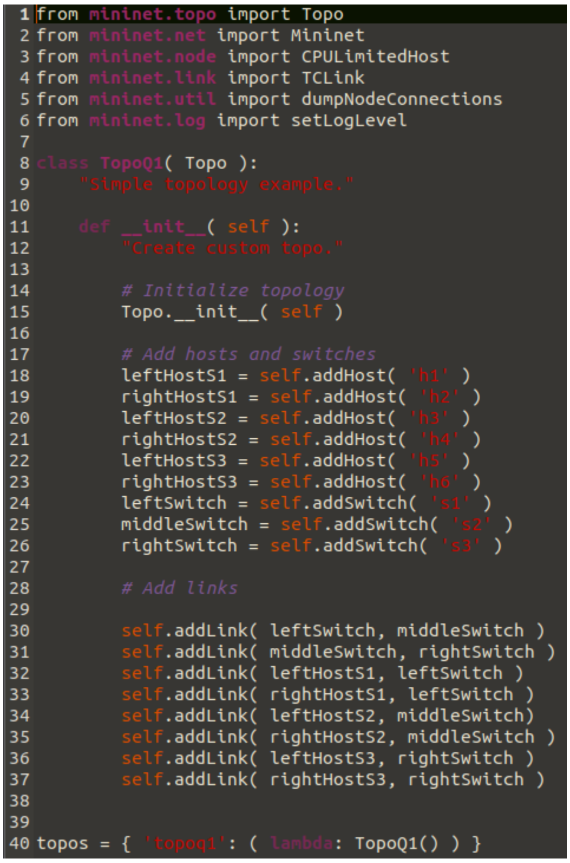
Charton Clément 2592257
Lemercier Nelson 2593887

Communication networks II

Mininet exercise

Task 1

a. Python script to make the topology :



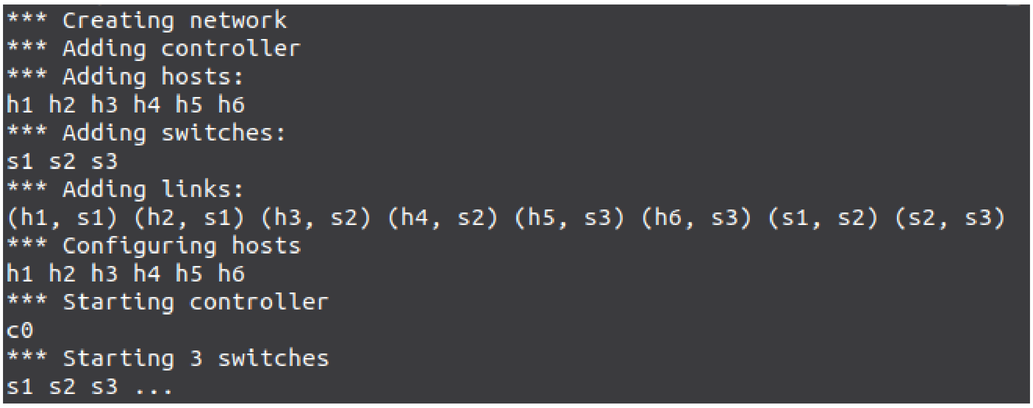
```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel

class TopoQ1( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHostS1 = self.addHost( 'h1' )
        rightHostS1 = self.addHost( 'h2' )
        leftHostS2 = self.addHost( 'h3' )
        rightHostS2 = self.addHost( 'h4' )
        leftHostS3 = self.addHost( 'h5' )
        rightHostS3 = self.addHost( 'h6' )
        leftSwitch = self.addSwitch( 's1' )
        middleSwitch = self.addSwitch( 's2' )
        rightSwitch = self.addSwitch( 's3' )

        # Add links

        self.addLink( leftSwitch, middleSwitch )
        self.addLink( middleSwitch, rightSwitch )
        self.addLink( leftHostS1, leftSwitch )
        self.addLink( rightHostS1, leftSwitch )
        self.addLink( leftHostS2, middleSwitch)
        self.addLink( rightHostS2, middleSwitch )
        self.addLink( leftHostS3, rightSwitch )
        self.addLink( rightHostS3, rightSwitch )


topos = { 'topoq1': ( lambda: TopoQ1() ) }
```

Command line to run this script :

```
sudo mn --custom ex1_CNII.py --topo topoq1
```

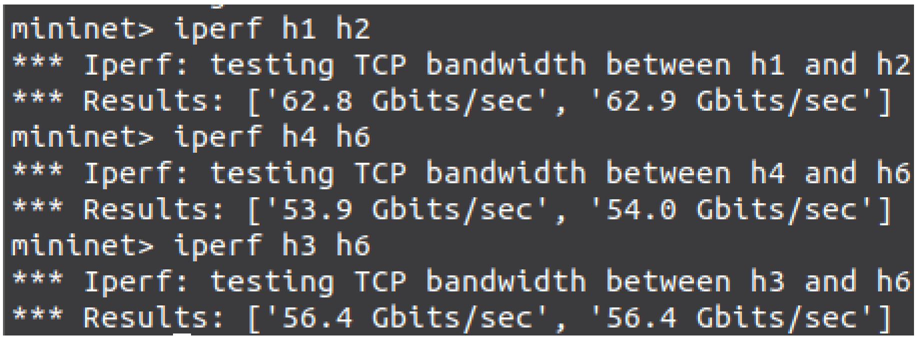
Here are the results of this command :

```
sudo mn --custom ex1_CNII.py --topo topoq1
```



```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s3) (h6, s3) (s1, s2) (s2, s3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
```

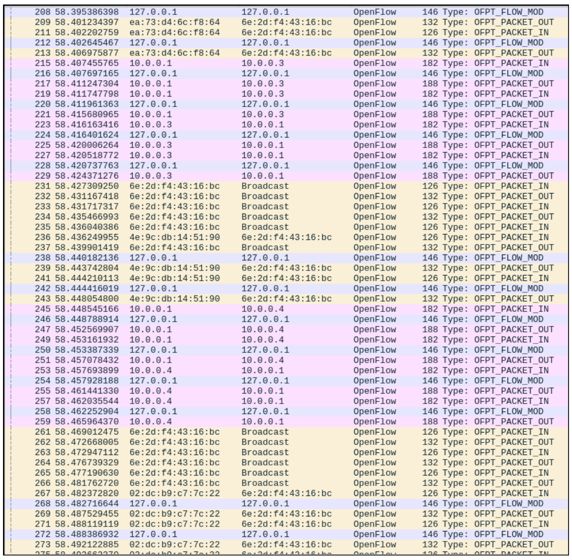
The bandwidth test performed with iperf give these results :



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['62.8 Gbits/sec', '62.9 Gbits/sec']
mininet> iperf h4 h6
*** Iperf: testing TCP bandwidth between h4 and h6
*** Results: ['53.9 Gbits/sec', '54.0 Gbits/sec']
mininet> iperf h3 h6
*** Iperf: testing TCP bandwidth between h3 and h6
*** Results: ['56.4 Gbits/sec', '56.4 Gbits/sec']
```

We have an average of 58 gigabits per second.

b. With wireshark, we can see the packet relating to mininet with the openflow protocol :

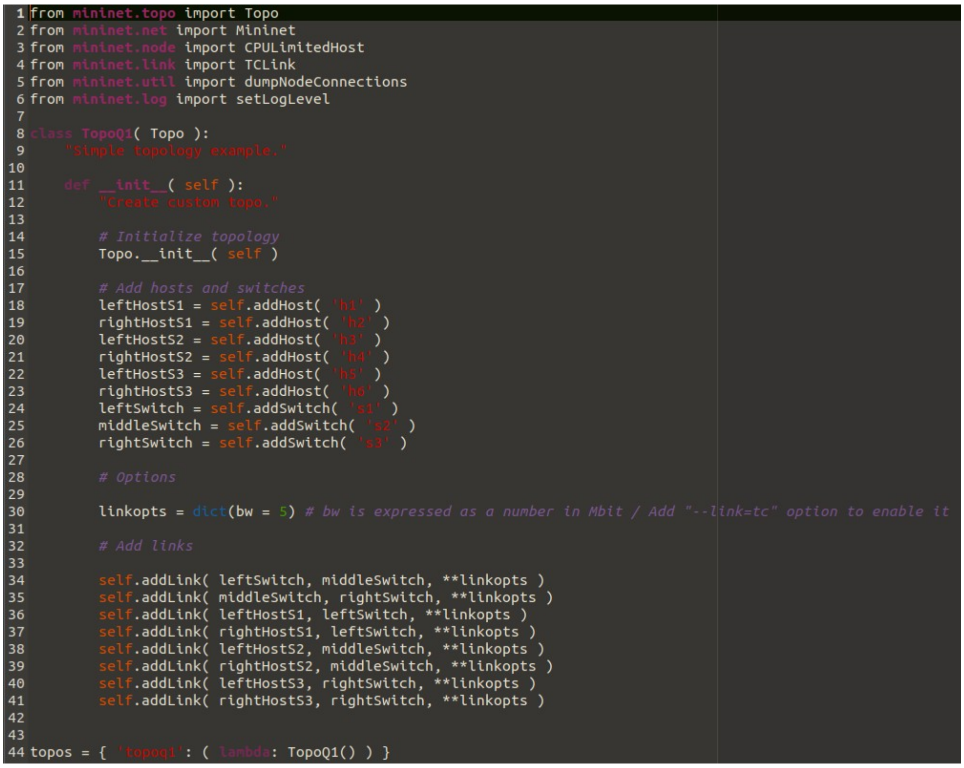


```
208 58.395386398  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
209 58.401234397  ea:73:d4:6c:f8:64  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
211 58.402202759  ea:73:d4:6c:f8:64  6e:2d:f4:43:16:bc  OpenFlow     126 Type: OFPT_PACKET_IN
212 58.402645467  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
213 58.406975877  ea:73:d4:6c:f8:64  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
215 58.407455765  10.0.0.1           10.0.0.3           OpenFlow     182 Type: OFPT_PACKET_IN
216 58.407697165  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
217 58.411247304  10.0.0.1           10.0.0.3           OpenFlow     188 Type: OFPT_PACKET_OUT
219 58.411747798  10.0.0.1           10.0.0.3           OpenFlow     182 Type: OFPT_PACKET_IN
220 58.411961363  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
221 58.415680965  10.0.0.1           10.0.0.3           OpenFlow     188 Type: OFPT_PACKET_OUT
223 58.416163416  10.0.0.3           10.0.0.1           OpenFlow     182 Type: OFPT_PACKET_IN
224 58.416401624  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
225 58.420006264  10.0.0.3           10.0.0.1           OpenFlow     188 Type: OFPT_PACKET_OUT
227 58.420518772  10.0.0.3           10.0.0.1           OpenFlow     182 Type: OFPT_PACKET_IN
228 58.420737763  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
229 58.424371276  10.0.0.3           10.0.0.1           OpenFlow     188 Type: OFPT_PACKET_OUT
231 58.427309250  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
232 58.431167418  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
233 58.431717317  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
234 58.435466993  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
235 58.436040386  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
236 58.436249955  4e:9c:db:14:51:90  6e:2d:f4:43:16:bc  OpenFlow     126 Type: OFPT_PACKET_IN
237 58.439901419  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
238 58.440182136  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
239 58.443742804  4e:9c:db:14:51:90  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
241 58.444210113  4e:9c:db:14:51:90  6e:2d:f4:43:16:bc  OpenFlow     126 Type: OFPT_PACKET_IN
242 58.444416019  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
243 58.448054800  4e:9c:db:14:51:90  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
245 58.448545166  10.0.0.1           10.0.0.4           OpenFlow     182 Type: OFPT_PACKET_IN
246 58.448788914  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
247 58.452569907  10.0.0.1           10.0.0.4           OpenFlow     188 Type: OFPT_PACKET_OUT
249 58.453161932  10.0.0.1           10.0.0.4           OpenFlow     182 Type: OFPT_PACKET_IN
250 58.453387339  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
251 58.457078432  10.0.0.1           10.0.0.4           OpenFlow     188 Type: OFPT_PACKET_OUT
253 58.457693899  10.0.0.4           10.0.0.1           OpenFlow     182 Type: OFPT_PACKET_IN
254 58.457928188  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
255 58.461441330  10.0.0.4           10.0.0.1           OpenFlow     188 Type: OFPT_PACKET_OUT
257 58.462035544  10.0.0.4           10.0.0.1           OpenFlow     182 Type: OFPT_PACKET_IN
258 58.462252904  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
259 58.465964370  10.0.0.4           10.0.0.1           OpenFlow     188 Type: OFPT_PACKET_OUT
261 58.469012475  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
262 58.472668005  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
263 58.472947112  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
264 58.476739329  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
265 58.477190630  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     126 Type: OFPT_PACKET_IN
266 58.481762720  6e:2d:f4:43:16:bc  Broadcast          OpenFlow     132 Type: OFPT_PACKET_OUT
267 58.482372820  02:dc:b9:c7:7c:22  6e:2d:f4:43:16:bc  OpenFlow     126 Type: OFPT_PACKET_IN
268 58.482716644  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
269 58.487529455  02:dc:b9:c7:7c:22  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
271 58.488119119  02:dc:b9:c7:7c:22  6e:2d:f4:43:16:bc  OpenFlow     126 Type: OFPT_PACKET_IN
272 58.488386932  127.0.0.1          127.0.0.1          OpenFlow     146 Type: OFPT_FLOW_MOD
273 58.492122885  02:dc:b9:c7:7c:22  6e:2d:f4:43:16:bc  OpenFlow     132 Type: OFPT_PACKET_OUT
```

c. To add a bandwidth limit, we modified our script so it behaves accordingly to the specifications.

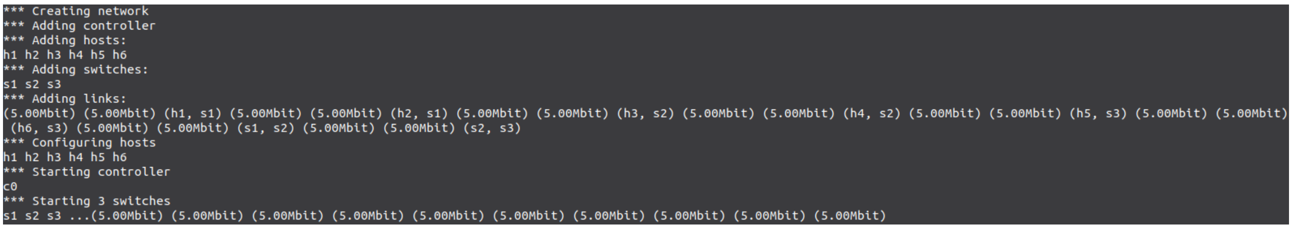
The command to run it is also modified :

sudo mn --custom ex1_CNII.py --topo topoq1 –link=tc



```python
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel

class TopoQ1( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHostS1 = self.addHost( 'h1' )
        rightHostS1 = self.addHost( 'h2' )
        leftHostS2 = self.addHost( 'h3' )
        rightHostS2 = self.addHost( 'h4' )
        leftHostS3 = self.addHost( 'h5' )
        rightHostS3 = self.addHost( 'h6' )
        leftSwitch = self.addSwitch( 's1' )
        middleSwitch = self.addSwitch( 's2' )
        rightSwitch = self.addSwitch( 's3' )

        # Options

        linkopts = dict(bw = 5) # bw is expressed as a number in Mbit / Add "--link=tc" option to enable it

        # Add links

        self.addLink( leftSwitch, middleSwitch, **linkopts )
        self.addLink( middleSwitch, rightSwitch, **linkopts )
        self.addLink( leftHostS1, leftSwitch, **linkopts )
        self.addLink( rightHostS1, leftSwitch, **linkopts )
        self.addLink( leftHostS2, middleSwitch, **linkopts )
        self.addLink( rightHostS2, middleSwitch, **linkopts )
        self.addLink( leftHostS3, rightSwitch, **linkopts )
        self.addLink( rightHostS3, rightSwitch, **linkopts )


topos = { 'topoq1': ( lambda: TopoQ1() ) }
```

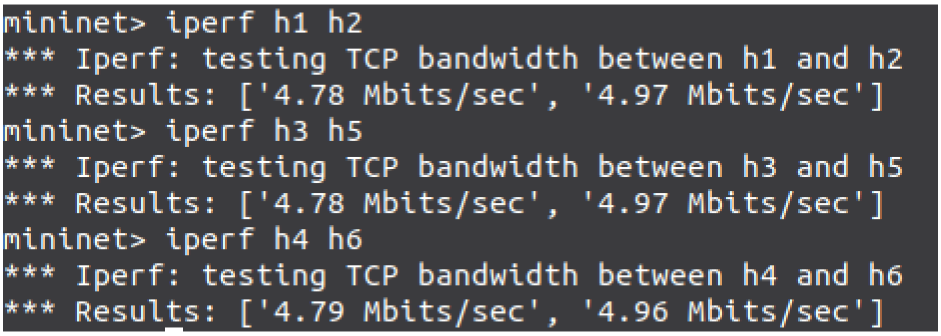
We can see that the badwidth value is taken into account when we lauch the script :



```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3
*** Adding links:
(5.00Mbit) (5.00Mbit) (h1, s1) (5.00Mbit) (5.00Mbit) (h2, s1) (5.00Mbit) (5.00Mbit) (h3, s2) (5.00Mbit) (5.00Mbit) (h4, s2) (5.00Mbit) (5.00Mbit) (h5, s3) (5.00Mbit) (5.00Mbit)
 (h6, s3) (5.00Mbit) (5.00Mbit) (s1, s2) (5.00Mbit) (5.00Mbit) (s2, s3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit) (5.00Mbit)
```

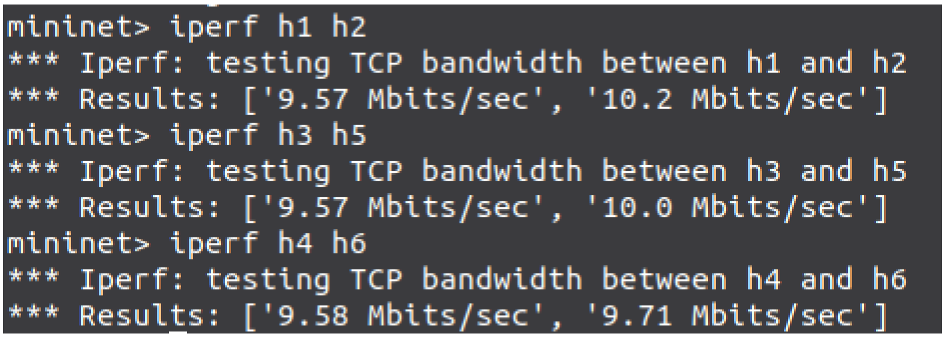
We can test the bandwidth with iperf once more for several values :

5 Mbit/s



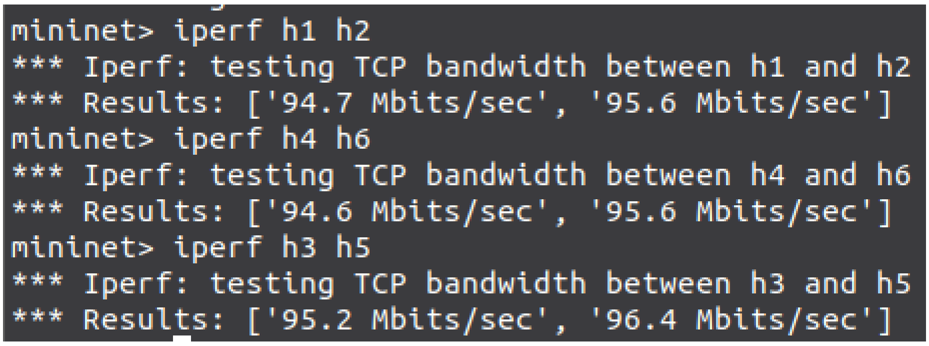
```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['4.78 Mbits/sec', '4.97 Mbits/sec']
mininet> iperf h3 h5
*** Iperf: testing TCP bandwidth between h3 and h5
*** Results: ['4.78 Mbits/sec', '4.97 Mbits/sec']
mininet> iperf h4 h6
*** Iperf: testing TCP bandwidth between h4 and h6
*** Results: ['4.79 Mbits/sec', '4.96 Mbits/sec']
```

10 Mbit/s



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.57 Mbits/sec', '10.2 Mbits/sec']
mininet> iperf h3 h5
*** Iperf: testing TCP bandwidth between h3 and h5
*** Results: ['9.57 Mbits/sec', '10.0 Mbits/sec']
mininet> iperf h4 h6
*** Iperf: testing TCP bandwidth between h4 and h6
*** Results: ['9.58 Mbits/sec', '9.71 Mbits/sec']
```

100 Mbit/s



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['94.7 Mbits/sec', '95.6 Mbits/sec']
mininet> iperf h4 h6
*** Iperf: testing TCP bandwidth between h4 and h6
*** Results: ['94.6 Mbits/sec', '95.6 Mbits/sec']
mininet> iperf h3 h5
*** Iperf: testing TCP bandwidth between h3 and h5
*** Results: ['95.2 Mbits/sec', '96.4 Mbits/sec']
```

We can see that the bandwidth limit is respected between hosts.

Task 2

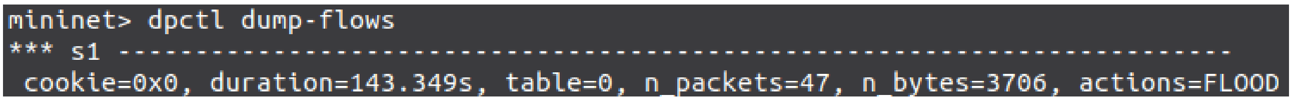
a) We launch the pox controller with the command

./pox.py pox.forwarding.hub

And then launch a simple mininet command :

sudo mn --topo single,3 --mac --controller remote --switch ovsk

We have these results for the dpctl dump-flows command :

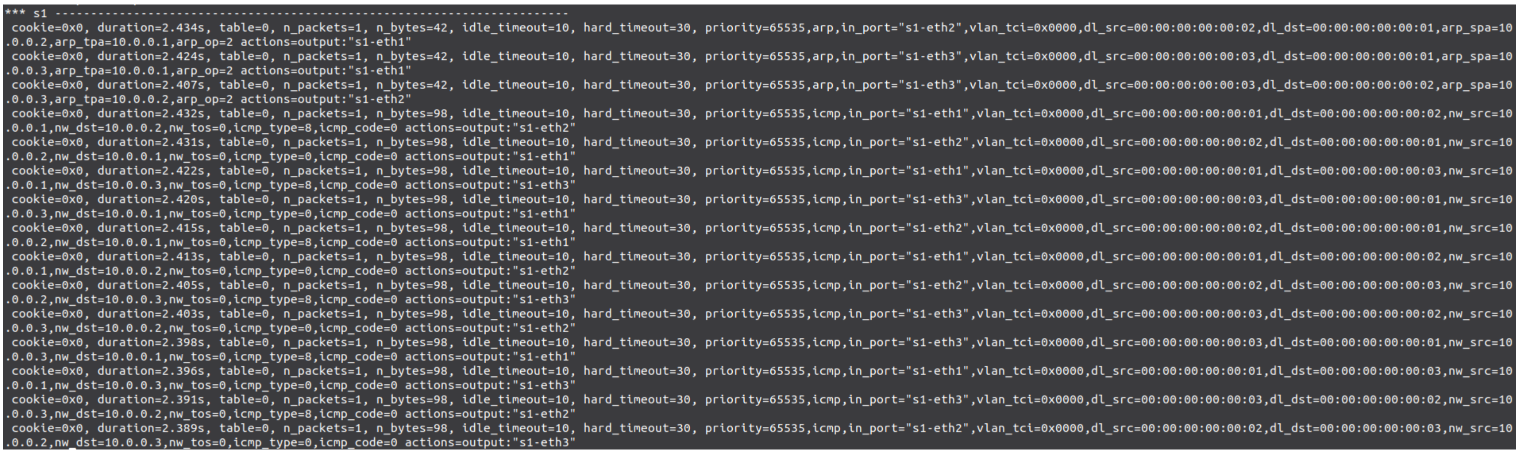


```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=143.349s, table=0, n_packets=47, n_bytes=3706, actions=FLOOD
```

b) This time, we lauch the controller with another command :

./pox.py pox.forwarding.l2_learning

These are the results of dpctl dump-flows :

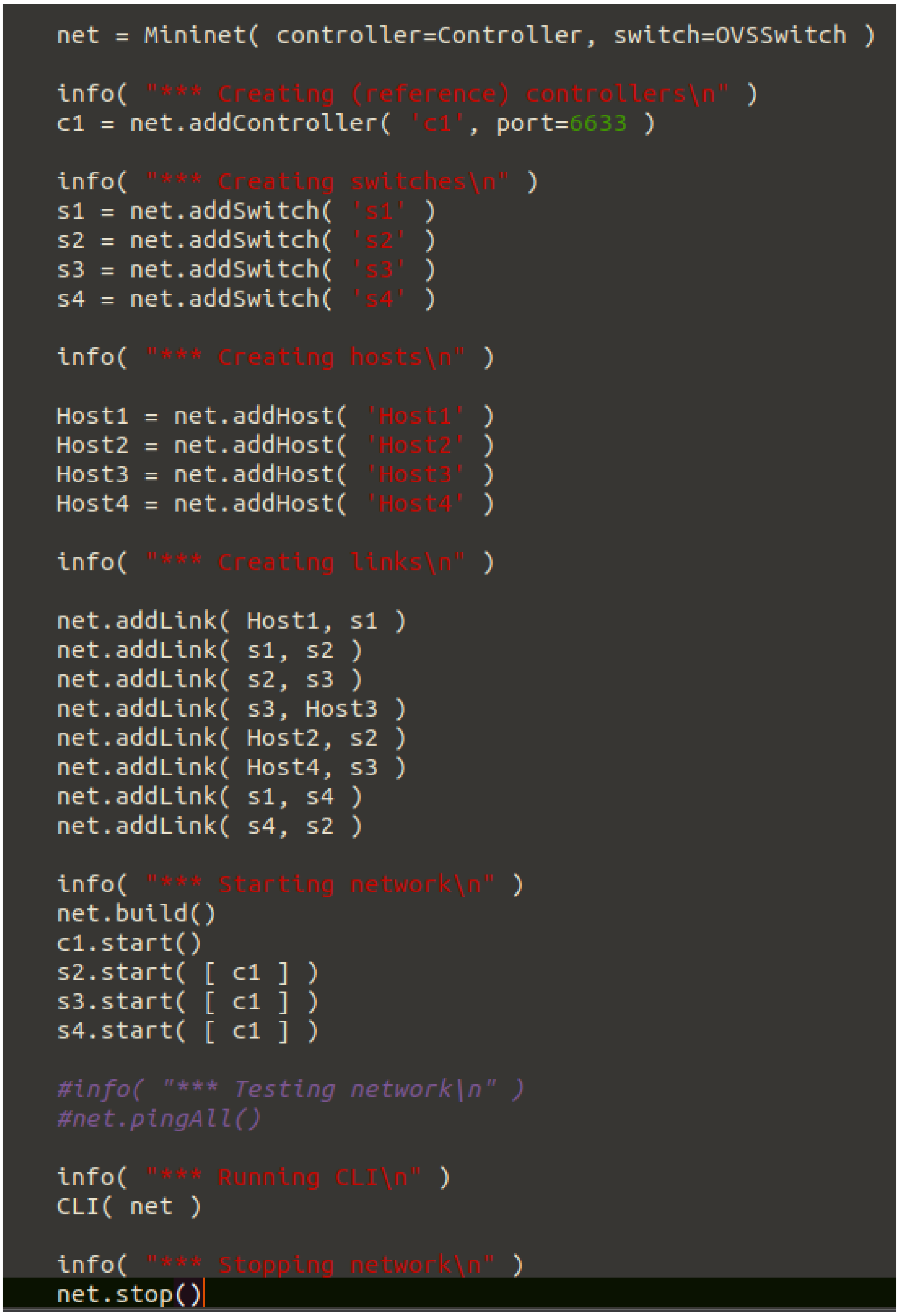


```
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=2.434s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, priority=65535,arp,in_port="s1-eth2",vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=10
.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:"s1-eth1"
 cookie=0x0, duration=2.424s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, priority=65535,arp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,arp_spa=10
.0.0.3,arp_tpa=10.0.0.1,arp_op=2 actions=output:"s1-eth1"
 cookie=0x0, duration=2.407s, table=0, n_packets=1, n_bytes=42, idle_timeout=10, hard_timeout=30, priority=65535,arp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02,arp_spa=10
.0.0.3,arp_tpa=10.0.0.2,arp_op=2 actions=output:"s1-eth2"
 cookie=0x0, duration=2.432s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10
.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=2.431s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10
.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=2.422s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10
.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth3"
 cookie=0x0, duration=2.420s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10
.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=2.415s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10
.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=2.413s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10
.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=2.405s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03,nw_src=10
.0.0.2,nw_dst=10.0.0.3,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth3"
 cookie=0x0, duration=2.403s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02,nw_src=10
.0.0.3,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=2.398s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10
.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth1"
 cookie=0x0, duration=2.396s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10
.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth3"
 cookie=0x0, duration=2.391s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02,nw_src=10
.0.0.3,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s1-eth2"
 cookie=0x0, duration=2.389s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03,nw_src=10
.0.0.2,nw_dst=10.0.0.3,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s1-eth3"
```

c) The hub controller only forward packets without any control on the flows so the table is very small. The learning hub is keeping many informations about the flows in this tables to learn the flow and adapt to it so there are much more data.

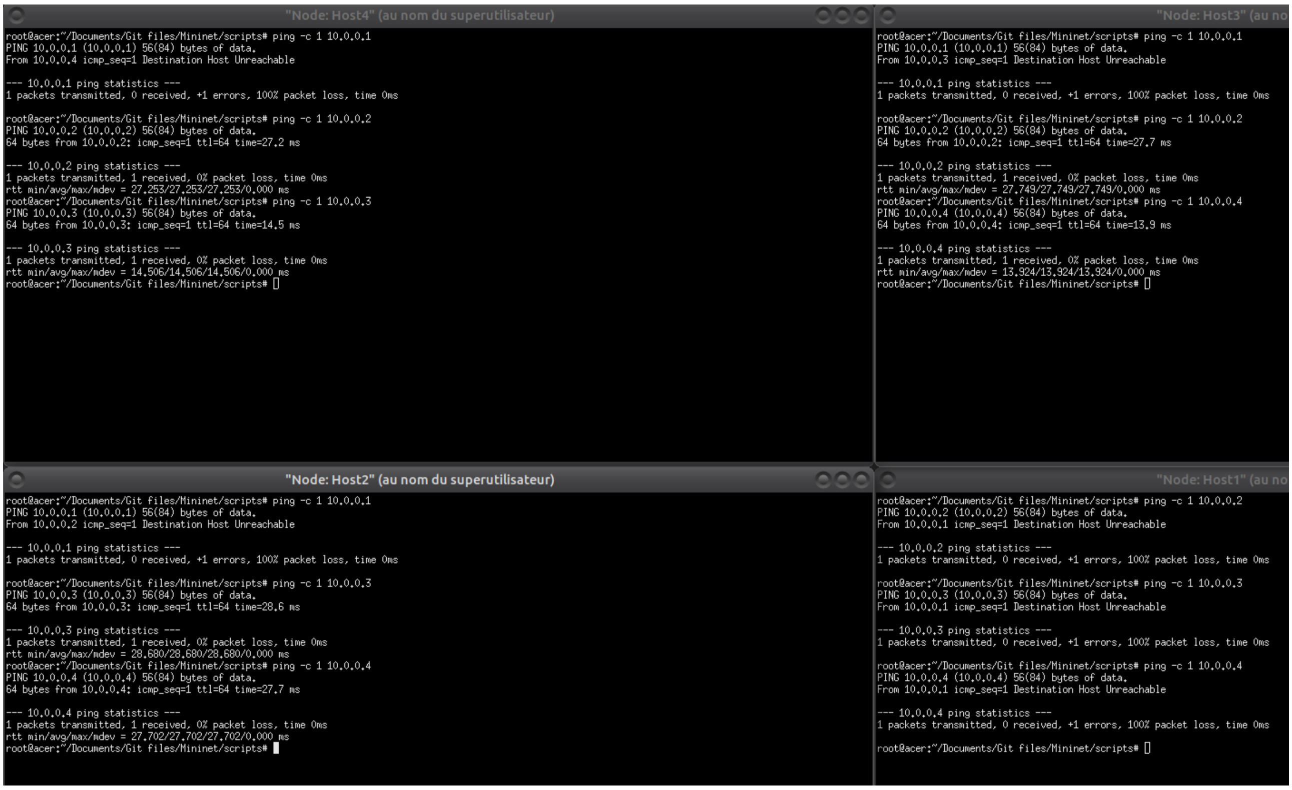
Task 3

To link the controller to three switches out of four, we made a script that includes not only the topology but also the mininet network building.



```
net = Mininet( controller=Controller, switch=OVSSwitch )

info( "*** Creating (reference) controllers\n" )
c1 = net.addController( 'c1', port=6633 )

info( "*** Creating switches\n" )
s1 = net.addSwitch( 's1' )
s2 = net.addSwitch( 's2' )
s3 = net.addSwitch( 's3' )
s4 = net.addSwitch( 's4' )

info( "*** Creating hosts\n" )

Host1 = net.addHost( 'Host1' )
Host2 = net.addHost( 'Host2' )
Host3 = net.addHost( 'Host3' )
Host4 = net.addHost( 'Host4' )

info( "*** Creating links\n" )

net.addLink( Host1, s1 )
net.addLink( s1, s2 )
net.addLink( s2, s3 )
net.addLink( s3, Host3 )
net.addLink( Host2, s2 )
net.addLink( Host4, s3 )
net.addLink( s1, s4 )
net.addLink( s4, s2 )

info( "*** Starting network\n" )
net.build()
c1.start()
s2.start( [ c1 ] )
s3.start( [ c1 ] )
s4.start( [ c1 ] )

#info( "*** Testing network\n" )
#net.pingAll()

info( "*** Running CLI\n" )
CLI( net )

info( "*** Stopping network\n" )
net.stop()
```

If we test to ping every host, we can notice that host 1 is unreachable.


"Node: Host4" (au nom du superutilisateur)
"Node: Host3" (au no
"Node: Host2" (au nom du superutilisateur)
"Node: Host1" (au no

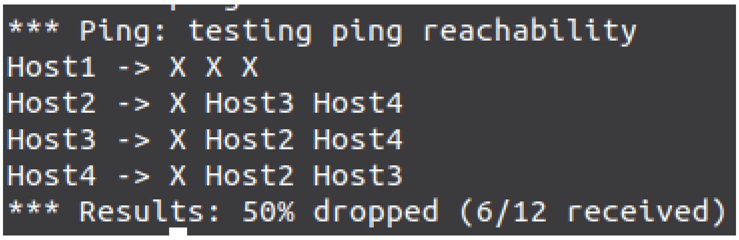
```
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.4 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.2
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=27.2 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 27.253/27.253/27.253/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=14.5 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.506/14.506/14.506/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts#
```

```
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.3 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.2
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=27.7 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 27.749/27.749/27.749/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=13.9 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.924/13.924/13.924/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts#
```

```
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=28.6 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 28.680/28.680/28.680/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=27.7 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 27.702/27.702/27.702/0.000 ms
root@acer:~/Documents/Git files/Mininet/scripts#
```

```
root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts# ping -c 1 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@acer:~/Documents/Git files/Mininet/scripts#
```

That is even more blatant when we make a pingall test :

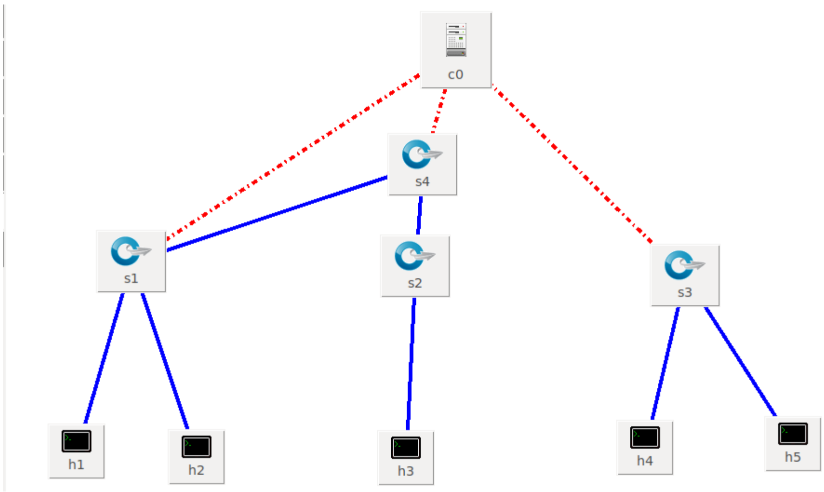


```
*** Ping: testing ping reachability
Host1 -> X X X
Host2 -> X Host3 Host4
Host3 -> X Host2 Host4
Host4 -> X Host2 Host3
*** Results: 50% dropped (6/12 received)
```

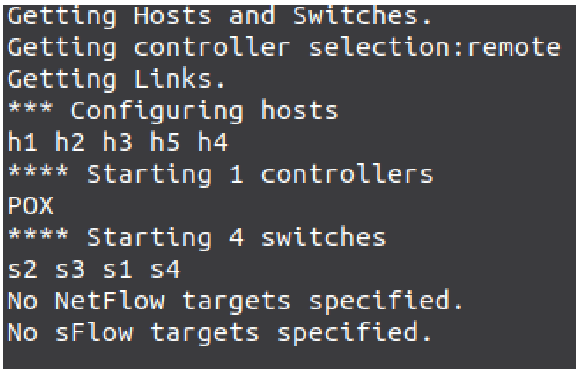
We can understand this result because the switch 1 is not connected to the controller.

Task 4

Here is our topology displayed on miniedit.

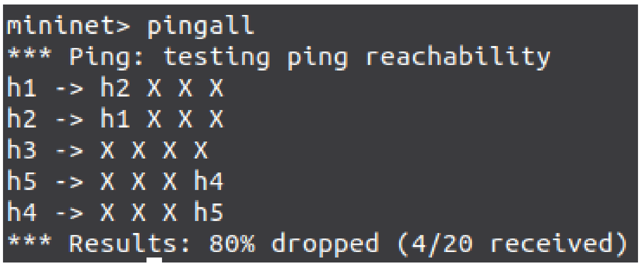


We can see that mininet is starting our five hosts, four switches and the pox controller.



```
Getting Hosts and Switches.
Getting controller selection:remote
Getting Links.
*** Configuring hosts
h1 h2 h3 h5 h4
**** Starting 1 controllers
POX
**** Starting 4 switches
s2 s3 s1 s4
No NetFlow targets specified.
No sFlow targets specified.
```

By running a pingall test, we have a result of 4 out of 20 due to s4 and s2 not connected to s3.



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X X
h2 -> h1 X X X
h3 -> X X X X
h5 -> X X X h4
h4 -> X X X h5
*** Results: 80% dropped (4/20 received)
```