

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

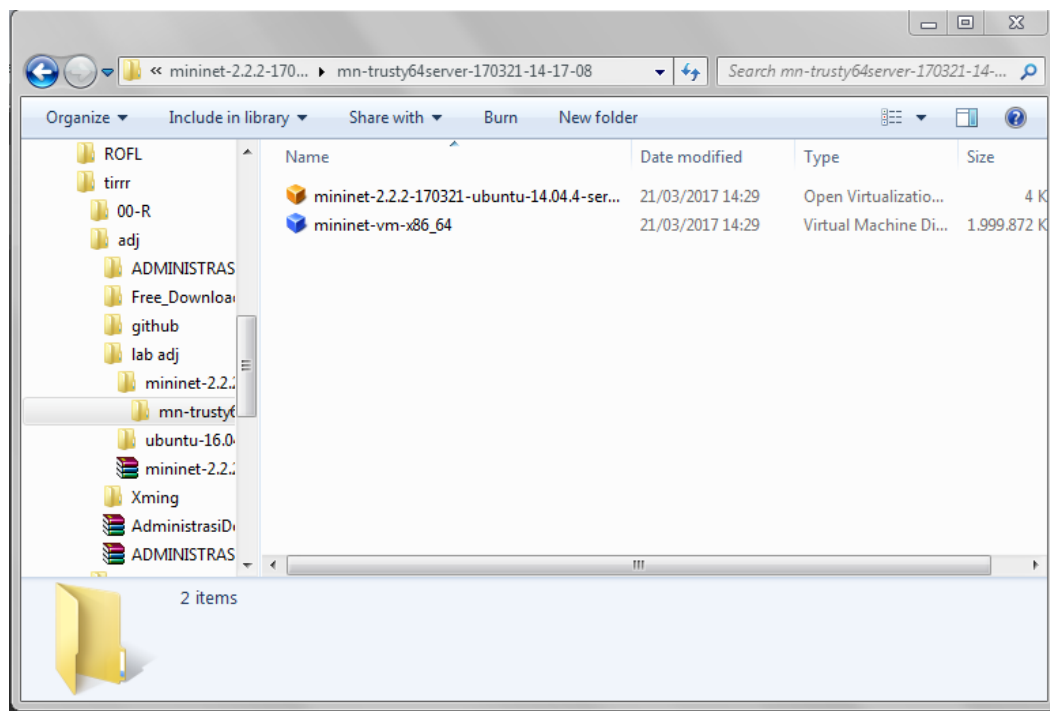
2018

MININET

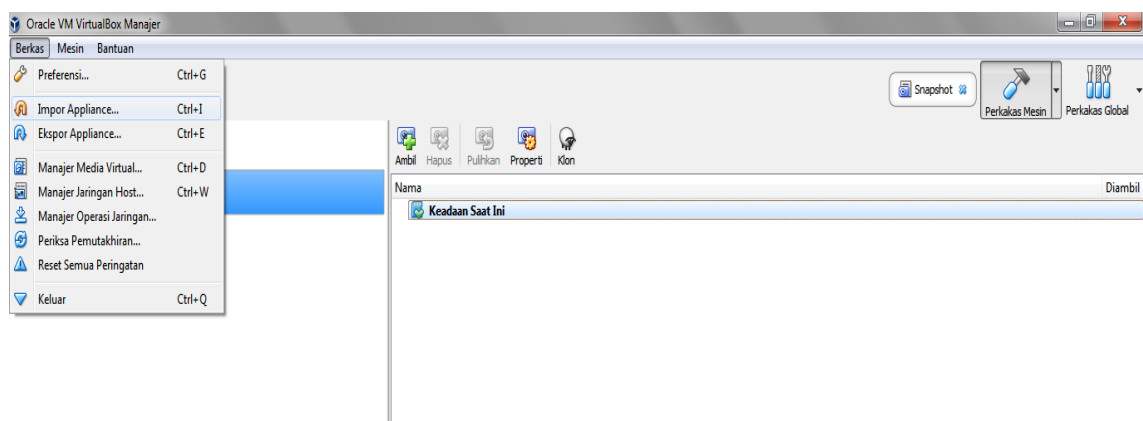
Mininet adalah emulator berbasis CLI yang digunakan untuk membuat sebuah topologi jaringan pada *Software Defined Network*. Dalam Mininet, terdapat beberapa topologi bawaan yang dapat langsung digunakan dengan menggunakan perintah tertentu, antara lain seperti topologi *single*, *tree*, dan *linear*.

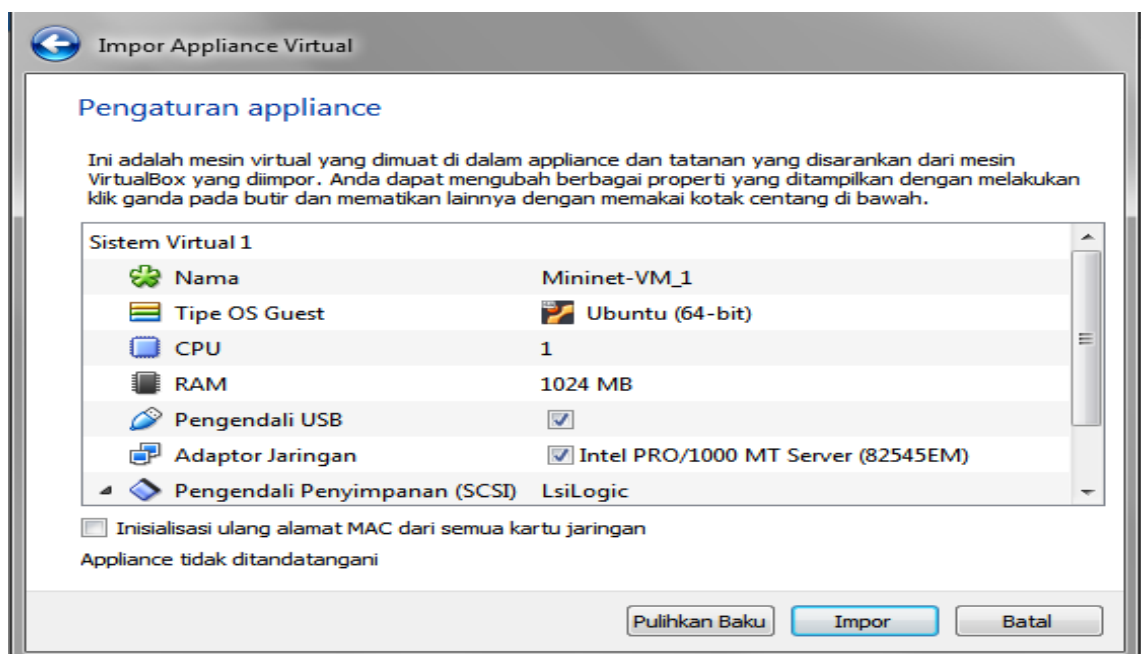
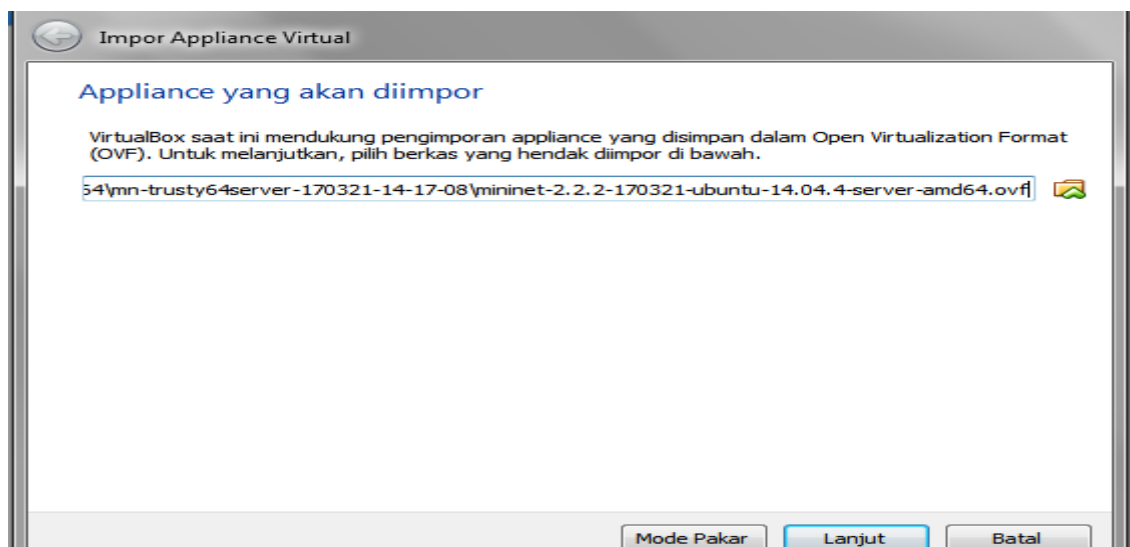
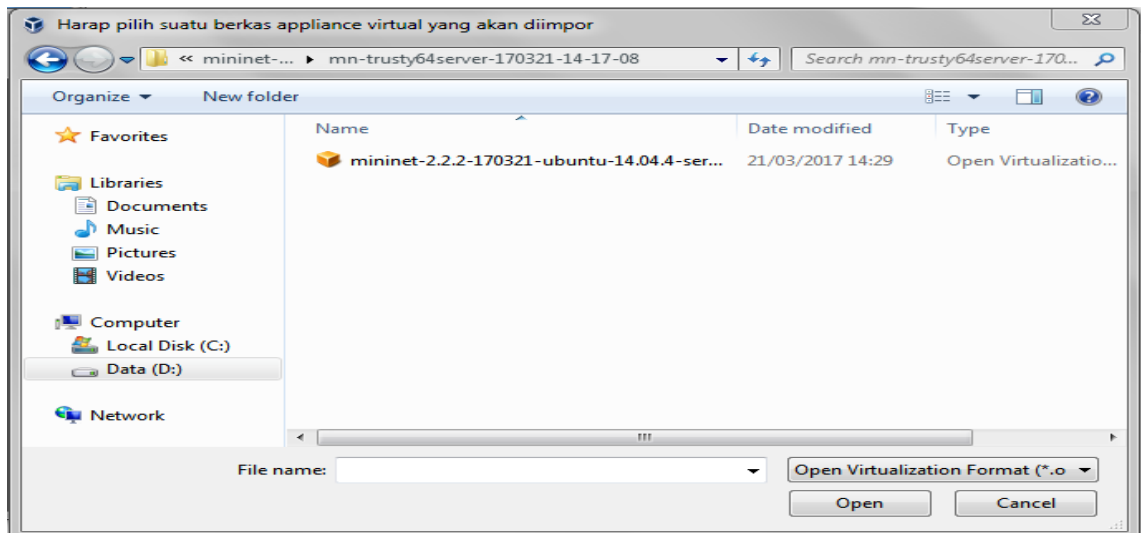
Download dan Import Mininet pada Virtual Machine (VM)

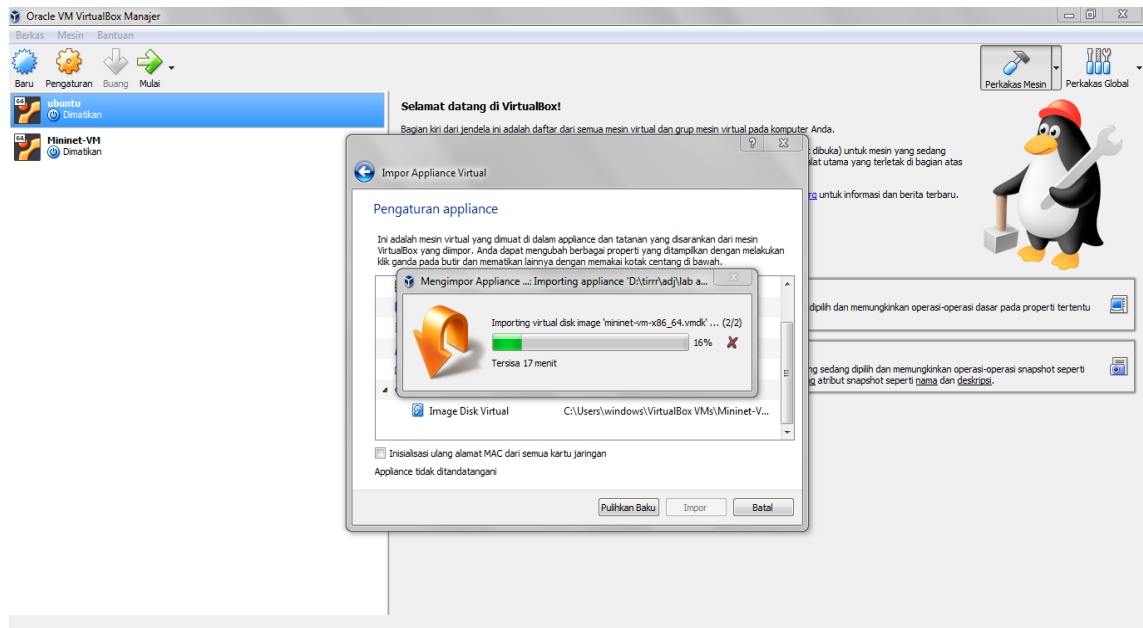
1. Download Mininet



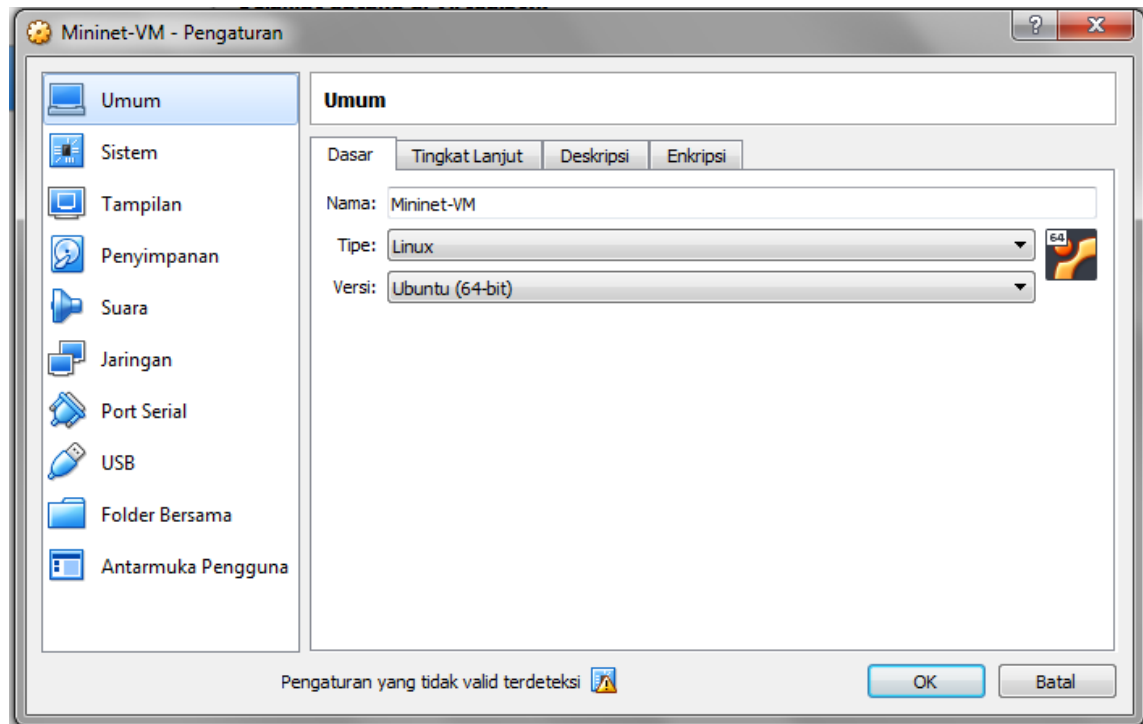
2. Import Mininet pada Virtual Machine (Virtual Box)

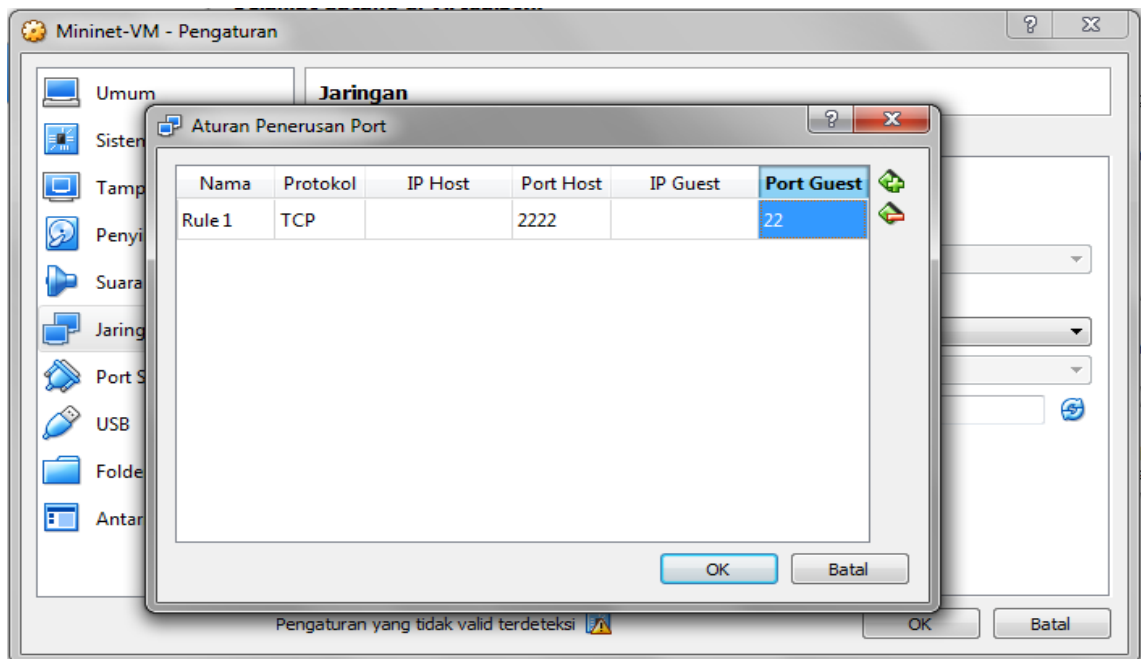
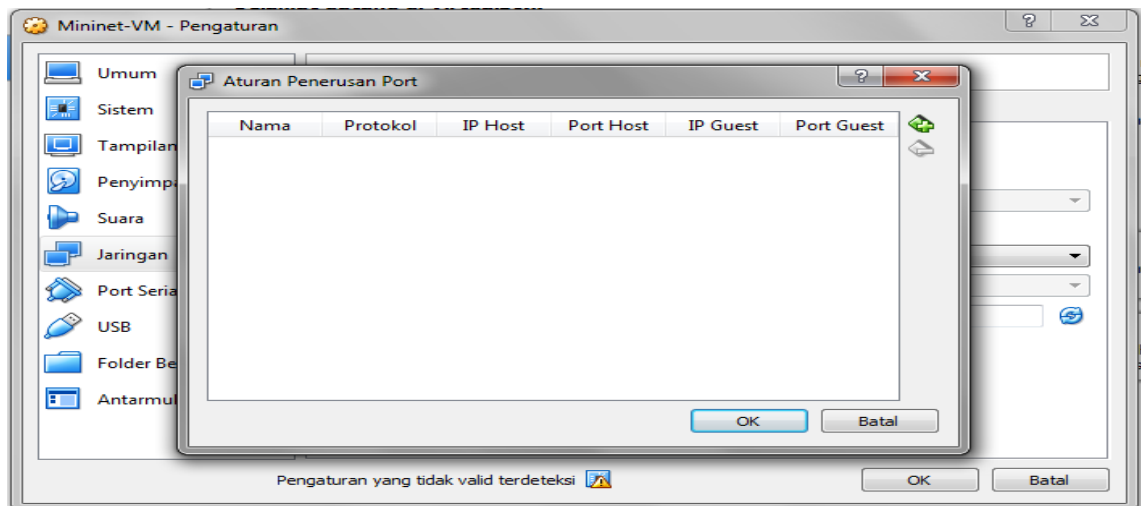
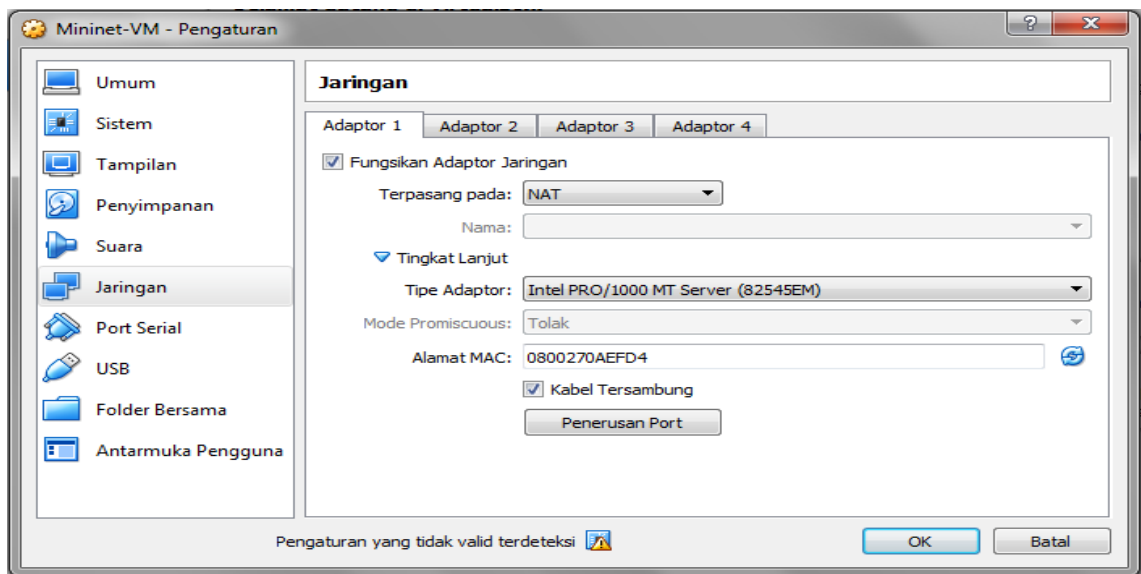




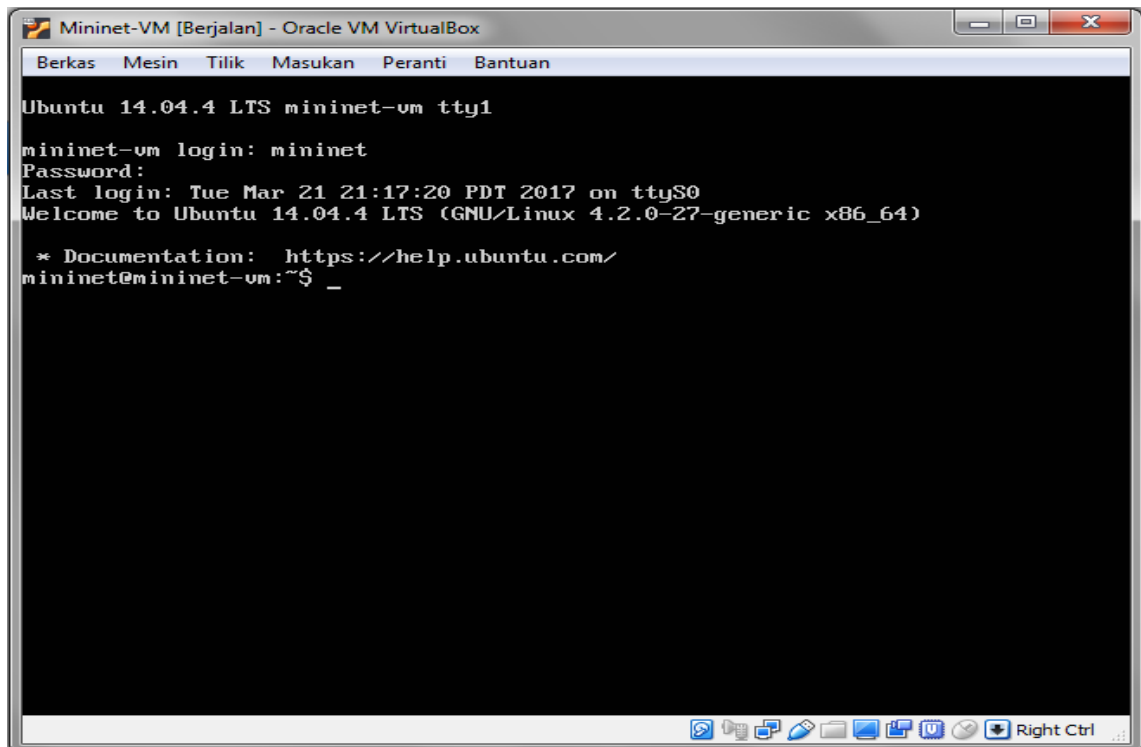
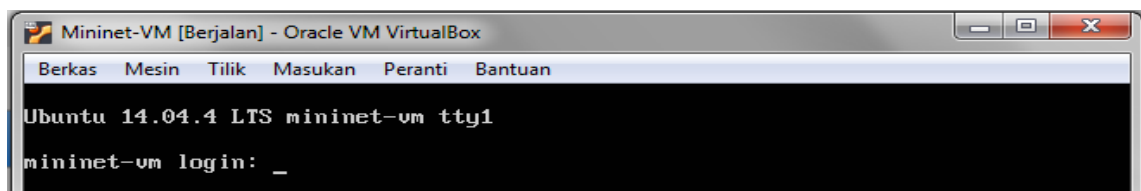
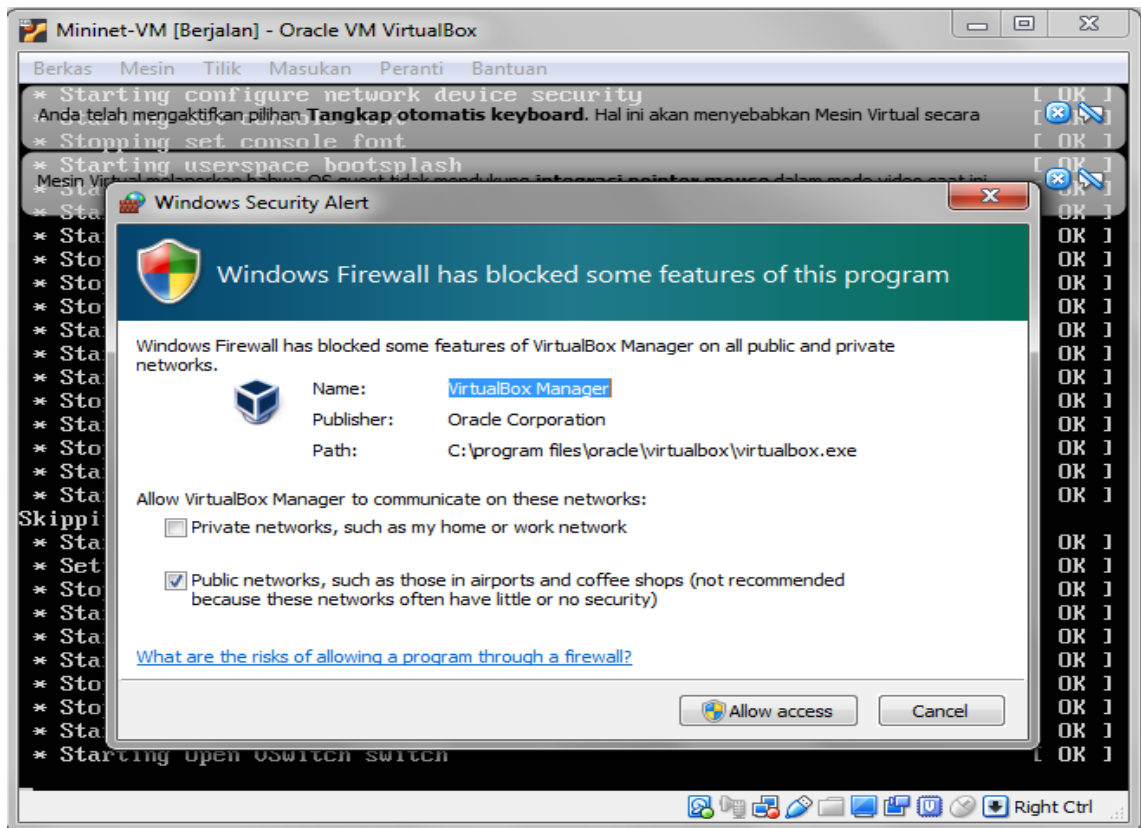


3. Konfigurasi (Pengaturan) Akses Mininet



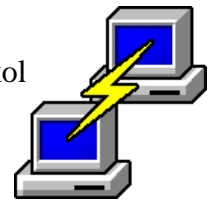


4. Mulai menjalankan Mininet



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

```
C:\Windows\system32\cmd.exe

D:\tirror\adj>putty.exe -p -l 2222 -l tirza localhost
D:\tirror\adj>putty.exe -X -P 2222 -l mininet localhost
D:\tirror\adj>
```

2. Aplikasi PuTTY berjalan

```
mininet@mininet-vm: ~
Using username "mininet".
mininet@localhost's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

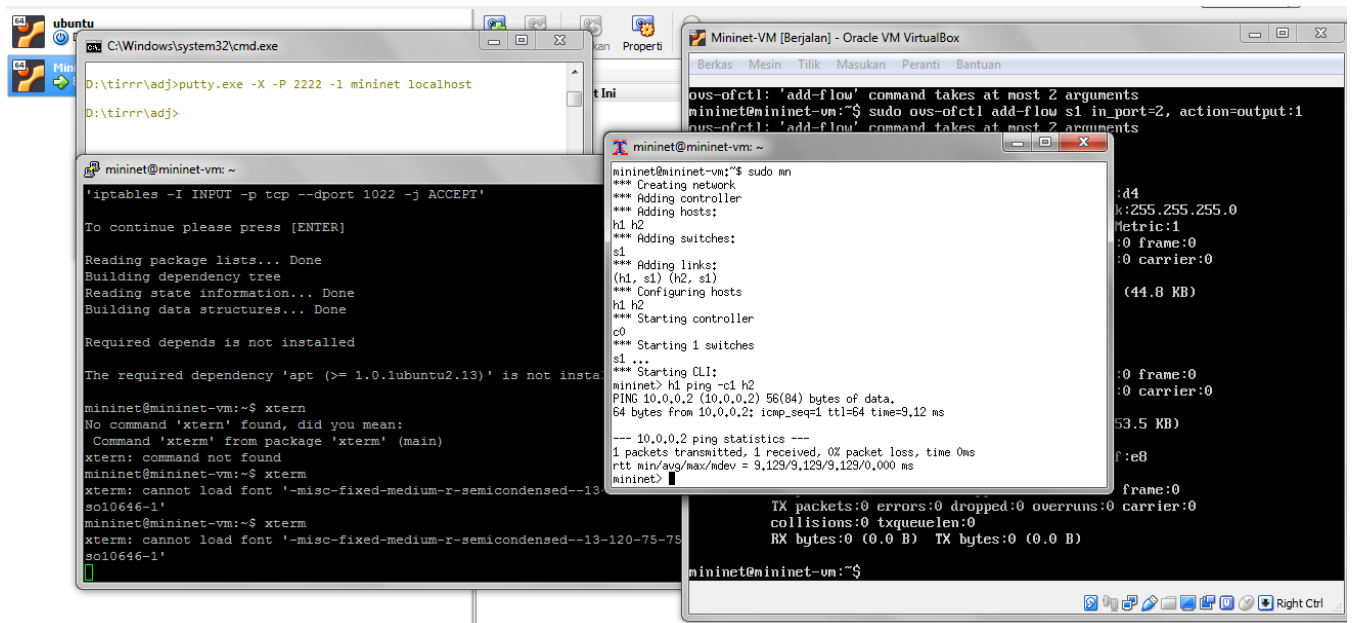
Last login: Sat Apr  7 01:32:17 2018
/usr/bin/xauth:  file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$
```

Menjalankan (run) Mininet

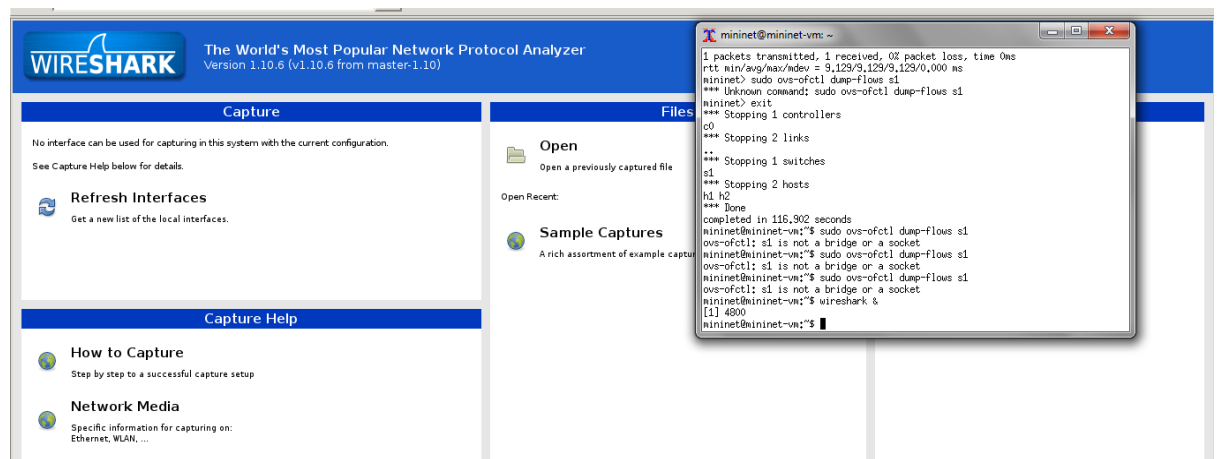
1. `mininet@mininet-vm:~$ sudo apt-get install tree`

```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ 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>
```

2. `h1 ping -c1 h2`



3. Test Wireshark

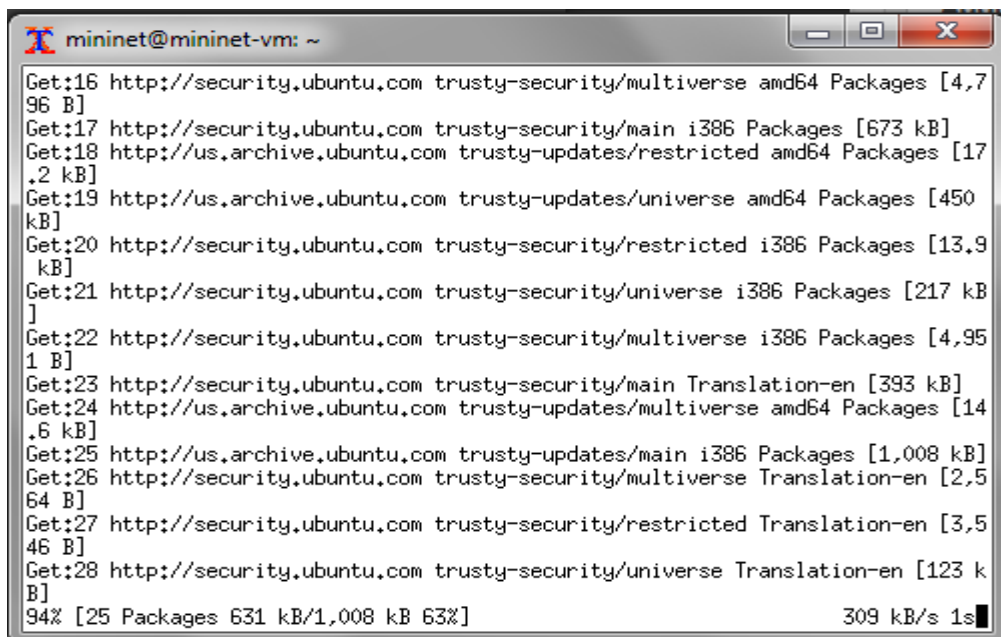
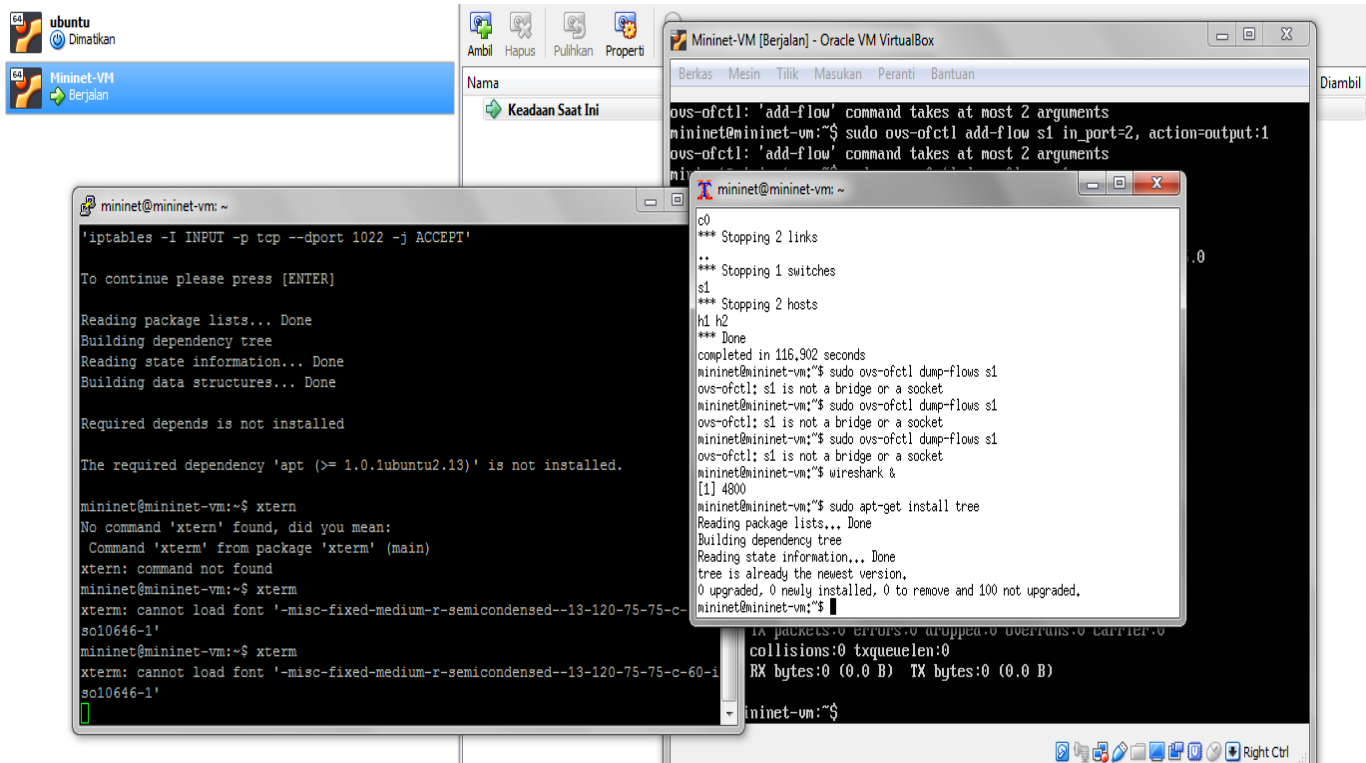


4. Update, Install, Check Tree dan Packages

```
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: ~
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.501 seconds
mininet@mininet-vm:~$ sudo mn --test iperf
*** 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 ...
*** Waiting for switches to connect
s1
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['5.57 Gbits/sec', '5.58 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 11.554 seconds
mininet@mininet-vm:~$
```

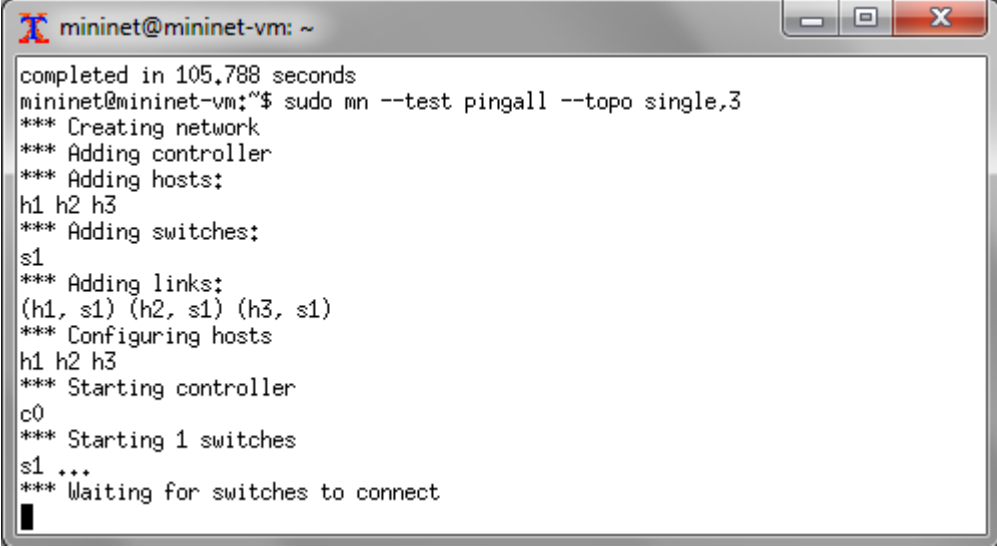
mininet@mininet-vm:~\$ sudo mn --link tc,bw=10,delay=10ms

```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ sudo mn --link tc,bw=10,delay=10ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (h1, s1) (10.00Mbit 10ms delay) (1
0.00Mbit 10ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (10.00Mbit 10ms delay) (10.00Mbit 10ms delay)
*** Starting CLI:
mininet>
```

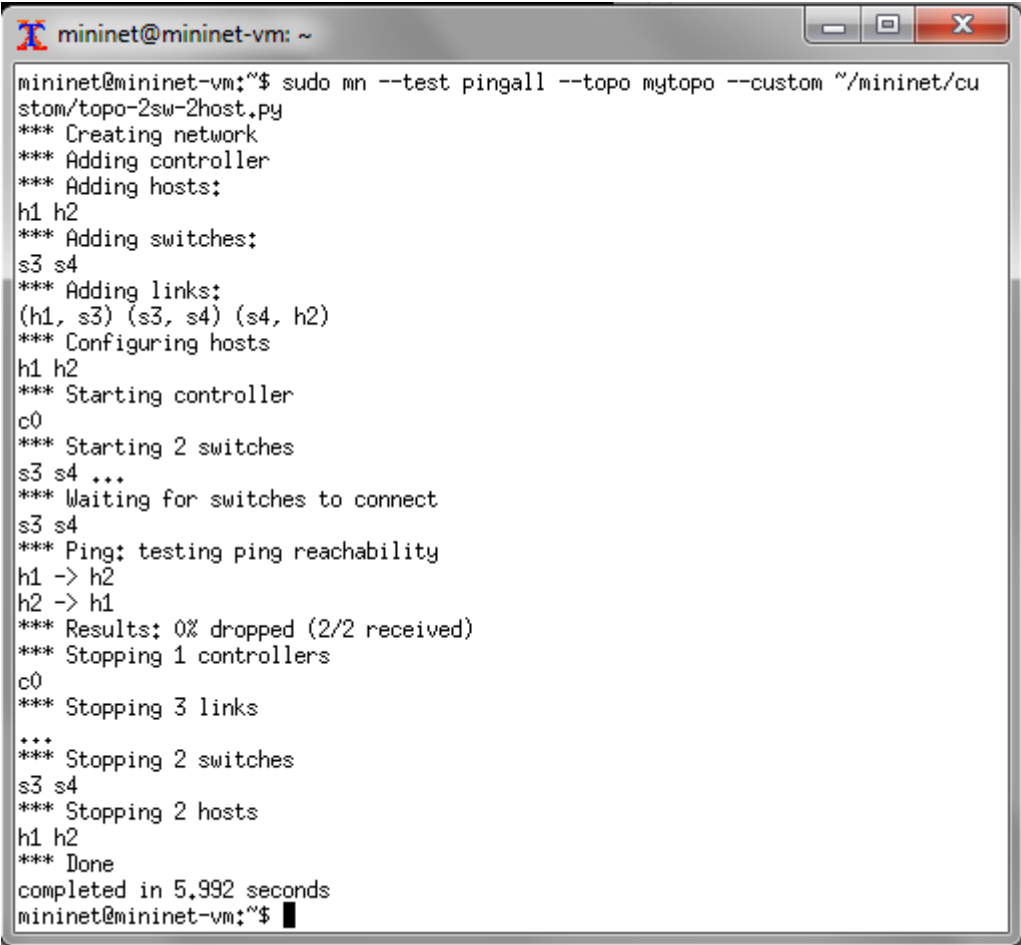
6. Ping all single dan ping all topo

```
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
```



```
mininet@mininet-vm: ~
completed in 105,788 seconds
mininet@mininet-vm:~$ 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
```



```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ sudo mn --test pingall --topo mytopo --custom ~/mininet/cu
stom/topo-2sw-2host.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Waiting for switches to connect
s3 s4
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 3 links
***
*** Stopping 2 switches
s3 s4
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.992 seconds
mininet@mininet-vm:~$
```

7. Create Network

```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ sudo mn --switch ovs --controller ref --topo tree,depth=2,fanout=3 --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, h1) (s2, h2) (s2, h3) (s3, h4) (s3, h5) (s3, h6) (s4, h7) (s4, h8) (s4, h9)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** 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 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
*** Stopping 1 controllers
c0
*** Stopping 12 links
*****
*** Stopping 4 switches
s1 s2 s3 s4
*** Stopping 9 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Done
completed in 8.265 seconds
mininet@mininet-vm:~$
```

8. Interacting with a Network

```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ #Interacting with a Network
mininet@mininet-vm:~$ sudo mn --link tc,bw=5,delay=10ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(5.00Mbit 10ms delay) (5.00Mbit 10ms delay) (h1, s1) (5.00Mbit 10ms delay) (5.00Mbit 10ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (5.00Mbit 10ms delay) (5.00Mbit 10ms delay)
*** Starting CLI:
mininet>
```

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: ~
c0
*** Stopping 2 links
**
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 223.185 seconds
mininet@mininet-vm:~$ git clone https://github.com/uccmisl/MiniNAM.git
fatal: destination path 'MiniNAM' already exists and is not an empty directory.
mininet@mininet-vm:~$ tree -L 1 -d
.
├── loxigen
├── MiniNAM
├── mininet
├── oflops
├── oftest
├── openflow
└── pox

7 directories
mininet@mininet-vm:~$
```

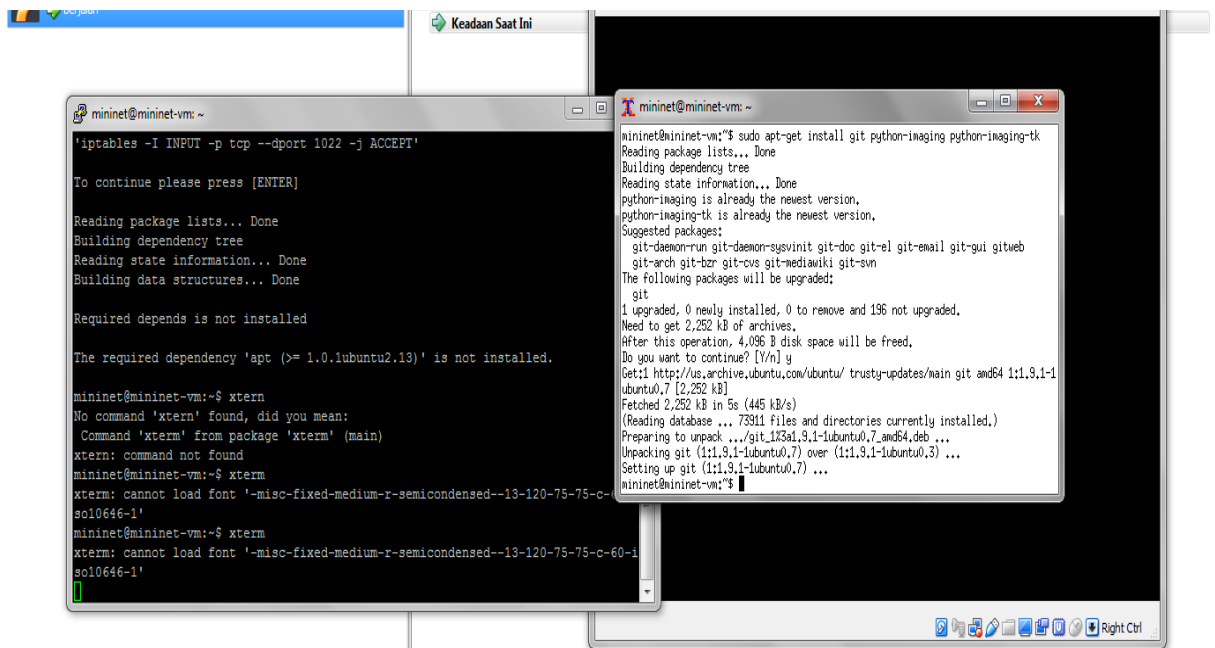


```
mininet@mininet-vm:~$ tree MiniNAM
MiniNAM
├── conf.config
├── Examples
│   ├── LoadBalancer
│   │   ├── install.sh
│   │   ├── MiniNAM.py
│   │   ├── paping
│   │   └── README.md
│   ├── NAT
│   │   ├── badNAT.py
│   │   ├── conf.config
│   │   ├── goodNAT.py
│   │   ├── MiniNAM.py
│   │   └── README.md
│   └── Routing
│       ├── MiniNAM.py
│       ├── README.md
│       ├── simple_switch_13.py
│       ├── simple_switch_stp_13.py
│       └── spanning_tree.py
├── LICENSE
├── MiniNAM.py
└── README.md

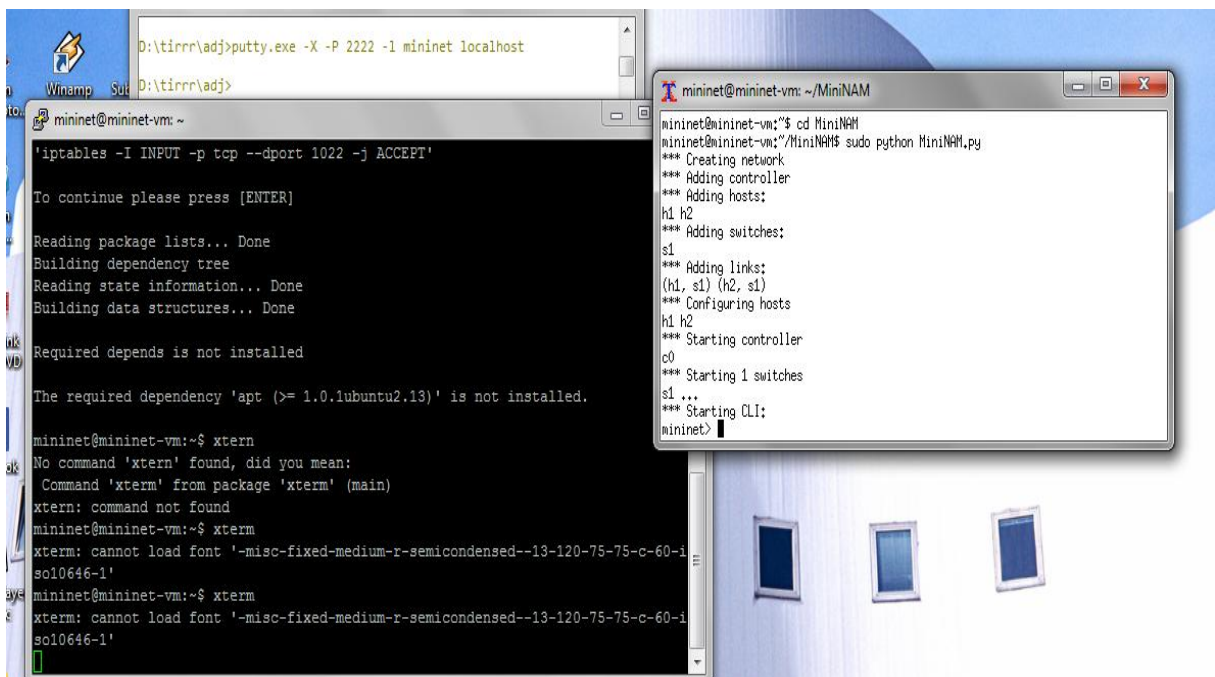
4 directories, 18 files
mininet@mininet-vm:~$
```

```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ sudo apt-get install python-imaging  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-imaging is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 197 not upgraded.  
mininet@mininet-vm:~$
```

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

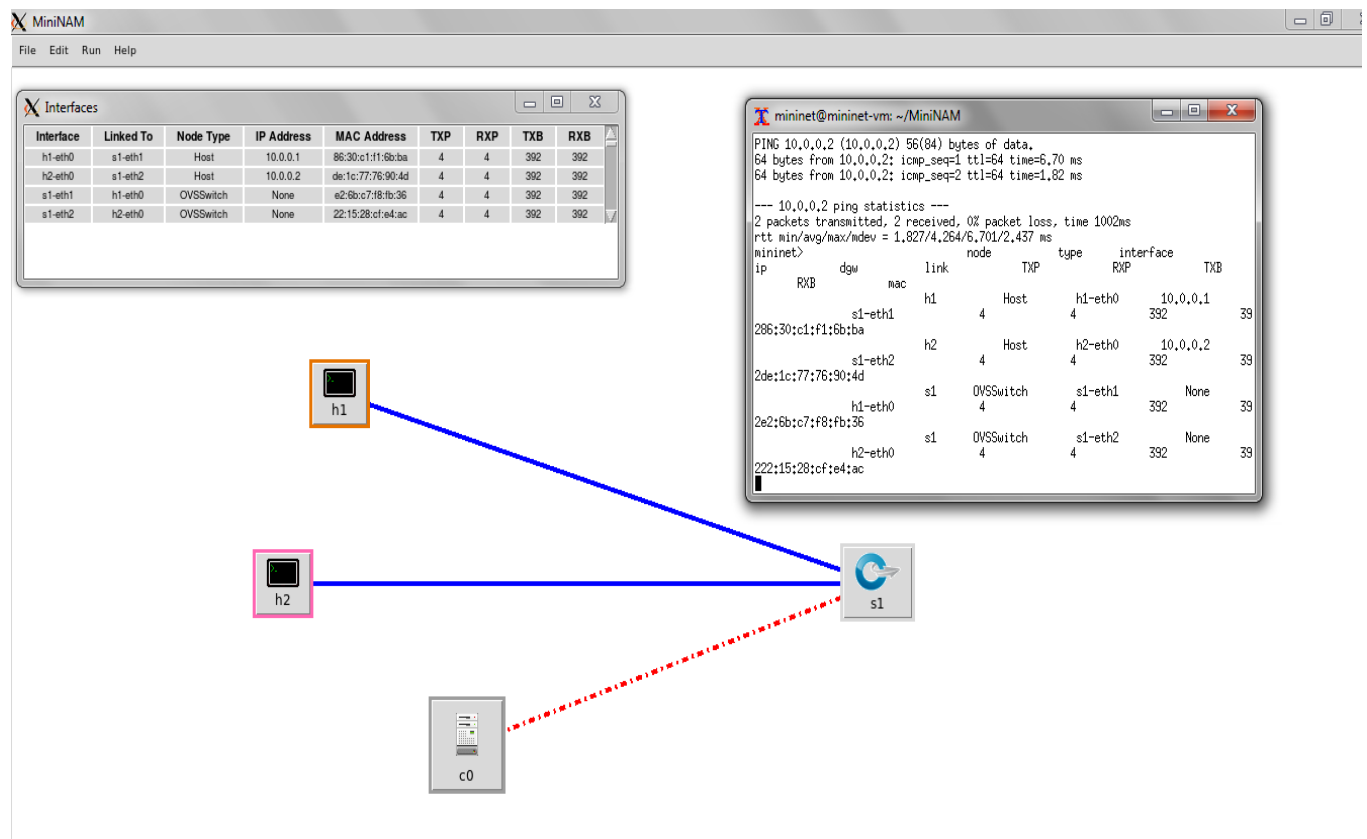


```
mininet@mininet-vm: ~  
'iptables -I INPUT -p tcp --dport 1022 -j ACCEPT'  
To continue please press [ENTER]  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Building data structures... Done  
Required depends is not installed  
The required dependency 'apt (>= 1.0.1ubuntu2.13)' is not installed.  
mininet@mininet-vm:~$ xterm  
No command 'xterm' found, did you mean:  
Command 'xterm' from package 'xterm' (main)  
xterm: command not found  
mininet@mininet-vm:~$ xterm  
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-1  
so10646-1'  
mininet@mininet-vm:~$ xterm  
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-1  
so10646-1'  
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ sudo apt-get install git python-imaging python-imaging-tk  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-imaging is already the newest version.  
python-imaging-tk is already the newest version.  
Suggested packages:  
git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitweb  
git-arch git-bzr git-cvs git-mediawiki git-svn  
The following packages will be upgraded:  
git  
1 upgraded, 0 newly installed, 0 to remove and 136 not upgraded.  
Need to get 2,252 kB of archives.  
After this operation, 4,096 B disk space will be freed.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1  
ubuntu0.7 [2,252 kB]  
Fetched 2,252 kB in 5s (445 kB/s)  
(Reading database ... 73911 files and directories currently installed.)  
Preparing to unpack .../git_1:1.9.1-1ubuntu0.7_amd64.deb ...  
Unpacking git (1:1.9.1-1ubuntu0.7) over (1:1.9.1-1ubuntu0.3) ...  
Setting up git (1:1.9.1-1ubuntu0.7) ...  
mininet@mininet-vm:~$
```



```
mininet@mininet-vm: ~  
'iptables -I INPUT -p tcp --dport 1022 -j ACCEPT'  
To continue please press [ENTER]  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Building data structures... Done  
Required depends is not installed  
The required dependency 'apt (>= 1.0.1ubuntu2.13)' is not installed.  
mininet@mininet-vm:~$ xterm  
No command 'xterm' found, did you mean:  
Command 'xterm' from package 'xterm' (main)  
xterm: command not found  
mininet@mininet-vm:~$ xterm  
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-1  
so10646-1'  
mininet@mininet-vm:~$ xterm  
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-1  
so10646-1'  
mininet@mininet-vm: ~/MiniNAM  
mininet@mininet-vm:~$ cd MiniNAM  
mininet@mininet-vm:~/MiniNAM$ sudo python MiniNAM.py  
*** 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>
```

2. h1 ping -c2 h2



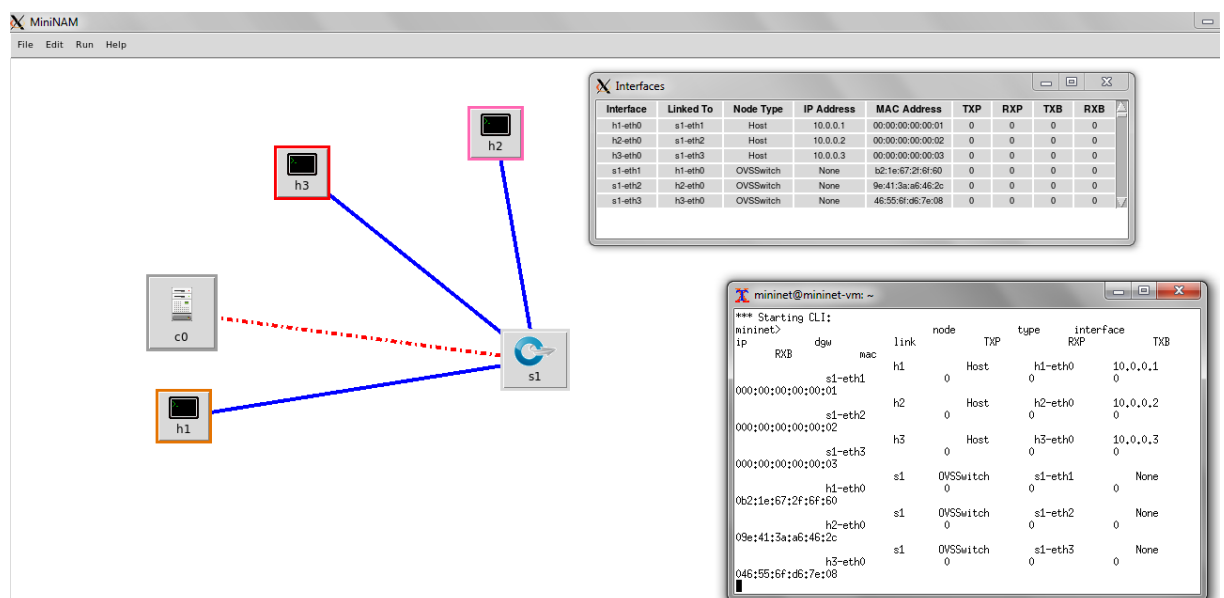
OPEN FLOW

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
```

```
mininet@mininet-vm: ~
mininet@mininet-vm:~$ sudo python ~/MiniNAM/MiniNAM.py --topo single,3 --mac --swi
tch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6653
*** 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>
```

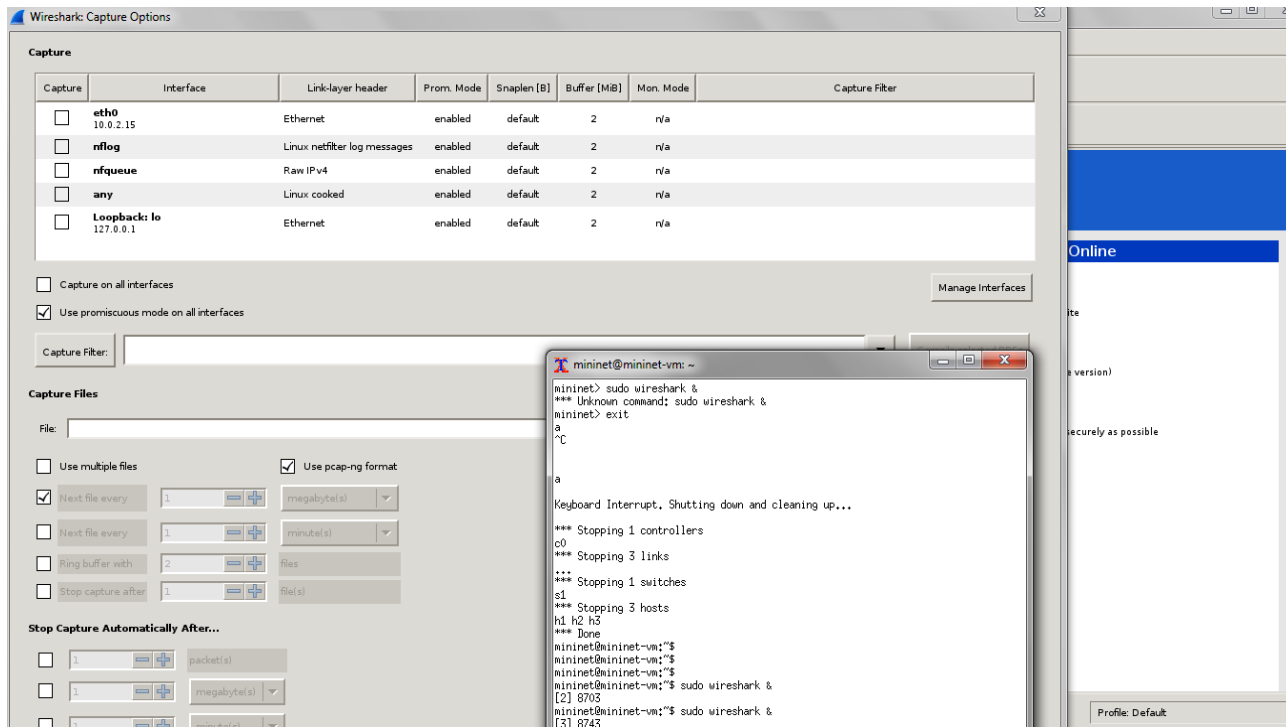
```
mininet@mininet-vm: ~
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=6.21 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.713 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.076 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.076/2.334/6.215/2.756 ms
mininet>
```

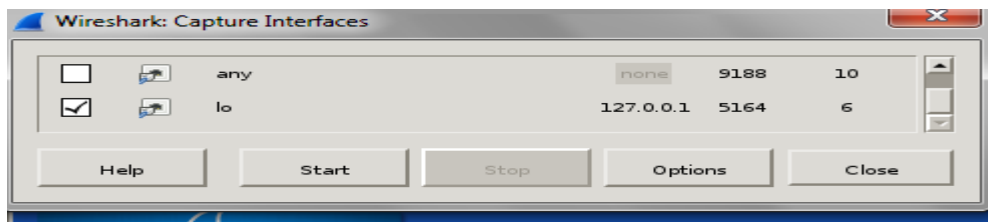
```
mininet@mininet-vm: ~
rtt min/avg/max/mdev = 0.074/3.474/9.584/4.329 ms
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 in_port=2,actions=output:1
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=47.957s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
  idle_age=47, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:01
,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=52.979s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
  idle_age=52, priority=65535,arp,in_port=2,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:1
  cookie=0x0, duration=47.96s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
  idle_age=47, priority=65535,arp,in_port=2,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=1 actions=output:1
  cookie=0x0, duration=18.725s, table=0, n_packets=0, n_bytes=0, idle_age=18, in_port=1
actions=output:2
  cookie=0x0, duration=8.331s, table=0, n_packets=0, n_bytes=0, idle_age=8, in_port=2
actions=output:1
  cookie=0x0, duration=52.976s, table=0, n_packets=3, n_bytes=294, idle_timeout=60,
  idle_age=50, priority=65535,icmp,in_port=1,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:2
  cookie=0x0, duration=52.972s, table=0, n_packets=3, n_bytes=294, idle_timeout=60,
  idle_age=50, priority=65535,icmp,in_port=2,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:1
mininet>
```

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)

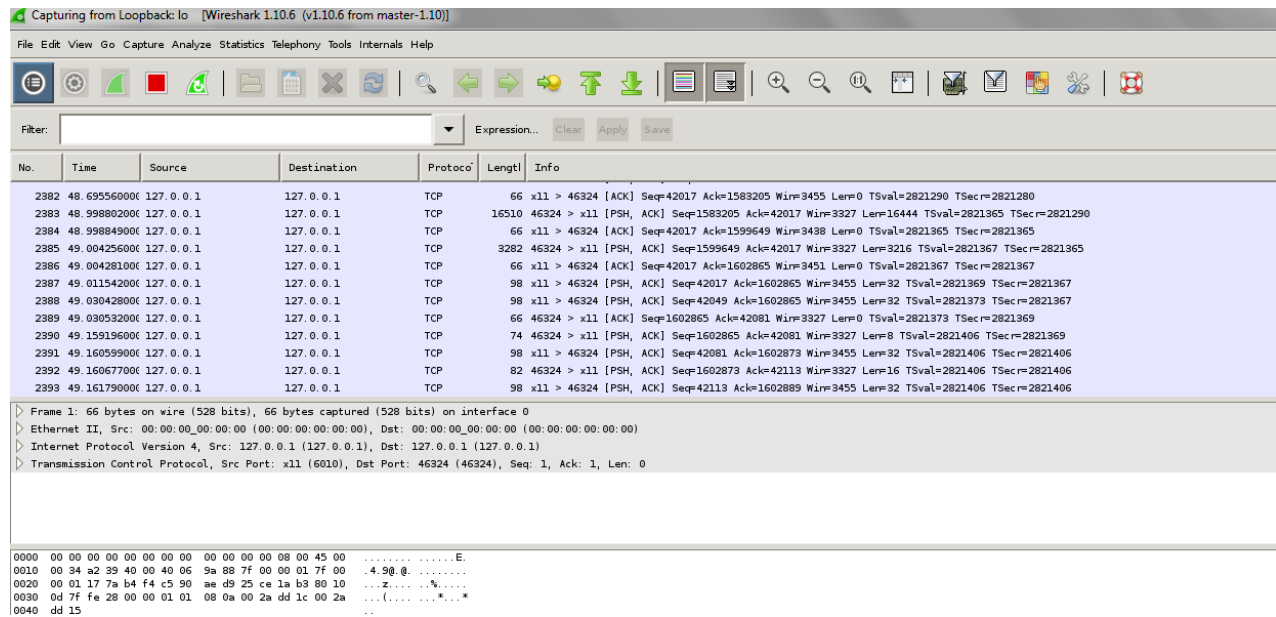
mininet@mininet-vm:~\$ controller ptcp: &

```
mininet@mininet-vm:~$ controller ptcp: &
[4] 9119
mininet@mininet-vm:~$ Apr 07 03:45:20|00001|vconn_tcp|ERR|ptcp:: bind: Address already 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 remote
*** 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>
```

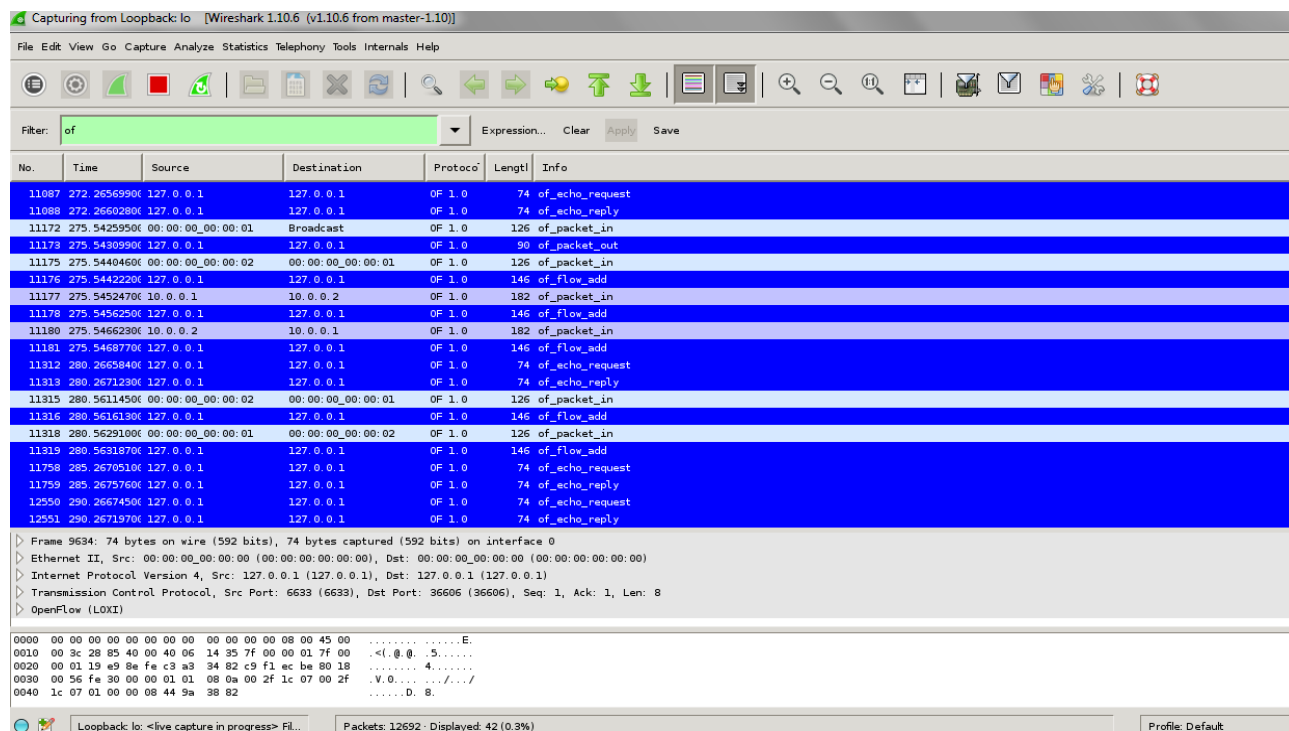
```
mininet@mininet-vm: ~
mininet> h1 ip -s -s neigh flush all
Nothing to flush.
mininet> h2 ip -s -s neigh flush all
Nothing to flush.
mininet> sh ovs-ofctl del-flows s1
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=6.39 ms

--- 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>
```

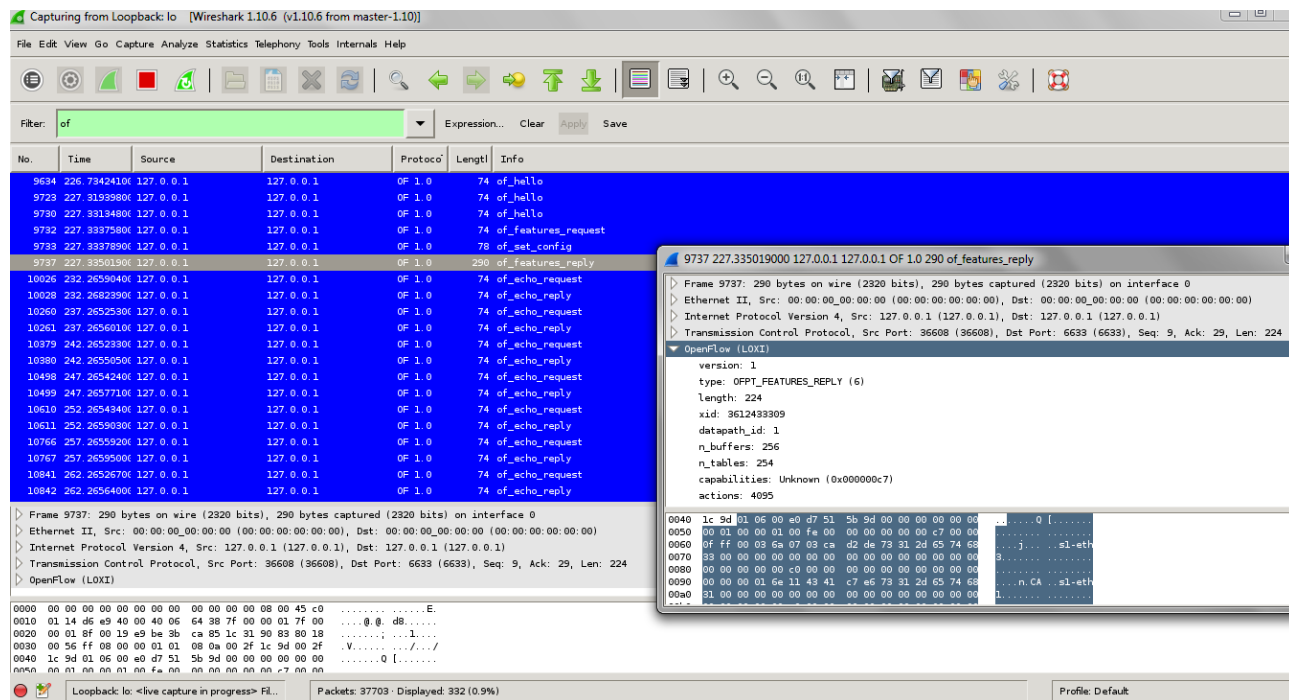
Berikut tampilan pada wireshark;



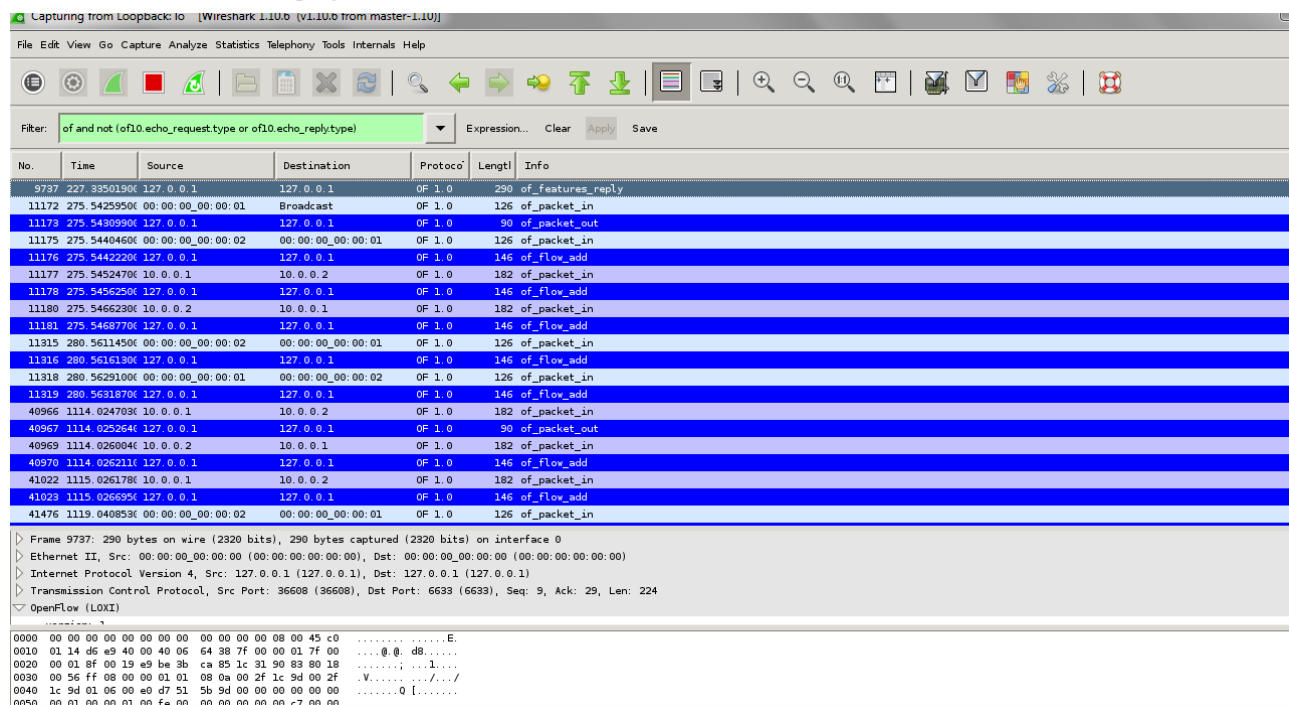
5. Lakukan Filter 'OF' pada Wireshark



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

```

mininet@mininet-vm: ~
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=244.939s, table=0, n_packets=10, n_bytes=420, idle_timeout=60, idle_age=4, priority=65535, arp, in_port=1, vlan_tci=0x0000, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:02, arp_spa=10.0.0.1, arp_tpa=10.0.0.2, arp_op=2 actions=output:2
  cookie=0x0, duration=244.941s, table=0, n_packets=10, n_bytes=420, idle_timeout=60, idle_age=4, priority=65535, arp, in_port=2, 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=1 actions=output:1
  cookie=0x0, duration=248.955s, table=0, n_packets=240, n_bytes=23520, idle_timeout=60, idle_age=9, priority=65535, icmp, in_port=1, 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:2
  cookie=0x0, duration=249.956s, table=0, n_packets=241, n_bytes=23618, idle_timeout=60, idle_age=9, priority=65535, icmp, in_port=2, 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:1
mininet>

```

9. Benchmark kernel- vs. user-space

```

mininet@mininet-vm: ~
: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:1
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
.e*** Results: ['6.26 Gbits/sec', '6.26 Gbits/sec']
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
***
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 1202.889 seconds
mininet@mininet-vm:~$ sudo mn --topo single,3 --controller remote --switch user
*** 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> sh ovs-ofctl add-flow s1 in_port=1,actions=output:3
ovs-ofctl: s1 is not a bridge or a socket
mininet> sh ovs-ofctl add-flow s1 in_port=3,actions=output:1
ovs-ofctl: s1 is not a bridge or a socket
mininet>

```


POX CONTROLLER

1. Mengaktifkan POX *Controller*

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

```
mininet@mininet-vm: ~/pox/pox/forwarding  
mininet@mininet-vm:~$ cd pox  
mininet@mininet-vm:~/pox$ cd pox  
mininet@mininet-vm:~/pox/pox$ cd misc  
mininet@mininet-vm:~/pox/pox/misc$ ll  
total 76  
drwxrwxr-x 3 mininet mininet 4096 Mar 21 2017 ./  
drwxrwxr-x 15 mininet mininet 4096 Mar 21 2017 ../  
-rw-rw-r-- 1 mininet mininet 1240 Mar 21 2017 cbench.py  
-rw-rw-r-- 1 mininet mininet 1079 Mar 21 2017 full_payload.py  
-rw-rw-r-- 1 mininet mininet 5214 Mar 21 2017 gephi_topo.py  
-rw-rw-r-- 1 mininet mininet 689 Mar 21 2017 __init__.py  
-rw-rw-r-- 1 mininet mininet 10251 Mar 21 2017 ip_loadbalancer.py  
-rw-rw-r-- 1 mininet mininet 3794 Mar 21 2017 mac_blocker.py  
-rw-rw-r-- 1 mininet mininet 14375 Mar 21 2017 nat.py  
-rw-rw-r-- 1 mininet mininet 4582 Mar 21 2017 of_tutorial.py  
-rw-rw-r-- 1 mininet mininet 2096 Mar 21 2017 pidfile.py  
drwxrwxr-x 2 mininet mininet 4096 Mar 21 2017 telnetd/  
mininet@mininet-vm:~/pox/pox/misc$ cd  
mininet@mininet-vm:~$ cd pox/pox/forwarding  
mininet@mininet-vm:~/pox/pox/forwarding$ ll  
total 96  
drwxrwxr-x 2 mininet mininet 4096 Mar 21 2017 ./  
drwxrwxr-x 15 mininet mininet 4096 Mar 21 2017 ../  
-rw-rw-r-- 1 mininet mininet 1092 Mar 21 2017 hub.py  
-rw-rw-r-- 1 mininet mininet 651 Mar 21 2017 __init__.py  
-rw-rw-r-- 1 mininet mininet 4426 Mar 21 2017 l2_flowvisor.py  
-rw-rw-r-- 1 mininet mininet 6692 Mar 21 2017 l2_learning.py  
-rw-rw-r-- 1 mininet mininet 15558 Mar 21 2017 l2_multi.py  
-rw-rw-r-- 1 mininet mininet 4324 Mar 21 2017 l2_nx.py  
-rw-rw-r-- 1 mininet mininet 2105 Mar 21 2017 l2_nx_self_learning.py  
-rw-rw-r-- 1 mininet mininet 2882 Mar 21 2017 l2_pairs.py  
-rw-rw-r-- 1 mininet mininet 12330 Mar 21 2017 l3_learning.py  
-rw-rw-r-- 1 mininet mininet 14102 Mar 21 2017 topo_proactive.py  
mininet@mininet-vm:~/pox/pox/forwarding$
```

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.addListener(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
```



```

root@mininet-vm:~# ping -c1 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=83.0 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 83.049/83.049/83.049/0.000 ms
root@mininet-vm:~# ping -c1 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=65.4 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 65.428/65.428/65.428/0.000 ms
root@mininet-vm:~#

```

```

root@mininet-vm:~# tcpdump -XX -n -i h2-eth0 > h2.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

```

```

root@mininet-vm:~# tcpdump -XX -n -i h3-eth0 > h3.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h3-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

```

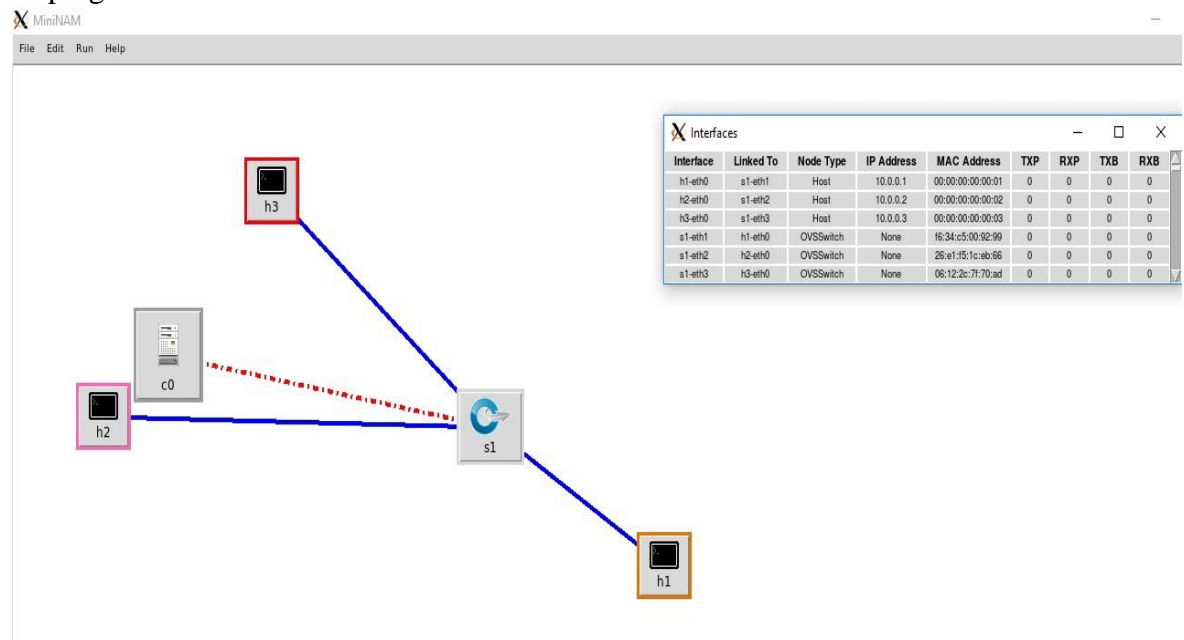
```

mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
Usage: mn [options]
(type mn -h for details)

mn: error: no such option: -m
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
*** 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> xterm h1 h2 h3
mininet>

```

4. h1 ping -c1 10.0.0.2



5. Manual Entry

```

mininet> pingpair
mininet> pingall

```

mininet> iperf

```
mininet@mininet-vm: ~
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
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: ['4.88 Mbits/sec', '5.36 Mbits/sec']
mininet> █
```

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>