Nama Kelompok : **Laperbaper**

Nama Anggota : 1. Dewi Safrida (161402008)

2. Khairunnisa Sitanggang (161402014)

3. Bora Sejati Siboro (161402020)

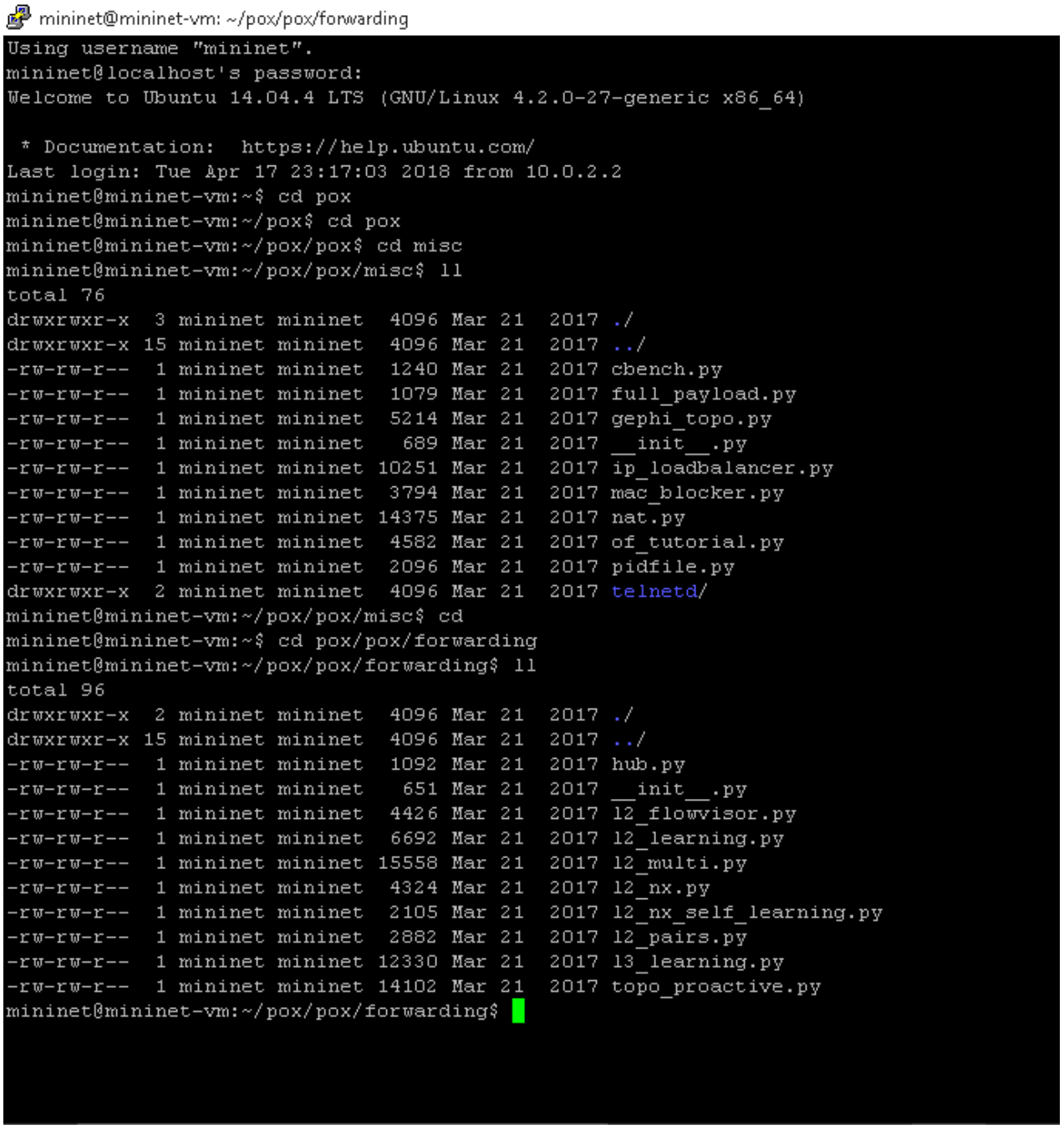
4. Haryati (161402047)

5. Mayang Dyah Azurah (161402050)

POX CONTROLLER

POX controller adalah salah satu controller SDN yang mendukung protokol OpenFlow. POX adalah controller yang berbasis bahasa Python. Tahapannya sebagai berikut.

1. Melakukan POX Controller files pada Mininet



2. Memilih of_tutorial.py pada pox maupun pox dan misc

mininet@mininet-vm: ~/pox/pox/misc

```
# Copyright 2012 James McCauley
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at:
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

"""
This component is for use with the OpenFlow tutorial.

It acts as a simple hub, but can be modified to act like an L2
learning switch.

It's roughly similar to the one Brandon Heller did for NOX.
"""

from pox.core import core
import pox.openflow.libopenflow_01 as of

log = core.getLogger()



class Tutorial (object):
  """
  A Tutorial object is created for each switch that connects.
  A Connection object for that switch is passed to the __init__ function.
  """
  def __init__ (self, connection):
    # Keep track of the connection to the switch so that we can
    # send it messages!
    self.connection = connection

    # This binds our PacketIn event listener
    connection.addListeners(self)

    # Use this table to keep track of which ethernet address is on
    # which switch port (keys are MACs, values are ports).
    self.mac_to_port = {}


  def resend_packet (self, packet_in, out_port):
    """
    Instructs the switch to resend a packet that it had sent to us.
    "packet_in" is the ofp_packet_in object the switch had sent to the
```

3. memulai pox controller dengan '**./pox.py log.level --DEBUG misc.of_tutorial pada directory pox**'. Kemudian mengetikkan lagi perintah **sudo mn --topo single,3 --mac --witch ovsk --controller remote**'.

4. Setelah itu, ketikkan perintah sebagai berikut **'tcpdump -XX -n -i h2-eth0 > h2.txt'** pada terminal node 2, dan **'tcpdump -XX -n -i h3-eth0 > h3.txt'** pada terminal node 3. Lalu lakukan ping pada terminal node 1 dengan perintah **'ping -c1 10.0.0.2'**

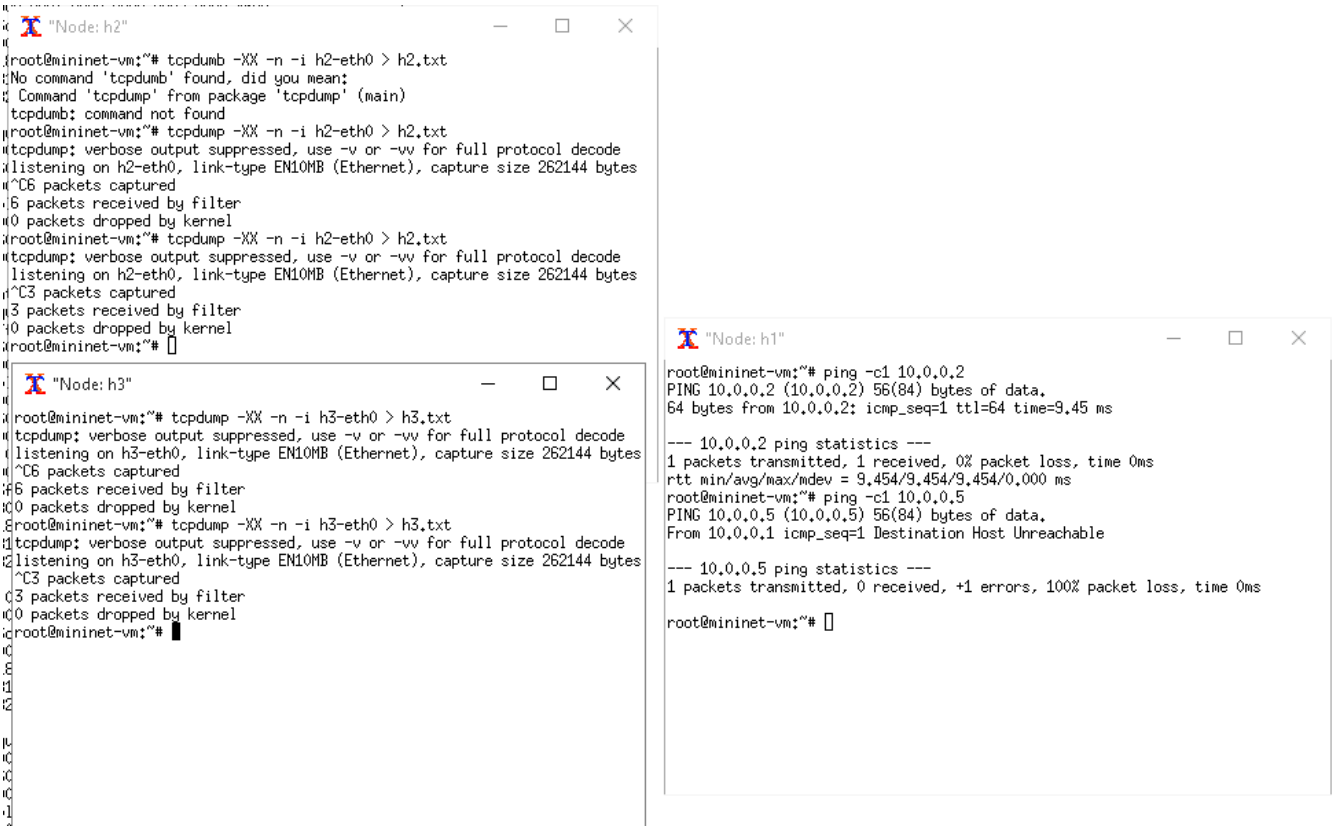
```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ cat h2.txt
08:39:34.828743 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0002                 ..........
08:39:34.828766 ARP, Reply 10.0.0.2 is-at 00:00:00:00:00:02, length 28
        0x0000:  0000 0000 0001 0000 0000 0002 0806 0001  ................
        0x0010:  0800 0604 0002 0000 0000 0002 0a00 0002  ................
        0x0020:  0000 0000 0001 0a00 0001                 ..........
08:39:34.831160 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 5099, seq 1, lengt
h 64
        0x0000:  0000 0000 0002 0000 0000 0001 0800 4500  ..............E.
        0x0010:  0054 13fb 4000 4001 12ac 0a00 0001 0a00  .T..@.@.........
        0x0020:  0002 0800 69f2 13eb 0001 b666 d75a 0000  ....i......f.Z..
        0x0030:  0000 218d 0c00 0000 0000 1011 1213 1415  ..!.............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
08:39:34.831393 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 5099, seq 1, length
64
        0x0000:  0000 0000 0001 0000 0000 0002 0800 4500  ..............E.
        0x0010:  0054 46d0 0000 4001 1fd7 0a00 0002 0a00  .TF...@.........
        0x0020:  0001 0000 71f2 13eb 0001 b666 d75a 0000  ....q......f.Z..
        0x0030:  0000 218d 0c00 0000 0000 1011 1213 1415  ..!.............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
08:39:39.835283 ARP, Request who-has 10.0.0.1 tell 10.0.0.2, length 28
        0x0000:  0000 0000 0001 0000 0000 0002 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0002 0a00 0002  ................
        0x0020:  0000 0000 0000 0a00 0001                 ..........
08:39:39.883178 ARP, Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28
        0x0000:  0000 0000 0002 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0002 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0002 0a00 0002                 ..........

mininet@mininet-vm:~$
```

```
mininet@mininet-vm:~$ cat h3.txt
08:39:34.828740 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0002                 ..........
08:39:34.829769 ARP, Reply 10.0.0.2 is-at 00:00:00:00:00:02, length 28
        0x0000:  0000 0000 0001 0000 0000 0002 0806 0001  ................
        0x0010:  0800 0604 0002 0000 0000 0002 0a00 0002  ................
        0x0020:  0000 0000 0001 0a00 0001                 ..........
08:39:34.831159 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 5099, seq 1, length 64
        0x0000:  0000 0000 0002 0000 0000 0001 0800 4500  ..............E.
        0x0010:  0054 13fb 4000 4001 12ac 0a00 0001 0a00  .T..@.@.........
        0x0020:  0002 0800 69f2 13eb 0001 b666 d75a 0000  ....i......f.Z..
        0x0030:  0000 218d 0c00 0000 0000 1011 1213 1415  ..!.............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
08:39:34.832014 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 5099, seq 1, length 64
        0x0000:  0000 0000 0001 0000 0000 0002 0800 4500  ..............E.
        0x0010:  0054 46d0 0000 4001 1fd7 0a00 0002 0a00  .TF...@.........
        0x0020:  0001 0000 71f2 13eb 0001 b666 d75a 0000  ....q......f.Z..
        0x0030:  0000 218d 0c00 0000 0000 1011 1213 1415  ..!.............
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
08:39:39.877184 ARP, Request who-has 10.0.0.1 tell 10.0.0.2, length 28
        0x0000:  0000 0000 0001 0000 0000 0002 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0002 0a00 0002  ................
        0x0020:  0000 0000 0000 0a00 0001                 ..........
08:39:39.883173 ARP, Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28
        0x0000:  0000 0000 0002 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0002 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0002 0a00 0002                 ..........

mininet@mininet-vm:~$
```

5. Mengetikkan kembali perintah untuk merekam aktivitas node lagi pada node 2 dan 3. Untuk lebih jelasnya, kita bisa menyaksikan gambar dibawah ini



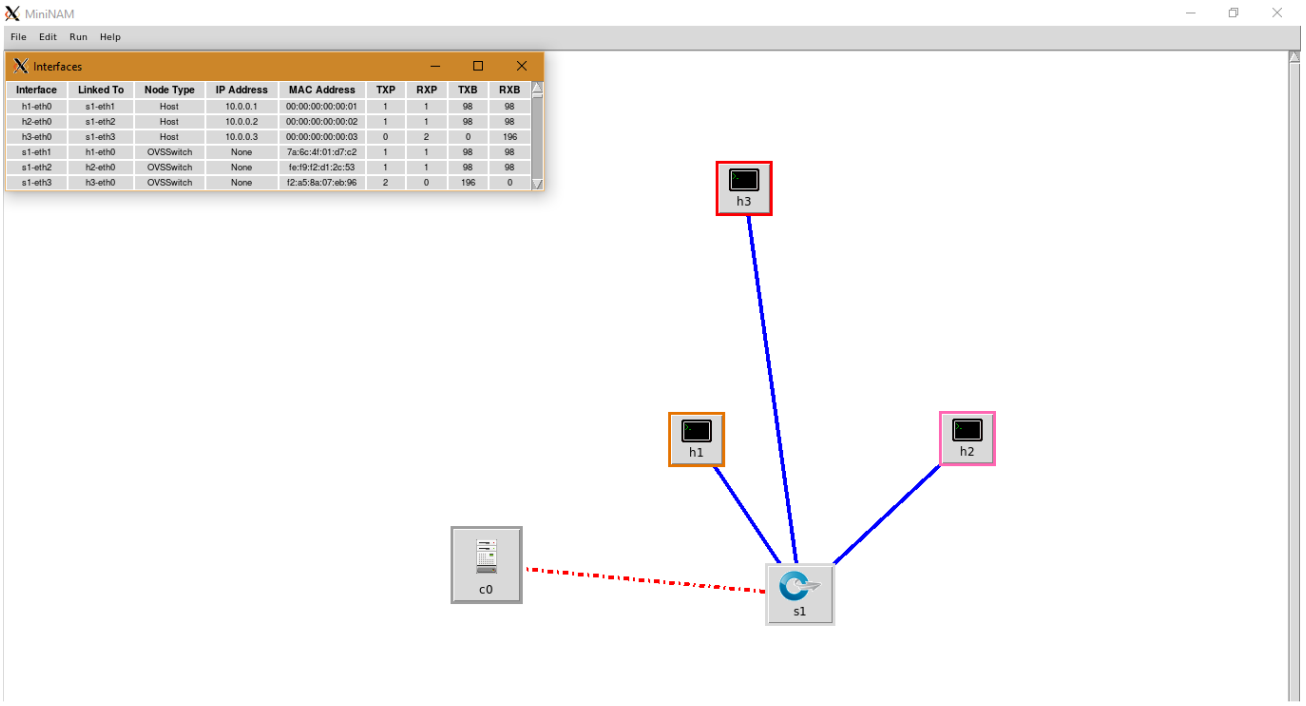

```
mininet@mininet-vm:~$ cat h2.txt
08:49:13.332689 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........
08:49:14.331482 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........
08:49:15.332863 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........

mininet@mininet-vm:~$
```

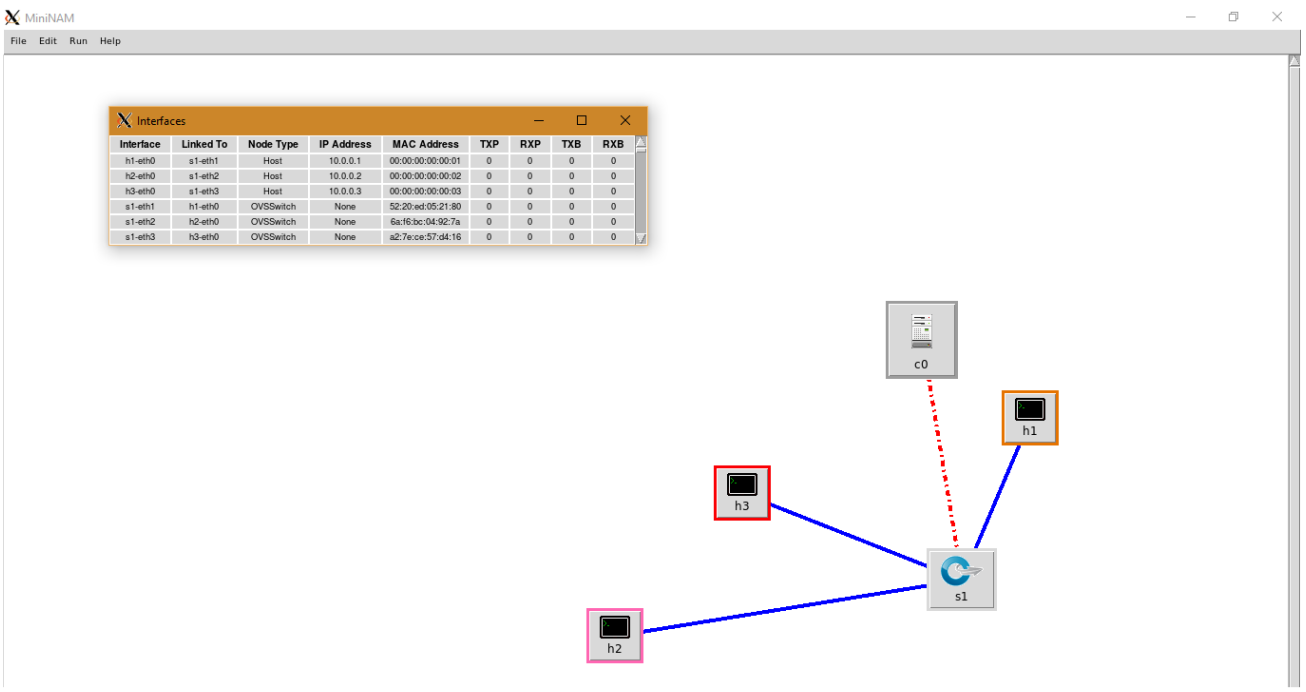
```
mininet@mininet-vm:~$ cat h3.txt
08:49:13.332686 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........
08:49:14.331478 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........
08:49:15.332860 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........

mininet@mininet-vm:~$
```

6. Tampilan jaringan yg kita buat tadi pada MiniNAM ketika ping dari 10.0.0.1 ke 10.0.0.2. terlihat ada komunikasi antara host 1 dan 2.



7. Tampilan jaringan yg kita buat tadi pada MiniNAM ketika ping dari 10.0.0.1 ke 10.0.0.5. terlihat tidak ada komunikasi antar host.



8. disini kita akan melakukan perbandingan antara kecepatan menggunakan hub dan switch. berikan perintah iperf pada console.

```
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
mininet> pingpair
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['6.61 Mbits/sec', '7.32 Mbits/sec']
mininet>
```

9. sehingga dapat kita simpulkan bahwa switch dapat bekerja lebih cepat daripada hub