ADMINISTRASI DAN DESAIN JARINGAN

MININET, MININAM, OPENFLOW, POXCONTROLLER

LAPORAN

DISUSUN OLEH

DESI TAMBUNAN	**161402005**
SONIA ELISA TELAUMBANUA	**161402045**
NOVALINA SIMBOLON	**161402077**
CHYNTIA CLAUDIA	**161402101**
TIRZA PRISKILA KINANTI SIBUEA	**161402110**

KOM B

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

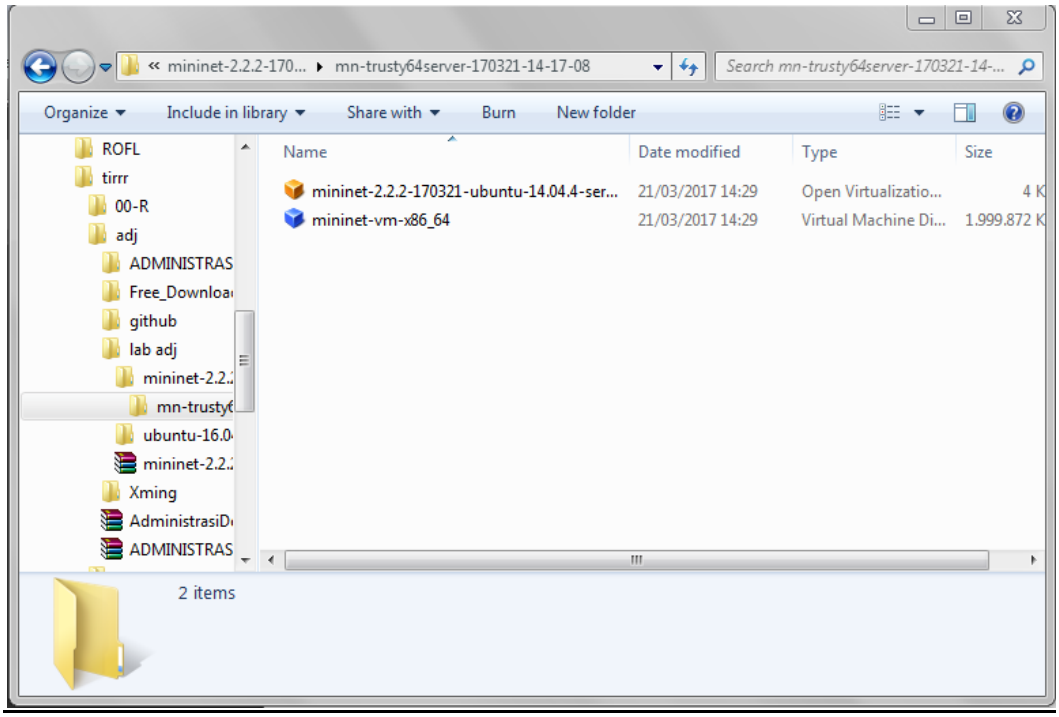
2018

MININET

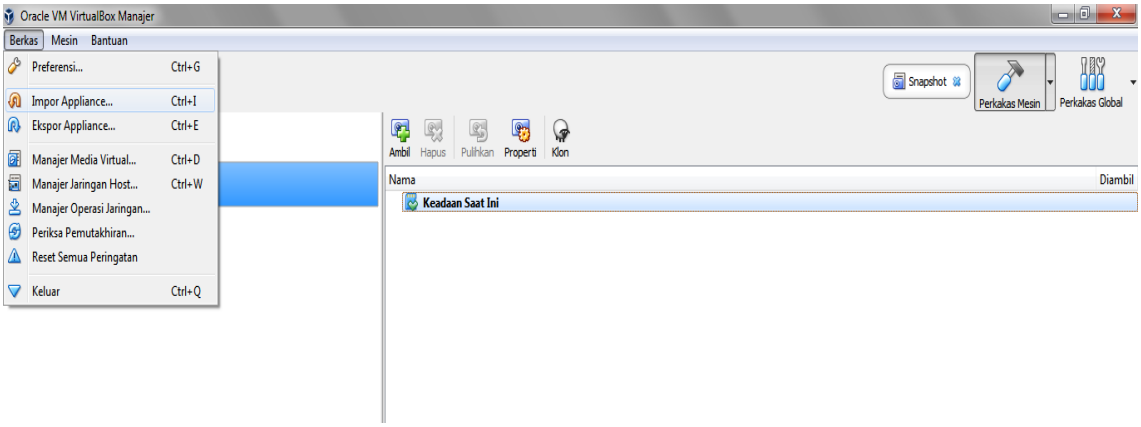
Mininet adalah emulator berbasis CLI yang digunakan untuk membuat sebuah topologi jaringan pada *Software Defined Network*. Dalam Mininet, terdapat beberapa topologi bawaan yang dapat langsung digunakan dengan menggunakan perintah tertentu, antara lain seperti topologi *single*, *tree*, dan *linear*.

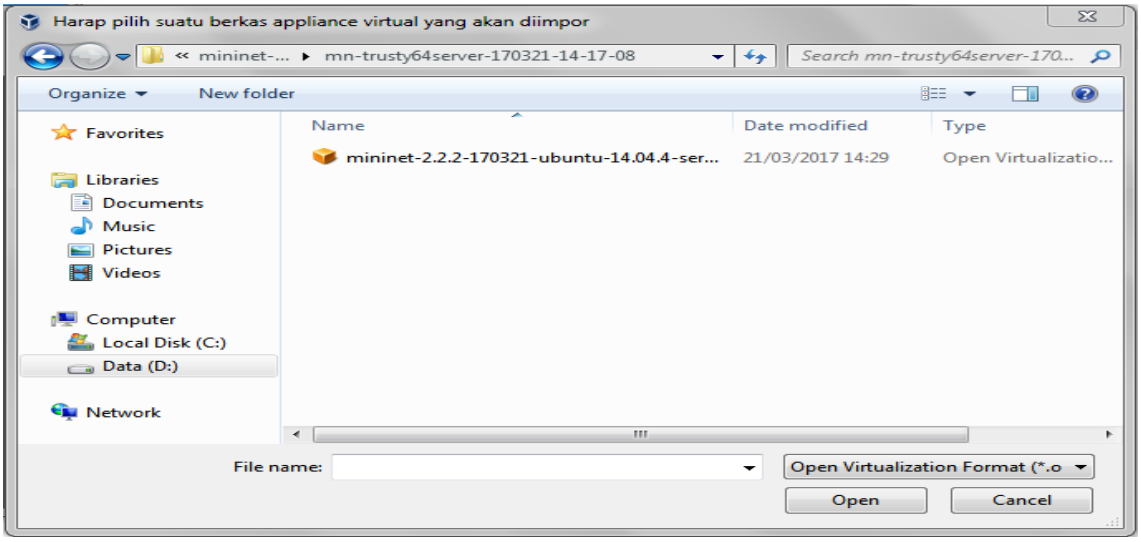
Download dan *Import* Mininet pada *Virtual Machine* (VM)

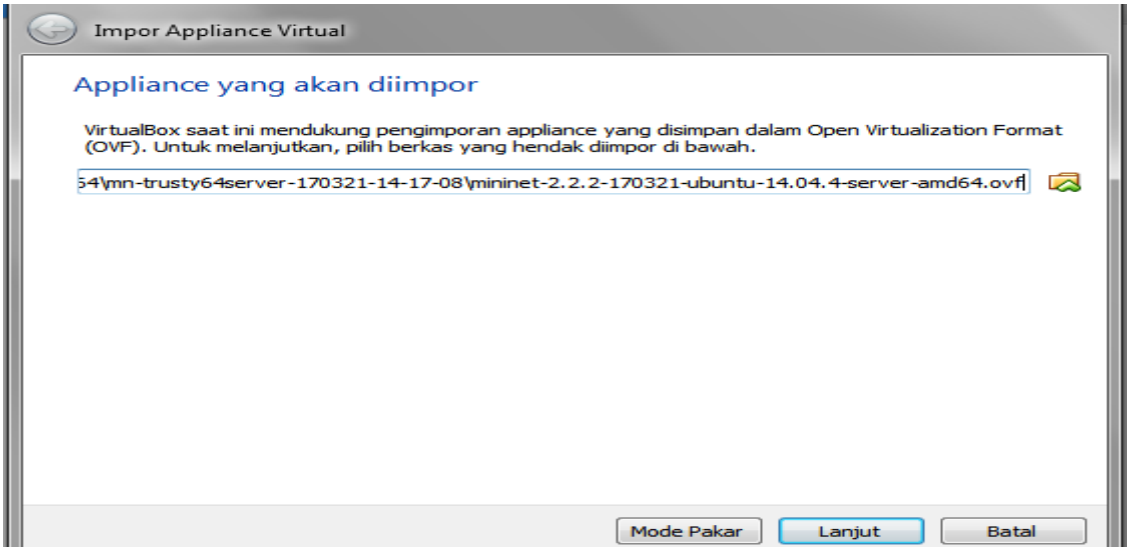
1. Download Mininet

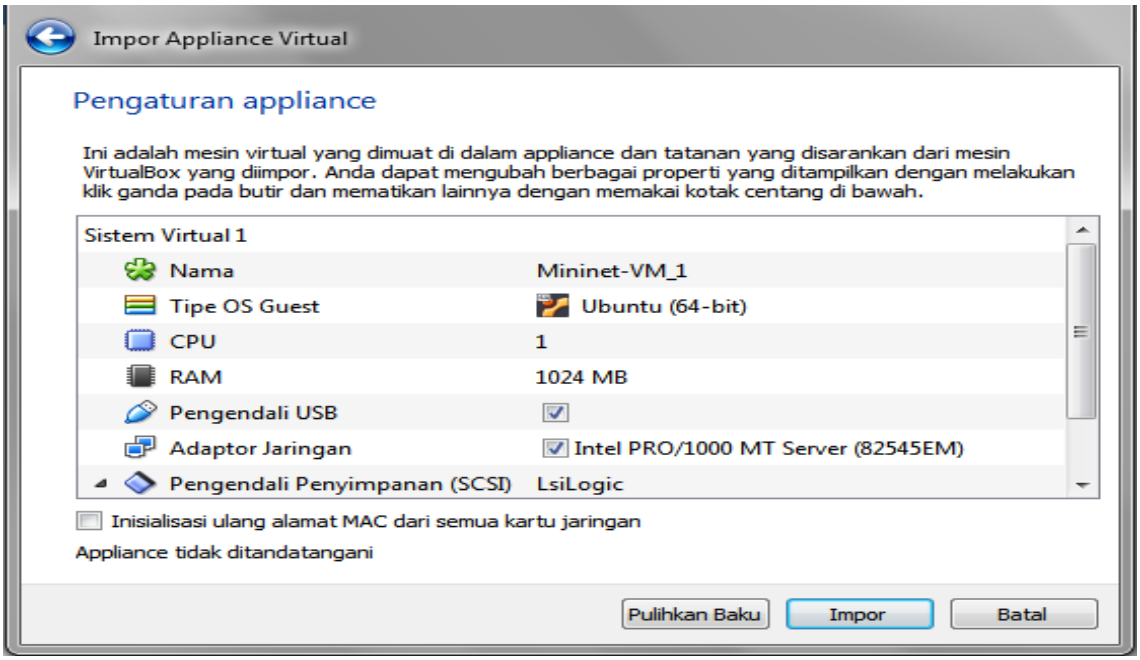


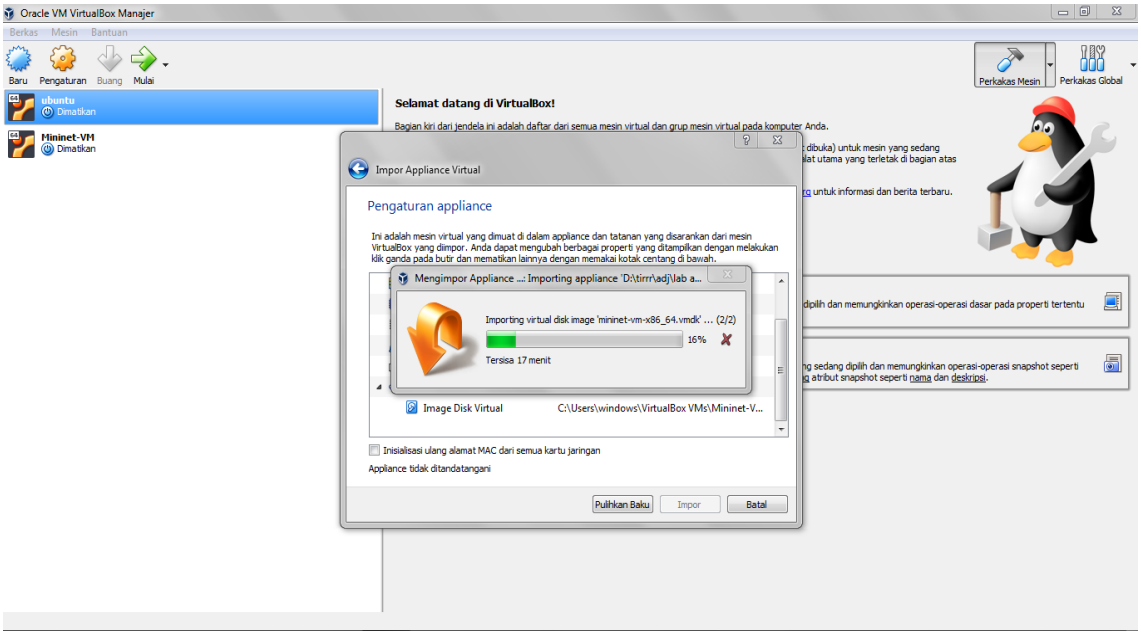
2. Import Mininet pada *Virtual Machine* (*Virtual Box*)



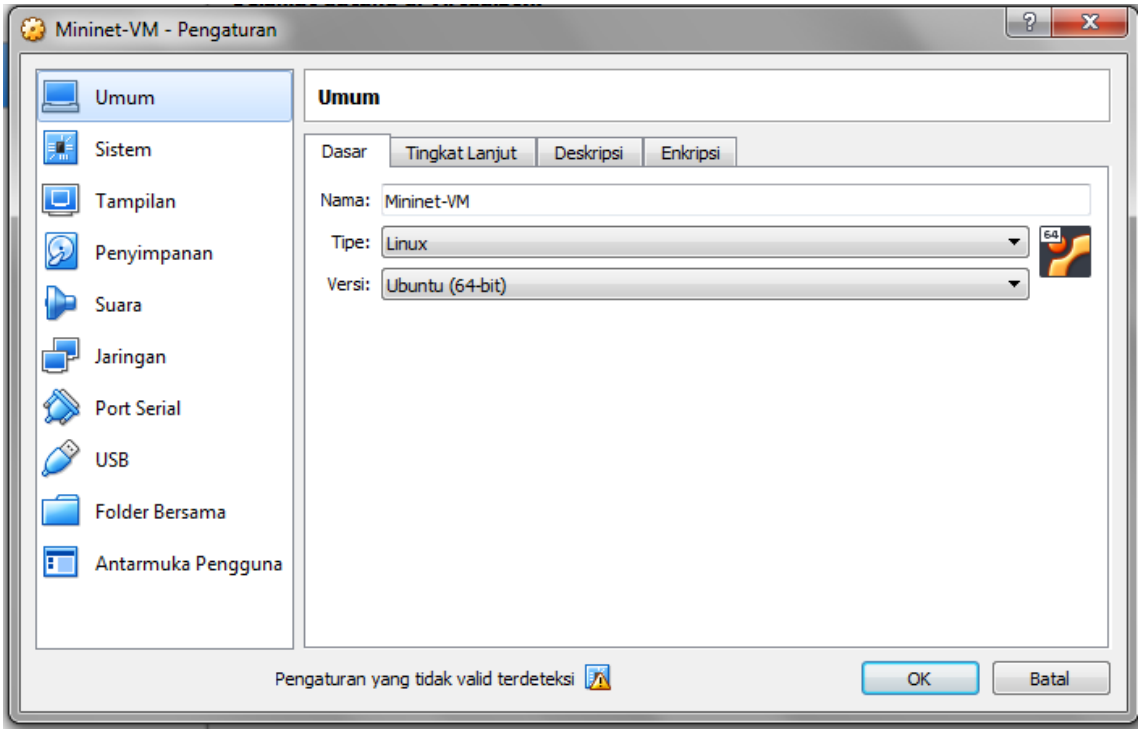

Harap pilih suatu berkas appliance virtual yang akan diimpor
mininet-... ▸ mn-trusty64server-170321-14-17-08
Search mn-trusty64server-170...
Organize
New folder
Favorites
Libraries
Documents
Music
Pictures
Videos
Computer
Local Disk (C:)
Data (D:)
Network
Name
Date modified
Type
mininet-2.2.2-170321-ubuntu-14.04.4-ser...
21/03/2017 14:29
Open Virtualizatio...
File name:
Open Virtualization Format (*.o
Open
Cancel

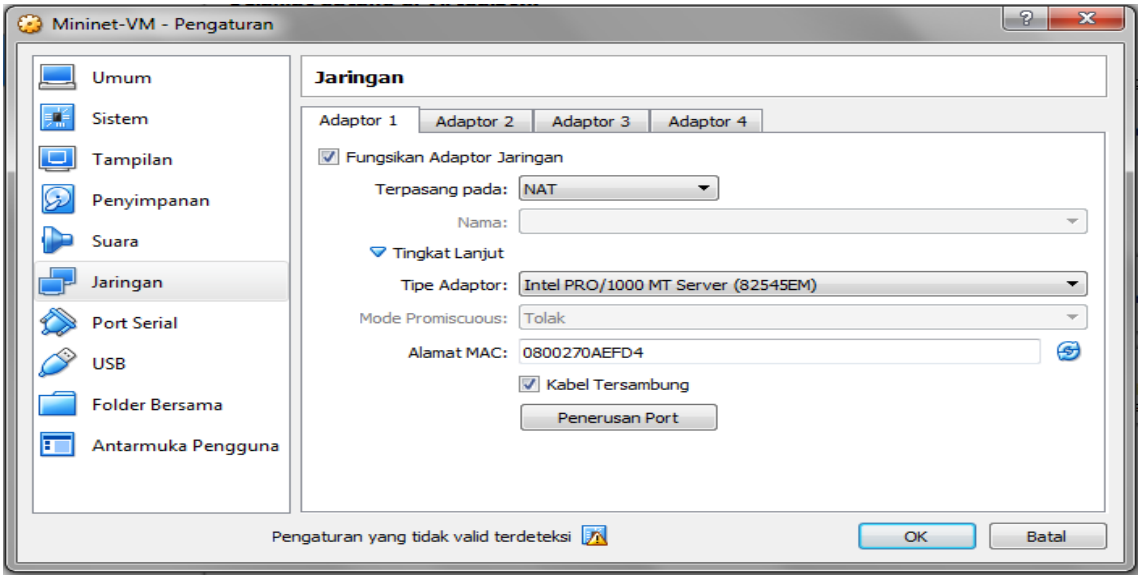

Impor Appliance Virtual
Appliance yang akan diimpor
VirtualBox saat ini mendukung pengimporan appliance yang disimpan dalam Open Virtualization Format (OVF). Untuk melanjutkan, pilih berkas yang hendak diimpor di bawah.
54\mn-trusty64server-170321-14-17-08\mininet-2.2.2-170321-ubuntu-14.04.4-server-amd64.ovf
Mode Pakar
Lanjut
Batal

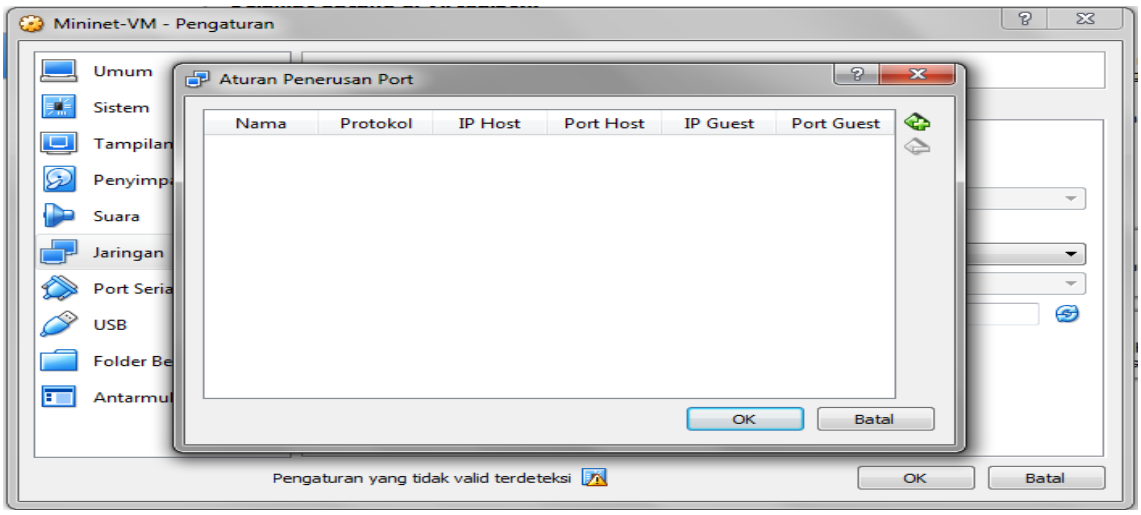

Impor Appliance Virtual
Pengaturan appliance
Ini adalah mesin virtual yang dimuat di dalam appliance dan tatanan yang disarankan dari mesin VirtualBox yang diimpor. Anda dapat mengubah berbagai properti yang ditampilkan dengan melakukan klik ganda pada butir dan mematikan lainnya dengan memakai kotak centang di bawah.
Sistem Virtual 1
Nama
Mininet-VM_1
Tipe OS Guest
Ubuntu (64-bit)
CPU
1
RAM
1024 MB
Pengendali USB
Adaptor Jaringan
Intel PRO/1000 MT Server (82545EM)
Pengendali Penyimpanan (SCSI)
LsiLogic
Inisialisasi ulang alamat MAC dari semua kartu jaringan
Appliance tidak ditandatangani
Pulihkan Baku
Impor
Batal

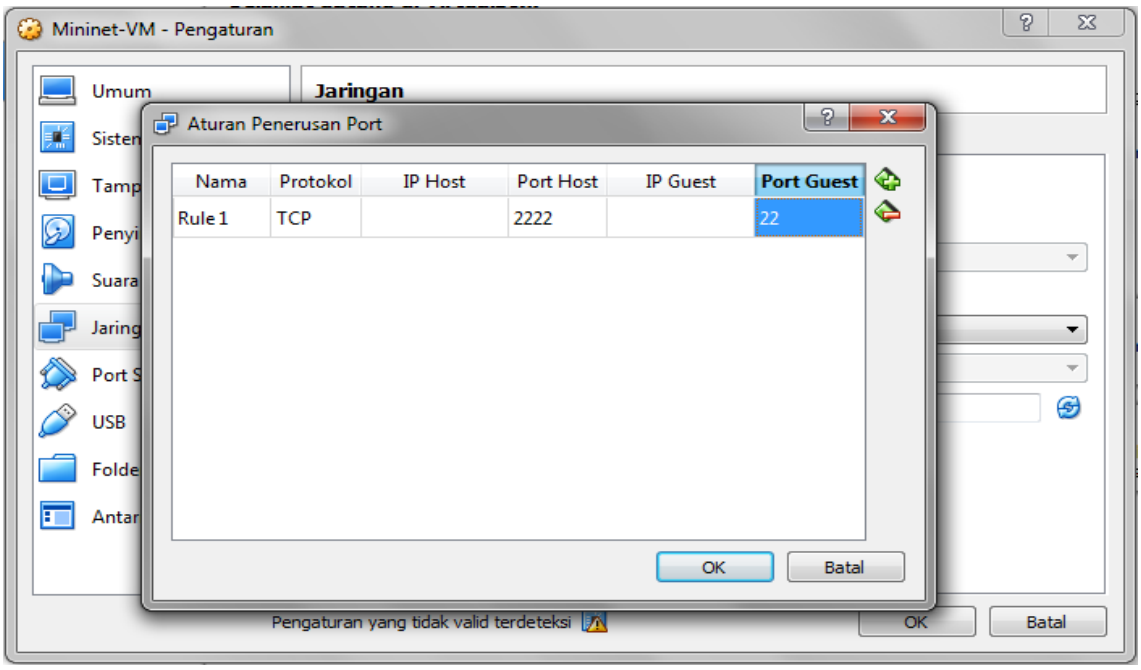


3. Konfigurasi (Pengaturan) Akses Mininet

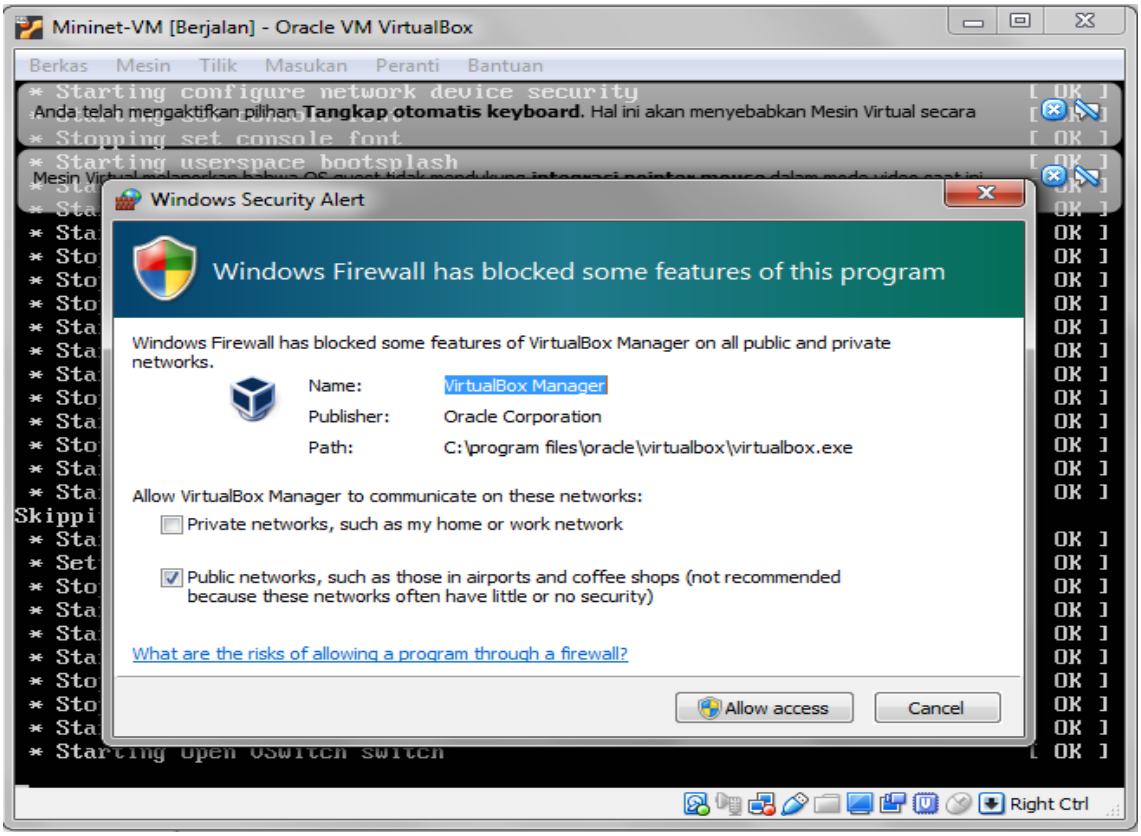


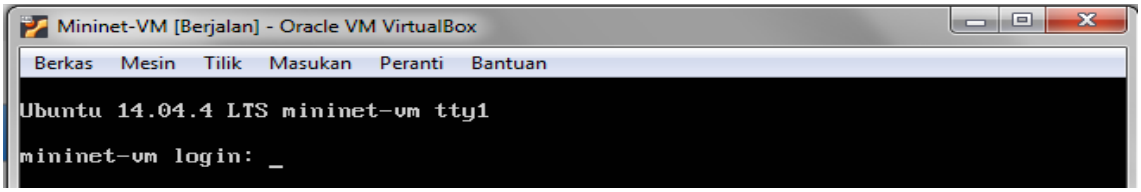


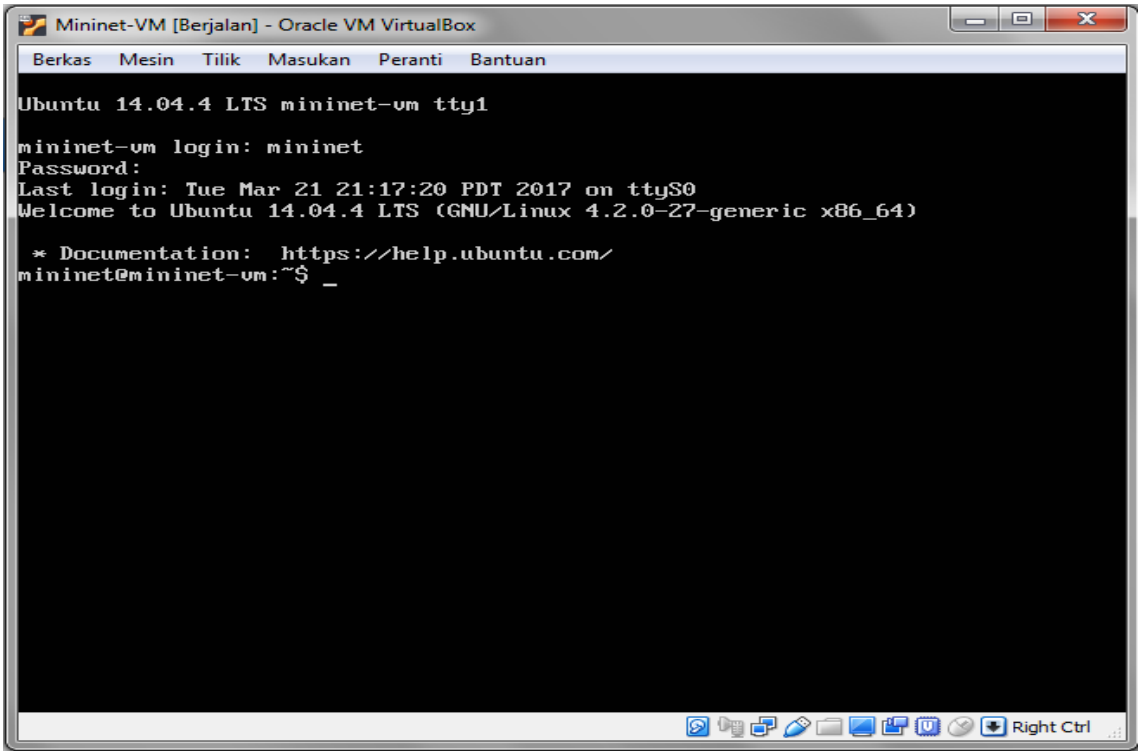




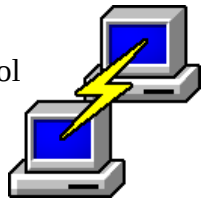
4. Mulai menjalankan Mininet


Mininet-VM [Berjalan] - Oracle VM VirtualBox
Berkas Mesin Tilik Masukan Peranti Bantuan
Anda telah mengaktifkan pilihan Tangkap otomatis keyboard. Hal ini akan menyebabkan Mesin Virtual secara
* Starting configure network device security [OK]
* Stopping set console font [OK]
* Starting userspace bootsplash [OK]
Windows Security Alert
Windows Firewall has blocked some features of this program
Windows Firewall has blocked some features of VirtualBox Manager on all public and private networks.
Name: VirtualBox Manager
Publisher: Oracle Corporation
Path: C:\program files\oracle\virtualbox\virtualbox.exe
Allow VirtualBox Manager to communicate on these networks:
Private networks, such as my home or work network
Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)
What are the risks of allowing a program through a firewall?
Allow access
Cancel
Right Ctrl


Mininet-VM [Berjalan] - Oracle VM VirtualBox
Berkas Mesin Tilik Masukan Peranti Bantuan
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login: _


Mininet-VM [Berjalan] - Oracle VM VirtualBox
Berkas Mesin Tilik Masukan Peranti Bantuan
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Tue Mar 21 21:17:20 PDT 2017 on ttyS0
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)
* Documentation: https://help.ubuntu.com/
mininet@mininet-vm:~$ _
Right Ctrl

PuTTy dan Xming

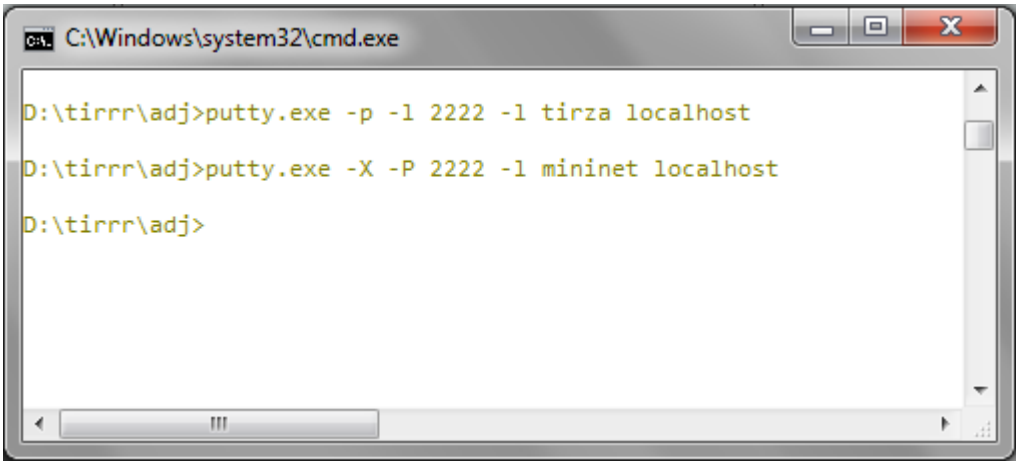


PuTTY adalah sebuah aplikasi open-source yang memanfaatkan protokol jaringan seperti SSH dan Telnet. PuTTY memanfaatkan protokol tersebut untuk mengaktifkan sesi remote pada komputer.

XMING adalah software yang mengimplementasikan X Server pada sistem operasi Windows. Dengan XMING Kita dapat mengakses X Server dari komputer lain dengan jalur SSH. Untuk mengakses SSH, XMING memerlukan software lain yang berperan sebagai alat untuk melakukan transfer data pada jalur SSH, seperti Putty misalnya. XMING berlisensi GPL (GNU *General Public License*) sehingga bebas untuk digunakan, dimodifikasi dan disebarkan.

Lakukan aktivasi PuTTy dengan cara memberikan perintah pada *command prompt* seperti di bawah ini;

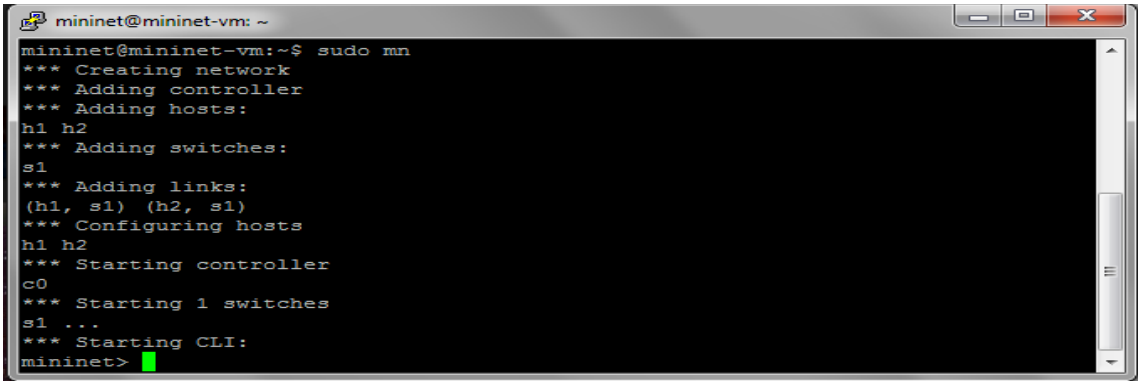
1. Aktivasi PuTTy melalui CMD



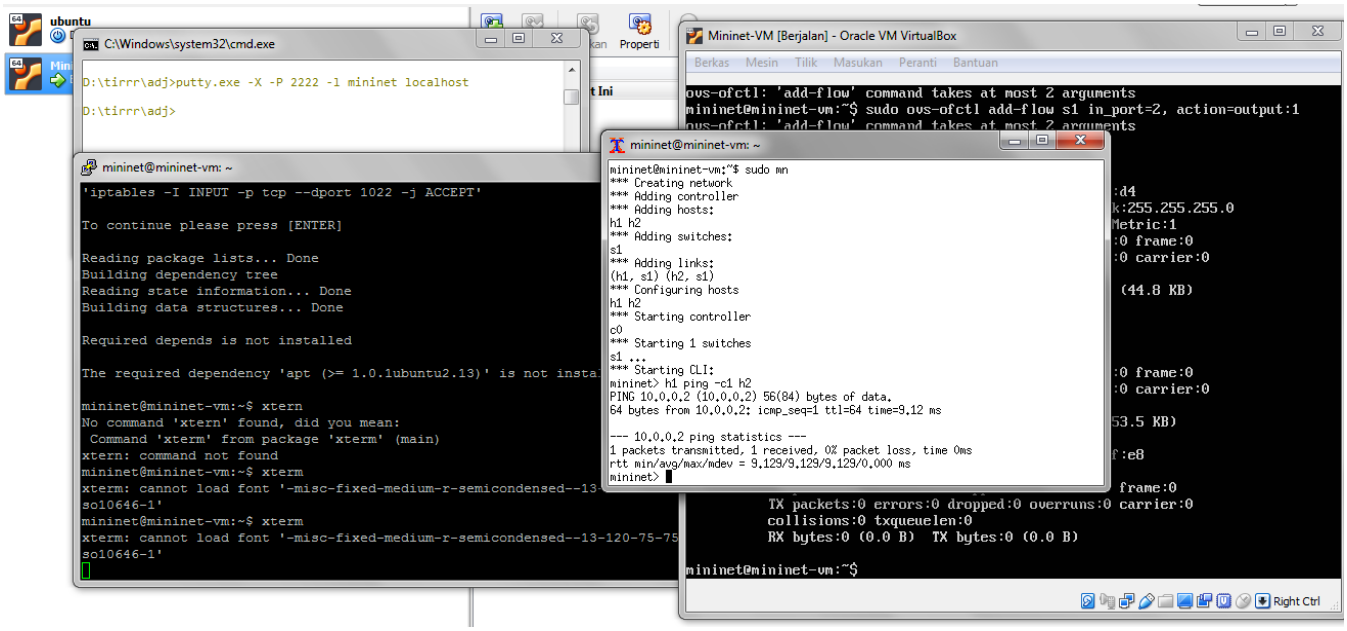
2. Aplikasi PuTTy berjalan

Menjalankan (run) Mininet

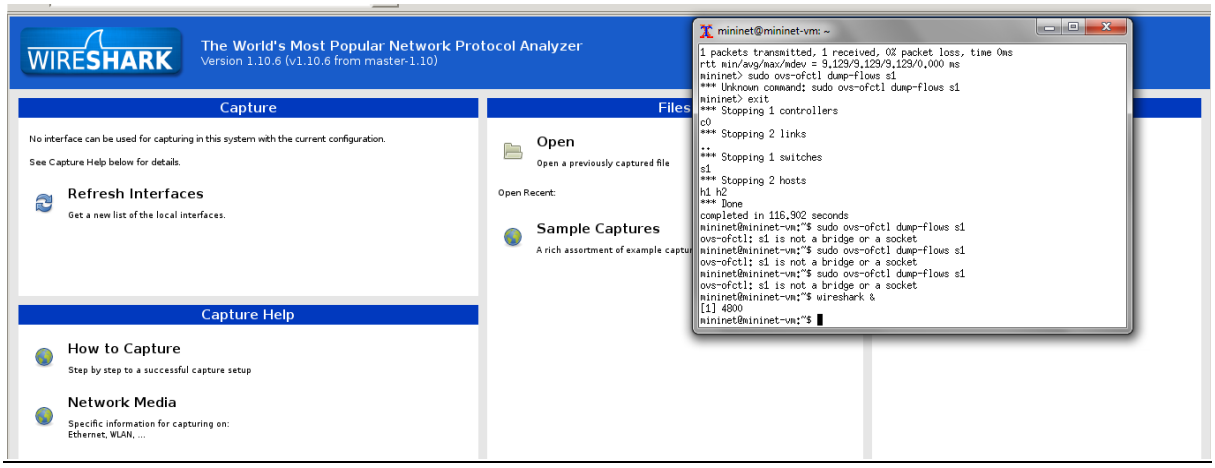
1. mininet@mininet-vm:~$ sudo apt-get install tree



2. h1 ping –c1 h2

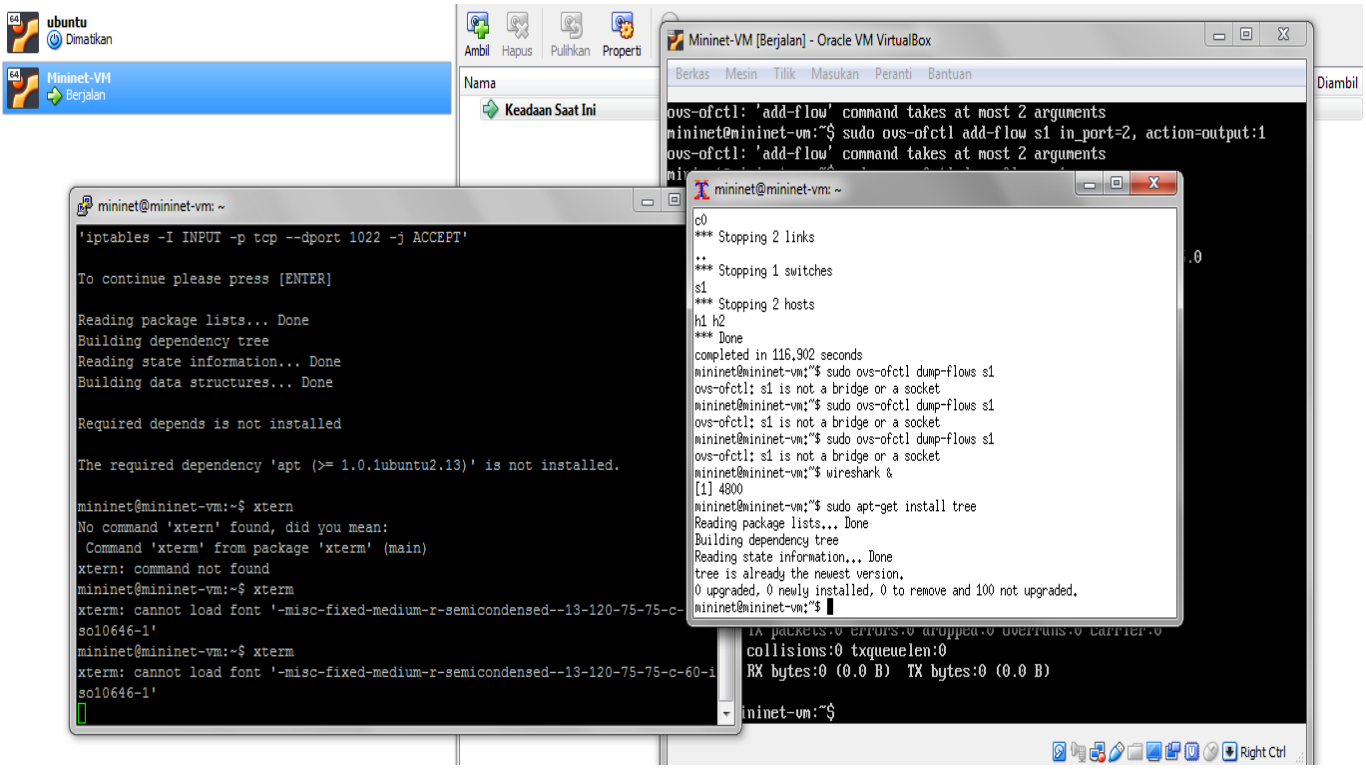


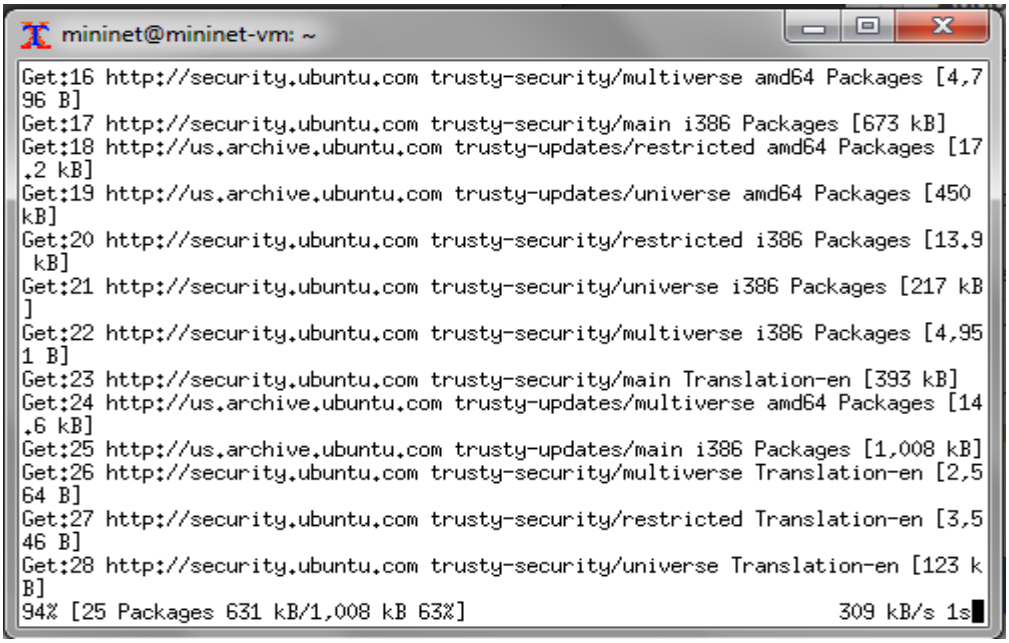
3. Test *Wireshark*

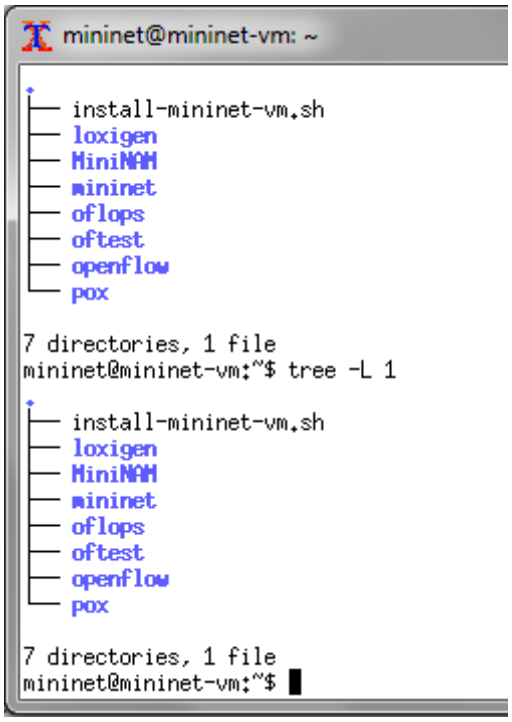


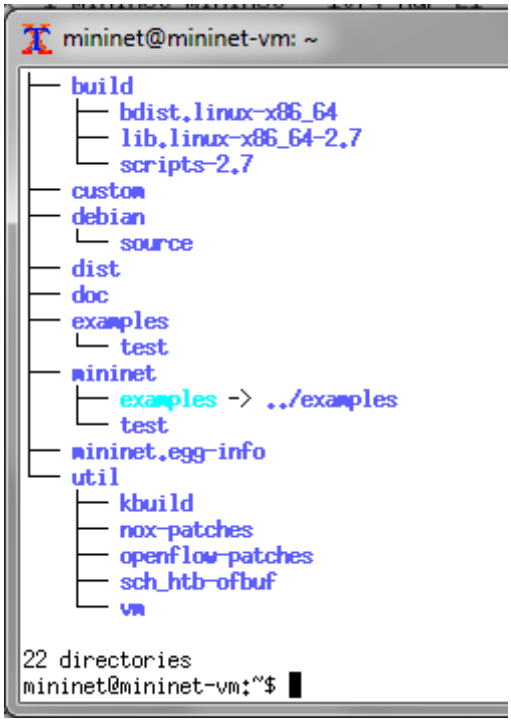
4. *Update*, *Install*, *Check* Tree dan Packages

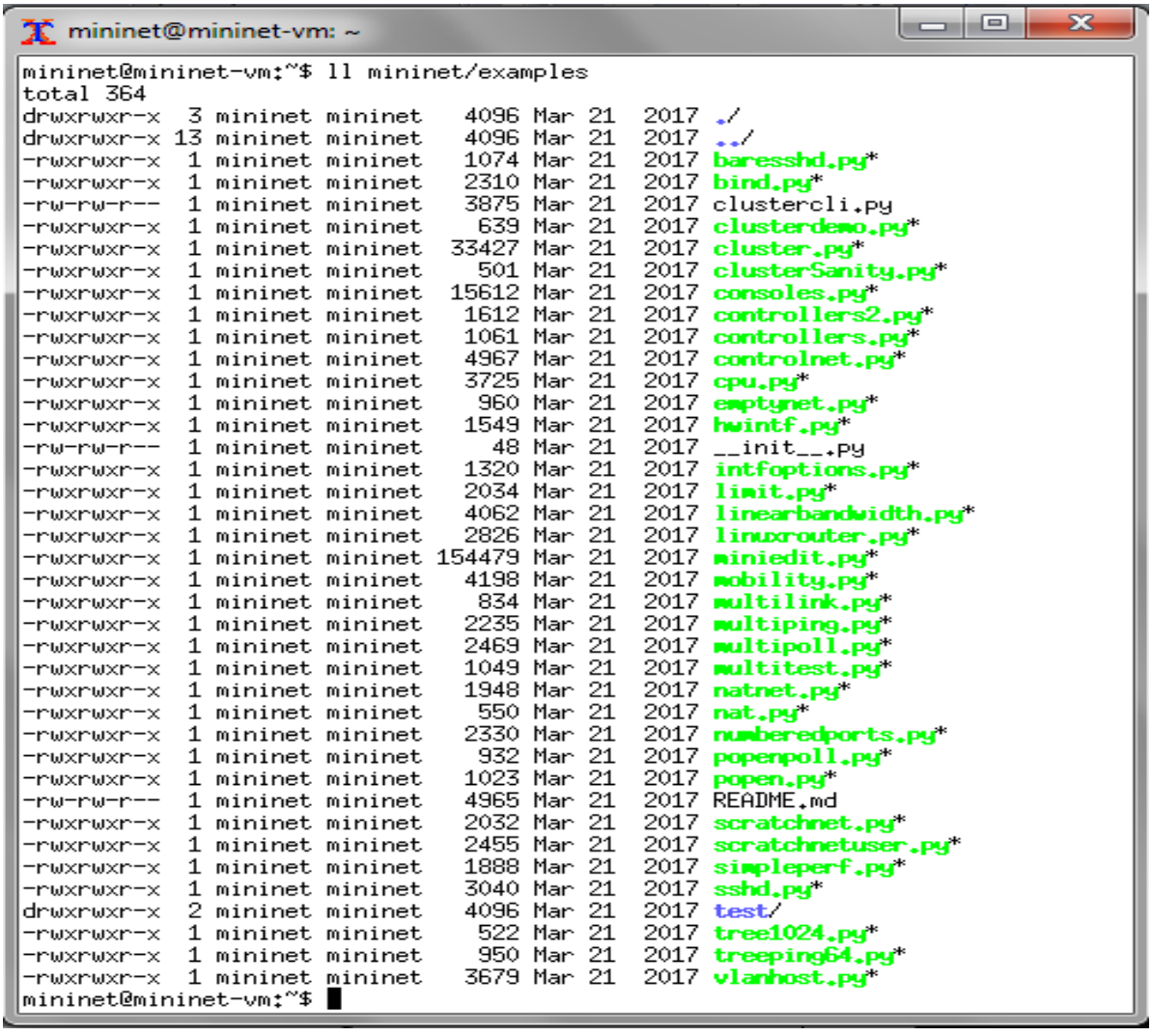
```
mininet@mininet-vm:~$ sudo apt-get install tree

mininet@mininet-vm:~$ sudo apt-get update

mininet@mininet-vm:~/mininet$ tree -L 2 -d
```



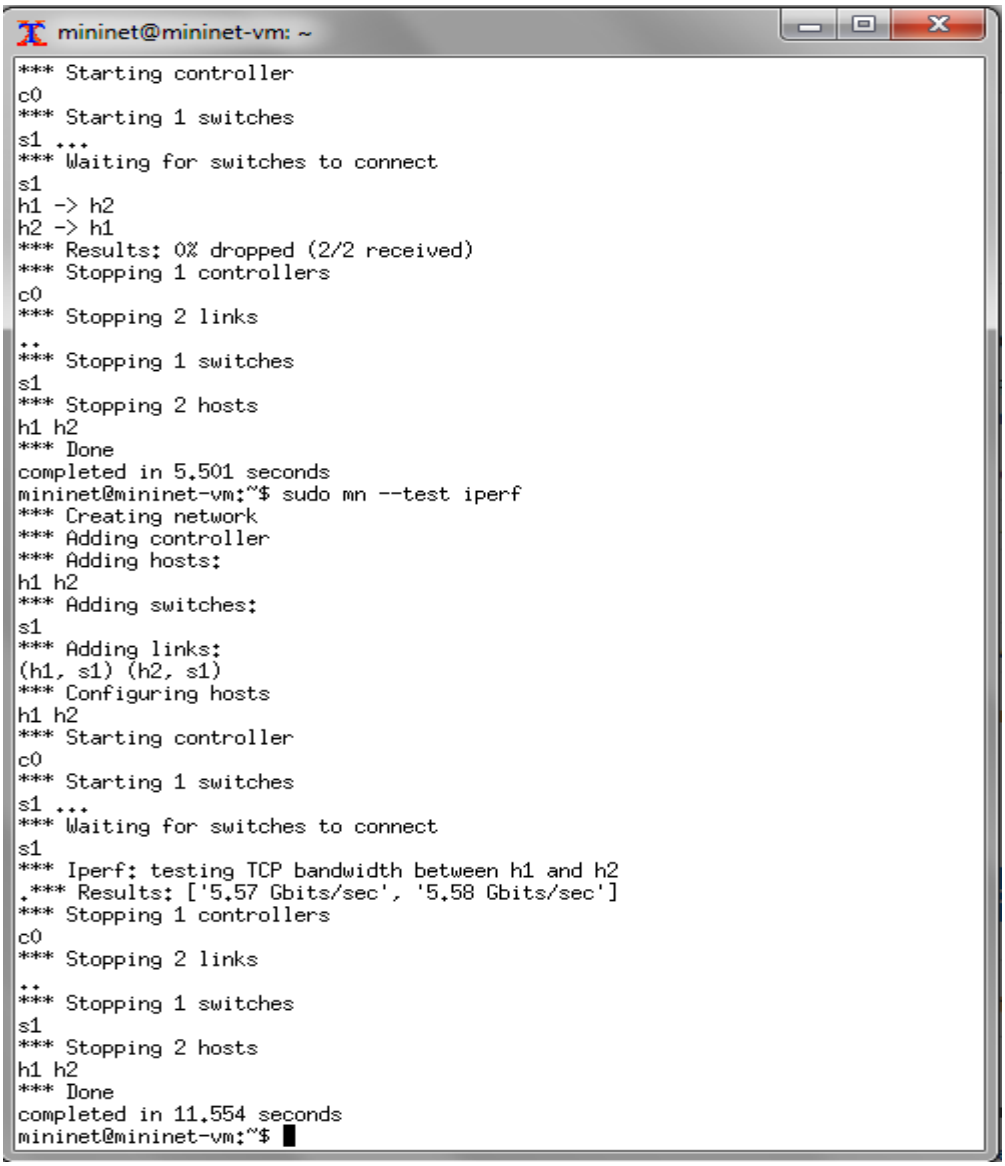



mininet@mininet-vm: ~
.
├── install-mininet-vm.sh
├── loxigen
├── MiniNAM
├── mininet
├── oflops
├── oftest
├── openflow
└── pox
7 directories, 1 file
mininet@mininet-vm:~$ tree -L 1
.
├── install-mininet-vm.sh
├── loxigen
├── MiniNAM
├── mininet
├── oflops
├── oftest
├── openflow
└── pox
7 directories, 1 file
mininet@mininet-vm:~$

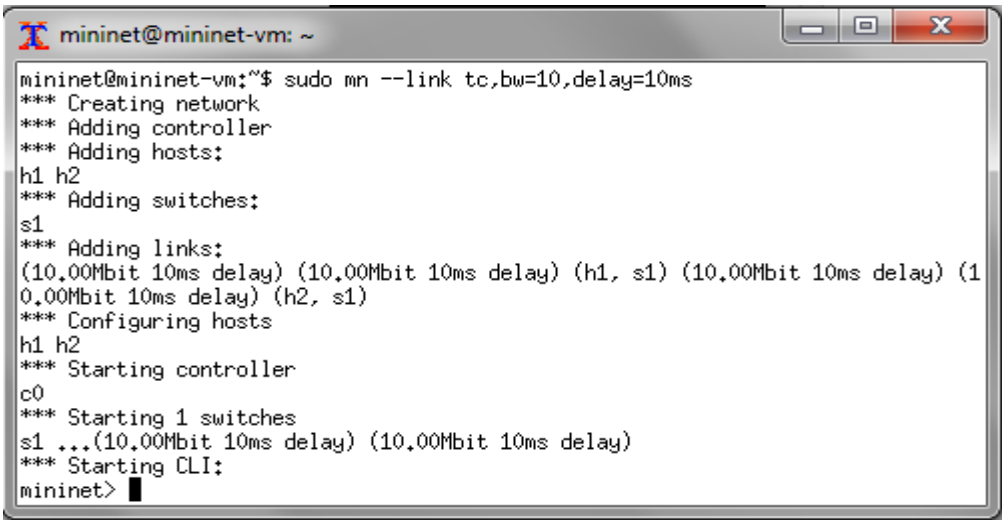

mininet@mininet-vm: ~
├── build
│ ├── bdist.linux-x86_64
│ ├── lib.linux-x86_64-2.7
│ └── scripts-2.7
├── custom
├── debian
│ └── source
├── dist
├── doc
├── examples
│ └── test
├── mininet
│ ├── examples -> ../examples
│ └── test
├── mininet.egg-info
└── util
├── kbuild
├── nox-patches
├── openflow-patches
├── sch_htb-ofbuf
└── vm
22 directories
mininet@mininet-vm:~$


mininet@mininet-vm: ~
mininet@mininet-vm:~$ ll mininet/examples
total 364
drwxrwxr-x 3 mininet mininet 4096 Mar 21 2017 ./
drwxrwxr-x 13 mininet mininet 4096 Mar 21 2017 ../
-rwxrwxr-x 1 mininet mininet 1074 Mar 21 2017 baresshd.py*
-rwxrwxr-x 1 mininet mininet 2310 Mar 21 2017 bind.py*
-rw-rw-r-- 1 mininet mininet 3875 Mar 21 2017 clustercli.py
-rwxrwxr-x 1 mininet mininet 639 Mar 21 2017 clusterdemo.py*
-rwxrwxr-x 1 mininet mininet 33427 Mar 21 2017 cluster.py*
-rwxrwxr-x 1 mininet mininet 501 Mar 21 2017 clustersanity.py*
-rwxrwxr-x 1 mininet mininet 15612 Mar 21 2017 consoles.py*
-rwxrwxr-x 1 mininet mininet 1612 Mar 21 2017 controllers2.py*
-rwxrwxr-x 1 mininet mininet 1061 Mar 21 2017 controllers.py*
-rwxrwxr-x 1 mininet mininet 4967 Mar 21 2017 controlnet.py*
-rwxrwxr-x 1 mininet mininet 3725 Mar 21 2017 cpu.py*
-rwxrwxr-x 1 mininet mininet 960 Mar 21 2017 emptynet.py*
-rwxrwxr-x 1 mininet mininet 1549 Mar 21 2017 hwintf.py*
-rw-rw-r-- 1 mininet mininet 48 Mar 21 2017 __init__.py
-rwxrwxr-x 1 mininet mininet 1320 Mar 21 2017 intfoptions.py*
-rwxrwxr-x 1 mininet mininet 2034 Mar 21 2017 limit.py*
-rwxrwxr-x 1 mininet mininet 4062 Mar 21 2017 linearbandwidth.py*
-rwxrwxr-x 1 mininet mininet 2826 Mar 21 2017 linuxrouter.py*
-rwxrwxr-x 1 mininet mininet 154479 Mar 21 2017 miniedit.py*
-rwxrwxr-x 1 mininet mininet 4198 Mar 21 2017 mobility.py*
-rwxrwxr-x 1 mininet mininet 834 Mar 21 2017 multilink.py*
-rwxrwxr-x 1 mininet mininet 2235 Mar 21 2017 multiping.py*
-rwxrwxr-x 1 mininet mininet 2469 Mar 21 2017 multipoll.py*
-rwxrwxr-x 1 mininet mininet 1049 Mar 21 2017 multitest.py*
-rwxrwxr-x 1 mininet mininet 1948 Mar 21 2017 natnet.py*
-rwxrwxr-x 1 mininet mininet 550 Mar 21 2017 nat.py*
-rwxrwxr-x 1 mininet mininet 2330 Mar 21 2017 numberedports.py*
-rwxrwxr-x 1 mininet mininet 932 Mar 21 2017 popenpoll.py*
-rwxrwxr-x 1 mininet mininet 1023 Mar 21 2017 popen.py*
-rw-rw-r-- 1 mininet mininet 4965 Mar 21 2017 README.md
-rwxrwxr-x 1 mininet mininet 2032 Mar 21 2017 scratchnet.py*
-rwxrwxr-x 1 mininet mininet 2455 Mar 21 2017 scratchnetuser.py*
-rwxrwxr-x 1 mininet mininet 1888 Mar 21 2017 simpleperf.py*
-rwxrwxr-x 1 mininet mininet 3040 Mar 21 2017 sshd.py*
drwxrwxr-x 2 mininet mininet 4096 Mar 21 2017 test/
-rwxrwxr-x 1 mininet mininet 522 Mar 21 2017 tree1024.py*
-rwxrwxr-x 1 mininet mininet 950 Mar 21 2017 treeping64.py*
-rwxrwxr-x 1 mininet mininet 3679 Mar 21 2017 vlanhost.py*
mininet@mininet-vm:~$

5. Link TC



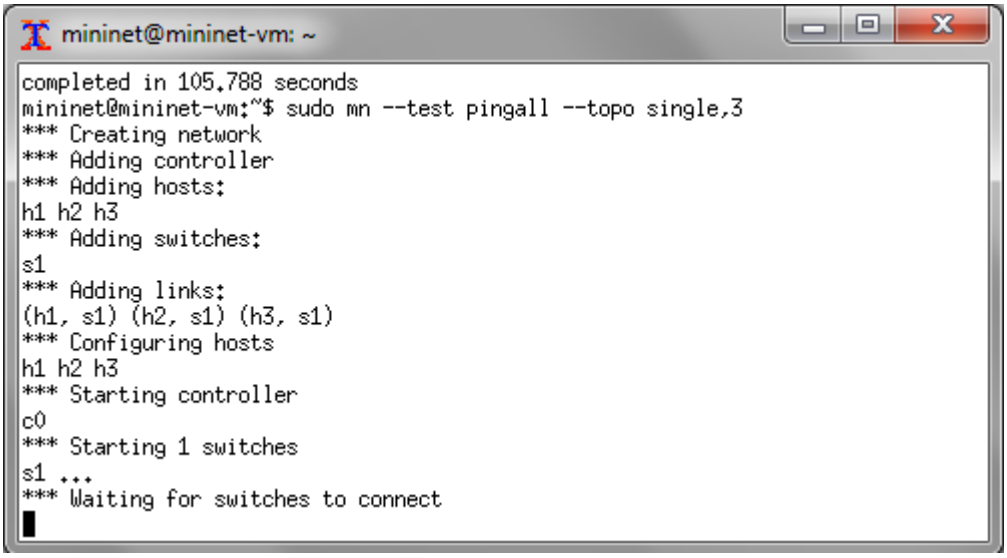
mininet@mininet-vm:~$ sudo mn --link tc,bw=10,delay=10ms

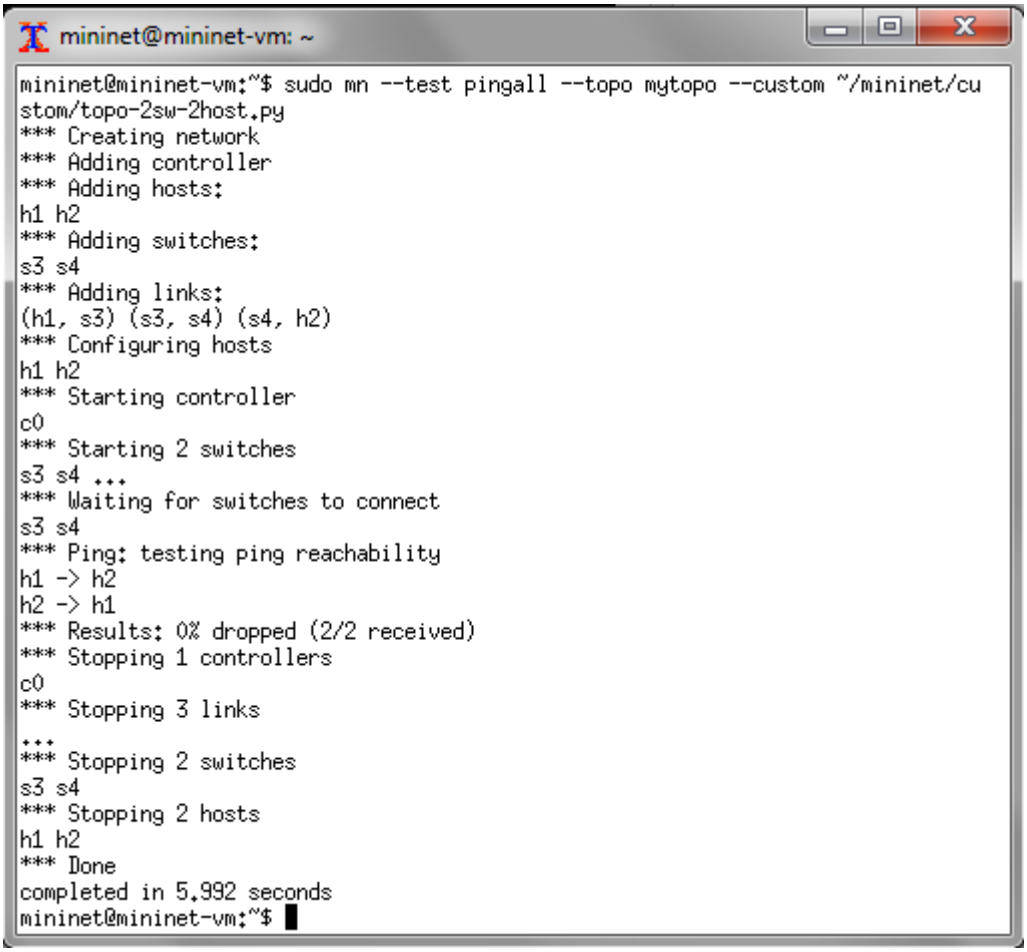


6. Ping all single dan ping all topo

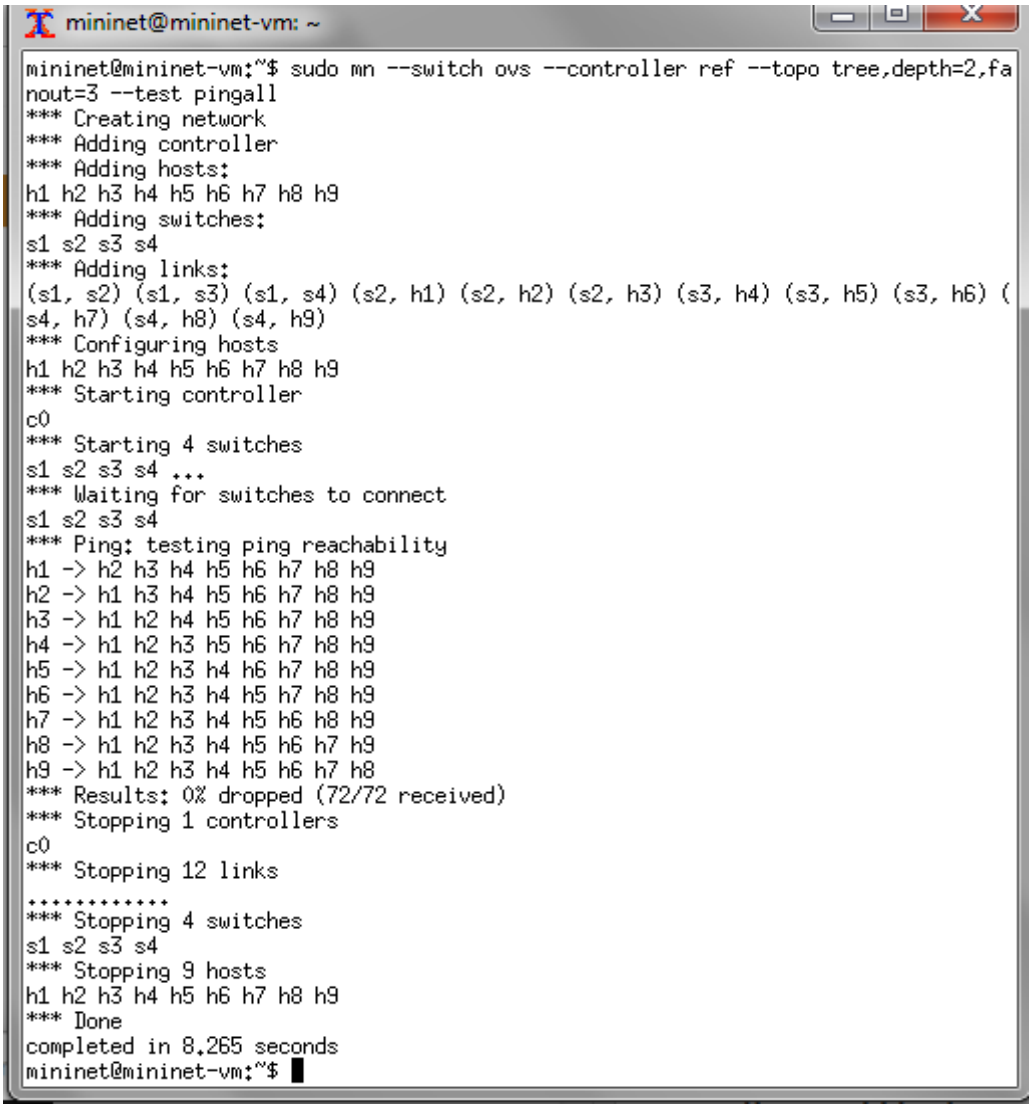
mininet@mininet-vm:~$ sudo mn --test pingall --topo single,3

mininet@mininet-vm:~$ sudo mn --test pingall --topo mytopo --custom ~/mininet/custom/topo-2sw-2host.py

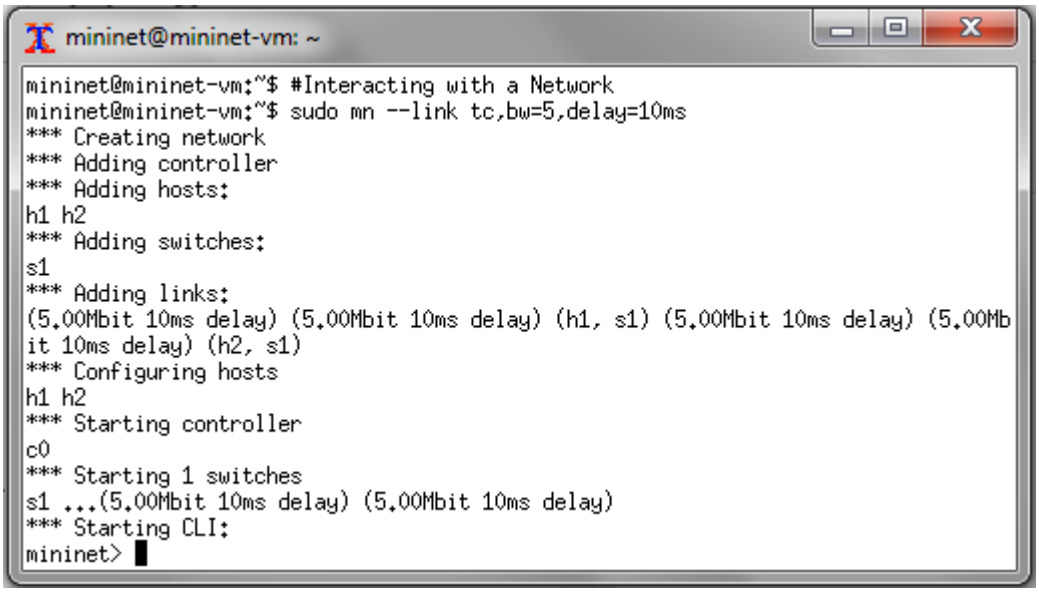




7. *Create Network*



8. *Interacting with a Network*

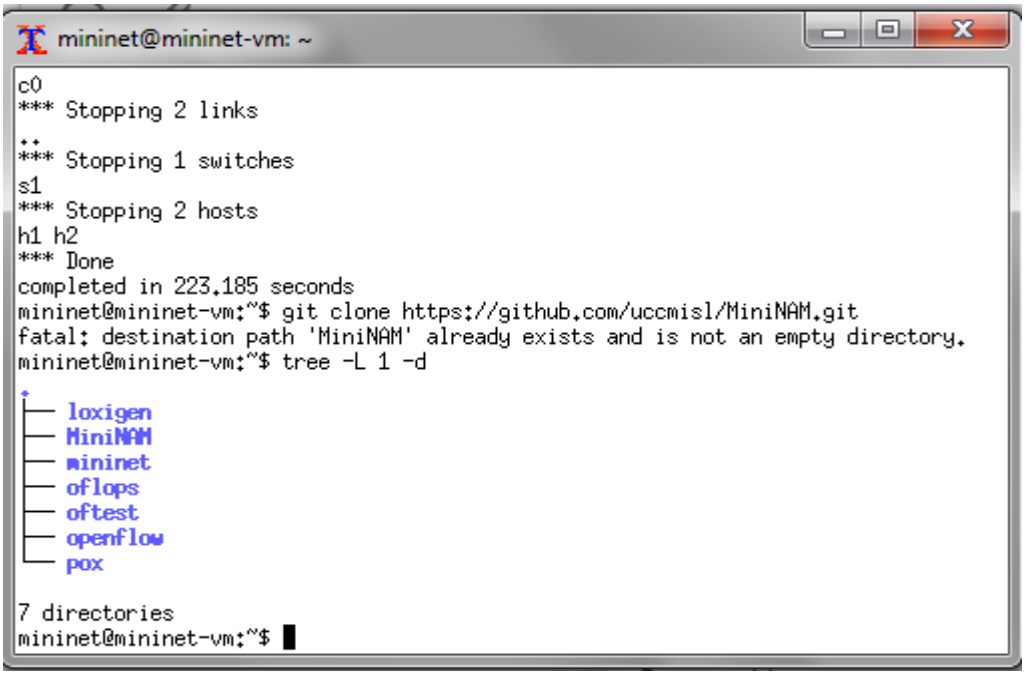


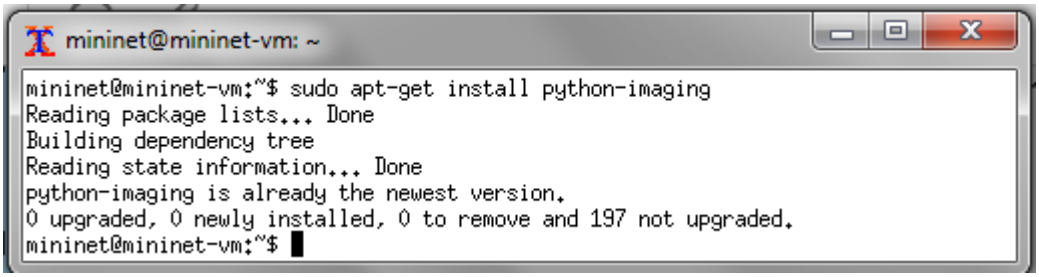
MINI NAM

MiniNAM adalah alat berbasis GUI yang ditulis dengan *Python Tkinter* yang menyediakan animasi real-time dari jaringan apa pun yang dibuat oleh emulator Mininet. Alat ini mencakup semua komponen yang diperlukan untuk memulai, memvisualisasikan dan memodifikasi aliran jaringan Mininet secara real-time.

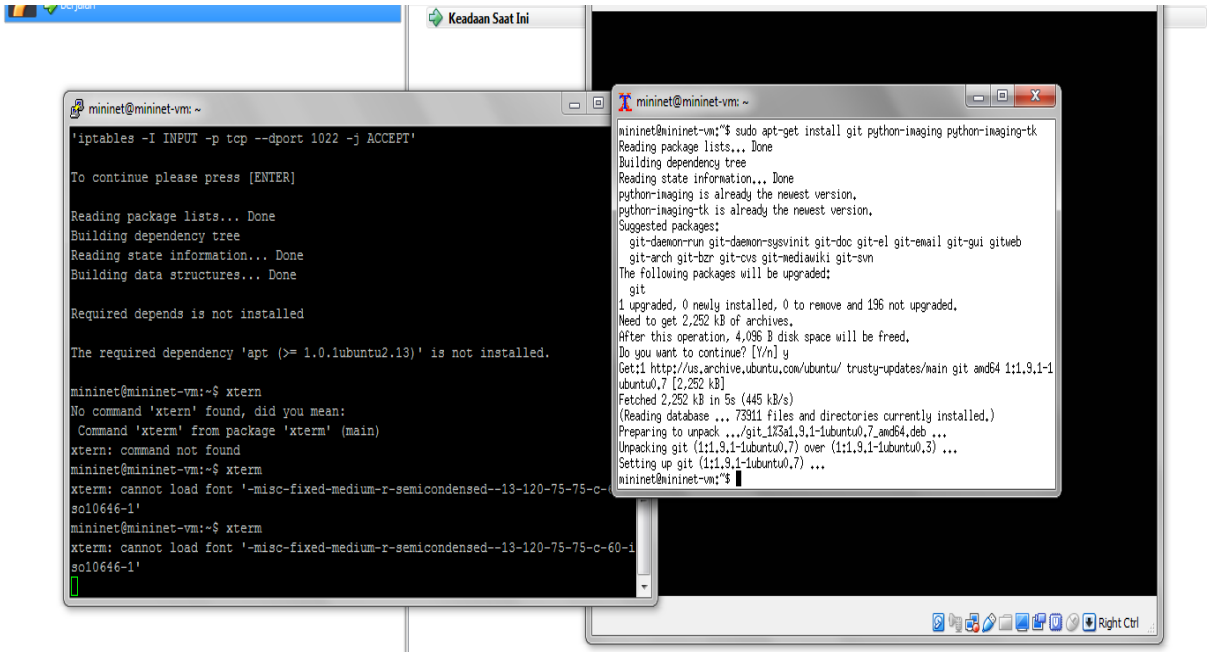
1. MiniNAM *Setup*

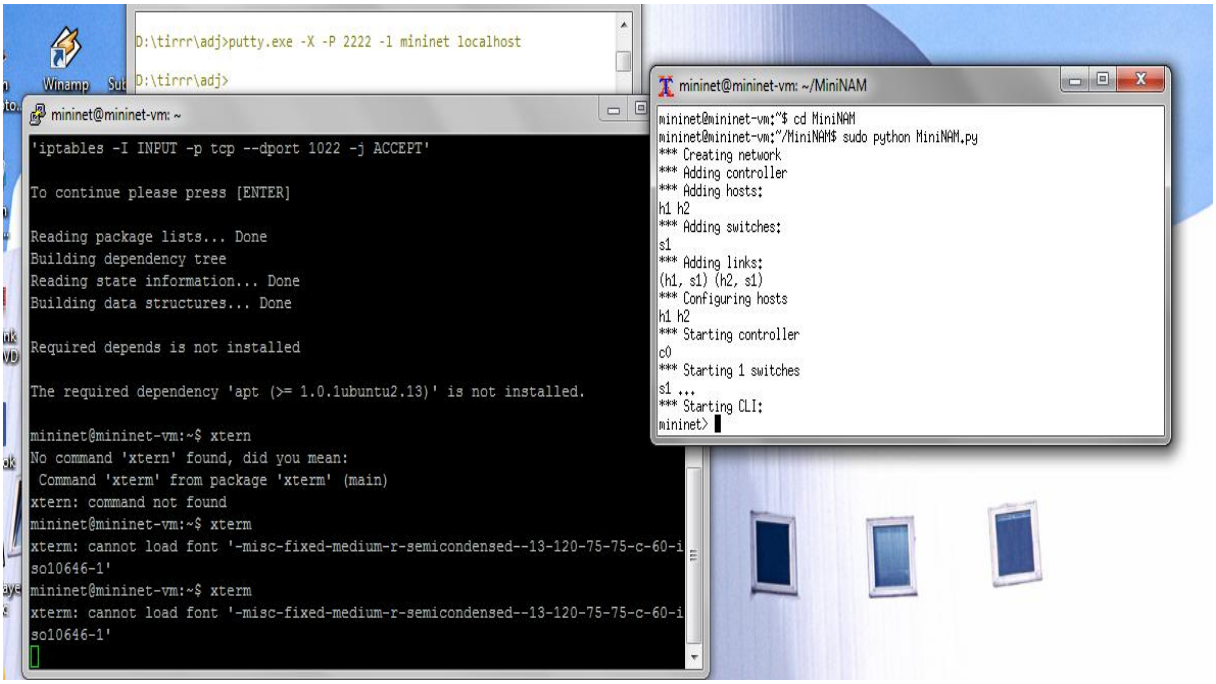
```
mininet@mininet-vm:~$ git clone https://github.com/uccmisl/MiniNAM.git
```



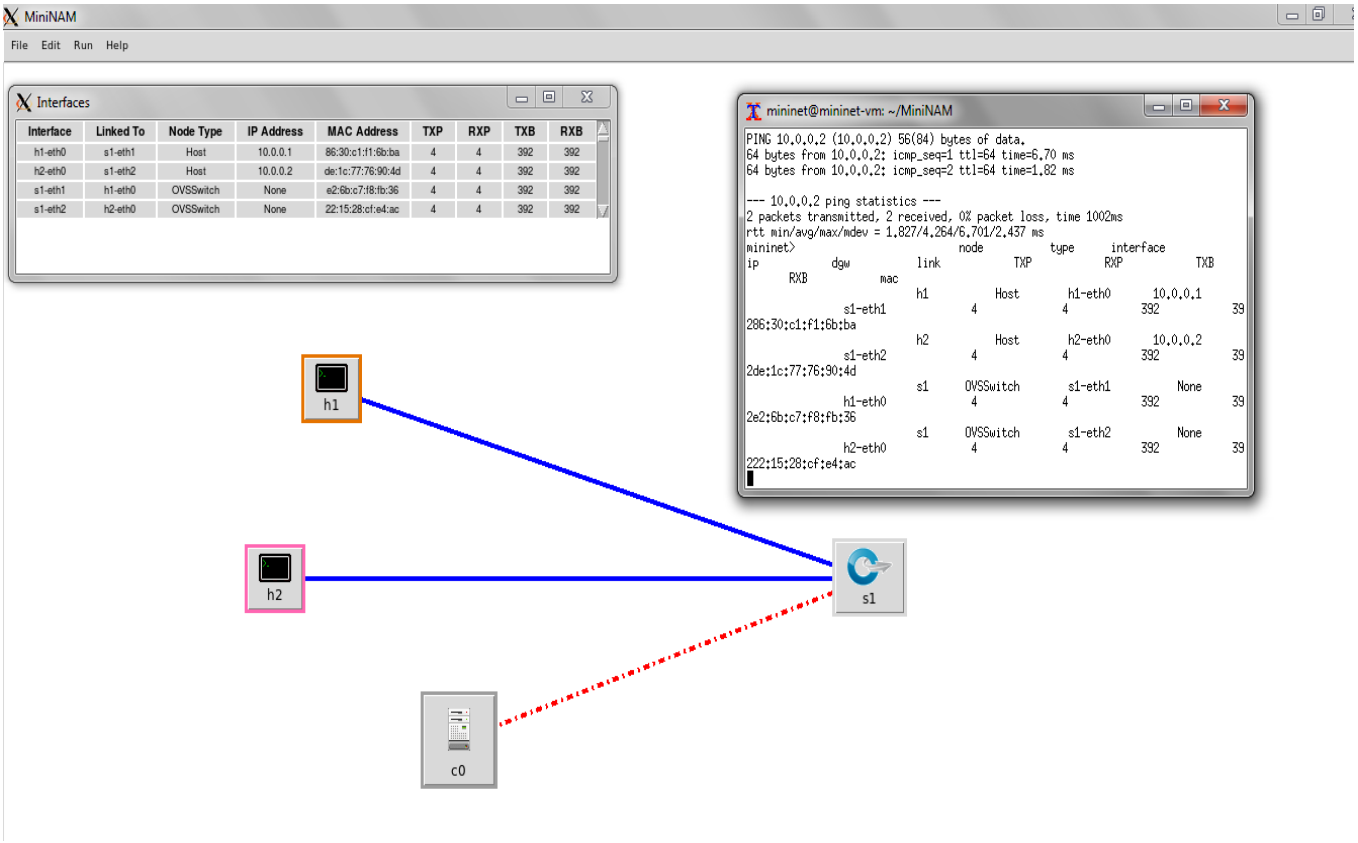


mininet@mininet-vm:~$ git apt-get install git python-imaging python-imaging-tk





2. h1 ping –c2 h2


MiniNAM
File Edit Run Help
Interfaces
Interface Linked To Node Type IP Address MAC Address TXP RXP TXB RXB
h1-eth0 s1-eth1 Host 10.0.0.1 86:30:c1:f1:6b:ba 4 4 392 392
h2-eth0 s1-eth2 Host 10.0.0.2 de:1c:77:76:90:4d 4 4 392 392
s1-eth1 h1-eth0 OVSSwitch None e2:6b:c7:f8:fb:36 4 4 392 392
s1-eth2 h2-eth0 OVSSwitch None 22:15:28:cf:e4:ac 4 4 392 392
mininet@mininet-vm: ~/MiniNAM
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=6.70 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.82 ms
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.827/4.264/6.701/2.437 ms
mininet> node type interface
ip dgw link TXP RXP TXB
RXB mac
h1 Host h1-eth0 10.0.0.1
s1-eth1 4 4 392 39
286:30:c1:f1:6b:ba
h2 Host h2-eth0 10.0.0.2
s1-eth2 4 4 392 39
2de:1c:77:76:90:4d
s1 OVSSwitch s1-eth1 None
h1-eth0 4 4 392 39
2e2:6b:c7:f8:fb:36
s1 OVSSwitch s1-eth2 None
h2-eth0 4 4 392 39
222:15:28:cf:e4:ac
h1
h2
s1
c0

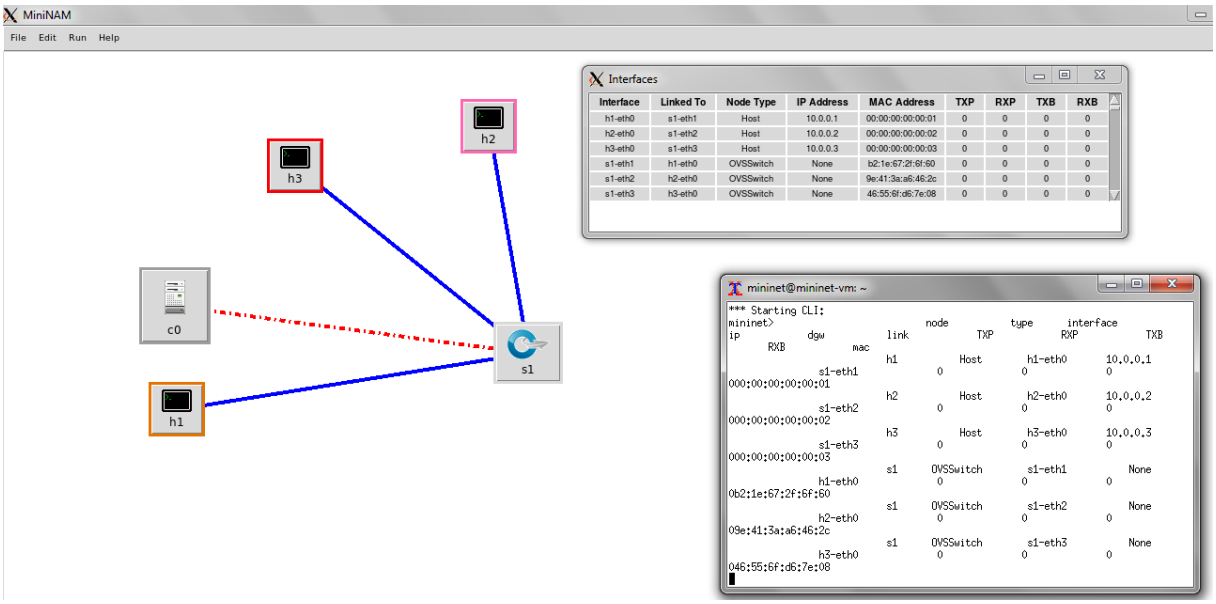
OPEN FLOW

OpenFlow adalah sebuah protokol yang memungkinkan pengaturan penjaluran dan pengiriman paket ketika melalui sebuah *switch*. Dalam sebuah jaringan konvensional, setiap *switch* hanya berfungsi meneruskan paket yang lewat ke port yang sesuai tanpa dapat membedakan tipe protokol data yang dikirimkan, misalnya *elastic* atau *inelastic traffic*.

Berikut contoh tahap-tahapannya;

1. mininet@mininet~vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote

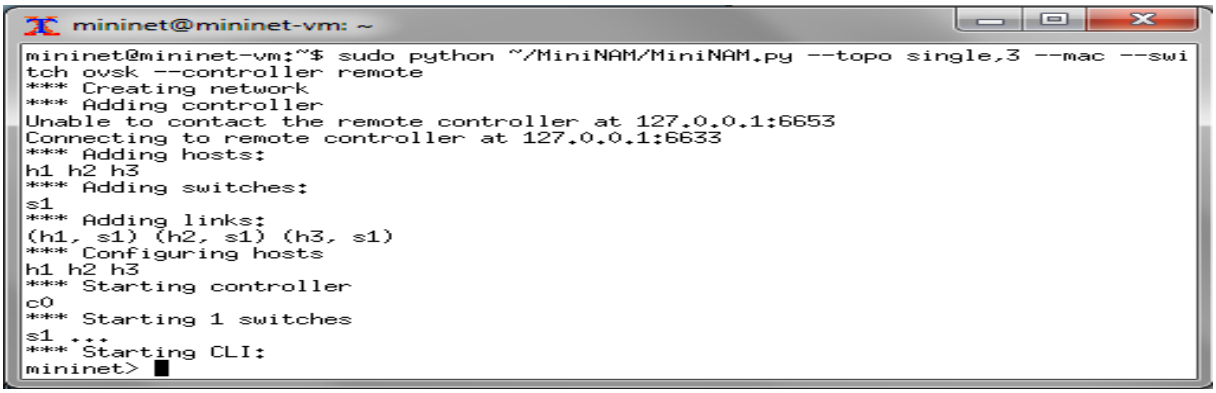
   Membuat 3 *host* dengan 1 *switch*

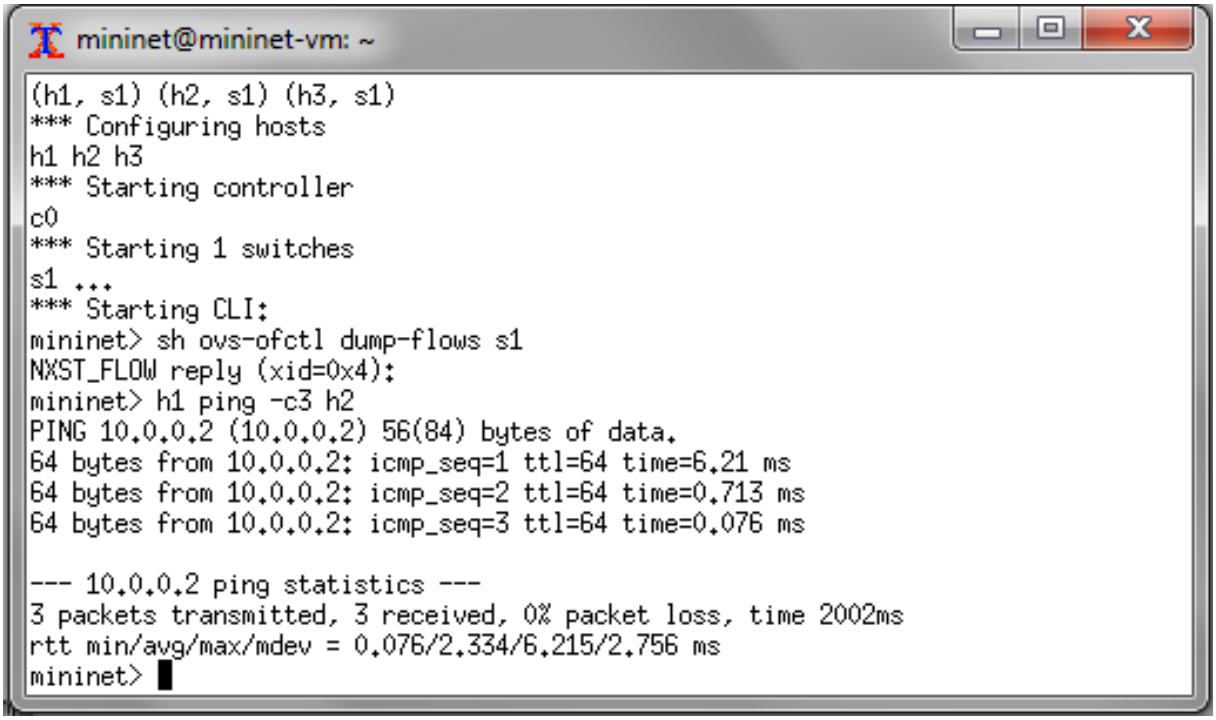


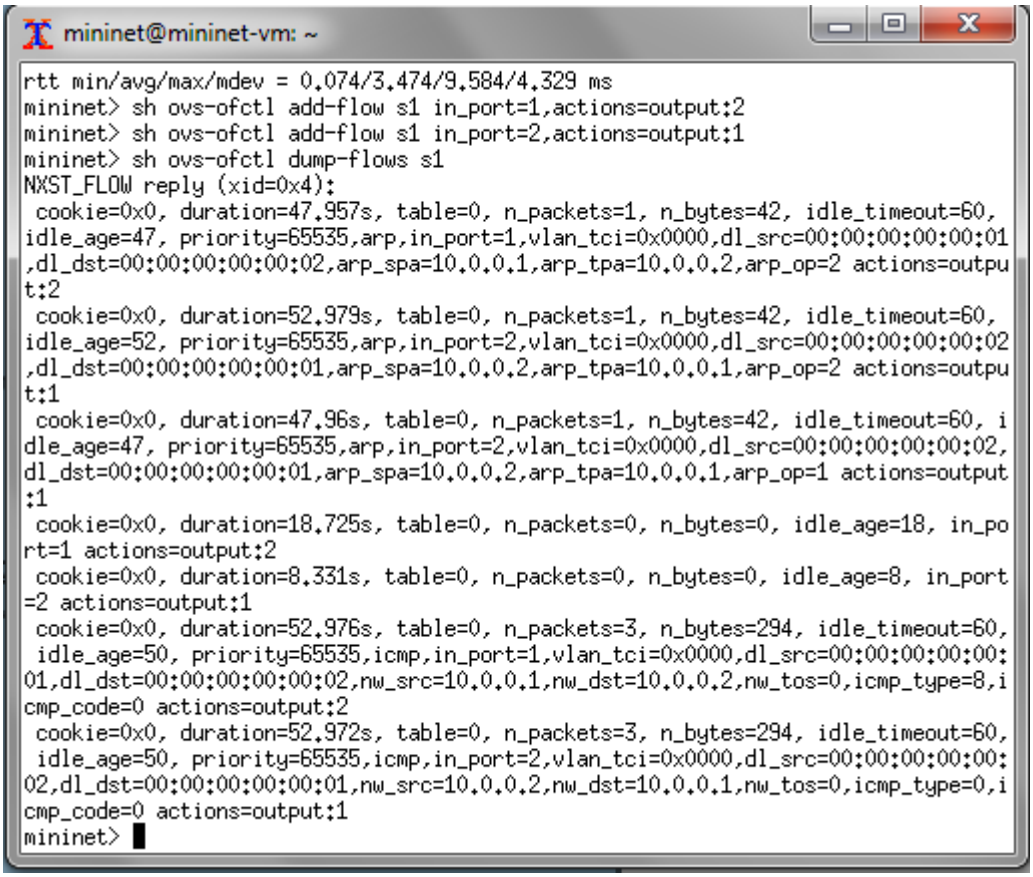
2. Manual Flow Entry

   mininet@mininet~vm:~$ sudo python ~/MiniNAM/MiniNAM.py --topo single,3 --mac --switch ovsk --controller remote

   mininet> sh ovs-ofctl dump-flows s1

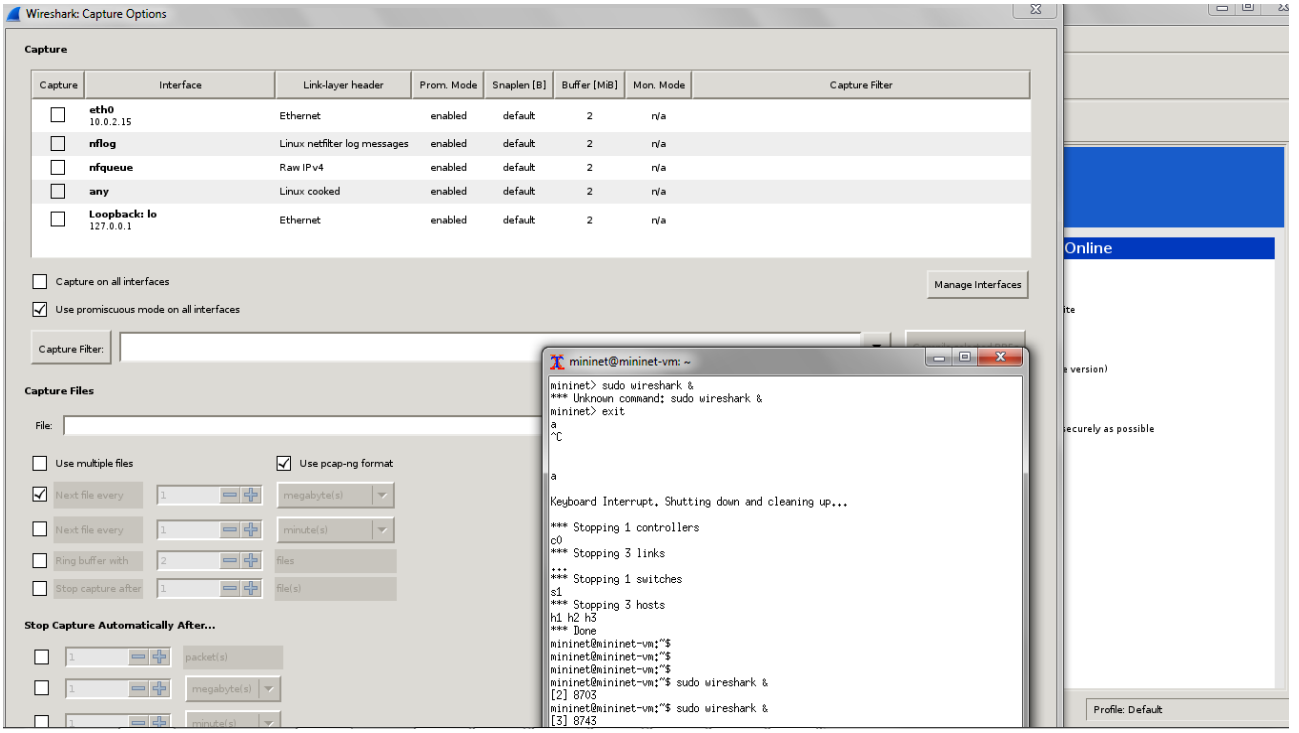




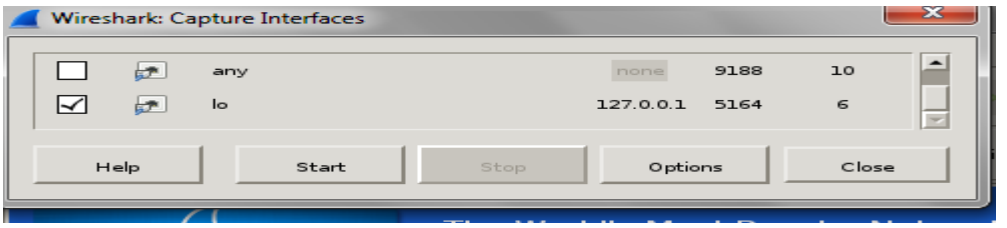


3. Wireshark

mininet@mininet~vm:~$ sudo wireshark &



Klik ‘Capture’ lalu pada *menubar* di atas lalu pilih ‘Capture Interfaces’ dan pilih (centang) ‘lo’ kemudian mulai (‘Start’).



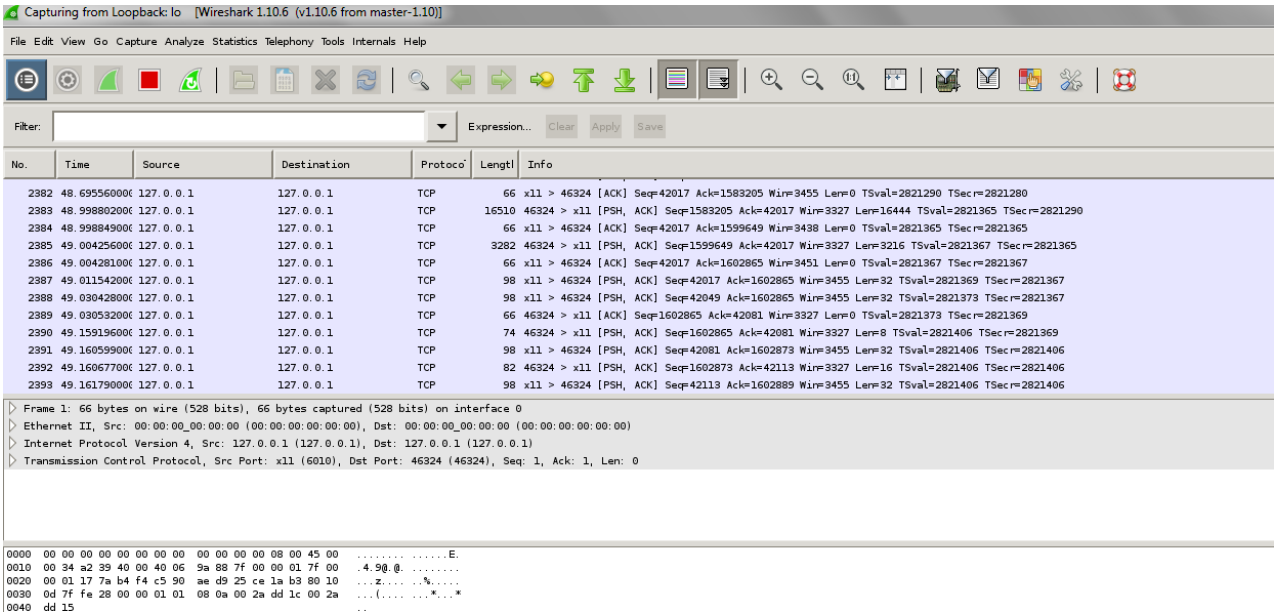
4. Starting Controller (OF Reference Controller)

mininet@mininet~vm:~$ controller ptcp: &

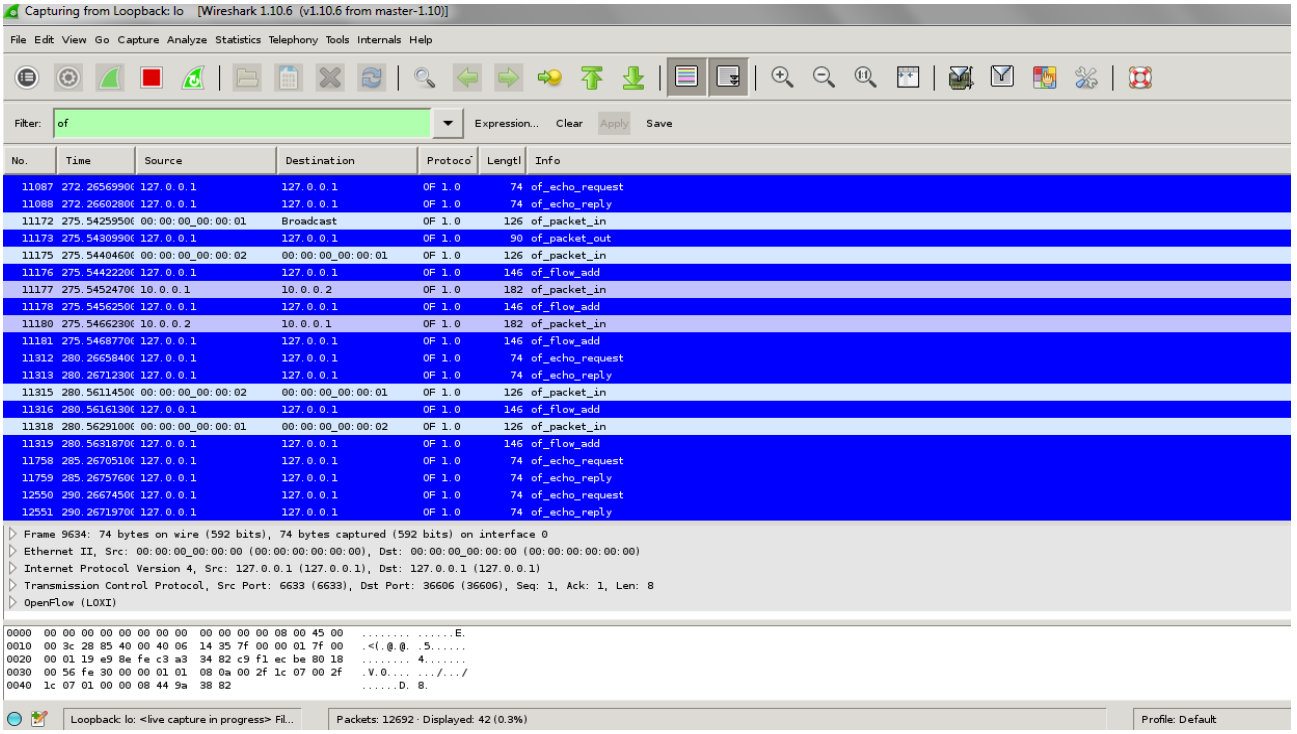
```
mininet@mininet-vm:~$ controller ptcp: &
[4] 9119
mininet@mininet-vm:~$ Apr 07 03:45:20|00001|vconn_tcp|ERR|ptcp:: bind: Address alr
eady in use
Apr 07 03:45:20|00002|controller|ERR|ptcp:: connect: Address already in use
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller rem
ote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

```
mininet@mininet-vm: ~
mininet> h1 ip -s -s neigh flush all
Nothing to flush.
mininet> h2 ip -s -s neigh flush all
Nothing to flush.
mininet> sh ovs-ofctl del-flows s1
mininet> h1 ping -c1 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=6.39 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.393/6.393/6.393/0.000 ms
mininet>
```

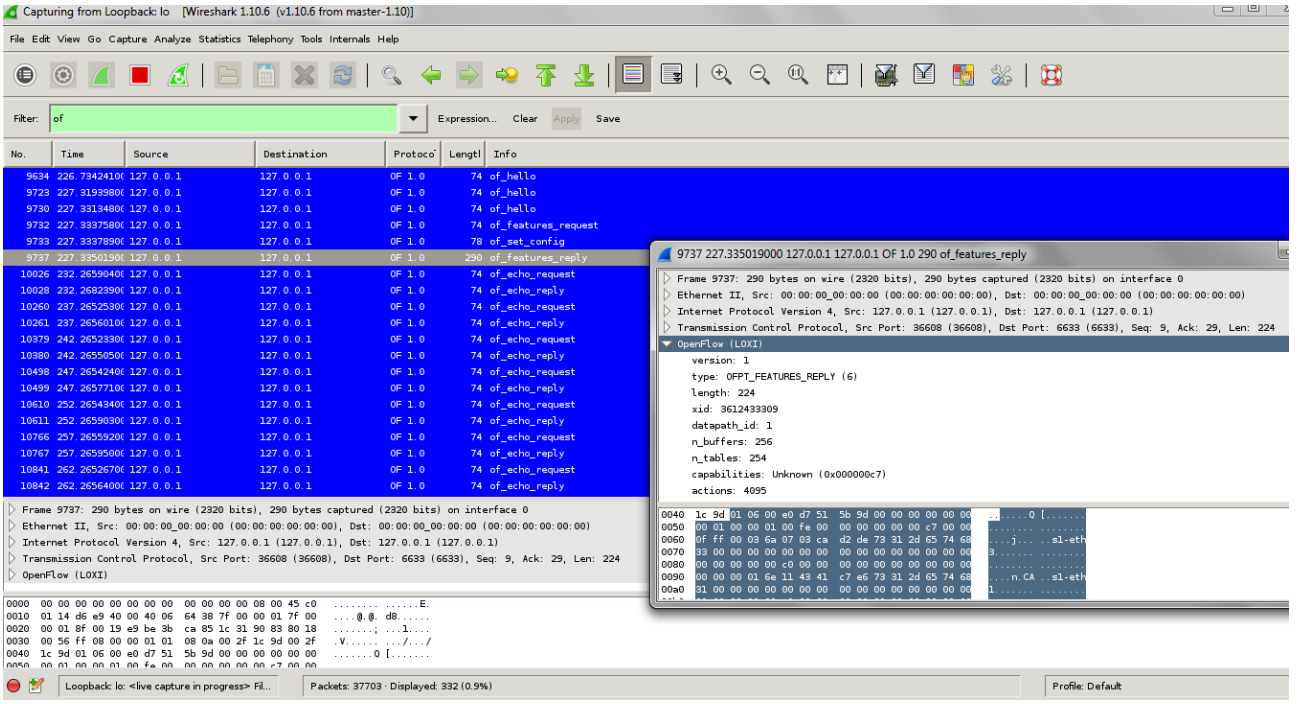
Berikut tampilan pada wireshark;



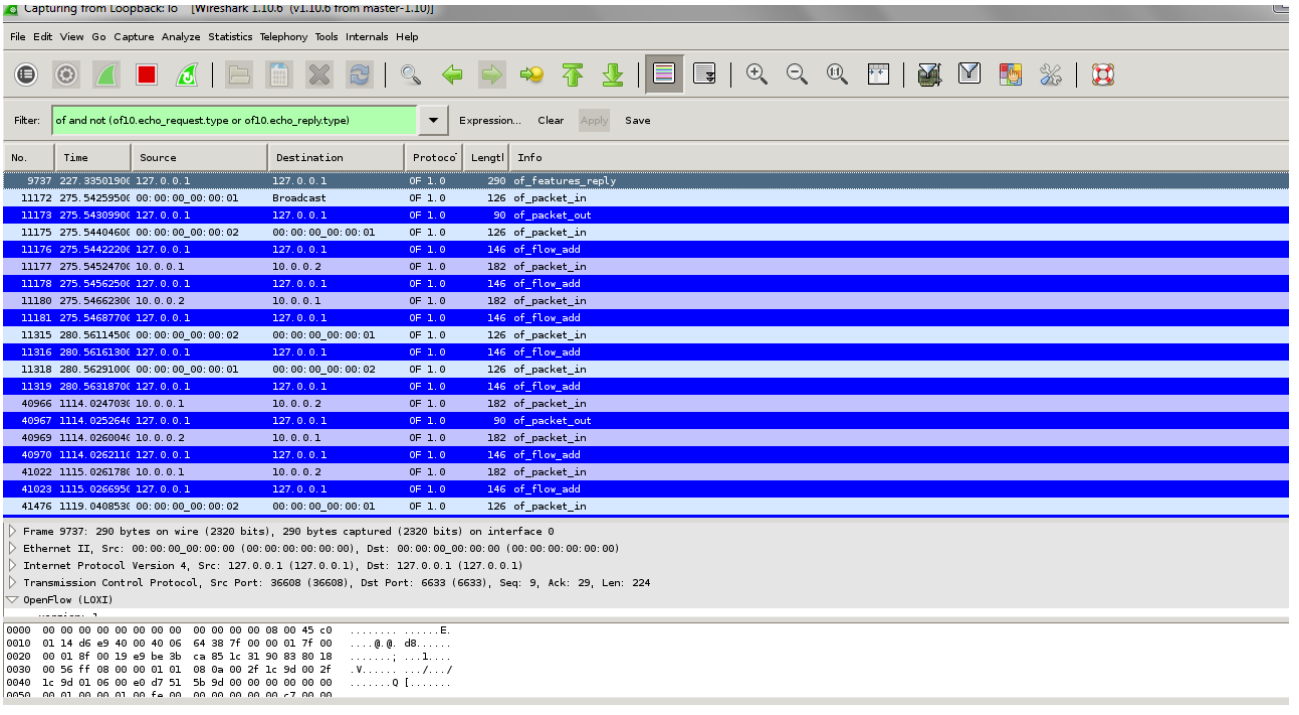
5. Lakukan Filter ‘OF’ pada Wireshark



6. Inspect Packet



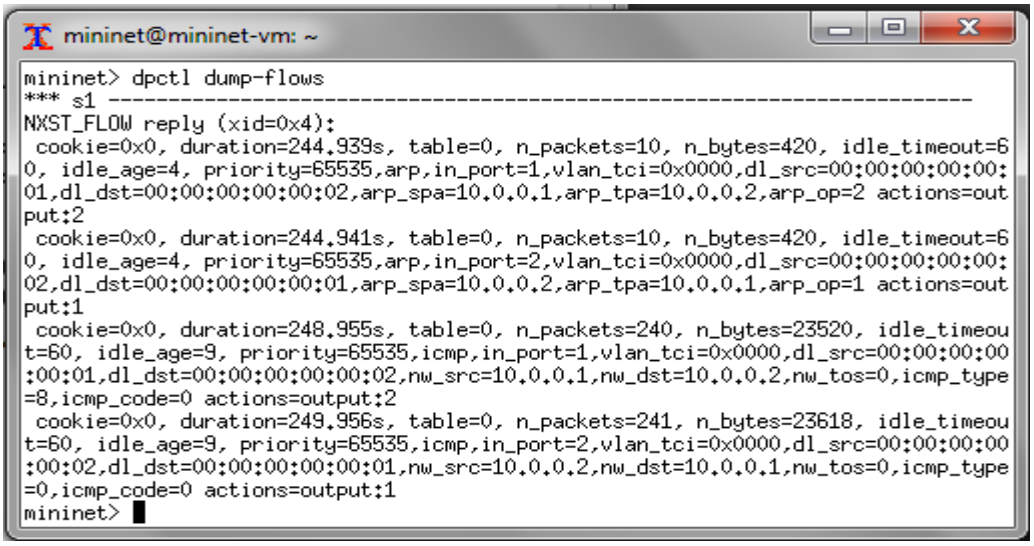
7. Filter ‘of and not (of10.echo_request.type or of10.echo_reply.type)’ pada tampilan WireShark setelah ‘h1 ping -c1 h2’.



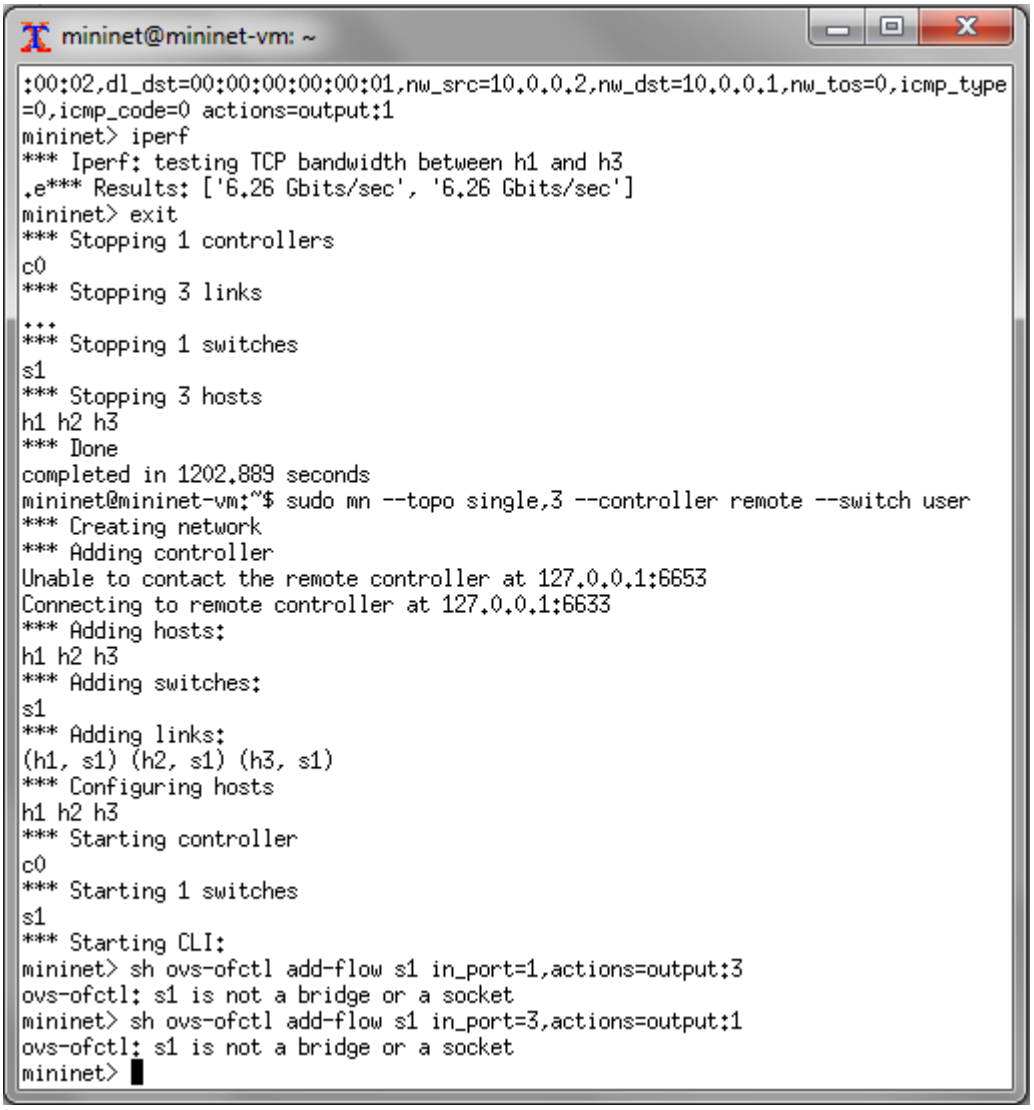
8. Flow Entries

mininet> dpctl dump-flows

mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1



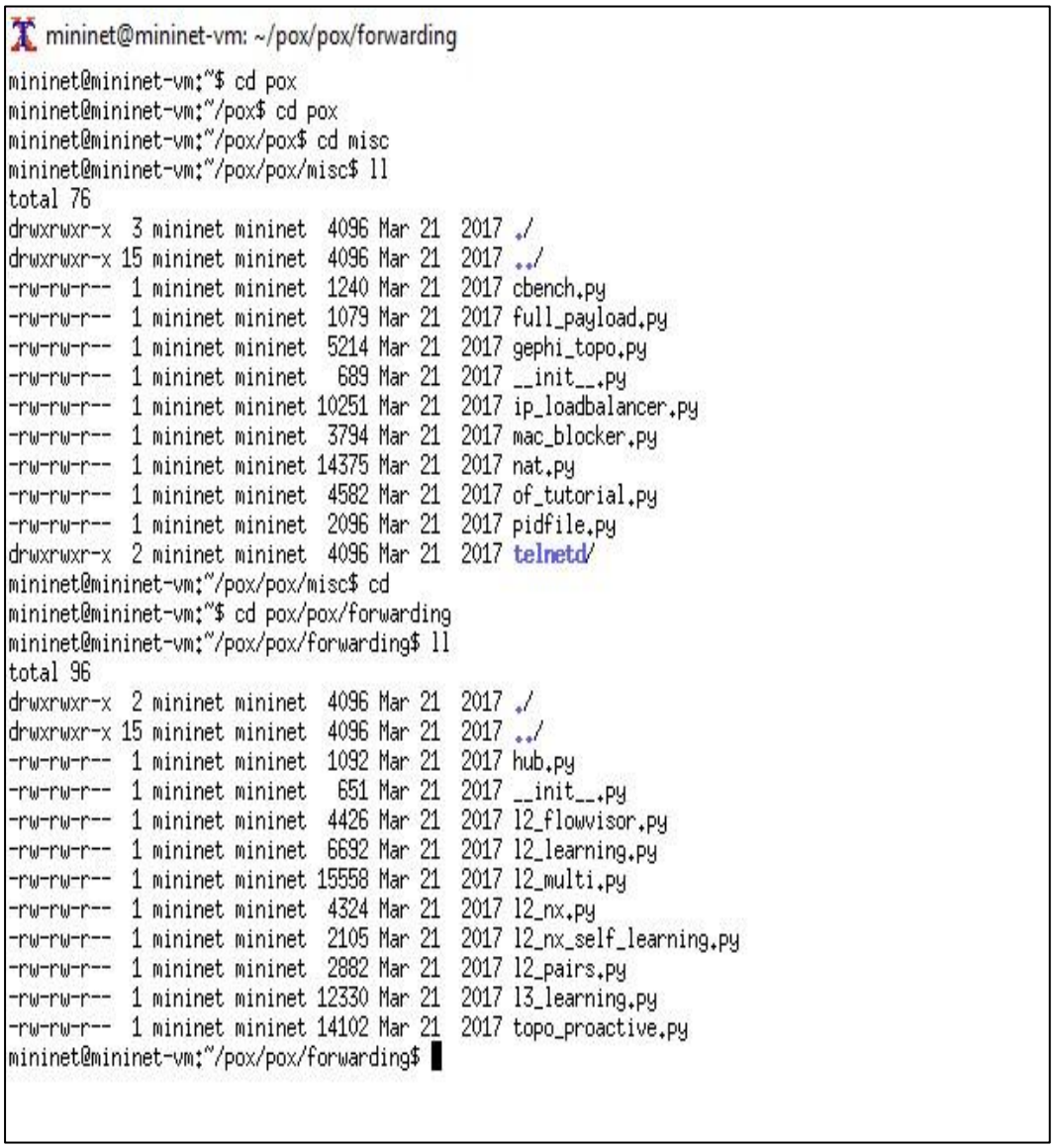
9. Benchmark kernel- vs. user-space



POX CONTROLLER

1. Mengaktifkan POX *Controller*

   root@ubuntu:~# cd /poxroot@ubuntu:~/pox# ./pox.py log.level --DEBUG misc.of_tutorial

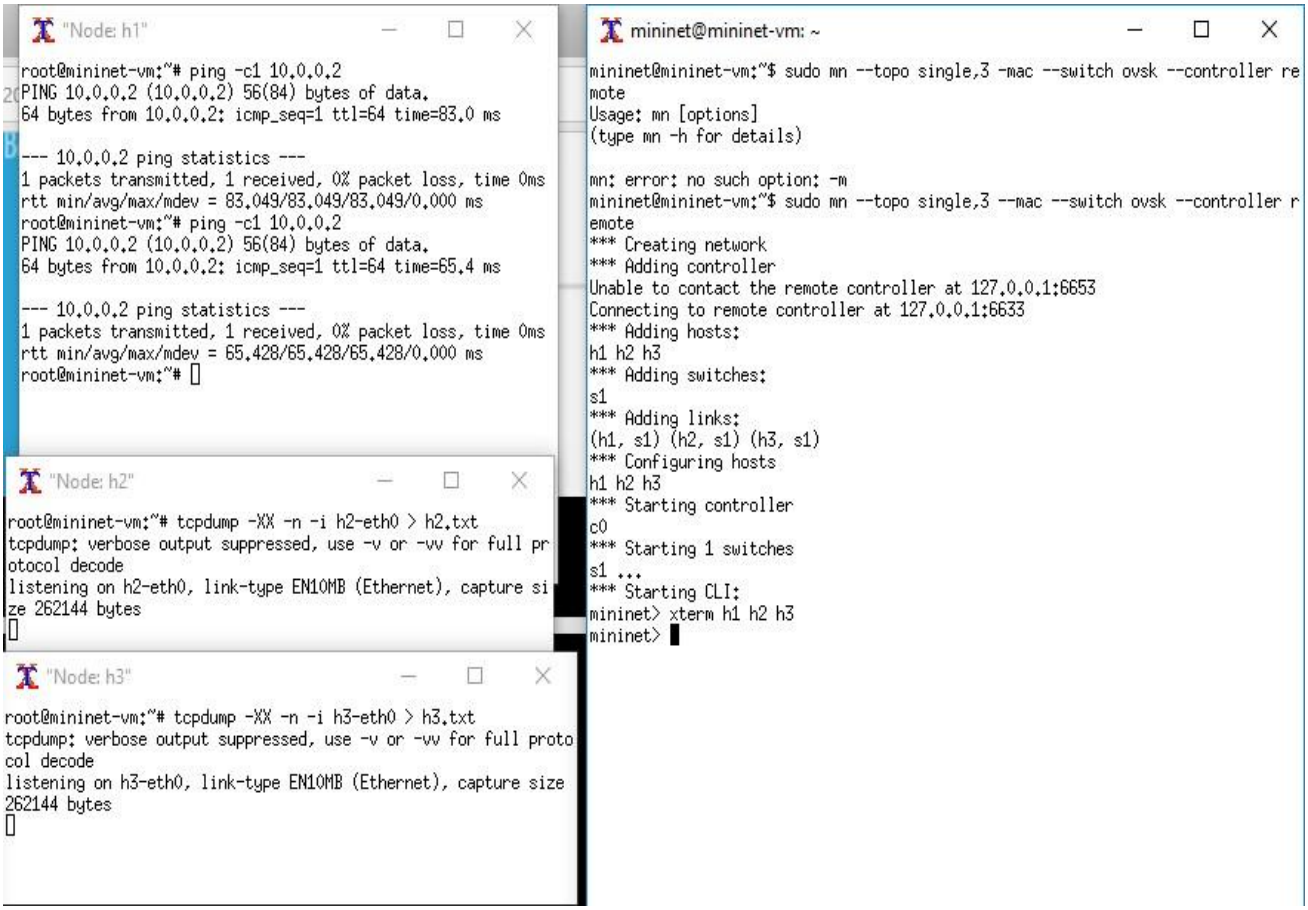


2. Of_tutorial.py

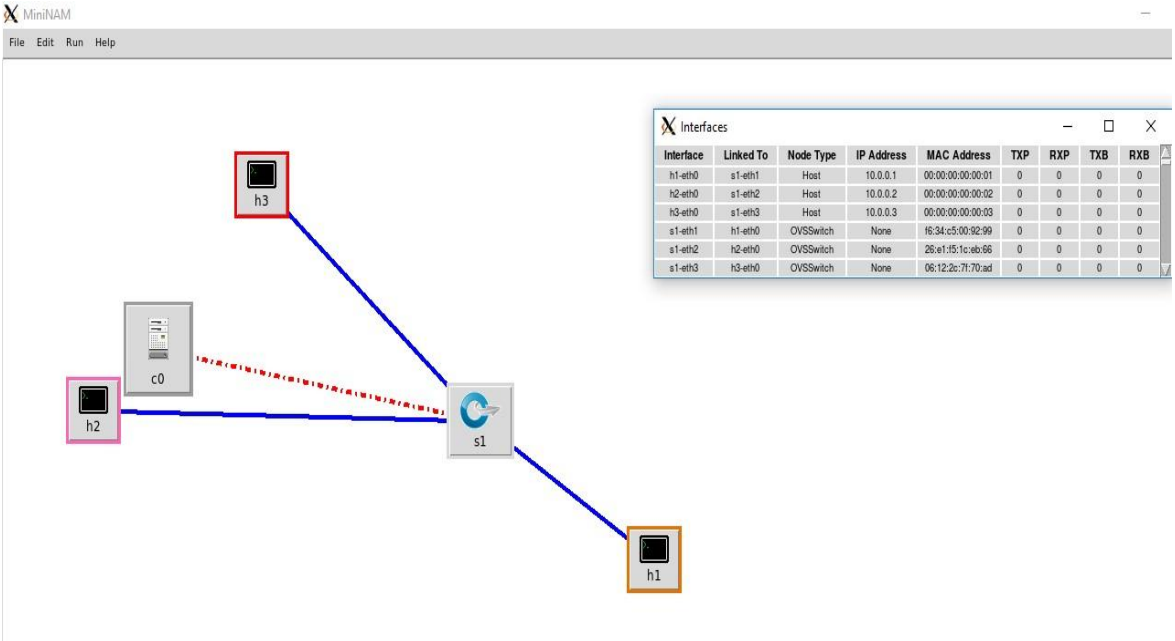
```
mininet@mininet-vm: ~/pox/pox/misc

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
    controller due to a table-miss.
    """
    msg = of.ofp_packet_out()
    msg.data = packet_in

    # Add an action to send to the specified port
    action = of.ofp_action_output(port = out_port)
    msg.actions.append(action)

    # Send message to switch
    self.connection.send(msg)


  def act_like_hub (self, packet, packet_in):
    """
    Implement hub-like behavior -- send all packets to all ports besides
    the input port.
    """

    # We want to output to all ports -- we do that using the special
    # OFPP_ALL port as the output port.  (We could have also used
    # OFPP_FLOOD.)
    self.resend_packet(packet_in, of.OFPP_ALL)

    # Note that if we didn't get a valid buffer_id, a slightly better
    # implementation would check that we got the full data before
```

3. Membuat 3 host dengan 1 switch

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



4. h1 ping –c1 10.0.0.2

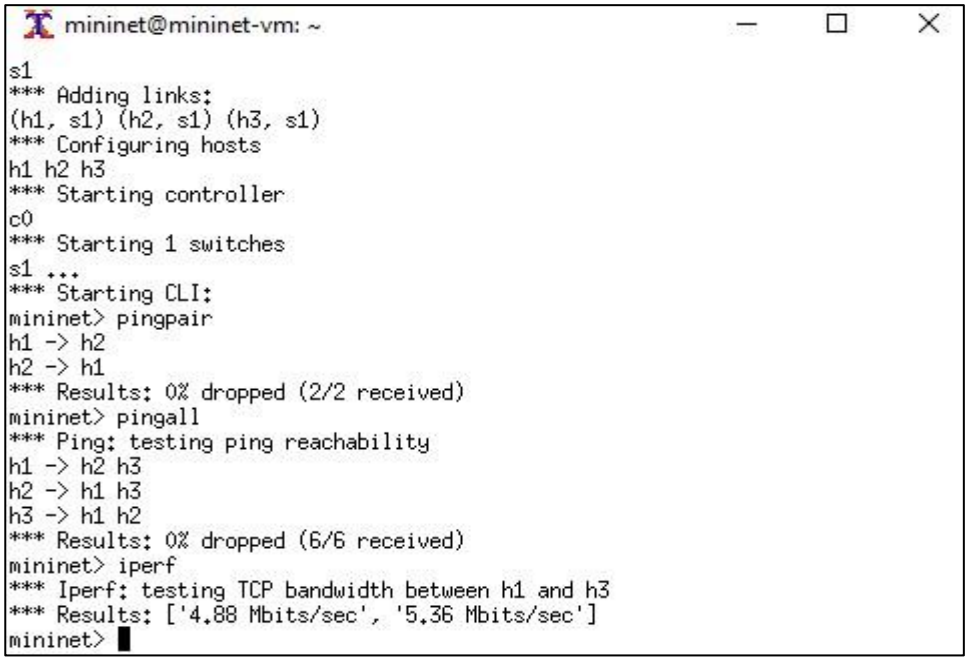


5. Manual Entry

mininet> pingpair

mininet> pingall

mininet> iperf



6. Of_tutorial.py

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

```
  def act_like_hub (self, packet, packet_in):
    """
    Implement hub-like behavior -- send all packets to all ports besides
    the input port.
    """

    # We want to output to all ports -- we do that using the special
    # OFPP_ALL port as the output port.  (We could have also used
    # OFPP_FLOOD.)
    self.resend_packet(packet_in, of.OFPP_ALL)

    # Note that if we didn't get a valid buffer_id, a slightly better
    # implementation would check that we got the full data before
    # sending it (len(packet_in.data) should be == packet_in.total_len)).


  def act_like_switch (self, packet, packet_in):
    self.mac_to_port[packet.src] = packet, packet_in.in_port
        if packet.dst in self.mac_to_port:
                print("Packet sent to Control Plane")

        self.resend_packet(packet_in,
                                self.mac_to_port[packet.dst])

        # log.debug(*Installing flow...*)
msg = of.ofp_flow_mod()
msg.match.dl_dst = packet.dst
# msg.match = of.ofp_match.from_packet(packet)

        msg.actions.append(of.ofp_action_output(port=self.mac_to_port[packet.dst]))
        self.connection.send(msg)
        else:
                self.resend_packet(packet_in, of.OFPP_ALL)

    """ # DELETE THIS LINE TO START WORKING ON THIS #


  def _handle_PacketIn (self, event):
    """
    Handles packet in messages from the switch.
    """

    packet = event.parsed # This is the parsed packet data.
    if not packet.parsed:
      log.warning("Ignoring incomplete packet")
      return

    packet_in = event.ofp # The actual ofp_packet_in message.

    # Comment out the following line and uncomment the one after
    # when starting the exercise.
    self.act_like_hub(packet, packet_in)
    #self.act_like_switch(packet, packet_in)
```

Github Link

https://github.com/novalinasimbolon

https://github.com/desi31

https://github.com/soniaelisa98

https://github.com/chyntiaclaudia

https://github.com/tirzapks

https://github.com/ayopushrank