')
*** h2 : ('l3config', 'h2-eth0', '10.0.0.2/8', 'up')

*** Running tunnels on :0
*** errRun: ['pk11kill', '-g', '-f', 'mnexec.*socat']
1*** errRun: ['xhost', '+:1;localuser:root']
localuser:root being added to access control list
O _open(['mnexec', '-da', '8753B', 'xterm', '-title', '"controller: c0 (root)', '-display', ':0', '-e', 'env TERM=ansi bash'] 87600*** errRun: ['xhost', '+:1;localuser:root']
localuser:root being added to access control list
O _open(['mnexec', '-da', '87553', 'xterm', '-title', '"switch: s1 (root)", "-display", ':0', '-e', 'env TERM=ansi bash'] 87602*** errRun: ['xhost', '+:1;localuser:root']
localuser:root being added to access control list
O _open(['mnexec', '-da', '87546', 'xterm', '-title', '"host: h1", "-display", ':0', '-e', 'env TERM=ansi bash'] 87604*** errRun: ['xhost', '+:1;localuser:root']
localuser:root being added to access control list
O _open(['mnexec', '-da', '87548', 'xterm', '-title', '"host: h2", "-display", ':0', '-e', 'env TERM=ansi bash'] 87606*** Starting controller
co _errRun: ['which', 'controller']
/usr/local/bin/controller
*** c0 : ('controller:tcp:6653 1>/tmp/c0.log 2>/tmp/c0.log &')

*** Starting 1 switches
s1 *** errRun: ovs-vsctl -- -id=@s1 create Controller target="tcp:127.0.0.1:6653" max_backoff_f=1000 -- --id=@s1-listen create Controller target="tcp:6654" max_backoff=1000 -- --if-exists delete br s1 -- add-br s1 -- set bridge si1 controller=@s1c0,@s1-listen[ ] other_config:datapath_id=0000000000000001 fall_mode=secure other-config:disabled-in-band=true other-config:dp-desc=s1 -- add-port s1 s1-eth1 -- set Interface s1-eth1 ofport_request=1 -- add-port si1-eth2 -- set Interface si1-eth2 ofp ort_request=2
dc3669fb-92e7-403f-b231-602514a47c35
019a723a-177a-4bf6-ab2b-c93e58929971
0
*** Starting CLI:
*** errRun: ['stty', 'echo', ' sane', 'intr', '^C']
mininet>

```

9 Python Interpreter

```
[fabian:Lab1]
$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> py 'hello' + 'world'
helleworld
mininet> py locals()
{'h2': <Host h2: h2-eth0:10.0.0.2 pid=121778>, 'net': <mininet.net.Mininet object at 0x7f7c2c110b90>, 'h1': <Host h1: h1-eth0:10.0.0.1 pid=121776>, 'c0': <Controller c0 : 127.0.0.1:6653 pid=121766>, 's1': <OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2 :None pid=121785> }
mininet> py dir(s1)
['IP', 'MAC', 'OVSVersion', 'TCReapply', '__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_popen', '_uids', 'addIntf', 'argmax', 'attach', 'batch', 'batchShutdown', 'batchStartup', 'bridgeOpts', 'checkSetup', 'cleanup', 'cmd', 'cmdPrint', 'cmds', 'commands', 'config', 'configDefault', 'connected', 'connectionsTo', 'controlIntf', 'controllerUUIDs', 'datapath', 'decoder', 'defaultDpid', 'defaultIntf', 'delIntf', 'deleteIntfs', 'detach', 'dpctl', 'dpid', 'dpidLen', 'execed', 'failMode', 'fdToNode', 'inNamespace', 'inToNode', 'inband', 'intf', 'intfIsUp', 'intfList', 'intfNames', 'intfOpts', 'intfs', 'isOldOVS', 'isSetup', 'lastCmd', 'lastPid', 'linkTo', 'listenPort', 'master', 'monitor', 'mountPrivateDirs', 'name', 'nameToI_ntf', 'newPort', 'opts', 'outToNode', 'params', 'pexec', 'pid', 'pollout', 'popen', 'portBase', 'ports', 'privateDirs', 'protocols', 'read', 'readbuf', 'readline', 'reconnects', 'sendCmd', 'sendInt', 'setARP', 'setDefaultRoute', 'setHostRoute', 'setIP', 'setMAC', 'setParam', 'setup', 'shell', 'slave', 'start', 'startShell', 'stdin', 'stdout', 'stop', 'stp', 'terminate', 'unmountPrivateDirs', 'vsctl', 'waitExited', 'waitOutput', 'waitForReadable', 'waiting', 'write']
mininet> py help(h1)

mininet> py h1.IP()
10.0.0.1
mininet>
```

10 ip link show

```
[Fabian:Lab1]
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:42:24:10 brd ff:ff:ff:ff:ff:ff
81: s1-eth1@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master ovs-system state UP mode DEFAULT group default qlen 1000
    link/ether a2:99:2c:34:c9:74 brd ff:ff:ff:ff:ff:ff link-netnsid 0
82: s1-eth2@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master ovs-system state UP mode DEFAULT group default qlen 1000
    link/ether ea:e5:9d:d9:77:8c brd ff:ff:ff:ff:ff:ff link-netnsid 1
83: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 5e:74:22:d2:98:6d brd ff:ff:ff:ff:ff:ff
84: s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether ea:97:dd:1a:1e:4f brd ff:ff:ff:ff:ff:ff

[Fabian:Lab1]
$ |
```

11 Running the script

```
[fabian:Lab1]
$ sudo python3 emptynet.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Running CLI
*** 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> py print("Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC())
Host h1 has IP address 10.0.0.1 and MAC address 8e:8a:62:d1:c8:5a
mininet> |
```

Pruebas que validen la realización - Tutorial 2. (<https://hackmd.io/@pmanzoni/SyWm3n0HH>)

1 create network

```
[fabian:Lab2]
$ sudo mn --topo=single,3 --controller=none --mac
[sudo] contraseña para fabian:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3208>
<Host h2: h2-eth0:10.0.0.2 pid=3210>
<Host h3: h3-eth0:10.0.0.3 pid=3212>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=3217>
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0
mininet> sh ovs-ofctl show s1
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: output enqueue set_vlan_vid set_vlan_pcp strip_vlan mod_dl_src mod_dl_dst mod_nw_src mod_nw_dst mod_nw_tos mod_tp_src mod_tp_dst
1(s1-eth1): addr:aa:19:c5:54:e3:48
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:fe:55:fo:16:31:86
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
3(s1-eth3): addr:4e:81:fb:c7:32:55
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:b2:b2:f9:92:01:4e
    config: PORT_DOWN
    state: LINK_DOWN
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet> |
```

2 dump-flows

```
:/home/fabian/Escritorio/Mininet/ProyectoP2/Lab2
mininet> sh ovs-ofctl add-flow s1 action=normal
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> sh ovs-ofctl dump-flows s1
cookie=0x0, duration=69.377s, table=0, n_packets=24, n_bytes=1680, actions=NORMAL
mininet> sh ovs-ofctl del-flows s1
mininet> sh ovs-ofctl dump-flows s1
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
mininet>
```

3 Using layer 1 data

```
:/home/fabian/Escritorio/Mininet/ProyectoP2/Lab2
mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=2,actions=output:1
mininet> h1 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.100 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.100/0.592/1.084/0.492 ms
mininet> h3 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.3 icmp_seq=1 Destination Host Unreachable
From 10.0.0.3 icmp_seq=2 Destination Host Unreachable

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1016ms
pipe 2
mininet> sh ovs-ofctl dump-flows s1
cookie=0x0, duration=180.998s, table=0, n_packets=4, n_bytes=280, priority=500,in_port="s1-eth1" actions=output:"s1-eth2"
cookie=0x0, duration=166.747s, table=0, n_packets=5, n_bytes=350, priority=500,in_port="s1-eth2" actions=output:"s1-eth1"
mininet> sh ovs-ofctl add-flow s1 priority=32768,action=drop
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
mininet> sh ovs-ofctl del-flows s1 --strict
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X
h2 -> h1 X
h3 -> X X
*** Results: 66% dropped (2/6 received)
mininet>
```

4 Using layer 2 data

```
:/home/fabian/Escritorio/Mininet/ProyectoP2/Lab2
mininet> sh ovs-ofctl del-flows s1
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1
mininet> sh ovs-ofctl add-flow s1 dl_type=0x806,nw_proto=1,action=flood
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X
h2 -> h1 X
h3 -> X X
*** Results: 66% dropped (2/6 received)
mininet> sh ovs-ofctl del-flows s1
mininet>
```

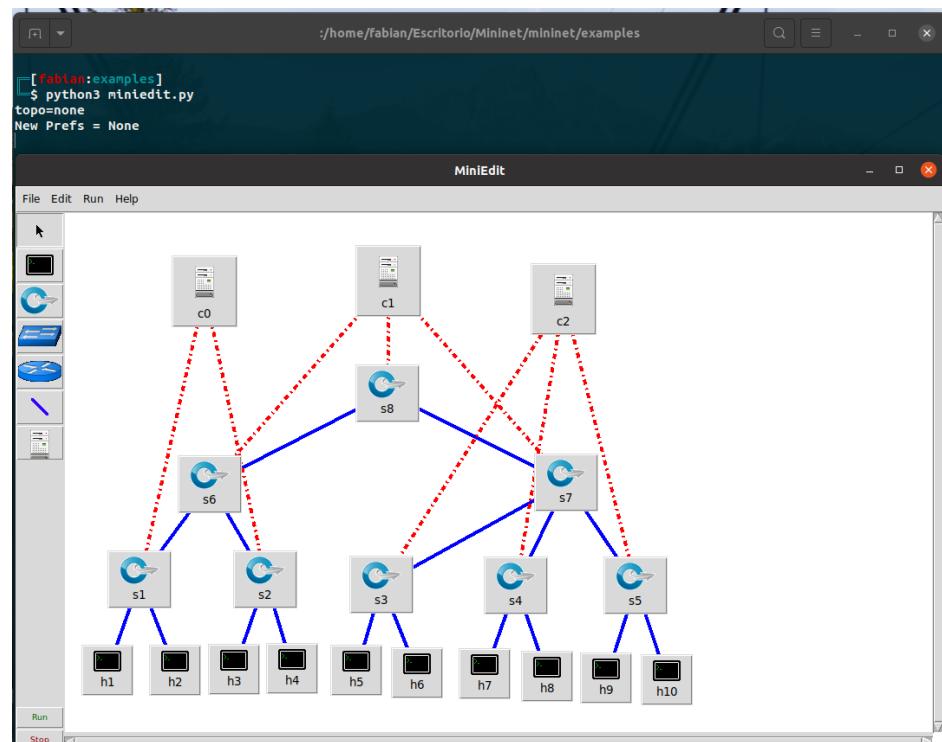
5 Using layer 3 data

```
./home/fabian/Escritorio/Mininet/ProyectoP2/Lab2
mininet> sh ovs-ofctl add-flow s1 priority=500,dl_type=0x800,nw_src=10.0.0.0/24,nw_dst=10.0.0.0/24,actions=normal
mininet> sh ovs-ofctl add-flow s1 priority=800,dl_type=0x800,nw_src=10.0.0.3,nw_dst=10.0.0.0/24,actions=mod_nw_tos:184,normal
mininet> sh ovs-ofctl add-flow s1 arp,nw_dst=10.0.0.1,actions=output:1
mininet> sh ovs-ofctl add-flow s1 arp,nw_dst=10.0.0.2,actions=output:2
mininet> sh ovs-ofctl add-flow s1 arp,nw_dst=10.0.0.3,actions=output:3
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

6 Using layer 4 data

```
./home/fabian/Escritorio/Mininet/ProyectoP2/Lab2
mininet> h3 python3 -m http.server 80 &
mininet> sh ovs-ofctl add-flow s1 arp,actions=normal
mininet> sh ovs-ofctl add-flow s1 priority=500,dl_type=0x800,nw_proto=6,tp_dst=80,actions=output:3
mininet> sh ovs-ofctl add-flow s1 priority=800,ip,nw_src=10.0.0.3,actions=normal
mininet> h1 curl h3
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
</ul>
<hr>
</body>
</html>
mininet> |
```

7 Create a custom network topology using MiniEditpng



8 Configure the controllers

The screenshot shows the Mininet CLI window and the MiniEdit graphical interface side-by-side. The CLI window displays the configuration of three controllers (c0, c1, c2) with their respective hostnames, remote IPs, and ports. The MiniEdit window shows a network diagram with three controllers at the top, each connected to a switch (s6, s8, s7). These switches are further connected to other switches (s1-s5) which are then connected to ten hosts (h1-h10) at the bottom. A blue line connects controller c2 to switch s8, while red dashed lines connect all controllers to their respective switches.

```
topo=none

[Fabian:examples]
$ python3 minitedit.py
topo=none
New Prefs = None
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6635, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c1 = {'hostname': 'c1', 'remoteIP': '127.0.0.1', 'remotePort': 6634, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c0 = {'hostname': 'c0', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
```

9 Experiments with the network

The screenshot shows the Mininet CLI window and the MiniEdit graphical interface side-by-side. The CLI window shows the configuration of three controllers (c0, c1, c2) and the starting of the Mininet CLI. The MiniEdit window shows the same network diagram as the previous screenshot, with three controllers (c0, c1, c2) connected to switches (s6, s8, s7) which are connected to switches (s1-s5) and then to hosts (h1-h10).

```
[Fabian:examples]
$ python3 minitedit.py
topo=none
New Prefs = None
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6635, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c1 = {'hostname': 'c1', 'remoteIP': '127.0.0.1', 'remotePort': 6634, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c0 = {'hostname': 'c0', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c0 = {'hostname': 'c0', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New Prefs = {'ipbase': '10.0.0.0/8', 'terminalType': 'xterm', 'dpctl': '', 'sflow': {'sFlowTarget': '', 'sFlowSampling': '0%', 'sflowHeader': '128', 'sflowPolling': '30'}, 'netflow': {'nflowTarget': '', 'nflowTimeout': '600', 'nflowAddid': '0'}, 'startCLI': '1', 'switchType': 'ovs', 'openFlowVersions': {'ovsOfl0': '1', 'ovsOfl1': '0', 'ovsOfl2': '0', 'ovsOfl3': '0'}}
*** Mininet must run as root.

[Fabian:examples]
$ sudo python3 minitedit.py
[Fabian:examples] contraseña para fabian:
topo=none
Getting Hosts and Switches.
Getting controller selection:ref
Getting controller selection:ref
Getting controller selection:ref
Getting controller selection:ref
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting 3 controllers
c0 c1 c2
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8
No NetFlow targets specified.
No sFlow targets specified.

NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent Mininet from quitting
and will prevent you from starting the network again during this session.

*** Starting CLI:
mininet> |
```

10 exit CLI

```
[fabian:examples]
$ python3 minitedit.py
topo=None
New Prefs = None
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c2 = {'hostname': 'c2', 'remoteIP': '127.0.0.1', 'remotePort': 6635, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c1 = {'hostname': 'c1', 'remoteIP': '127.0.0.1', 'remotePort': 6634, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New controller details for c0 = {'hostname': 'c0', 'remoteIP': '127.0.0.1', 'remotePort': 6633, 'controllerType': 'ref', 'controllerProtocol': 'tcp'}
New Prefs = [{'ipBase': '10.0.0.0/8', 'terminalType': 'xterm', 'dpctl': '', 'sflow': {'sflowTarget': '', 'sflowSampling': '400', 'sflowHeader': '128', 'sflowPolling': '30'}, 'netflow': {'nflowTarget': '', 'nflowTimeout': '600', 'nflowAddId': '0'}, 'startCLI': '1', 'switchType': 'ovs', 'openFlowVersions': {'ovsOf10': '1', 'ovsOf11': '0', 'ovsOf12': '0', 'ovsOf13': '0'}}]
*** Mininet must run as root.

[fabian:examples]
$ sudo python3 minitedit.py
[sudo] contraseña para fabian:
topo=None
Getting Hosts and Switches.
Getting controller selection:ref
Getting controller selection:ref
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
**** Starting 3 controllers
c0 c1 c2
**** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8
No NetFlow targets specified.
No sFlow targets specified.

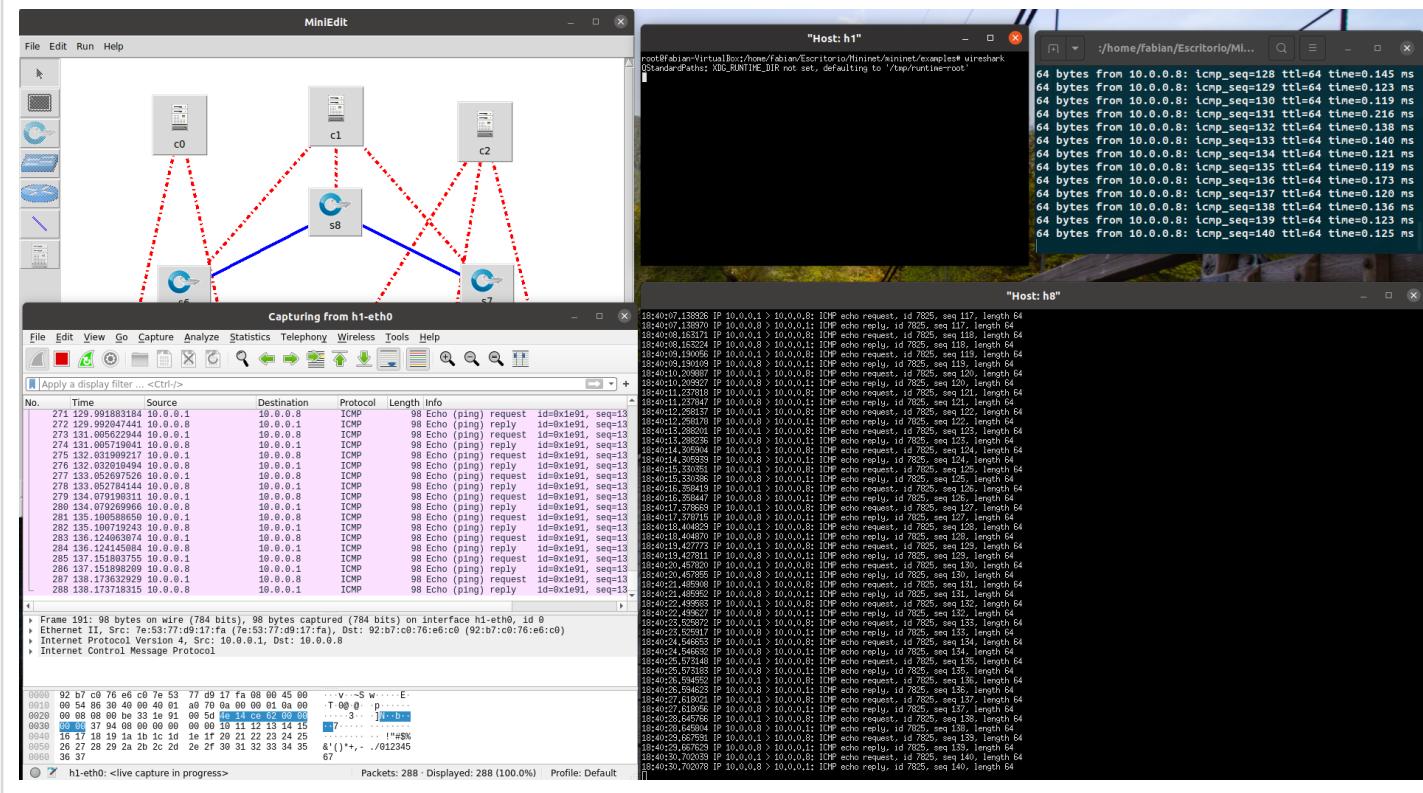
NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting
and will prevent you from starting the network again during this session.

*** Starting CLI:
mininet> exit
*** Stopping 3 controllers
c0 c1 c2
*** Stopping 17 links
.....
*** Stopping 8 switches
s1 s2 s3 s4 s5 s6 s7 s8
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done
```

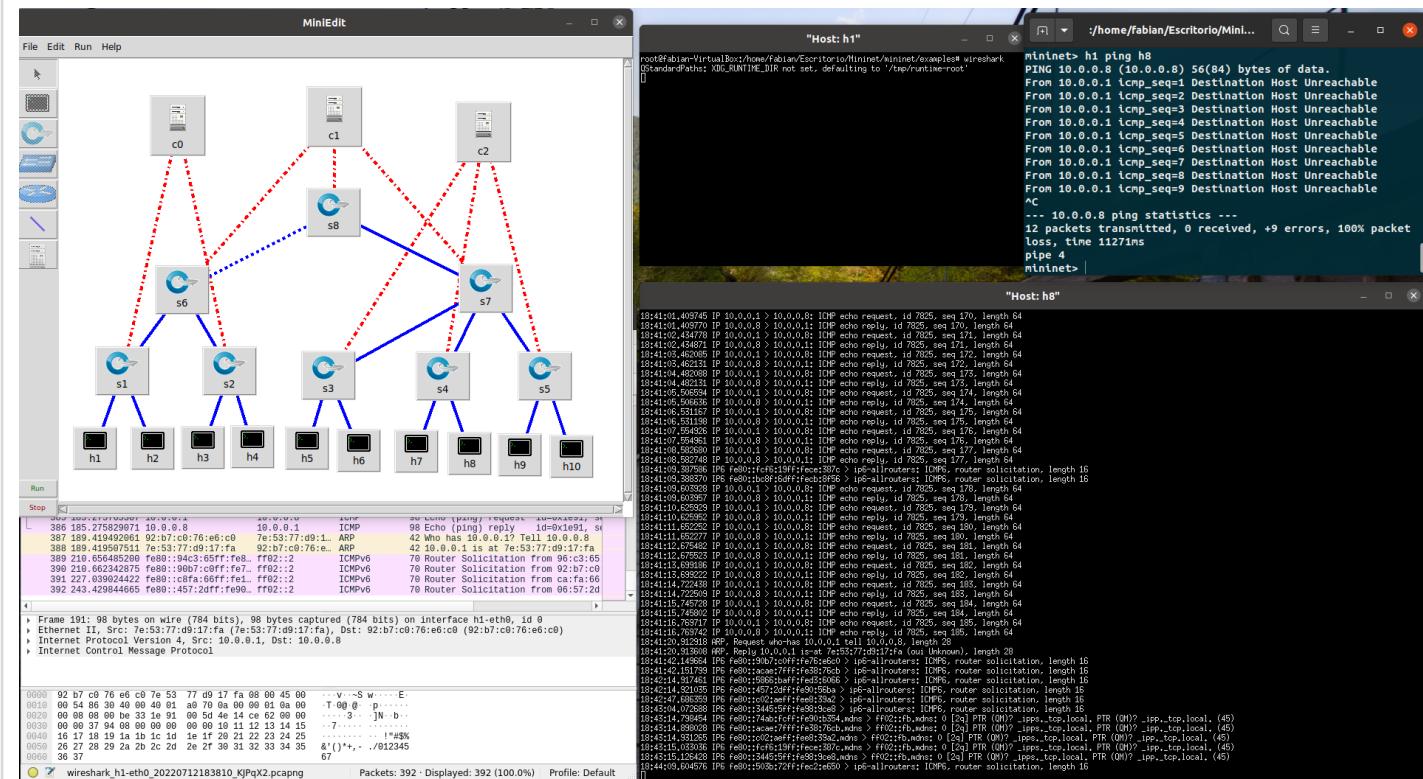
11 View Open vSwitch configurations

```
      fail_mode: secure
      Port s5
        Interface s5
          type: internal
      Port s5-eth3
        Interface s5-eth3
      Port s5-eth1
        Interface s5-eth1
      Port s5-eth2
        Interface s5-eth2
    Bridge s1
      Controller "tcp:127.0.0.1:6633"
      fail_mode: secure
      Port s1-eth1
        Interface s1-eth1
      Port s1-eth2
        Interface s1-eth2
      Port s1
        Interface s1
          type: internal
      Port s1-eth3
        Interface s1-eth3
    Bridge s4
      Controller "tcp:127.0.0.1:6635"
      fail_mode: secure
      Port s4-eth2
        Interface s4-eth2
      Port s4-eth3
        Interface s4-eth3
      Port s4-eth1
        Interface s4-eth1
      Port s4
        Interface s4
          type: internal
    Bridge s8
      Controller "tcp:127.0.0.1:6634"
      fail_mode: secure
      Port s8-eth2
        Interface s8-eth2
      Port s8-eth1
        Interface s8-eth1
      Port s8
        Interface s8
          type: internal
    Bridge s6
      Controller "tcp:127.0.0.1:6634"
      fail_mode: secure
      Port s6-eth2
        Interface s6-eth2
      Port s6
        Interface s6
          type: internal
      Port s6-eth3
        Interface s6-eth3
      Port s6-eth1
        Interface s6-eth1
    Bridge s3
      Controller "tcp:127.0.0.1:6635"
      fail_mode: secure
      Port s3-eth3
        Interface s3-eth3
      Port s3
        Interface s3
          type: internal
      Port s3-eth2
        Interface s3-eth2
      Port s3-eth1
        Interface s3-eth1
  ovs_version: "2.13.5"
Press Enter to close
```

12 monitor traffic



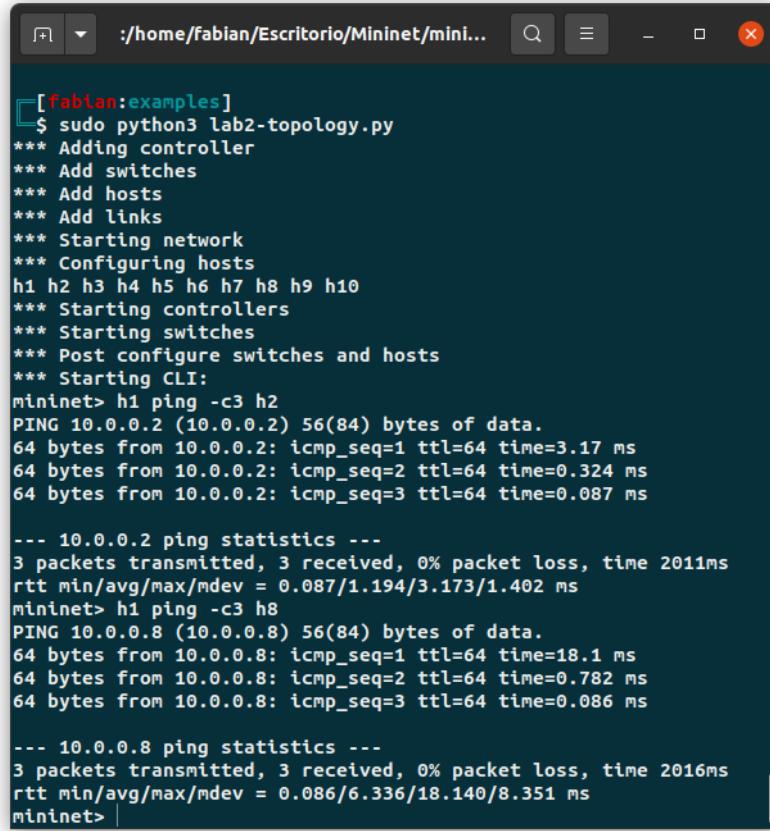
13 Simulate a broken link



14 Check flow tables

```
root@fabian-VirtualBox:/home/fabian/Escritorio/Mininet/mininet/examples# sudo ovs-ofctl dump-flows s1
cookie=0x0, duration=1.282s, table=0, n_packets=4, n_bytes=188, idle_timeout=60, priority=65535,arp,in_port="si-eth3",vlan_tci=0x0000,dl_src="52:00:7e:00:00:c0",dl_dst="e8:33:77:d8:17:fa",arp.op=2,actions=output:"si-eth1"
cookie=0x0, duration=1.282s, table=0, n_packets=4, n_bytes=188, idle_timeout=60, priority=65535,arp,in_port="si-eth3",vlan_tci=0x0000,dl_src="e8:33:77:d8:17:fa",dl_dst="52:00:7e:00:00:c0",arp.op=2,actions=output:"si-eth1"
cookie=0x0, duration=164.095s, table=0, n_packets=4, n_bytes=188, idle_timeout=60, priority=65535,arp,in_port="si-eth1",vlan_tci=0x0000,dl_src="7e:53:77:d9:17:fa",dl_dst="52:b7:e6:c0",arp.op=2,actions=output:"si-eth3"
cookie=0x0, duration=123.853s, table=0, n_packets=3, n_bytes=126, idle_timeout=60, priority=65535,arp,in_port="si-eth1",vlan_tci=0x0000,dl_src="7e:53:77:d9:17:fa",dl_dst="52:b7:e6:c0",arp.op=2,actions=output:"si-eth3"
cookie=0x0, duration=183.291s, table=0, n_packets=99, n_bytes=3408, idle_timeout=60, priority=65535,icmp,in_port="si-eth3",vlan_tci=0x0000,dl_src="7e:53:77:d9:17:fa",dl_dst="52:b7:e6:c0",nu_src=10.0.0.1,nu_dst=10.0.0.8,nu_tos=0,icmp_type=8,icmp_code=0,actions=output:"si-eth3"
cookie=0x0, duration=163.274s, table=0, n_packets=36, n_bytes=948, idle_timeout=60, priority=65535,icmp,in_port="si-eth3",vlan_tci=0x0000,dl_src="92:b7:e6:c0",dl_dst="7e:53:77:d9:17:fa",nu_src=10.0.0.8,nu_dst=10.0.0.1,nu_tos=0,icmp_type=8,icmp_code=0,actions=output:"si-eth1"
root@fabian-VirtualBox:/home/fabian/Escritorio/Mininet/mininet/examples#
```

15 Run the saved Mininet custom topology script



```
[fabian:examples]$ sudo python3 lab2-topology.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.17 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.324 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.087 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.087/1.194/3.173/1.402 ms
mininet> h1 ping -c3 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=18.1 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.782 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.086 ms

--- 10.0.0.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 0.086/6.336/18.140/8.351 ms
mininet> |
```