Software-Defined Networking

Dibuat untuk pemenuhan tugas besar matakuliah Administrasi dan Desain Jaringan:

Elwin Duha	161402011
Jhon Rendy Sortono	161402017
Sahat Gebima Sihotang	161402059
Aldo Stepanus Simarmata	161402065
Gebriel Juliendi Sitorus	161402143

Dosen Pembimbing : Baihaqi Siregar, S.Si., MT.

JURUSAN TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2018

KATA PENGANTAR

Puji dan syukur penyusun panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya tim penyusun mampu menyelesaikan tugas ini. Laporan ini diajukan guna memenuhi tugas mata kuliah Administrasi dan Desain Jaringan.

Tim penyusun mengucapkan terima kasih kepada Bapak Baihaqi Siregar, S.Si., MT. selaku Dosen mata kuliah Administrasi dan Desain Jaringan yang telah memberikan kami tugas besar Administrasi dan Desain Jaringan ini.

Tim penyusun juga mengucapkan terima kasih kepada semua pihak yang telah membantu sehingga tugas ini dapat diselesaikan tepat pada waktunya.

Tim penyusun juga mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan database ini. Semoga laporan ini dapat memberikan manfaat bagi masyarakat untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua.

Medan, April 2018

Penyusun

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Berbicara mengenai dunia jaringan komputer maka seringkali dikaitkan dengan berbagai vendor penyedia teknologi jaringan seperti CISCO, Juniper, NEC ataupun Huwawei. Sejak menjamurnya pengguna internet di dunia vendor-vendor tersebut memiliki peran penting dalanm mendukung terselenggaranya koneksi internet. Teknologinya diimplementasikan di perkantoran, sekolah, kampus, dan berbagai tempat lainya.

Software Defined Networking (SDN) adalah sebuah pendekatan baru dalam mendesain, membangun, dan mengelola jaringan komputer. Konsep dasar SDN berkaitan erat dengan arsitektur perangkat networking seperti Router, packet Switch, LAN Switch dan sebagainya

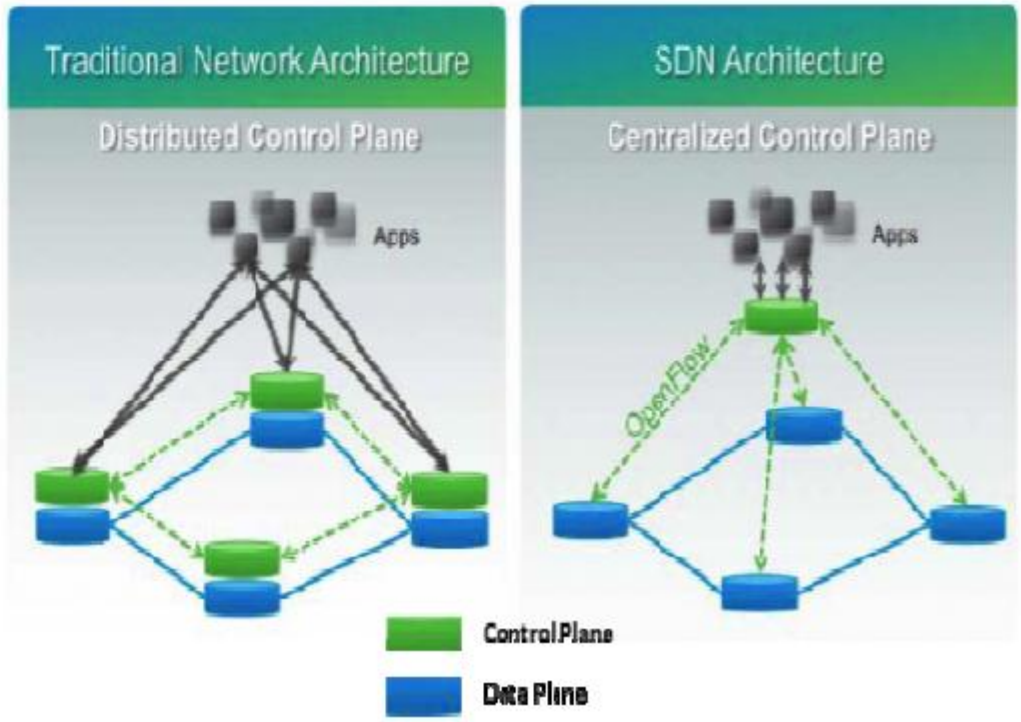
Inovasi dari SDN sangat diperlukan dalam bidang jaringan, manfaatnya antara lain yaitu kemampuan untuk mengatur dan mengelola ribuan perangkat melalui sebuah point of management, kemampuan untuk mengubah behaviour (perilaku) jaringan serta untuk dapat melakukan perubahan terebut secara otomatis, Kemampuan untuk dapat memonitor jaringan, baik dari sisi sumber daya, konektivitas dan lain-lain, dan masih banyak lagi. Hadirnya Software-Defined Networking sangat dibutuhkan untuk memenuhi keperluan inovasi dalam bidang jaringan

1.2 Konsep

Secara umum dalam perangkat *networking* terdapat dua bagian, yaitu *Control Plane* dan *Data Plane*. *Control Plane* adalah bagian yang berfungsi untuk mengatur logika pada perangkat networking seperti routing table, pemetaan jaringan, dan sebagainya. *Data Plane* adalah bagian yang berfungsi untuk meneruskan paket-paket yang masuk ke suatu port pada perangkat networking menuju port keluar dengan berkonsultasi kepada *Control Plane*.

Konsep jaringan SDN adalah melakukan pemisahan antara *Control Plane* dan *Data Plane*, dimana *Data Plane* tetap berada pada perangkat networking, sedang *Control Plane* berada pada sebuah entitas terpisah bernama “*Controller*” yang akan menentukan perilaku jaringan dengan cara memungkinkan Data Plane untuk di program sehingga terbentuklah istilah *Software Defined Networking* (SDN) yang mendefinisikan jaringan. Hal ini berbeda dengan konsep jaringan tradisional, yaitu Control Plane dan Data Plane berada dalam satu perangkat networking yang sama.

Gambar dibawah ini mengilustrasikan perbandingan antara konsep jaringan tradisional dengan konsep jaringan Software Defined Networking (SDN).


Traditional Network Architecture
Distributed Control Plane
Apps
SDN Architecture
Centralized Control Plane
Apps
OpenFlow
Control Plane
Data Plane

Gambar di sebelah kiri merupakan konsep jaringan tradisional, Control Plane dan Data Plane berada dalam satu tempat yang sama, yaitu di dalam perangkat networking.

Gambar sebelah kanan merupakan konsep jaringan SDN menunjukkan pemisahan Control Plane dari Data Plane, sehingga pada perangkat networking hanya terdapat Data Plane.Selain itu pada gambar di atas perbedaan antara konsep jaringan komputer tradisional dengan konsep jaringan komputer SDN juga di tunjukkan, yaitu pada struktur Control Plane.

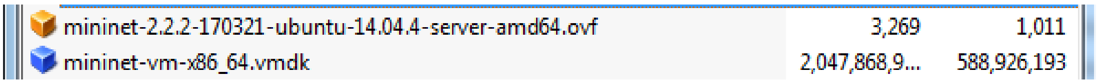
BAB II

PENGERJAAN SISTEM

2.1 Mininet

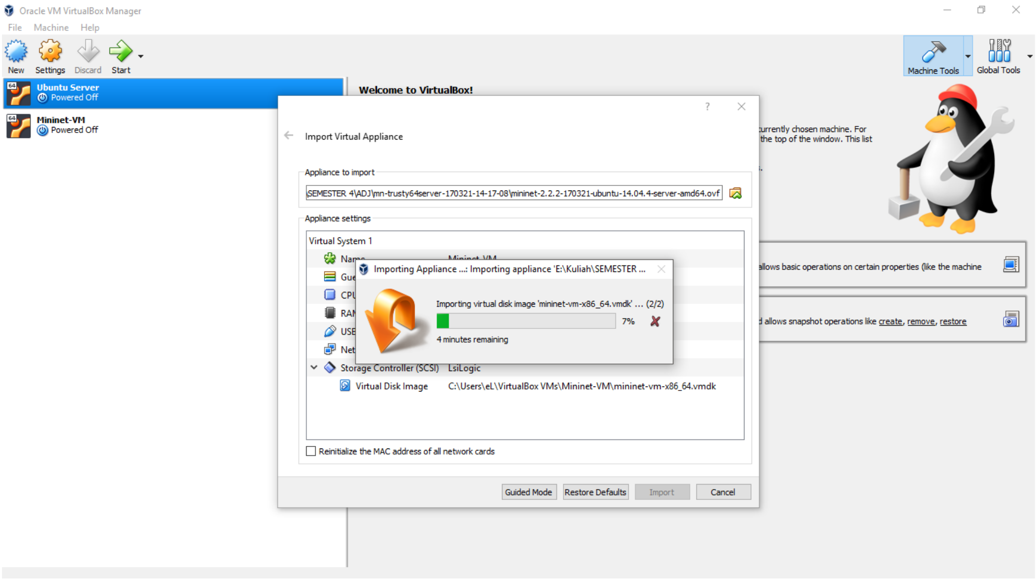
Dalam pengerjaan tugas kali ini, modules yang dipakai adalah Mininet, MiniNam, OpenFlow, POX COntoller. Mininet adalah emulator jaringan yang menciptakan jaringan virtual yang realistis yang berbasis Command Line Interface yang digunakan untuk membuat sebuah topologi jaringan pada Software Defined Network. Mininet berjalan dengan kernel yang sama seperti aslinya, switch dan kode aplikasi pada mesin yang seperti aslinya.

Langkah pertama adalah mengunduh Mininet VM Image. Hasil unduhan akan berupa seperti dibawah ini.



mininet-2.2.2-170321-ubuntu-14.04.4-server-amd64.ovf	3,269	1,011
mininet-vm-x86_64.vmdk	2,047,868,9...	588,926,193

Lalu, import image ke VirtualBox.



Import Virtual Appliance

Appliance to import: SEMESTER 4\ADJ\mn-trusty64server-170321-14-17-08\mininet-2.2.2-170321-ubuntu-14.04.4-server-amd64.ovf

Appliance settings

Virtual System 1

Name: Mininet-VM

Storage Controller (SCSI): LsiLogic

Virtual Disk Image: C:\Users\eL\VirtualBox VMs\Mininet-VM\mininet-vm-x86_64.vmdk

□ Reinitialize the MAC address of all network cards

Importing Appliance ...: Importing appliance 'E:\Kuliah\SEMESTER ...

Importing virtual disk image 'mininet-vm-x86_64.vmdk' ... (2/2)

7%

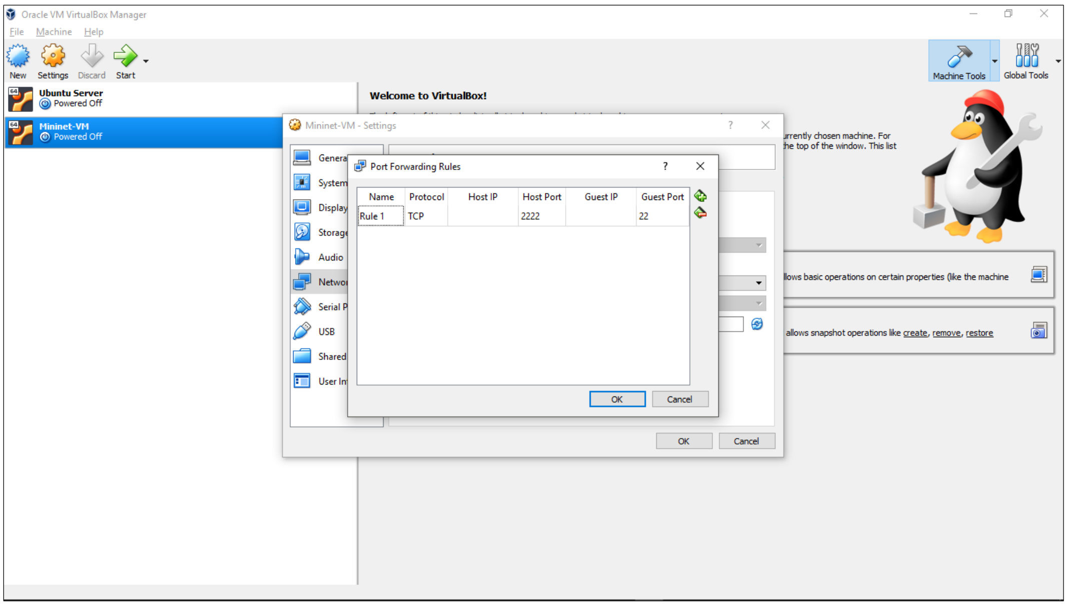
4 minutes remaining

currently chosen machine. For the top of the window. This list

allows basic operations on certain properties (like the machine

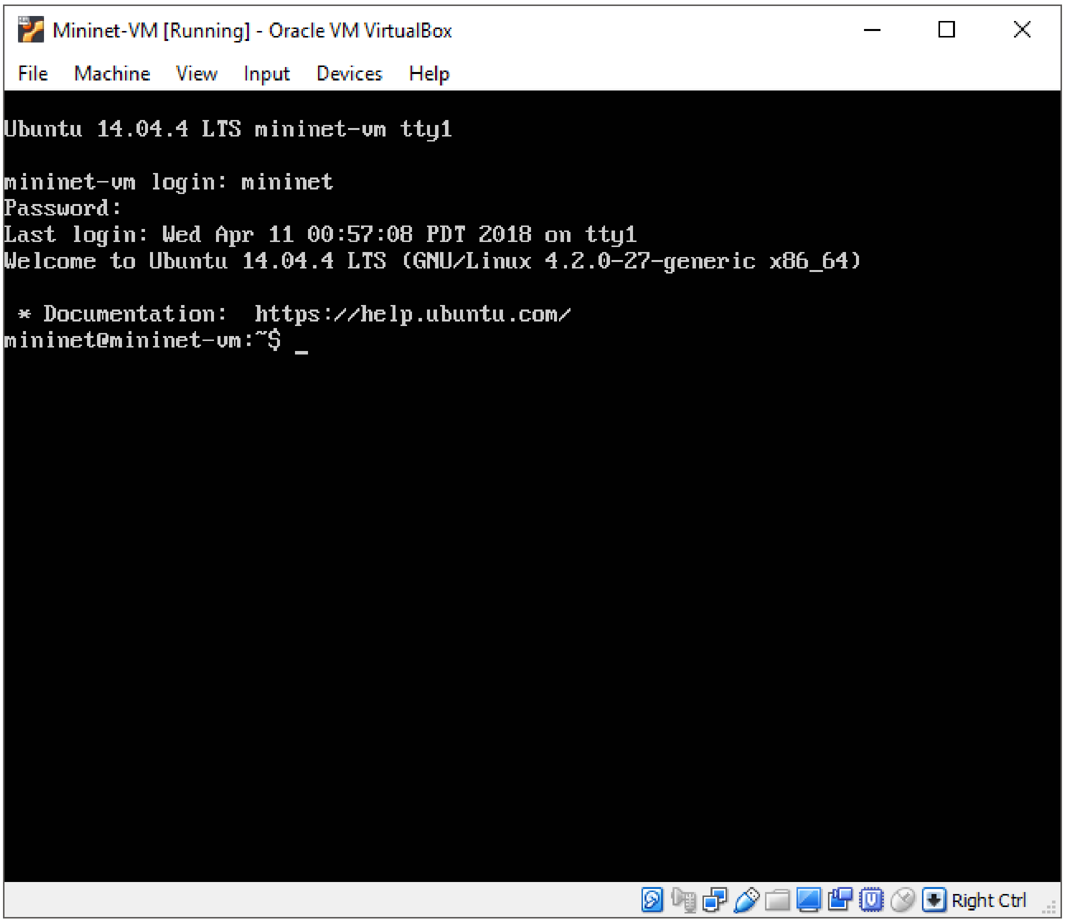
allows snapshot operations like create, remove, restore

Setelah itu, setting konfigurasi network dengan postforwarding



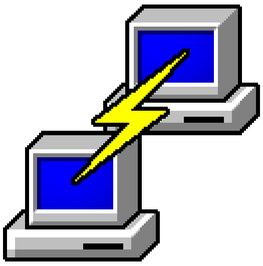
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
Rule 1	TCP		2222		22

Jalankan mininet dan login dengan username mininet dan password mininet.


Mininet-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
Ubuntu 14.04.4 LTS mininet-vm tty1

mininet-vm login: mininet
Password:
Last login: Wed Apr 11 00:57:08 PDT 2018 on tty1
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$ _
```

Lalu, akses via ssh dengan Putty dan Xming. PuTTY adalah sebuah program open source yang dapat digunakan untuk melakukan protokol jaringan SSH, Telnet dan Rlogin, atau dapat juga didefinisikan sebagai software remote console atau terminal yang digunakan untuk meremote komputer dengan terhubungnya menggunakan Port SSH dan sebagainya. Protokol ini dapat digunakan untuk menjalankan sesi remote pada sebuah komputer melalui sebuah jaringan, baik itu LAN, maupun internet.

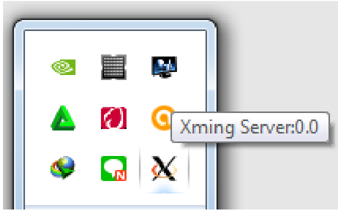


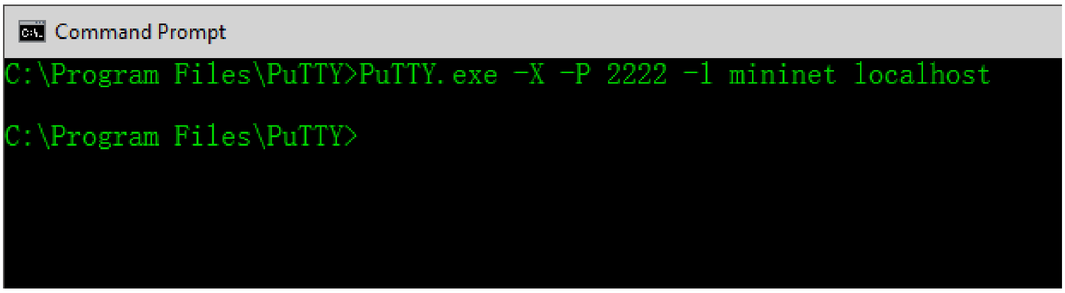
Icon Putty

Xming dalah software yang mengimplementasikan X Server pada sistem operasi Windows. Xming dan Putty dalam kasus ini harus digunakan berdampingan. Xming digunakan untuk menyediakan Xserver pada windows, sedangkan putty digunakan sebagai ssh client pada windows yang mendukung X Forwarding.

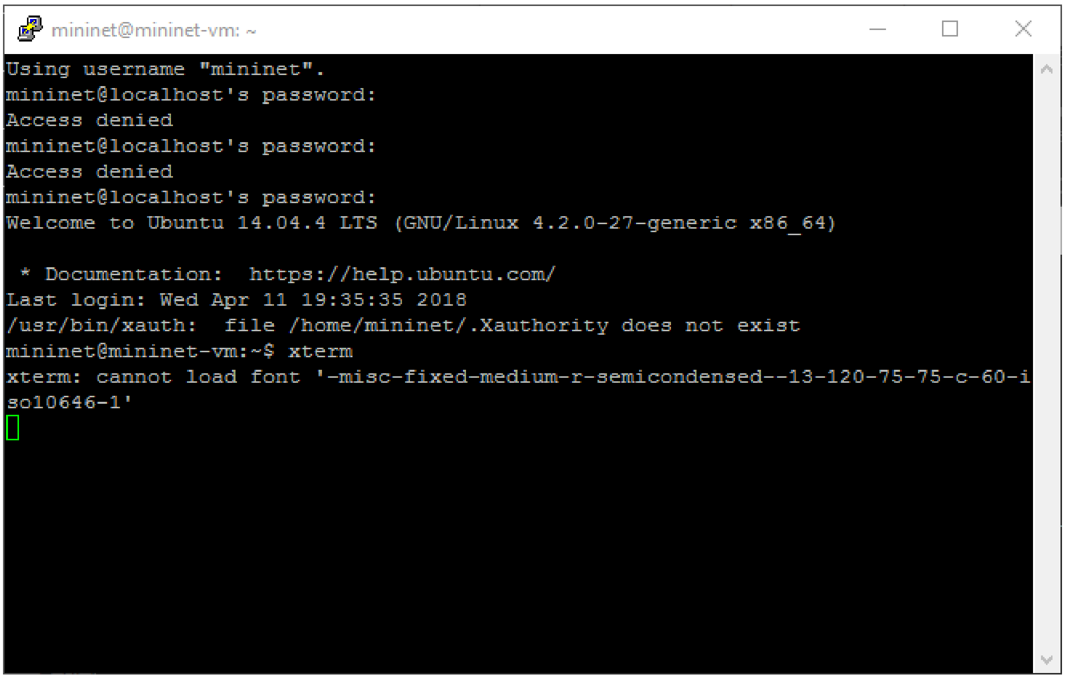
Icon Xming

Untuk mengakses via ssh, jalankan putty dan xming. Pertama buka dahulu Xming, setelah dibuka, tidak akan terjadi apa – apa, tetapi Xming telah berjalan pada toolbar windows. Lalu jalankan dan akses putty dengan cmd pada komputer.

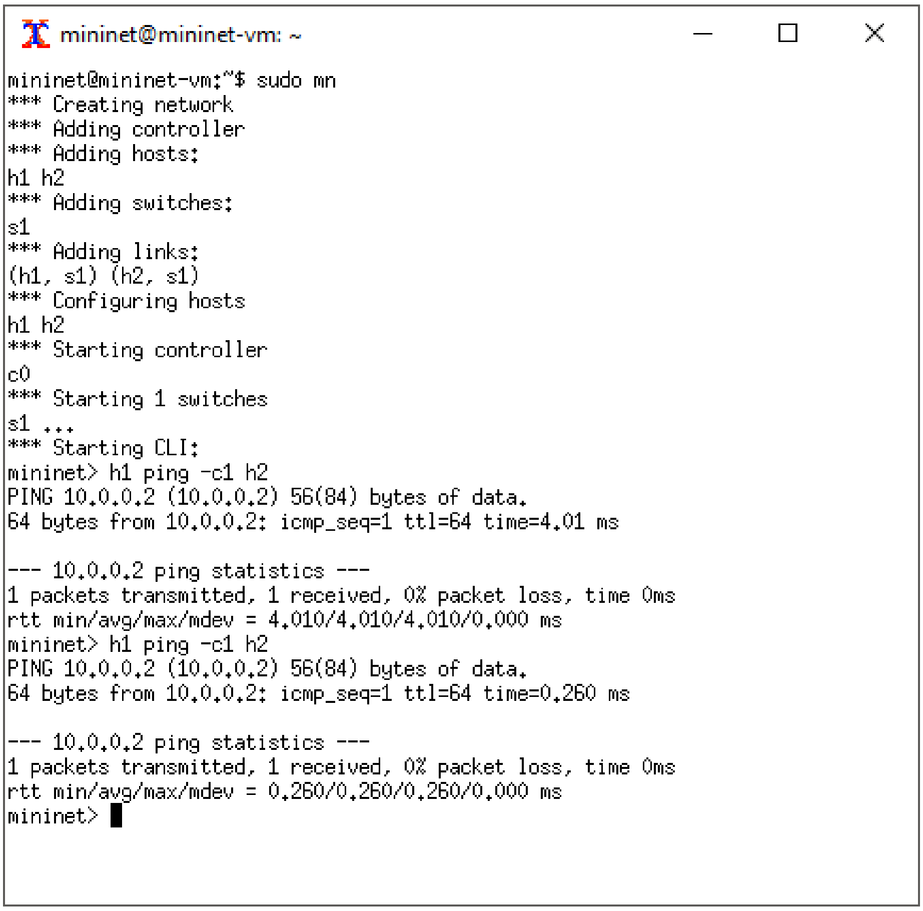




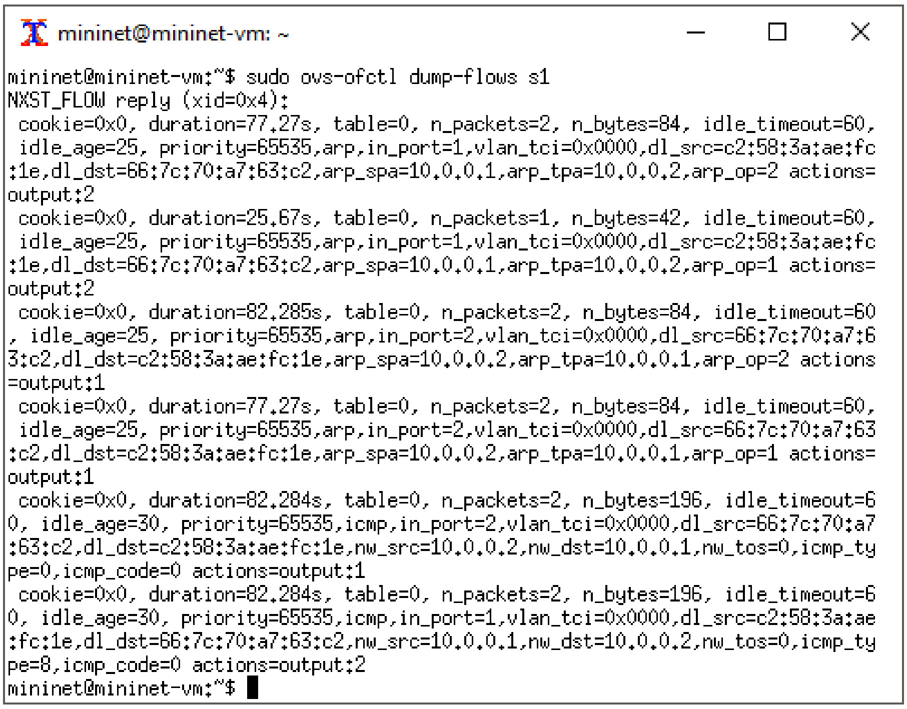
```
C:\Program Files\PuTTY>PuTTY.exe -X -P 2222 -l mininet localhost

C:\Program Files\PuTTY>
```



```
Using username "mininet".
mininet@localhost's password:
Access denied
mininet@localhost's password:
Access denied
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/
Last login: Wed Apr 11 19:35:35 2018
/usr/bin/xauth:  file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$ xterm
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-i
so10646-1'
```

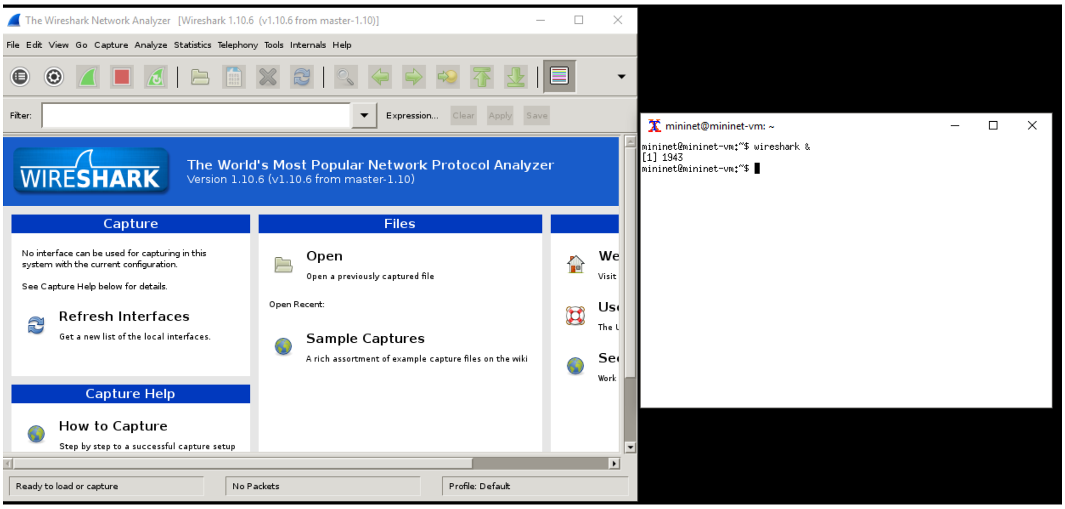


```
mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.01 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.010/4.010/4.010/0.000 ms
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=0.260 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.260/0.260/0.260/0.000 ms
mininet> █
```



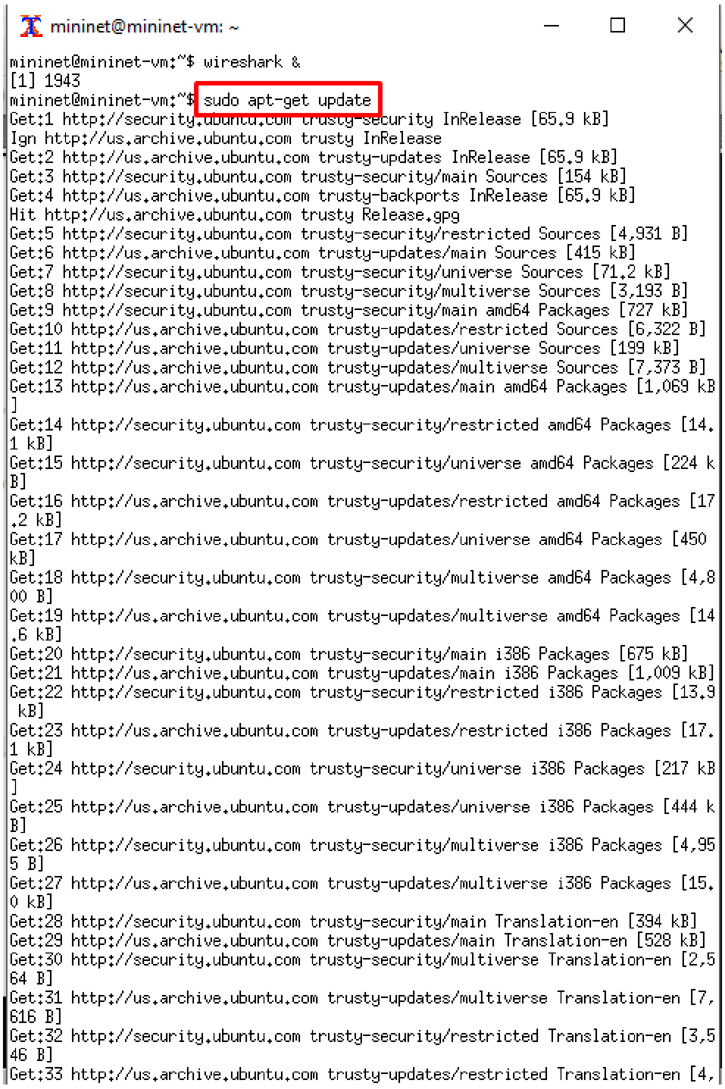
```
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=77.27s, table=0, n_packets=2, n_bytes=84, idle_timeout=60,
 idle_age=25, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=c2:58:3a:ae:fc
:1e,dl_dst=66:7c:70:a7:63:c2,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=
output:2
 cookie=0x0, duration=25.67s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
 idle_age=25, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=c2:58:3a:ae:fc
:1e,dl_dst=66:7c:70:a7:63:c2,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=1 actions=
output:2
 cookie=0x0, duration=82.285s, table=0, n_packets=2, n_bytes=84, idle_timeout=60
, idle_age=25, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=66:7c:70:a7:6
3:c2,dl_dst=c2:58:3a:ae:fc:1e,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions
=output:1
 cookie=0x0, duration=77.27s, table=0, n_packets=2, n_bytes=84, idle_timeout=60,
 idle_age=25, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=66:7c:70:a7:63
:c2,dl_dst=c2:58:3a:ae:fc:1e,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=
output:1
 cookie=0x0, duration=82.284s, table=0, n_packets=2, n_bytes=196, idle_timeout=6
0, idle_age=30, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=66:7c:70:a7
:63:c2,dl_dst=c2:58:3a:ae:fc:1e,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:1
 cookie=0x0, duration=82.284s, table=0, n_packets=2, n_bytes=196, idle_timeout=6
0, idle_age=30, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=c2:58:3a:ae
:fc:1e,dl_dst=66:7c:70:a7:63:c2,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_ty
pe=8,icmp_code=0 actions=output:2
mininet@mininet-vm:~$ █
```

Setelah itu, lakukan test Wireshark.

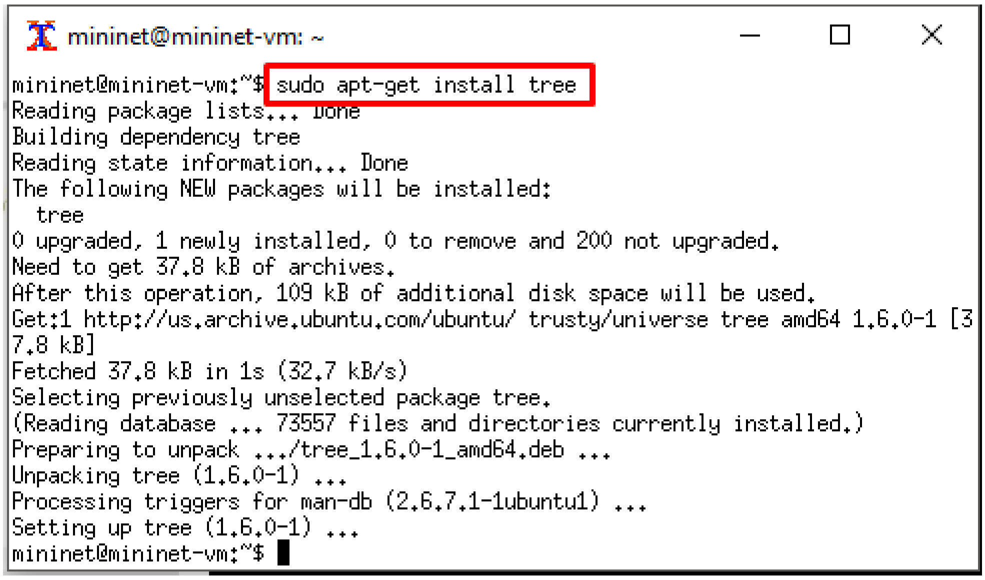

The Wireshark Network Analyzer [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.10.6 (v1.10.6 from master-1.10)
Capture
No interface can be used for capturing in this system with the current configuration.
See Capture Help below for details.
Refresh Interfaces
Get a new list of the local interfaces.
Capture Help
How to Capture
Step by step to a successful capture setup
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
Ready to load or capture
No Packets
Profile: Default
mininet@mininet-vm: ~

```
mininet@mininet-vm:~$ wireshark &
[1] 1943
mininet@mininet-vm:~$ 
```

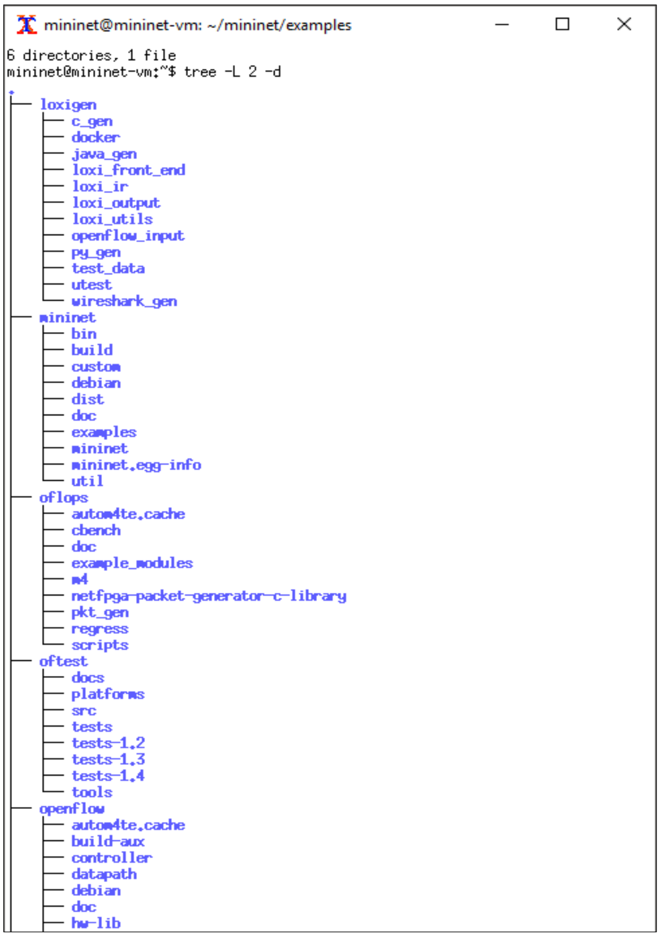
Lalu, update, install tree dan check tree.



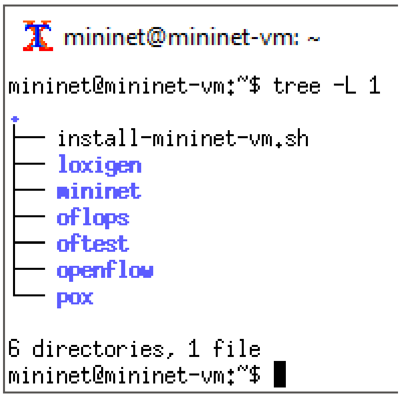
```
mininet@mininet-vm:~$ wireshark &
[1] 1943
mininet@mininet-vm:~$ sudo apt-get update
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://us.archive.ubuntu.com trusty InRelease
Get:2 http://us.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:3 http://security.ubuntu.com trusty-security/main Sources [154 kB]
Get:4 http://us.archive.ubuntu.com trusty-backports InRelease [65.9 kB]
Hit http://us.archive.ubuntu.com trusty Release.gpg
Get:5 http://security.ubuntu.com trusty-security/restricted Sources [4,931 B]
Get:6 http://us.archive.ubuntu.com trusty-updates/main Sources [415 kB]
Get:7 http://security.ubuntu.com trusty-security/universe Sources [71.2 kB]
Get:8 http://security.ubuntu.com trusty-security/multiverse Sources [3,193 B]
Get:9 http://security.ubuntu.com trusty-security/main amd64 Packages [727 kB]
Get:10 http://us.archive.ubuntu.com trusty-updates/restricted Sources [6,322 B]
Get:11 http://us.archive.ubuntu.com trusty-updates/universe Sources [199 kB]
Get:12 http://us.archive.ubuntu.com trusty-updates/multiverse Sources [7,373 B]
Get:13 http://us.archive.ubuntu.com trusty-updates/main amd64 Packages [1,069 kB
]
Get:14 http://security.ubuntu.com trusty-security/restricted amd64 Packages [14.
1 kB]
Get:15 http://security.ubuntu.com trusty-security/universe amd64 Packages [224 k
B]
Get:16 http://us.archive.ubuntu.com trusty-updates/restricted amd64 Packages [17
.2 kB]
Get:17 http://us.archive.ubuntu.com trusty-updates/universe amd64 Packages [450
kB]
Get:18 http://security.ubuntu.com trusty-security/multiverse amd64 Packages [4,8
00 B]
Get:19 http://us.archive.ubuntu.com trusty-updates/multiverse amd64 Packages [14
.6 kB]
Get:20 http://security.ubuntu.com trusty-security/main i386 Packages [675 kB]
Get:21 http://us.archive.ubuntu.com trusty-updates/main i386 Packages [1,009 kB]
Get:22 http://security.ubuntu.com trusty-security/restricted i386 Packages [13.9
 kB]
Get:23 http://us.archive.ubuntu.com trusty-updates/restricted i386 Packages [17.
1 kB]
Get:24 http://security.ubuntu.com trusty-security/universe i386 Packages [217 kB
]
Get:25 http://us.archive.ubuntu.com trusty-updates/universe i386 Packages [444 k
B]
Get:26 http://security.ubuntu.com trusty-security/multiverse i386 Packages [4,95
5 B]
Get:27 http://us.archive.ubuntu.com trusty-updates/multiverse i386 Packages [15.
0 kB]
Get:28 http://security.ubuntu.com trusty-security/main Translation-en [394 kB]
Get:29 http://us.archive.ubuntu.com trusty-updates/main Translation-en [528 kB]
Get:30 http://security.ubuntu.com trusty-security/multiverse Translation-en [2,5
64 B]
Get:31 http://us.archive.ubuntu.com trusty-updates/multiverse Translation-en [7,
616 B]
Get:32 http://security.ubuntu.com trusty-security/restricted Translation-en [3,5
46 B]
Get:33 http://us.archive.ubuntu.com trusty-updates/restricted Translation-en [4,
```



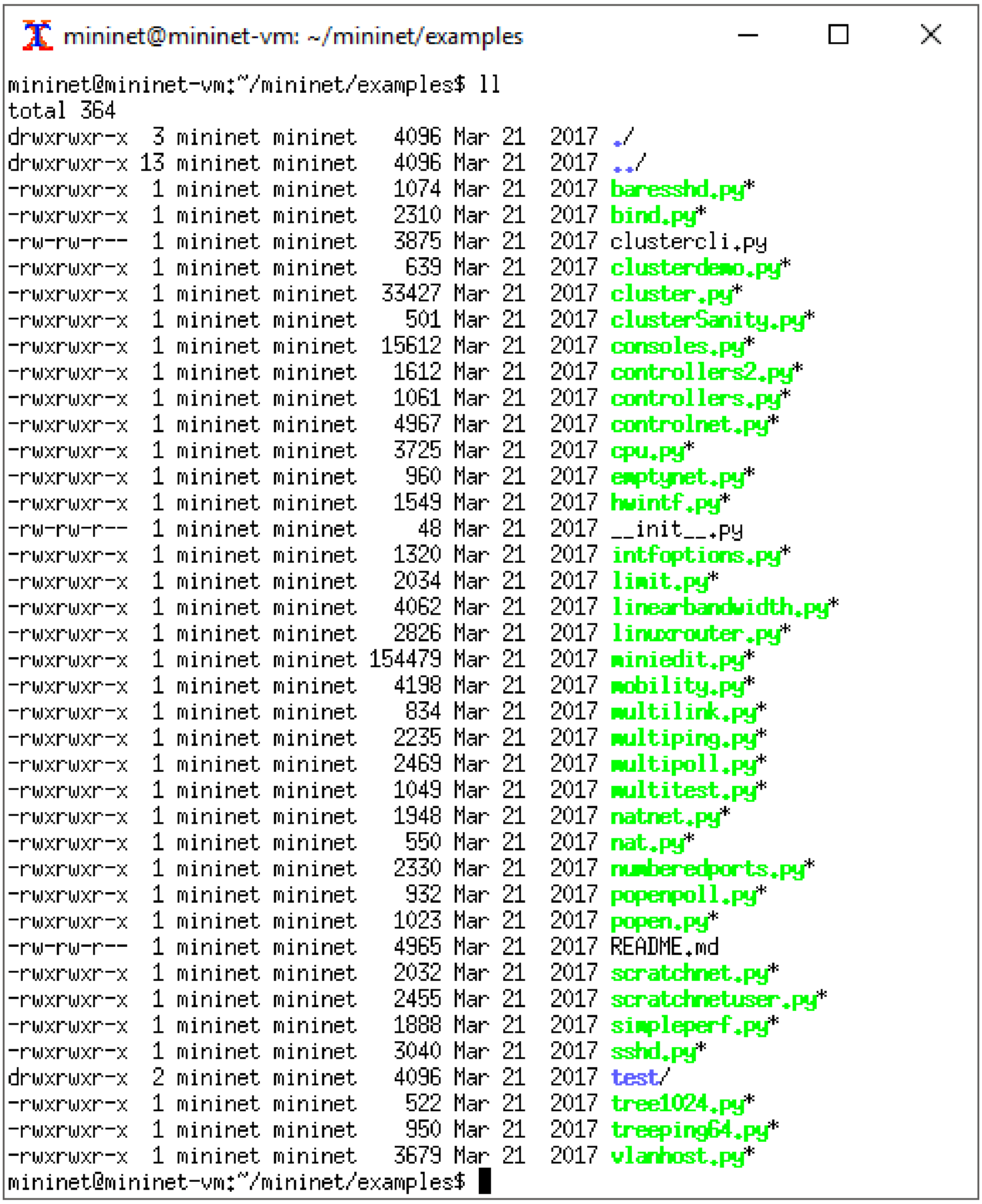
```
mininet@mininet-vm:~$ sudo apt-get install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 200 not upgraded.
Need to get 37.8 kB of archives.
After this operation, 109 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/universe tree amd64 1.6.0-1 [3
7.8 kB]
Fetched 37.8 kB in 1s (32.7 kB/s)
Selecting previously unselected package tree.
(Reading database ... 73557 files and directories currently installed.)
Preparing to unpack .../tree_1.6.0-1_amd64.deb ...
Unpacking tree (1.6.0-1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up tree (1.6.0-1) ...
mininet@mininet-vm:~$ 
```



```
6 directories, 1 file
mininet@mininet-vm:~$ tree -L 2 -d
.
├── loxigen
│   ├── c_gen
│   ├── docker
│   ├── java_gen
│   ├── loxi_front_end
│   ├── loxi_ir
│   ├── loxi_output
│   ├── loxi_utils
│   ├── openflow_input
│   ├── py_gen
│   ├── test_data
│   ├── utest
│   └── wireshark_gen
├── mininet
│   ├── bin
│   ├── build
│   ├── custom
│   ├── debian
│   ├── dist
│   ├── doc
│   ├── examples
│   ├── mininet
│   ├── mininet.egg-info
│   └── util
├── oflops
│   ├── autom4te.cache
│   ├── cbench
│   ├── doc
│   ├── example_modules
│   ├── m4
│   ├── netfpga-packet-generator-c-library
│   ├── pkt_gen
│   ├── regress
│   └── scripts
├── oftest
│   ├── docs
│   ├── platforms
│   ├── src
│   ├── tests
│   ├── tests-1.2
│   ├── tests-1.3
│   ├── tests-1.4
│   └── tools
├── openflow
│   ├── autom4te.cache
│   ├── build-aux
│   ├── controller
│   ├── datapath
│   ├── debian
│   ├── doc
│   ├── hw-lib
```

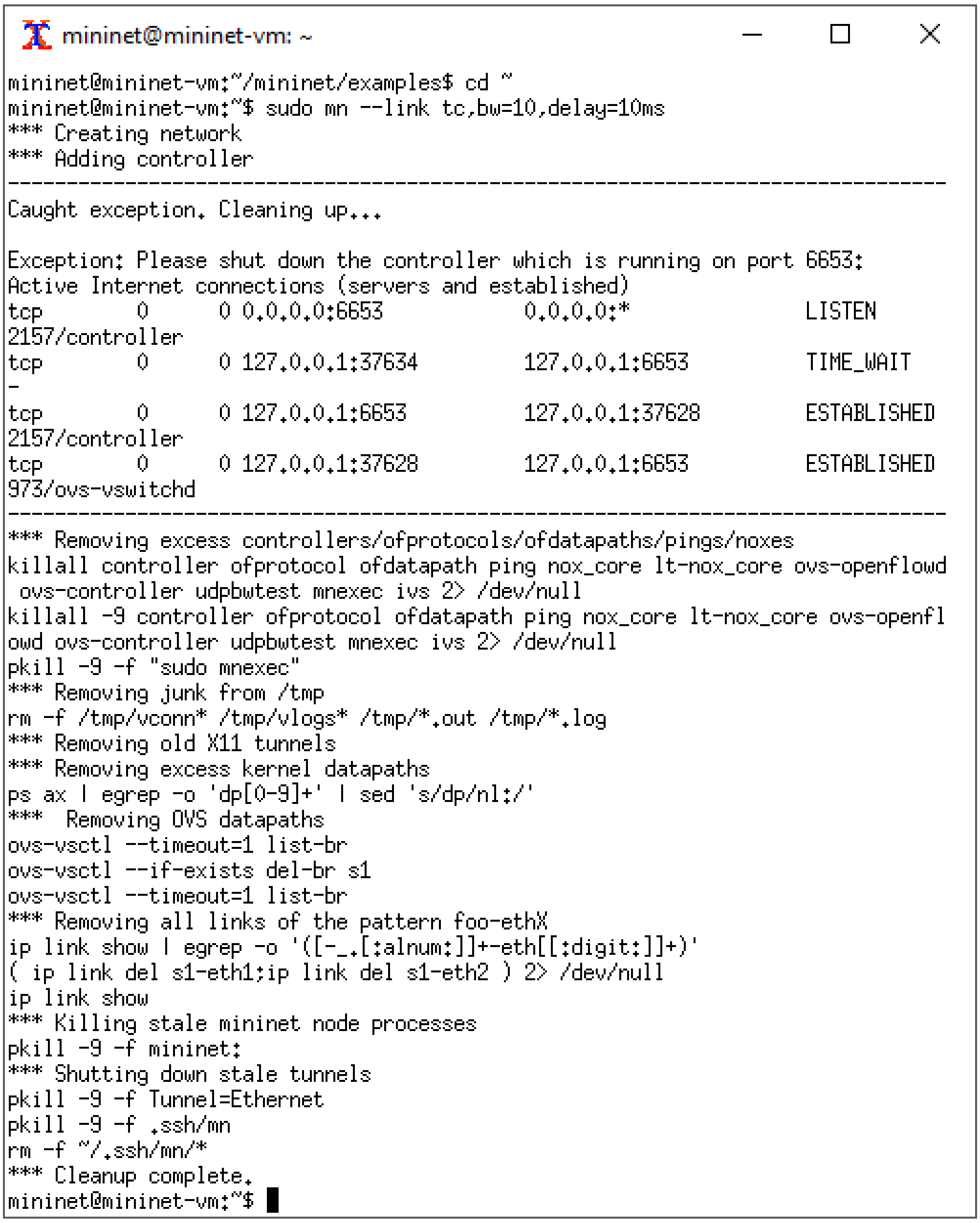


```
mininet@mininet-vm:~$ tree -L 1
.
├── install-mininet-vm.sh
├── loxigen
├── mininet
├── oflops
├── oftest
├── openflow
└── pox

6 directories, 1 file
mininet@mininet-vm:~$ 
```



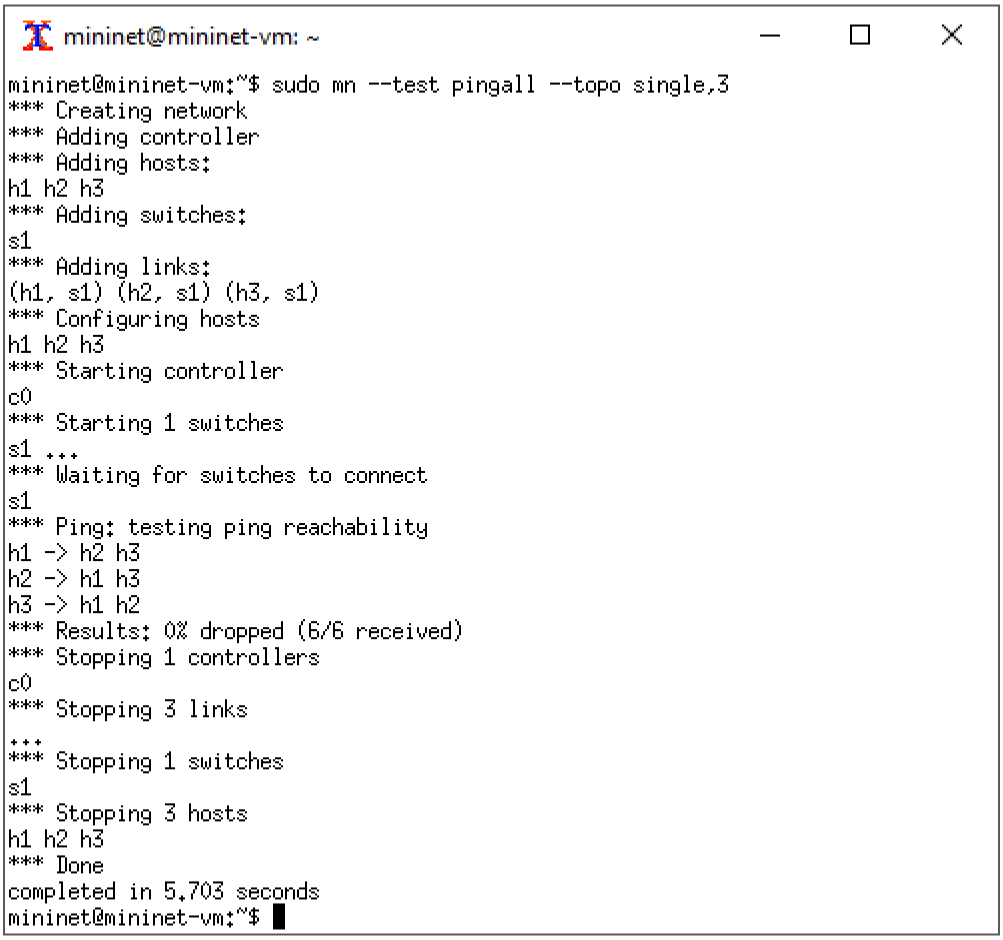
```
mininet@mininet-vm:~/mininet/examples$ ll
total 364
drwxrwxr-x  3 mininet mininet   4096 Mar 21  2017 ./
drwxrwxr-x 13 mininet mininet   4096 Mar 21  2017 ../
-rwxrwxr-x  1 mininet mininet   1074 Mar 21  2017 baresshd.py*
-rwxrwxr-x  1 mininet mininet   2310 Mar 21  2017 bind.py*
-rw-rw-r--  1 mininet 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:~/mininet/examples$ █
```

Link tc

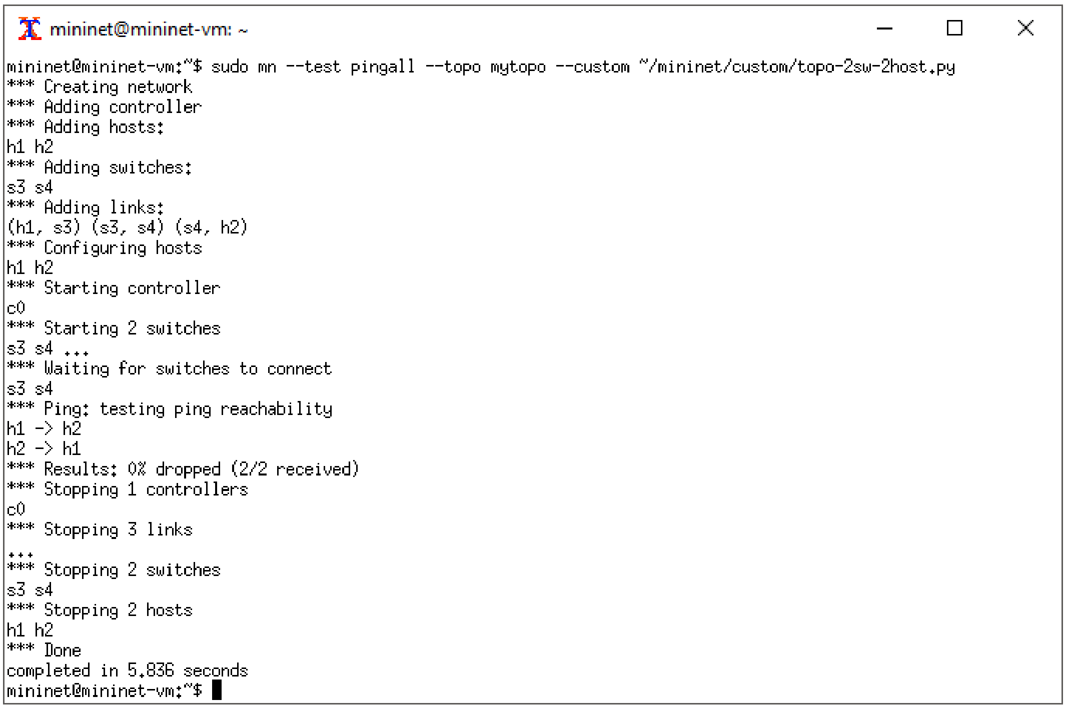


```
mininet@mininet-vm:~/mininet/examples$ cd ~
mininet@mininet-vm:~$ sudo mn --link tc,bw=10,delay=10ms
*** Creating network
*** Adding controller
--------------------------------------------------------------------------------
Caught exception. Cleaning up...

Exception: Please shut down the controller which is running on port 6653:
Active Internet connections (servers and established)
tcp        0      0 0.0.0.0:6653            0.0.0.0:*               LISTEN
2157/controller
tcp        0      0 127.0.0.1:37634         127.0.0.1:6653          TIME_WAIT
-
tcp        0      0 127.0.0.1:6653          127.0.0.1:37628         ESTABLISHED
2157/controller
tcp        0      0 127.0.0.1:37628         127.0.0.1:6653          ESTABLISHED
973/ovs-vswitchd
--------------------------------------------------------------------------------
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
 ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openfl
owd ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
***  Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --if-exists del-br s1
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
( ip link del s1-eth1;ip link del s1-eth2 ) 2> /dev/null
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
mininet@mininet-vm:~$
```

Test ping all



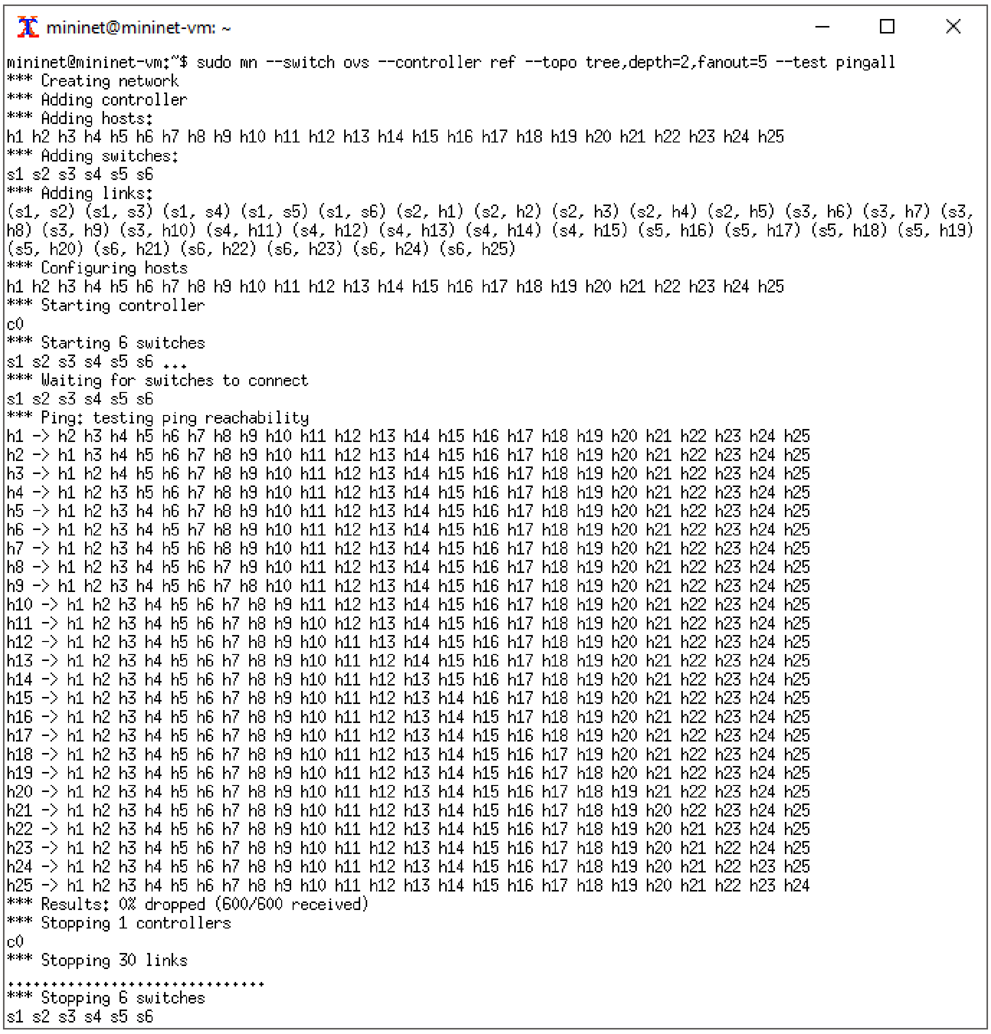
```
mininet@mininet-vm:~$ sudo mn --test pingall --topo single,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 5.703 seconds
mininet@mininet-vm:~$
```



```
mininet@mininet-vm:~$ sudo mn --test pingall --topo mytopo --custom ~/mininet/custom/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 5.836 seconds
mininet@mininet-vm:~$
```

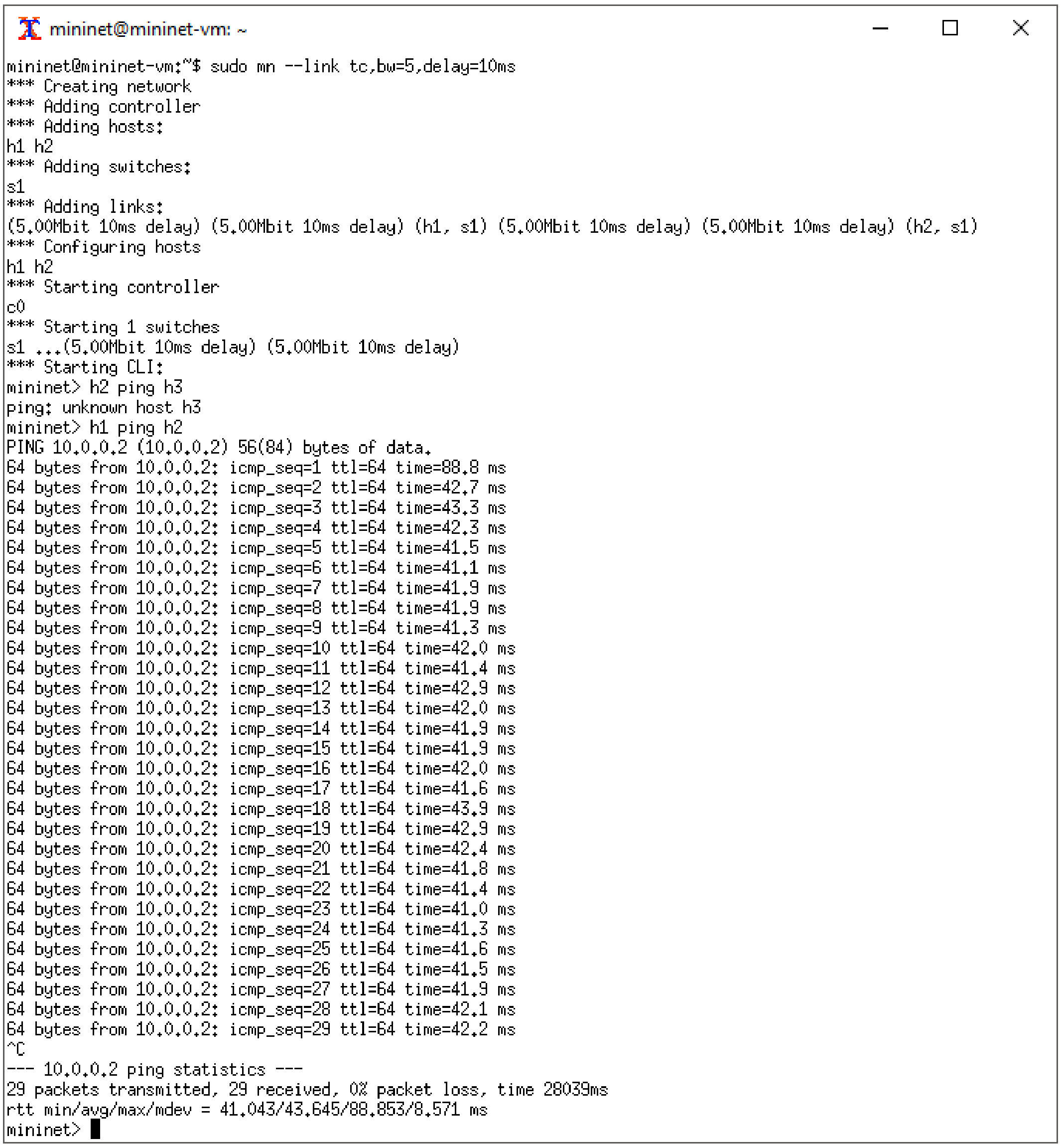
Notes Mininet Sample Workflow

Creating a network



```
mininet@mininet-vm:~$ sudo mn --switch ovs --controller ref --topo tree,depth=2,fanout=5 --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s1, s5) (s1, s6) (s2, h1) (s2, h2) (s2, h3) (s2, h4) (s2, h5) (s3, h6) (s3, h7) (s3,
h8) (s3, h9) (s3, h10) (s4, h11) (s4, h12) (s4, h13) (s4, h14) (s4, h15) (s5, h16) (s5, h17) (s5, h18) (s5, h19)
(s5, h20) (s6, h21) (s6, h22) (s6, h23) (s6, h24) (s6, h25)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Waiting for switches to connect
s1 s2 s3 s4 s5 s6
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h13 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h14 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h15 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25
h16 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h17 h18 h19 h20 h21 h22 h23 h24 h25
h17 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h18 h19 h20 h21 h22 h23 h24 h25
h18 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h19 h20 h21 h22 h23 h24 h25
h19 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h20 h21 h22 h23 h24 h25
h20 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h21 h22 h23 h24 h25
h21 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h22 h23 h24 h25
h22 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h23 h24 h25
h23 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h24 h25
h24 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h25
h25 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24
*** Results: 0% dropped (600/600 received)
*** Stopping 1 controllers
c0
*** Stopping 30 links
..............................
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
```

Interacting with network



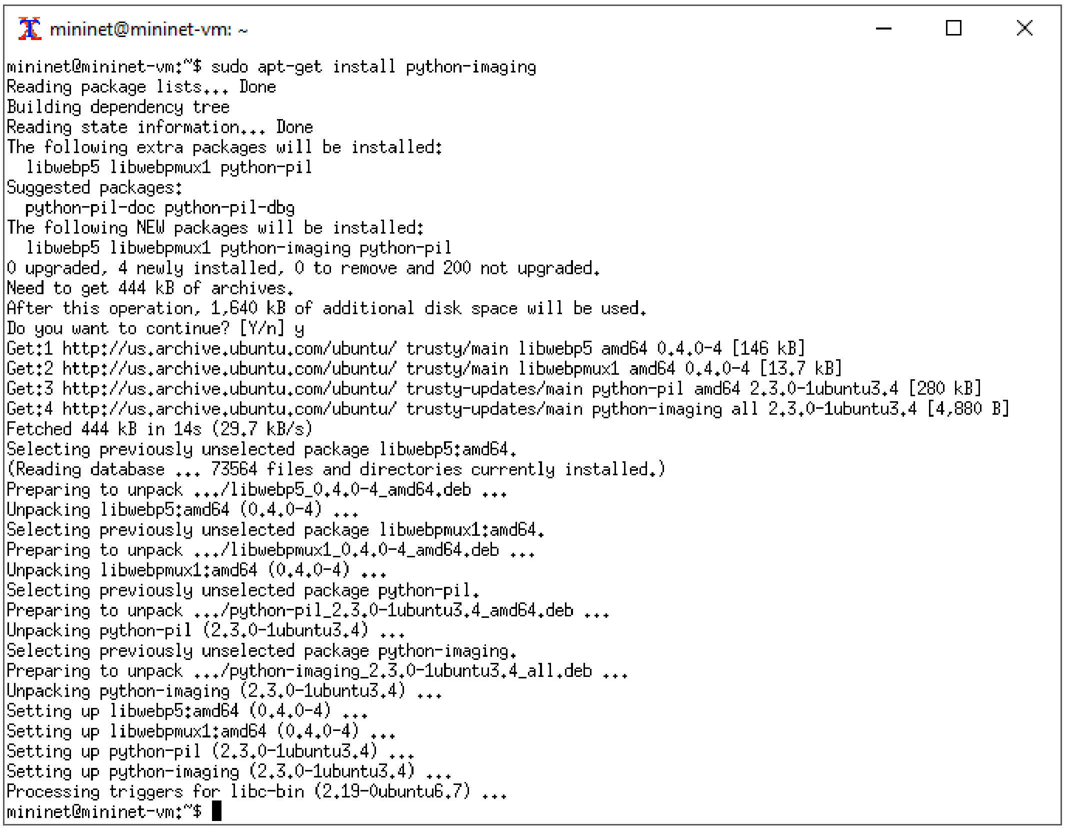
```
mininet@mininet-vm:~$ sudo mn --link tc,bw=5,delay=10ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(5.00Mbit 10ms delay) (5.00Mbit 10ms delay) (h1, s1) (5.00Mbit 10ms delay) (5.00Mbit 10ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...(5.00Mbit 10ms delay) (5.00Mbit 10ms delay)
*** Starting CLI:
mininet> h2 ping h3
ping: unknown host h3
mininet> h1 ping 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=88.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=42.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=43.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=42.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=41.5 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=41.1 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=41.3 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=42.0 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=41.4 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=42.9 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=42.0 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=42.0 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=41.6 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=43.9 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=42.9 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=42.4 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=41.8 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=41.4 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=41.0 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=41.3 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=41.6 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=41.5 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=42.2 ms
^C
--- 10.0.0.2 ping statistics ---
29 packets transmitted, 29 received, 0% packet loss, time 28039ms
rtt min/avg/max/mdev = 41.043/43.645/88.853/8.571 ms
mininet> █
```

2.2 MiniNAM

MiniNAM adalah sebuah utilitas yang menyediakan animasi real-time dari jaringan yang dibuat oleh Mininet. MiniNAM menyediakan user interface grafis yang memungkinkan modifikasi dan filter paket, pengguna juga dapat melihat arus selektif dengan opsi untuk paket kode warna berdasarkan jenis paket dan / atau node (host).

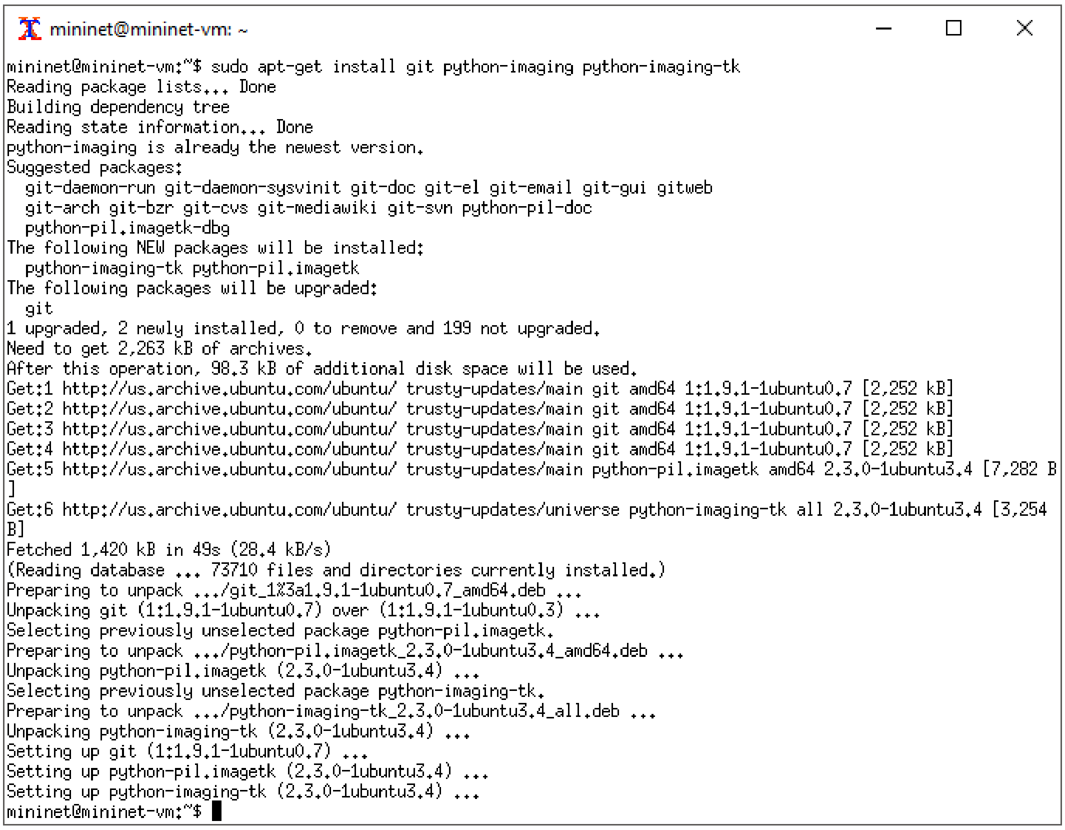
Setup MiniNAM

1. Install python-imaging



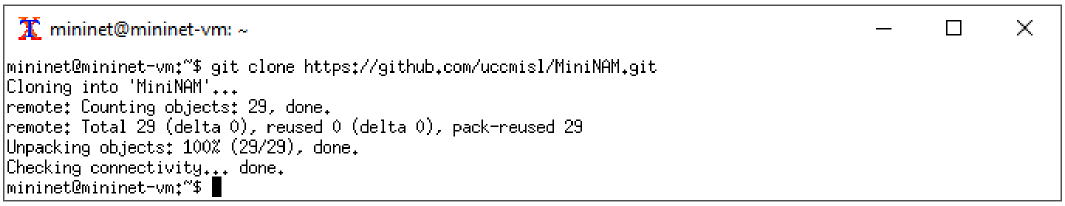
```
mininet@mininet-vm:~$ sudo apt-get install python-imaging
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libwebp5 libwebpmux1 python-pil
Suggested packages:
  python-pil-doc python-pil-dbg
The following NEW packages will be installed:
  libwebp5 libwebpmux1 python-imaging python-pil
0 upgraded, 4 newly installed, 0 to remove and 200 not upgraded.
Need to get 444 kB of archives.
After this operation, 1,640 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main libwebp5 amd64 0.4.0-4 [146 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty/main libwebpmux1 amd64 0.4.0-4 [13.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-pil amd64 2.3.0-1ubuntu3.4 [280 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-imaging all 2.3.0-1ubuntu3.4 [4,880 B]
Fetched 444 kB in 14s (29.7 kB/s)
Selecting previously unselected package libwebp5:amd64.
(Reading database ... 73564 files and directories currently installed.)
Preparing to unpack .../libwebp5_0.4.0-4_amd64.deb ...
Unpacking libwebp5:amd64 (0.4.0-4) ...
Selecting previously unselected package libwebpmux1:amd64.
Preparing to unpack .../libwebpmux1_0.4.0-4_amd64.deb ...
Unpacking libwebpmux1:amd64 (0.4.0-4) ...
Selecting previously unselected package python-pil.
Preparing to unpack .../python-pil_2.3.0-1ubuntu3.4_amd64.deb ...
Unpacking python-pil (2.3.0-1ubuntu3.4) ...
Selecting previously unselected package python-imaging.
Preparing to unpack .../python-imaging_2.3.0-1ubuntu3.4_all.deb ...
Unpacking python-imaging (2.3.0-1ubuntu3.4) ...
Setting up libwebp5:amd64 (0.4.0-4) ...
Setting up libwebpmux1:amd64 (0.4.0-4) ...
Setting up python-pil (2.3.0-1ubuntu3.4) ...
Setting up python-imaging (2.3.0-1ubuntu3.4) ...
Processing triggers for libc-bin (2.19-0ubuntu6.7) ...
mininet@mininet-vm:~$ █
```

2. Install git python-imaging python-imaging-tk



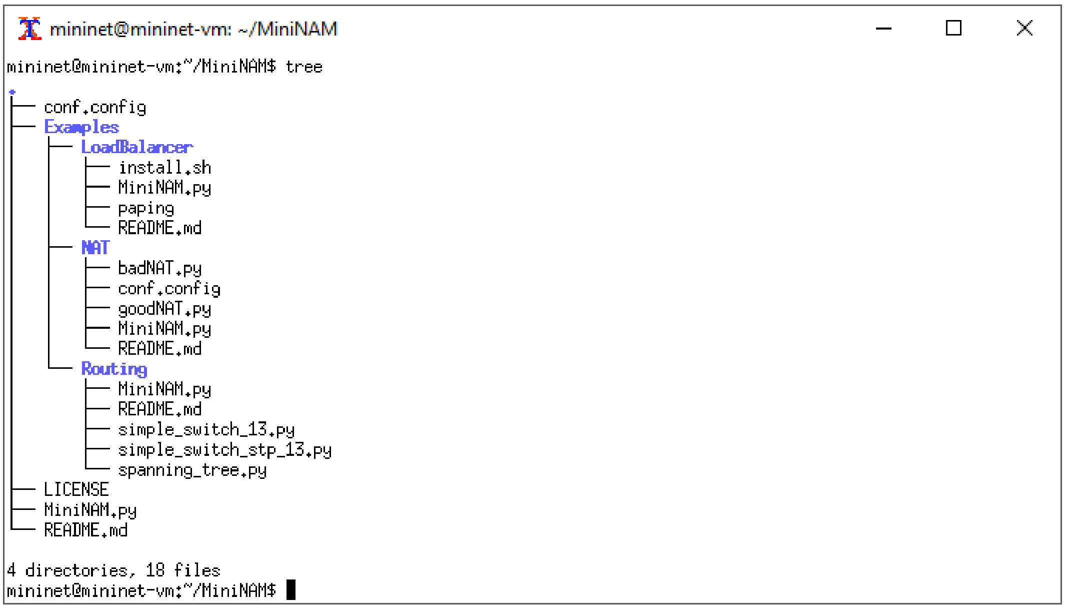
```
mininet@mininet-vm:~$ sudo apt-get install git python-imaging python-imaging-tk
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-imaging is already the newest version.
Suggested packages:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitweb
  git-arch git-bzr git-cvs git-mediawiki git-svn python-pil-doc
  python-pil.imagetk-dbg
The following NEW packages will be installed:
  python-imaging-tk python-pil.imagetk
The following packages will be upgraded:
  git
1 upgraded, 2 newly installed, 0 to remove and 199 not upgraded.
Need to get 2,263 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1ubuntu0.7 [2,252 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1ubuntu0.7 [2,252 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1ubuntu0.7 [2,252 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main git amd64 1:1.9.1-1ubuntu0.7 [2,252 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main python-pil.imagetk amd64 2.3.0-1ubuntu3.4 [7,282 B
]
Get:6 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/universe python-imaging-tk all 2.3.0-1ubuntu3.4 [3,254
B]
Fetched 1,420 kB in 49s (28.4 kB/s)
(Reading database ... 73710 files and directories currently installed.)
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.7_amd64.deb ...
Unpacking git (1:1.9.1-1ubuntu0.7) over (1:1.9.1-1ubuntu0.3) ...
Selecting previously unselected package python-pil.imagetk.
Preparing to unpack .../python-pil.imagetk_2.3.0-1ubuntu3.4_amd64.deb ...
Unpacking python-pil.imagetk (2.3.0-1ubuntu3.4) ...
Selecting previously unselected package python-imaging-tk.
Preparing to unpack .../python-imaging-tk_2.3.0-1ubuntu3.4_all.deb ...
Unpacking python-imaging-tk (2.3.0-1ubuntu3.4) ...
Setting up git (1:1.9.1-1ubuntu0.7) ...
Setting up python-pil.imagetk (2.3.0-1ubuntu3.4) ...
Setting up python-imaging-tk (2.3.0-1ubuntu3.4) ...
mininet@mininet-vm:~$
```

3. Install MiniNAM



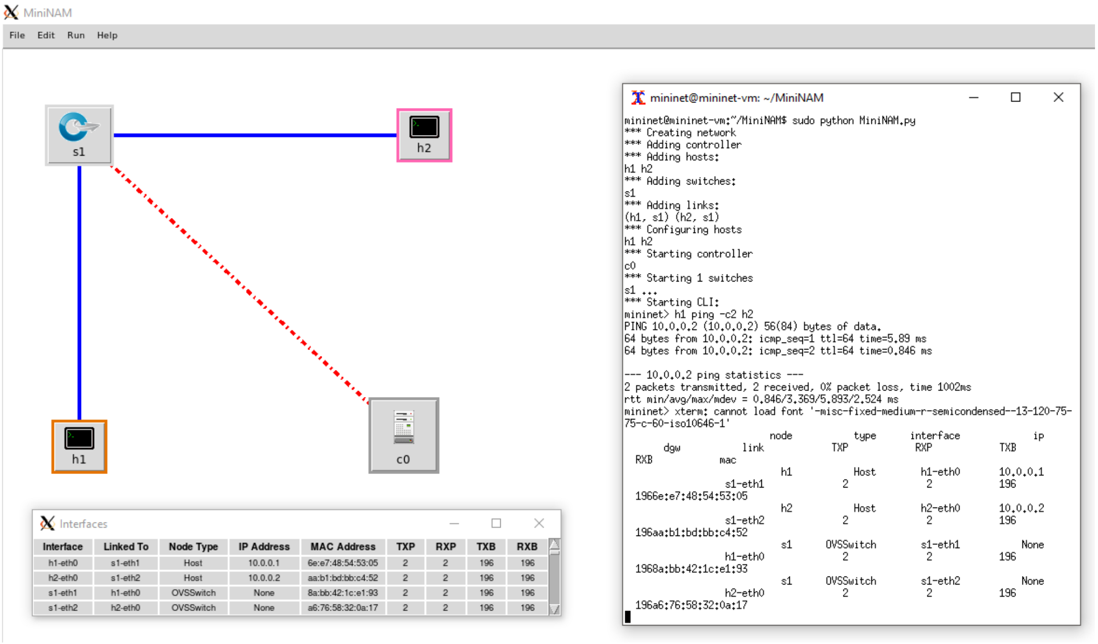
```
mininet@mininet-vm:~$ git clone https://github.com/uccmisl/MiniNAM.git
Cloning into 'MiniNAM'...
remote: Counting objects: 29, done.
remote: Total 29 (delta 0), reused 0 (delta 0), pack-reused 29
Unpacking objects: 100% (29/29), done.
Checking connectivity... done.
mininet@mininet-vm:~$
```

4. MiniNAM tree



```
mininet@mininet-vm:~/MiniNAM$ tree
.
├── 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:~/MiniNAM$
```

5. Run MiniNAM



Interface	Linked To	Node Type	IP Address	MAC Address	TXP	RXP	TXB	RXB
h1-eth0	s1-eth1	Host	10.0.0.1	6e:e7:48:54:53:05	2	2	196	196
h2-eth0	s1-eth2	Host	10.0.0.2	aa:b1:bd:bb:c4:52	2	2	196	196
s1-eth1	h1-eth0	OVSSwitch	None	8a:bb:42:1c:e1:93	2	2	196	196
s1-eth2	h2-eth0	OVSSwitch	None	a6:76:58:32:0a:17	2	2	196	196

mininet@mininet-vm: ~/MiniNAM

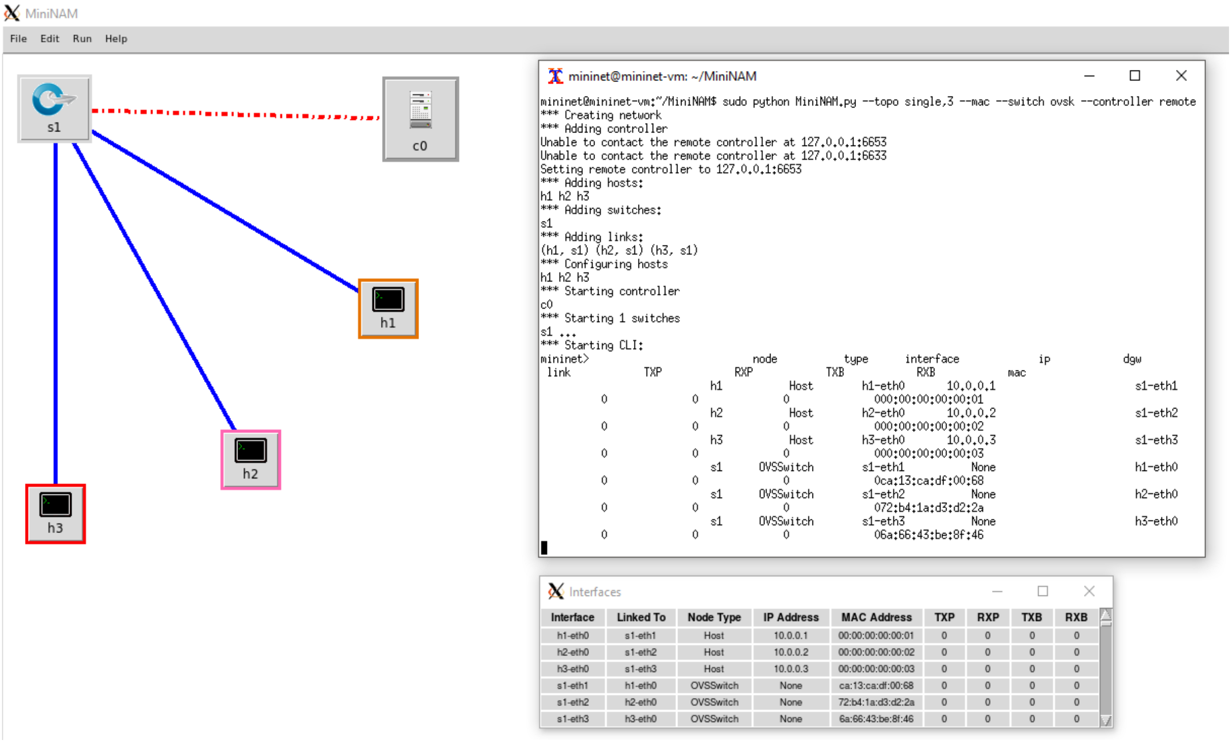
```
mininet@mininet-vm:~/MiniNAM$ sudo python MiniNAM.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping -c2 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=5.89 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.846 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.846/3.369/5.893/2.524 ms
mininet> xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-
75-c-60-iso10646-1'
                          node           type      interface             ip
       dgw            link             TXP            RXP              TXB
  RXB            mac
                            h1             Host        h1-eth0        10.0.0.1
                        s1-eth1              2              2          196
  1966e:e7:48:54:53:05
                            h2             Host        h2-eth0        10.0.0.2
                        s1-eth2              2              2          196
  196aa:b1:bd:bb:c4:52
                            s1        OVSSwitch        s1-eth1            None
                        h1-eth0              2              2          196
  1968a:bb:42:1c:e1:93
                            s1        OVSSwitch        s1-eth2            None
                        h2-eth0              2              2          196
  196a6:76:58:32:0a:17
```

2.3 OpenFlow

Secara global pengertian dari OpenFlow adalah protokol komunikasi (communication protocol) yang memberikan akses ke forwarding plane dari switch atau router melalui jaringan. OpenFlow merupakan communication protocol antara controller dengan OpenFlow Agent yang ada di dummy router atau switch. Jadi perangkat dummy hanya melakukan fungsi forwarding atau fungsi otot dan tidak melakukan fungsi control plane atau fungsi otak, karena untuk control plane itu hanya ada di controller sebagai fungsi otak tersebut.

Langkah-langkah yang dilakukan dalam OpenFlow:

1. Membuat topology dengan 3 host dan 1 switch

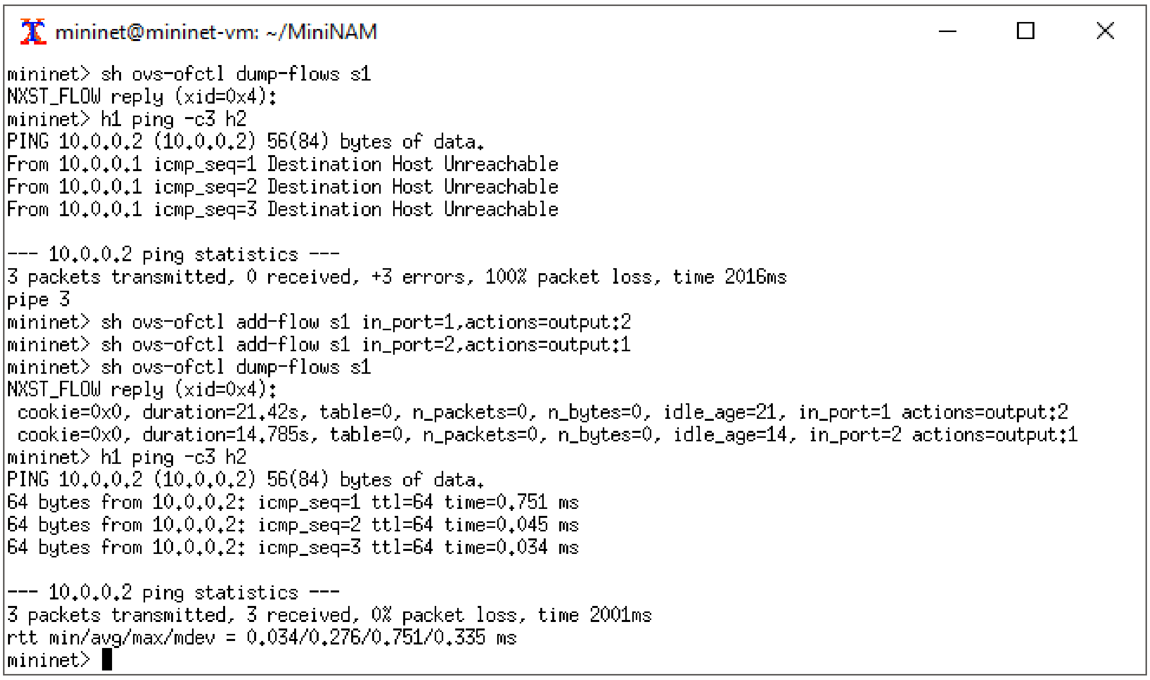


```
mininet@mininet-vm:~/MiniNAM$ sudo python MiniNAM.py --topo single,3 --mac --switch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 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>                   node          type     interface           ip           dgw
 link          TXP          RXP          TXB          RXB          mac
                        h1         Host        h1-eth0      10.0.0.1                  s1-eth1
        0           0           0           000:00:00:00:00:01
                        h2         Host        h2-eth0      10.0.0.2                  s1-eth2
        0           0           0           000:00:00:00:00:02
                        h3         Host        h3-eth0      10.0.0.3                  s1-eth3
        0           0           0           000:00:00:00:00:03
                        s1     OVSSwitch       s1-eth1          None                  h1-eth0
        0           0           0           0ca:13:ca:df:00:68
                        s1     OVSSwitch       s1-eth2          None                  h2-eth0
        0           0           0           072:b4:1a:d3:d2:2a
                        s1     OVSSwitch       s1-eth3          None                  h3-eth0
        0           0           0           06a:66:43:be:8f:46
```

Interfaces

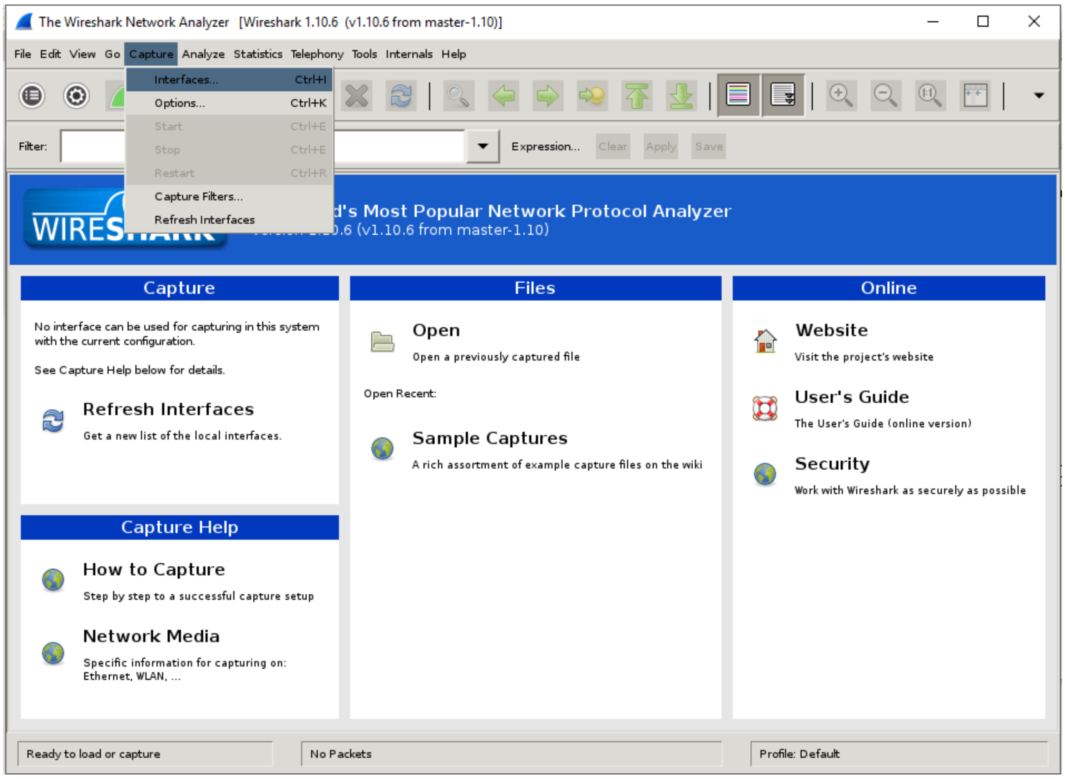
Interface	Linked To	Node Type	IP Address	MAC Address	TXP	RXP	TXB	RXB
h1-eth0	s1-eth1	Host	10.0.0.1	00:00:00:00:00:01	0	0	0	0
h2-eth0	s1-eth2	Host	10.0.0.2	00:00:00:00:00:02	0	0	0	0
h3-eth0	s1-eth3	Host	10.0.0.3	00:00:00:00:00:03	0	0	0	0
s1-eth1	h1-eth0	OVSSwitch	None	ca:13:ca:df:00:68	0	0	0	0
s1-eth2	h2-eth0	OVSSwitch	None	72:b4:1a:d3:d2:2a	0	0	0	0
s1-eth3	h3-eth0	OVSSwitch	None	6a:66:43:be:8f:46	0	0	0	0

2. Menambah openvswitch flow ke topology



```
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.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2016ms
pipe 3
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=21.42s, table=0, n_packets=0, n_bytes=0, idle_age=21, in_port=1 actions=output:2
 cookie=0x0, duration=14.785s, table=0, n_packets=0, n_bytes=0, idle_age=14, in_port=2 actions=output:1
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=0.751 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.034 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.034/0.276/0.751/0.335 ms
mininet> 
```

3. Membuka wireshark dengan perintah 'sudo wireshark &', lalu pilih Interfaces



The World's Most Popular Network Protocol Analyzer
Version 1.10.6 (v1.10.6 from master-1.10)

No interface can be used for capturing in this system with the current configuration.

See Capture Help below for details.

Get a new list of the local interfaces.

Open a previously captured file

A rich assortment of example capture files on the wiki

Visit the project's website

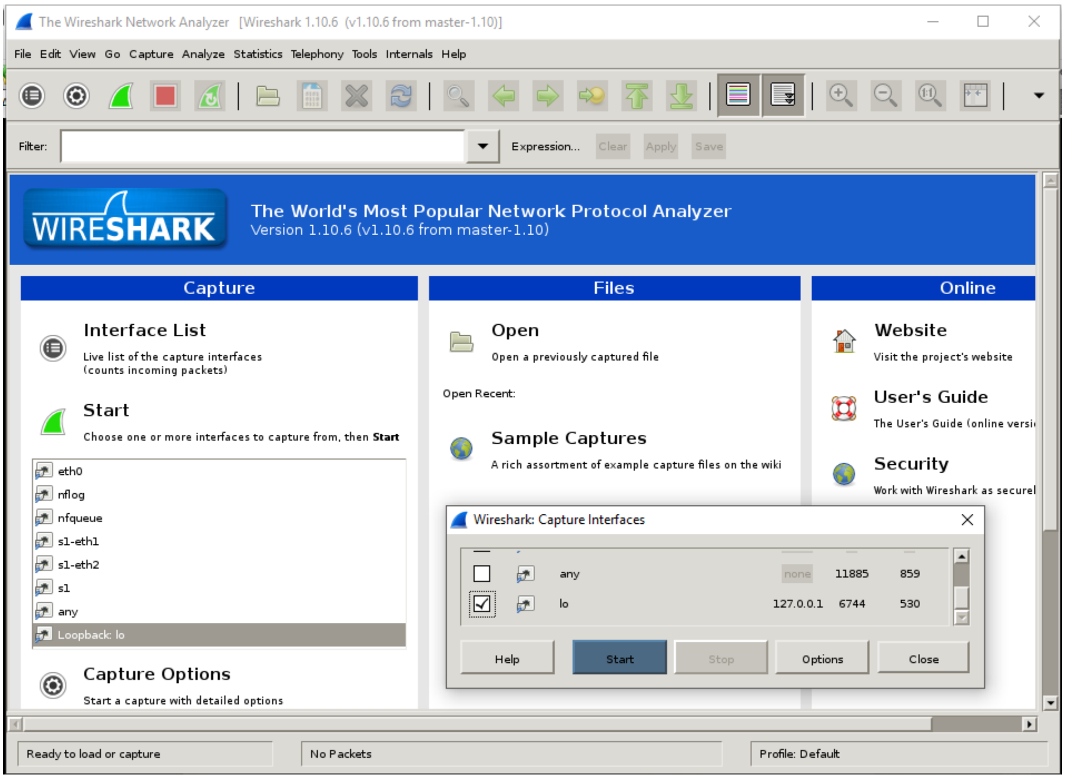
The User's Guide (online version)

Work with Wireshark as securely as possible

Step by step to a successful capture setup

Specific information for capturing on: Ethernet, WLAN, ...

4. Capture Interfaces



The Wireshark Network Analyzer [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
Capture
Files
Online
Interface List
Start
Capture Options
Open
Open Recent:
Sample Captures
Website
User's Guide
Security
Wireshark: Capture Interfaces
Help Start Stop Options Close
Ready to load or capture
No Packets
Profile: Default

The World's Most Popular Network Protocol Analyzer
Version 1.10.6 (v1.10.6 from master-1.10)

Live list of the capture interfaces (counts incoming packets)

Choose one or more interfaces to capture from, then **Start**

eth0
nflog
nfqueue
s1-eth1
s1-eth2
s1
any
Loopback: lo

Start a capture with detailed options

Open a previously captured file

A rich assortment of example capture files on the wiki

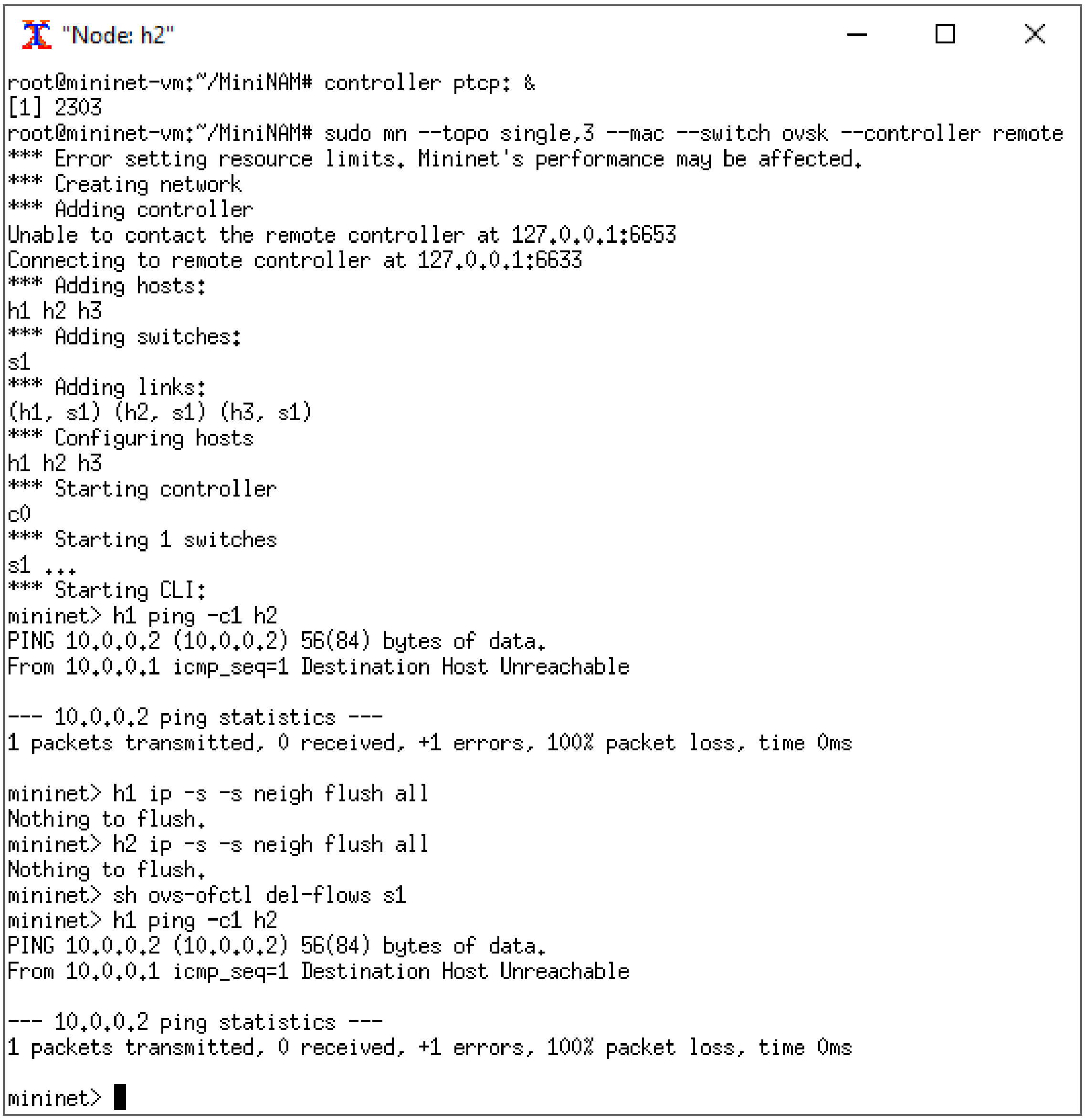
Visit the project's website

The User's Guide (online versi

Work with Wireshark as securel

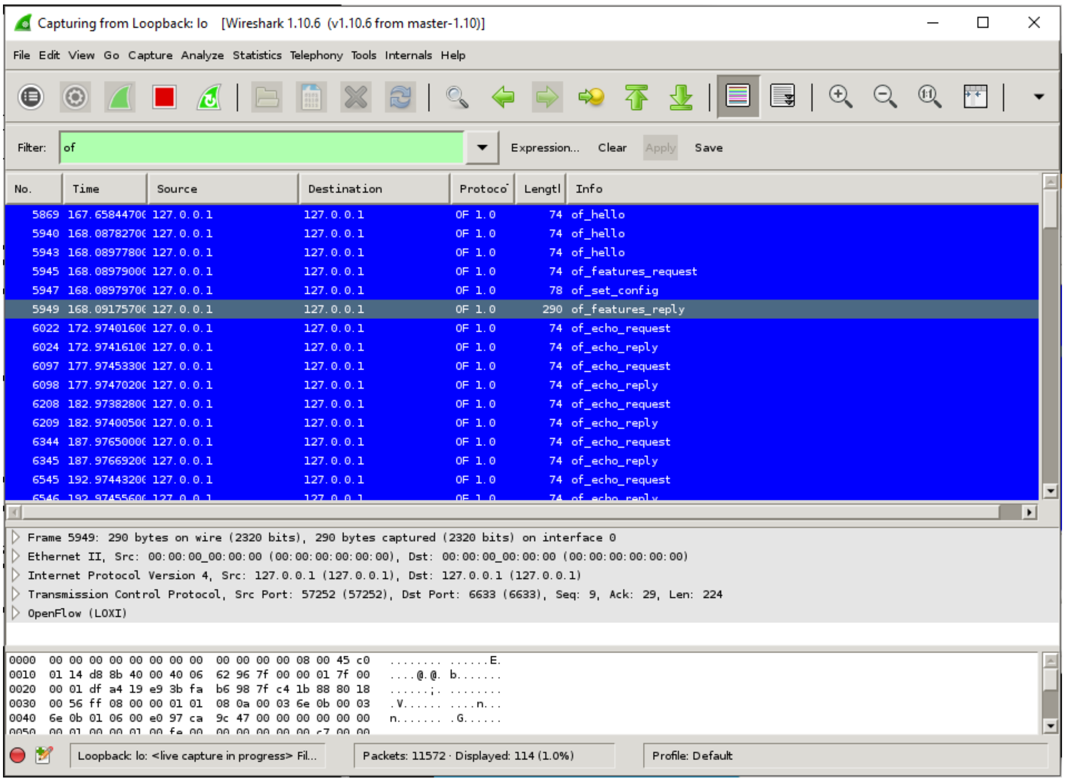
□	any	none	11885	859
☑	lo	127.0.0.1	6744	530

5. Starting Controller



```
root@mininet-vm:~/MiniNAM# controller ptcp: &
[1] 2303
root@mininet-vm:~/MiniNAM# sudo mn --topo single,3 --mac --switch ovsk --controller remote
*** Error setting resource limits. Mininet's performance may be affected.
*** 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> h1 ping -c1 h2
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

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.
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

mininet> 
```

6. Capturing from Loopback (filter of)


Capturing from Loopback: lo [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: of

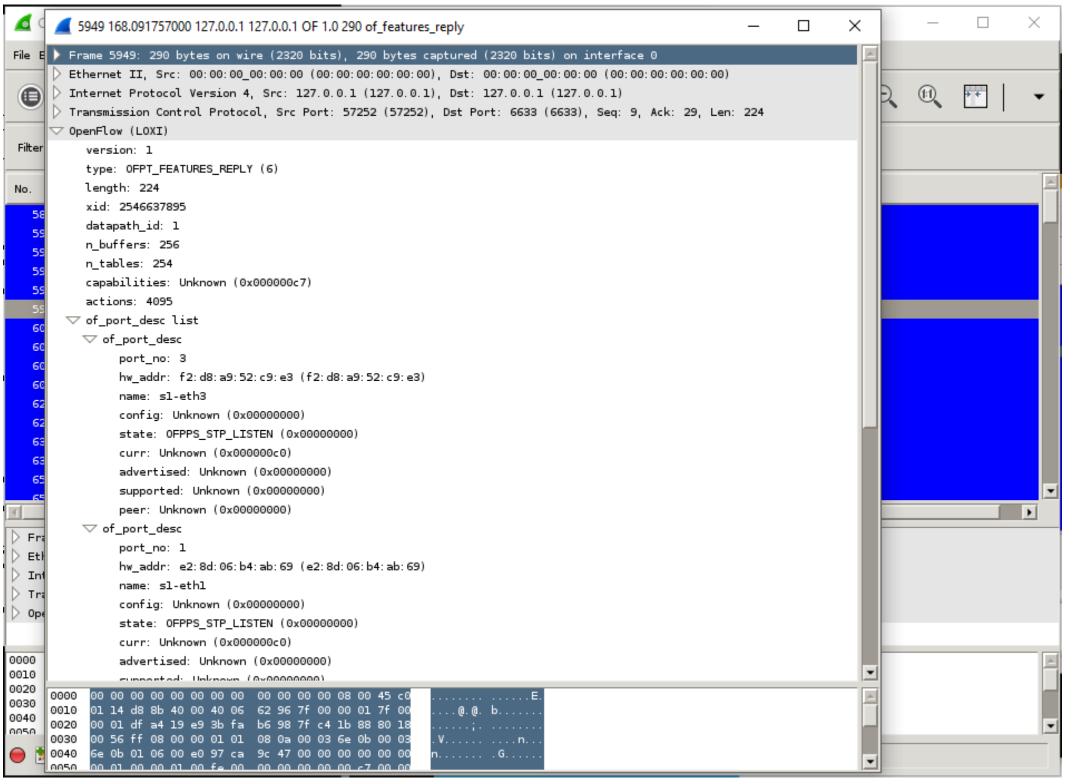
```
No.   Time        Source      Destination  Protocol Length Info
 5869 167.6584470( 127.0.0.1   127.0.0.1    OF 1.0      74 of_hello
 5940 168.0878270( 127.0.0.1   127.0.0.1    OF 1.0      74 of_hello
 5943 168.0897780( 127.0.0.1   127.0.0.1    OF 1.0      74 of_hello
 5945 168.0897900( 127.0.0.1   127.0.0.1    OF 1.0      74 of_features_request
 5947 168.0897970( 127.0.0.1   127.0.0.1    OF 1.0      78 of_set_config
 5949 168.0917570( 127.0.0.1   127.0.0.1    OF 1.0     290 of_features_reply
 6022 172.9740160( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_request
 6024 172.9741610( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_reply
 6097 177.9745330( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_request
 6098 177.9747020( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_reply
 6208 182.9738280( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_request
 6209 182.9740050( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_reply
 6344 187.9765000( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_request
 6345 187.9766920( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_reply
 6545 192.9744320( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_request
 6546 192.9745560( 127.0.0.1   127.0.0.1    OF 1.0      74 of_echo_reply
```

```
▷ Frame 5949: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0
▷ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 57252 (57252), Dst Port: 6633 (6633), Seq: 9, Ack: 29, Len: 224
▷ OpenFlow (LOXI)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ........ ......E.
0010  01 14 d8 8b 40 00 40 06  62 96 7f 00 00 01 7f 00   ....@.@. b.......
0020  00 01 df a4 19 e9 3b fa  b6 98 7f c4 1b 88 80 18   ......;. ........
0030  00 56 ff 08 00 00 01 01  08 0a 00 03 6e 0b 00 03   .V...... ....n...
0040  6e 0b 01 06 00 e0 97 ca  9c 47 00 00 00 00 00 00   n....... .G......
0050  00 01 00 00 01 00 fe 00  00 00 00 00 00 c7 00 00
```

Loopback: lo: <live capture in progress> Fil... Packets: 11572 · Displayed: 114 (1.0%) Profile: Default

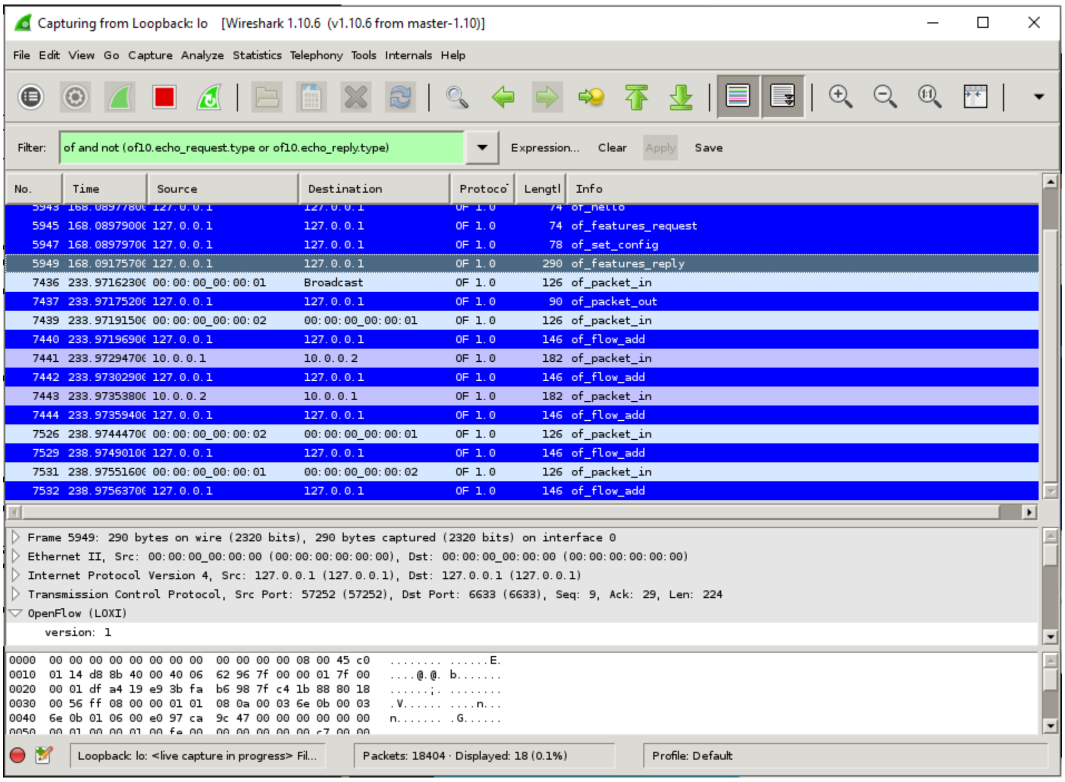
7. Inspect Packet


5949 168.091757000 127.0.0.1 127.0.0.1 OF 1.0 290 of_features_reply

```
▷ Frame 5949: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0
▷ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 57252 (57252), Dst Port: 6633 (6633), Seq: 9, Ack: 29, Len: 224
▽ OpenFlow (LOXI)
    version: 1
    type: OFPT_FEATURES_REPLY (6)
    length: 224
    xid: 2546637895
    datapath_id: 1
    n_buffers: 256
    n_tables: 254
    capabilities: Unknown (0x000000c7)
    actions: 4095
  ▽ of_port_desc list
    ▽ of_port_desc
        port_no: 3
        hw_addr: f2:d8:a9:52:c9:e3 (f2:d8:a9:52:c9:e3)
        name: s1-eth3
        config: Unknown (0x00000000)
        state: OFPPS_STP_LISTEN (0x00000000)
        curr: Unknown (0x000000c0)
        advertised: Unknown (0x00000000)
        supported: Unknown (0x00000000)
        peer: Unknown (0x00000000)
    ▽ of_port_desc
        port_no: 1
        hw_addr: e2:8d:06:b4:ab:69 (e2:8d:06:b4:ab:69)
        name: s1-eth1
        config: Unknown (0x00000000)
        state: OFPPS_STP_LISTEN (0x00000000)
        curr: Unknown (0x000000c0)
        advertised: Unknown (0x00000000)
        supported: Unknown (0x00000000)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ........ ......E.
0010  01 14 d8 8b 40 00 40 06  62 96 7f 00 00 01 7f 00   ....@.@. b.......
0020  00 01 df a4 19 e9 3b fa  b6 98 7f c4 1b 88 80 18   ......;. ........
0030  00 56 ff 08 00 00 01 01  08 0a 00 03 6e 0b 00 03   .V...... ....n...
0040  6e 0b 01 06 00 e0 97 ca  9c 47 00 00 00 00 00 00   n....... .G......
0050  00 01 00 00 01 00 fe 00  00 00 00 00 00 c7 00 00
```

8. Flter of and not



Capturing from Loopback: lo [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

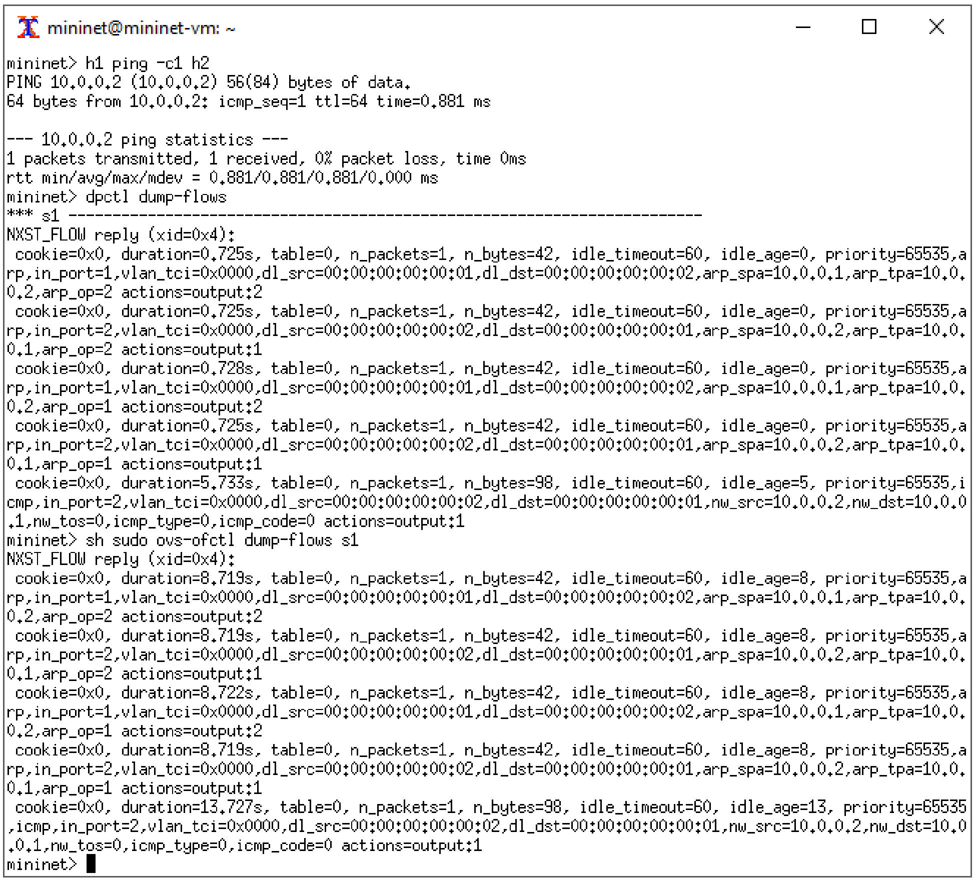
Filter: of and not (of10.echo_request.type or of10.echo_reply.type)

Expression... Clear Apply Save

```
No.   Time             Source              Destination         Protocol Length Info
 5943 168.0897780(     127.0.0.1           127.0.0.1           OF 1.0    74  of_hello
 5945 168.0897900(     127.0.0.1           127.0.0.1           OF 1.0    74  of_features_request
 5947 168.0897970(     127.0.0.1           127.0.0.1           OF 1.0    78  of_set_config
 5949 168.0917570(     127.0.0.1           127.0.0.1           OF 1.0   290  of_features_reply
 7436 233.9716230(     00:00:00_00:00:01   Broadcast           OF 1.0   126  of_packet_in
 7437 233.9717520(     127.0.0.1           127.0.0.1           OF 1.0    90  of_packet_out
 7439 233.9719150(     00:00:00_00:00:02   00:00:00_00:00:01   OF 1.0   126  of_packet_in
 7440 233.9719690(     127.0.0.1           127.0.0.1           OF 1.0   146  of_flow_add
 7441 233.9729470(     10.0.0.1            10.0.0.2            OF 1.0   182  of_packet_in
 7442 233.9730290(     127.0.0.1           127.0.0.1           OF 1.0   146  of_flow_add
 7443 233.9735380(     10.0.0.2            10.0.0.1            OF 1.0   182  of_packet_in
 7444 233.9735940(     127.0.0.1           127.0.0.1           OF 1.0   146  of_flow_add
 7526 238.9744470(     00:00:00_00:00:02   00:00:00_00:00:01   OF 1.0   126  of_packet_in
 7529 238.9749010(     127.0.0.1           127.0.0.1           OF 1.0   146  of_flow_add
 7531 238.9755160(     00:00:00_00:00:01   00:00:00_00:00:02   OF 1.0   126  of_packet_in
 7532 238.9756370(     127.0.0.1           127.0.0.1           OF 1.0   146  of_flow_add

▷ Frame 5949: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0
▷ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 57252 (57252), Dst Port: 6633 (6633), Seq: 9, Ack: 29, Len: 224
▽ OpenFlow (LOXI)
    version: 1

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ........ ......E.
0010  01 14 d8 8b 40 00 40 06  62 96 7f 00 00 01 7f 00   ....@.@. b.......
0020  00 01 df a4 19 e9 3b fa  b6 98 7f c4 1b 88 80 18   ......;. ........
0030  00 56 ff 08 00 00 01 01  08 0a 00 03 6e 0b 00 03   .V...... ....n...
0040  6e 0b 01 06 00 e0 97 ca  9c 47 00 00 00 00 00 00   n....... .G......
0050  00 01 00 00 01 00 fe 00  00 00 00 00 00 c7 00 00
```

Loopback: lo: <live capture in progress> Fil... Packets: 18404 · Displayed: 18 (0.1%) Profile: Default

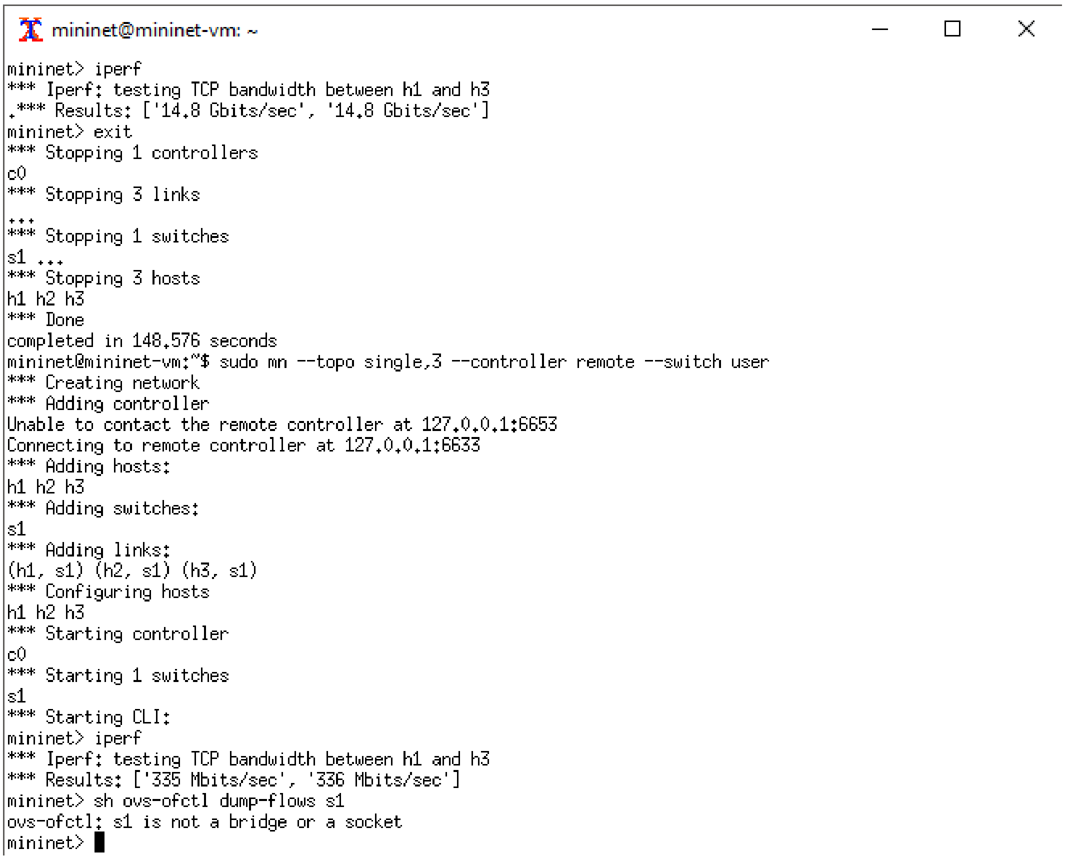
9. Flow entries.



mininet@mininet-vm: ~

```
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=0.881 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.881/0.881/0.881/0.000 ms
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=0.725s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,a
rp,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=0.725s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,a
rp,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=0.728s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,a
rp,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=1 actions=output:2
 cookie=0x0, duration=0.725s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=0, priority=65535,a
rp,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=5.733s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=5, priority=65535,i
cmp,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> sh sudo ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=8.719s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=8, priority=65535,a
rp,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=8.719s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=8, priority=65535,a
rp,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=8.722s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=8, priority=65535,a
rp,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=1 actions=output:2
 cookie=0x0, duration=8.719s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=8, priority=65535,a
rp,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=13.727s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=13, 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> █
```

10. Benchmark kernel- vs -user space



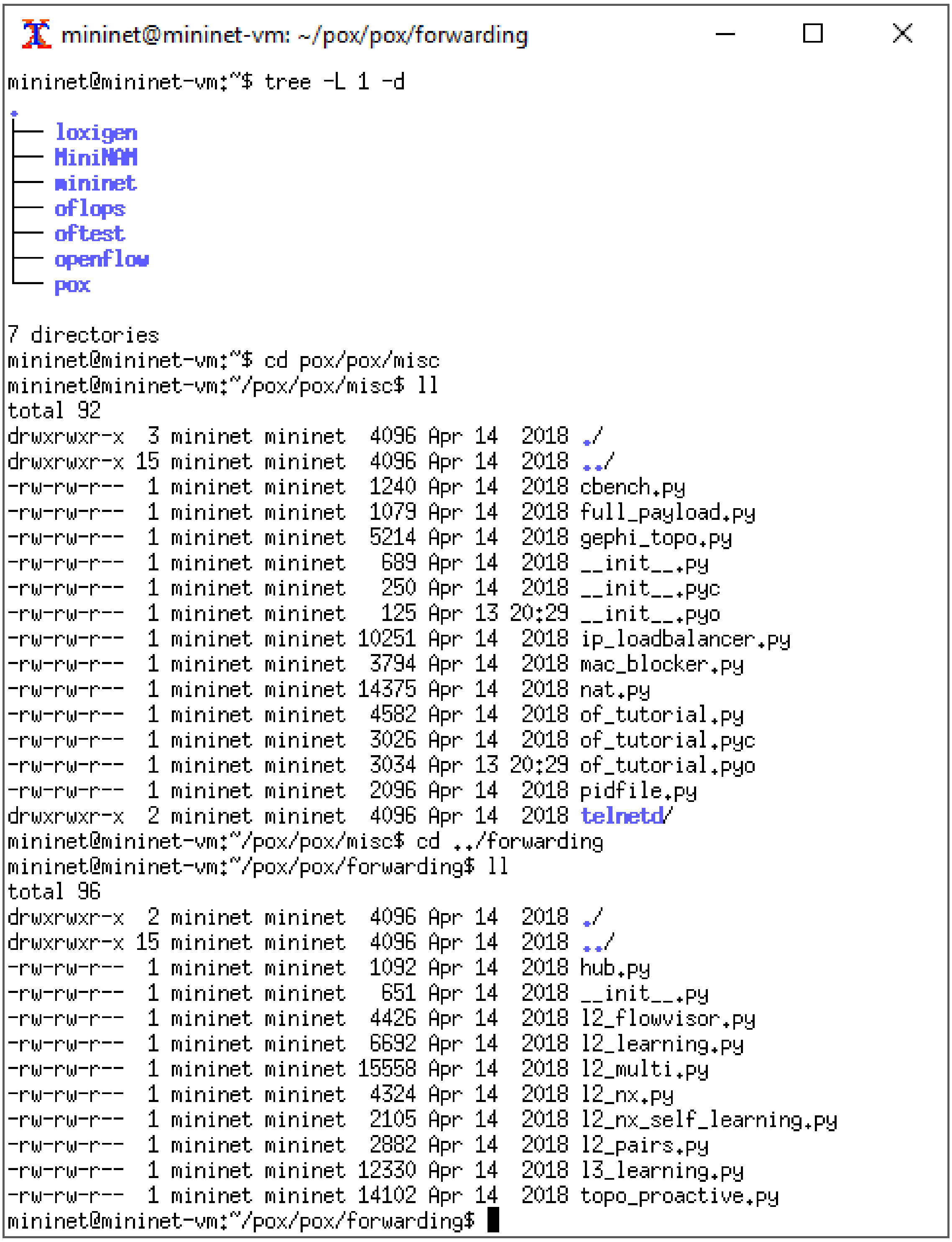
```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
.*** Results: ['14.8 Gbits/sec', '14.8 Gbits/sec']
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
s1 ...
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 148.576 seconds
mininet@mininet-vm:~$ 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> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['335 Mbits/sec', '336 Mbits/sec']
mininet> sh ovs-ofctl dump-flows s1
ovs-ofctl: s1 is not a bridge or a socket
mininet> █
```

2.4 POX Controller

Hal yang terpenting dalam system ini agar bekerja adalah adanya komunikasi antara Mininet Netwok dan POX Controller. POX Controller adalah controller yang open source dan berbasis Phyton untuk OpenFlow/SDN. Dengan menggunakan POX Controller, dumb OpenFlow devices dapat menjadi hub, switch, load balancer, firewall devices. POX Controller menyediakan cara yang mudah untuk menjalankan OpenFlow/SDN.

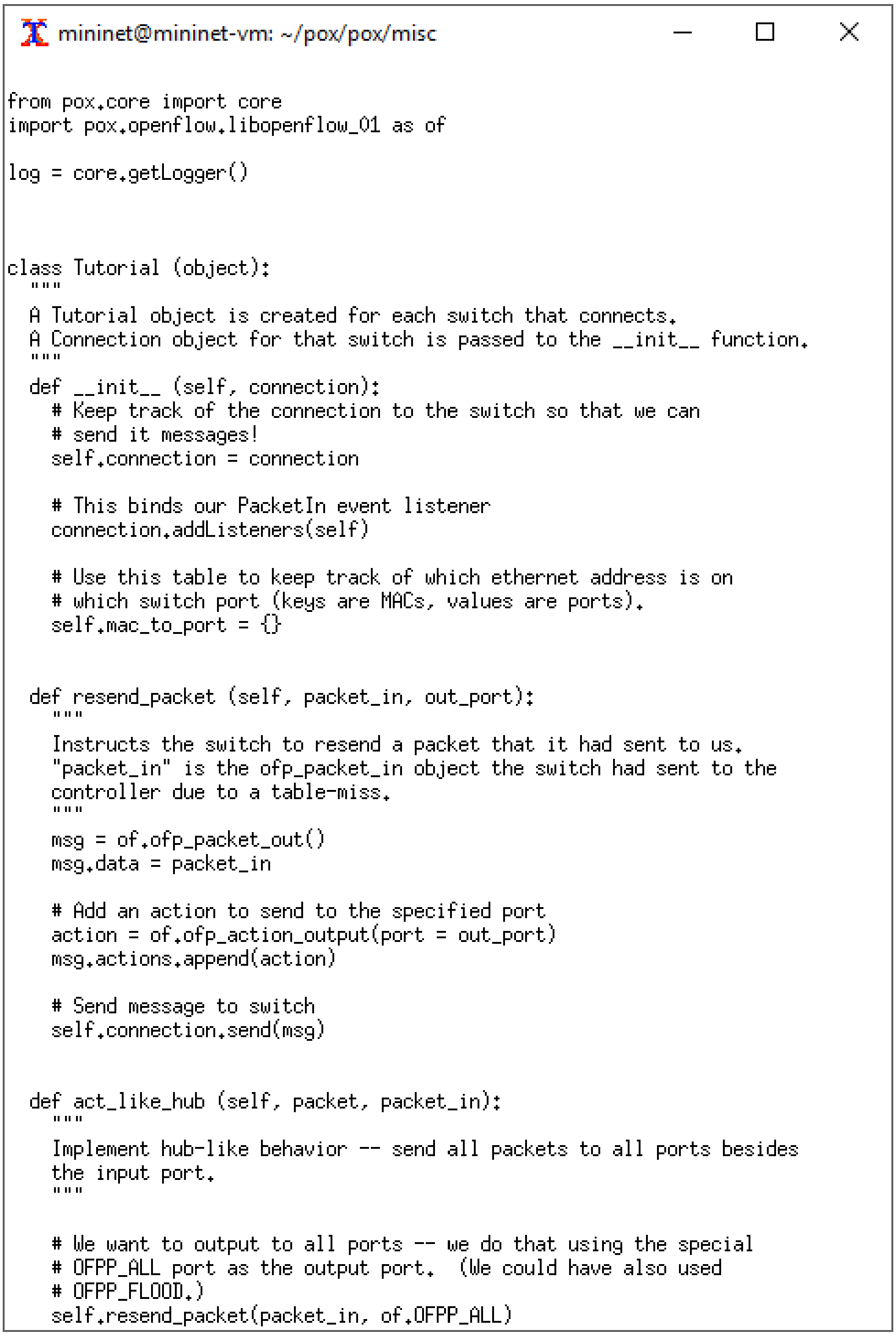
Berikut langkah-langkah yang dilakukan dengan POX Controller.

1. Membuka pox controller file



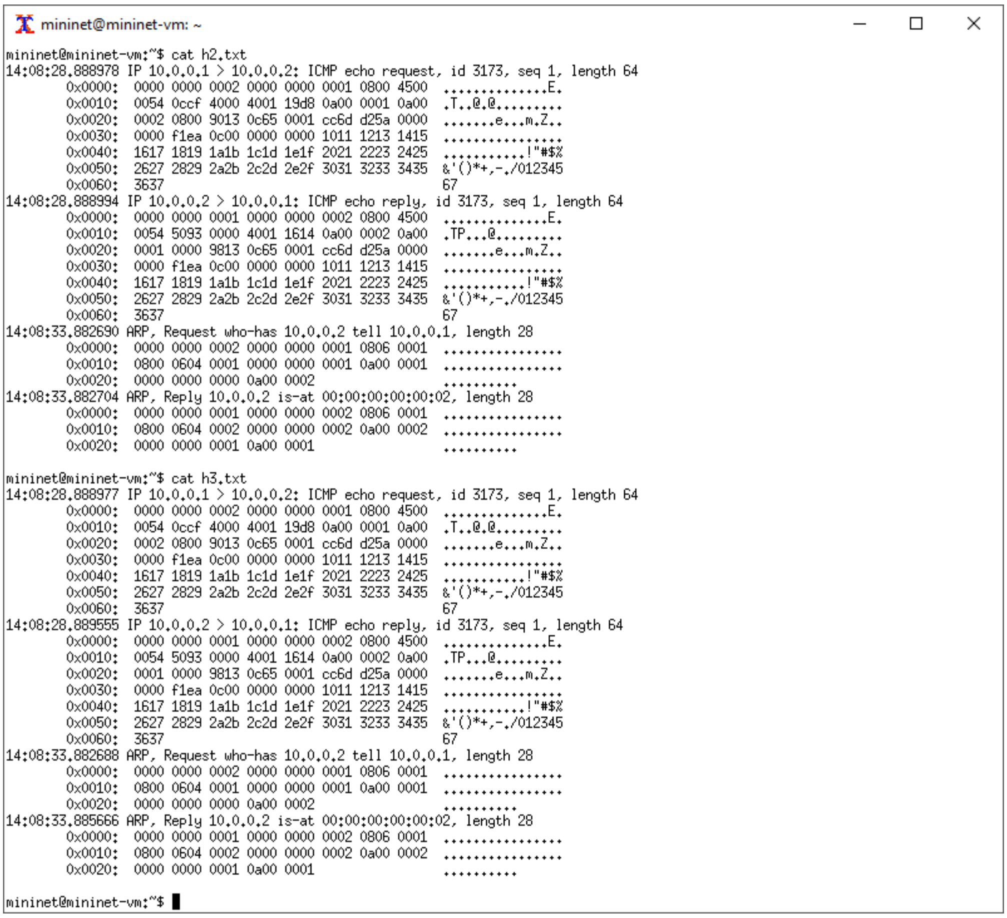
```
mininet@mininet-vm:~$ tree -L 1 -d
.
├── loxigen
├── MiniNAM
├── mininet
├── oflops
├── oftest
├── openflow
└── pox

7 directories
mininet@mininet-vm:~$ cd pox/pox/misc
mininet@mininet-vm:~/pox/pox/misc$ ll
total 92
drwxrwxr-x  3 mininet mininet  4096 Apr 14  2018 ./
drwxrwxr-x 15 mininet mininet  4096 Apr 14  2018 ../
-rw-rw-r--  1 mininet mininet  1240 Apr 14  2018 cbench.py
-rw-rw-r--  1 mininet mininet  1079 Apr 14  2018 full_payload.py
-rw-rw-r--  1 mininet mininet  5214 Apr 14  2018 gephi_topo.py
-rw-rw-r--  1 mininet mininet   689 Apr 14  2018 __init__.py
-rw-rw-r--  1 mininet mininet   250 Apr 14  2018 __init__.pyc
-rw-rw-r--  1 mininet mininet   125 Apr 13 20:29 __init__.pyo
-rw-rw-r--  1 mininet mininet 10251 Apr 14  2018 ip_loadbalancer.py
-rw-rw-r--  1 mininet mininet  3794 Apr 14  2018 mac_blocker.py
-rw-rw-r--  1 mininet mininet 14375 Apr 14  2018 nat.py
-rw-rw-r--  1 mininet mininet  4582 Apr 14  2018 of_tutorial.py
-rw-rw-r--  1 mininet mininet  3026 Apr 14  2018 of_tutorial.pyc
-rw-rw-r--  1 mininet mininet  3034 Apr 13 20:29 of_tutorial.pyo
-rw-rw-r--  1 mininet mininet  2096 Apr 14  2018 pidfile.py
drwxrwxr-x  2 mininet mininet  4096 Apr 14  2018 telnetd/
mininet@mininet-vm:~/pox/pox/misc$ cd ../forwarding
mininet@mininet-vm:~/pox/pox/forwarding$ ll
total 96
drwxrwxr-x  2 mininet mininet  4096 Apr 14  2018 ./
drwxrwxr-x 15 mininet mininet  4096 Apr 14  2018 ../
-rw-rw-r--  1 mininet mininet  1092 Apr 14  2018 hub.py
-rw-rw-r--  1 mininet mininet   651 Apr 14  2018 __init__.py
-rw-rw-r--  1 mininet mininet  4426 Apr 14  2018 l2_flowvisor.py
-rw-rw-r--  1 mininet mininet  6692 Apr 14  2018 l2_learning.py
-rw-rw-r--  1 mininet mininet 15558 Apr 14  2018 l2_multi.py
-rw-rw-r--  1 mininet mininet  4324 Apr 14  2018 l2_nx.py
-rw-rw-r--  1 mininet mininet  2105 Apr 14  2018 l2_nx_self_learning.py
-rw-rw-r--  1 mininet mininet  2882 Apr 14  2018 l2_pairs.py
-rw-rw-r--  1 mininet mininet 12330 Apr 14  2018 l3_learning.py
-rw-rw-r--  1 mininet mininet 14102 Apr 14  2018 topo_proactive.py
mininet@mininet-vm:~/pox/pox/forwarding$ █
```

2. Membuka of_tutorial.py



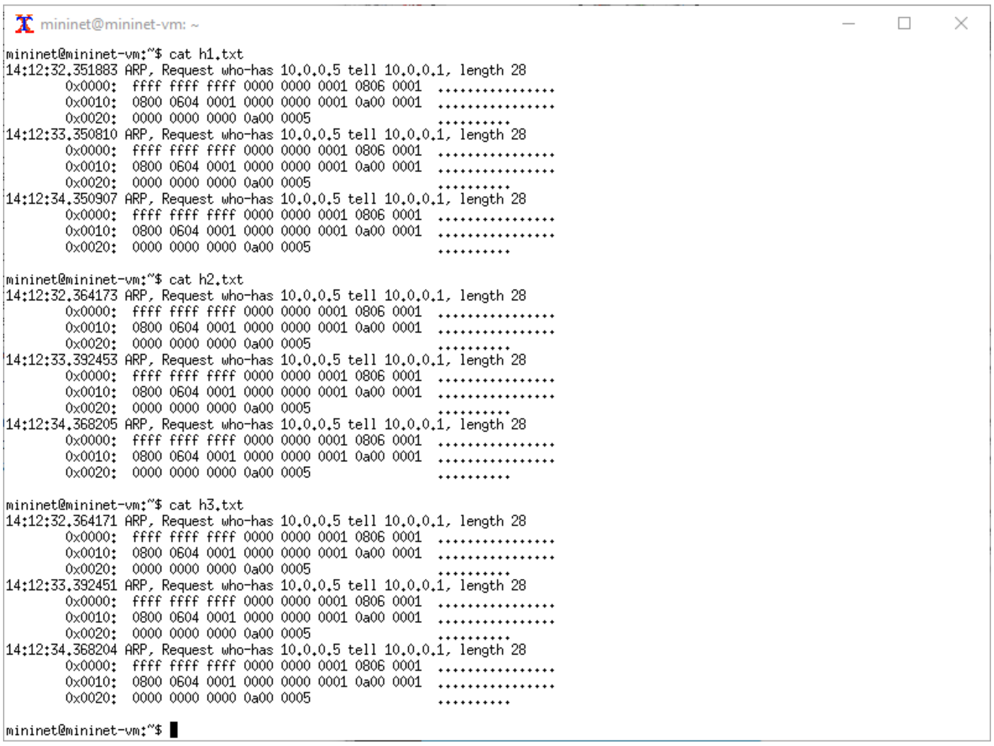
```
from pox.core import core
import pox.openflow.libopenflow_01 as of

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)
```

3. Cat h2.txt



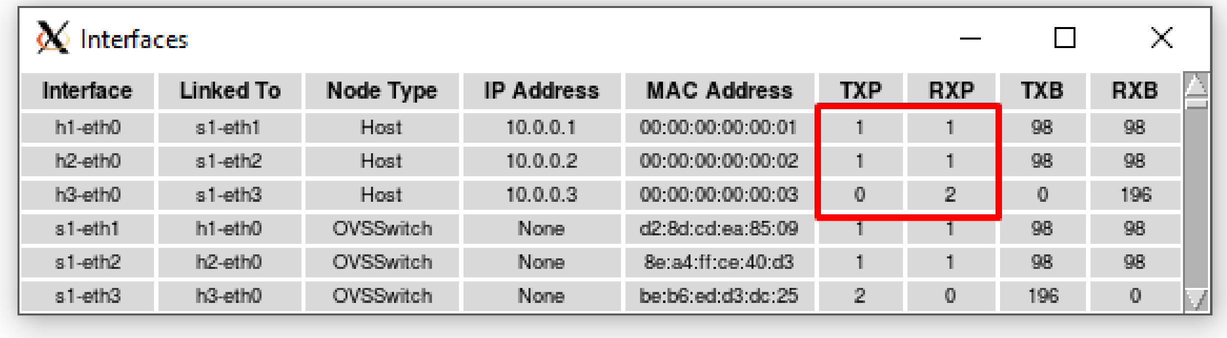
```
mininet@mininet-vm:~$ cat h2.txt
14:08:28.888978 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 3173, seq 1, length 64
        0x0000:  0000 0000 0002 0000 0000 0001 0800 4500  ..............E.
        0x0010:  0054 0ccf 4000 4001 19d8 0a00 0001 0a00  .T..@.@.........
        0x0020:  0002 0800 9013 0c65 0001 cc6d d25a 0000  .......e...m.Z..
        0x0030:  0000 f1ea 0c00 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .......... !"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
14:08:28.888994 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 3173, seq 1, length 64
        0x0000:  0000 0000 0001 0000 0000 0002 0800 4500  ..............E.
        0x0010:  0054 5093 0000 4001 1614 0a00 0002 0a00  .TP...@.........
        0x0020:  0001 0000 9813 0c65 0001 cc6d d25a 0000  .......e...m.Z..
        0x0030:  0000 f1ea 0c00 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .......... !"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
14:08:33.882690 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
        0x0000:  0000 0000 0002 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0002                 ..........
14:08:33.882704 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                 ..........

mininet@mininet-vm:~$ cat h3.txt
14:08:28.888977 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 3173, seq 1, length 64
        0x0000:  0000 0000 0002 0000 0000 0001 0800 4500  ..............E.
        0x0010:  0054 0ccf 4000 4001 19d8 0a00 0001 0a00  .T..@.@.........
        0x0020:  0002 0800 9013 0c65 0001 cc6d d25a 0000  .......e...m.Z..
        0x0030:  0000 f1ea 0c00 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .......... !"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
14:08:28.889555 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 3173, seq 1, length 64
        0x0000:  0000 0000 0001 0000 0000 0002 0800 4500  ..............E.
        0x0010:  0054 5093 0000 4001 1614 0a00 0002 0a00  .TP...@.........
        0x0020:  0001 0000 9813 0c65 0001 cc6d d25a 0000  .......e...m.Z..
        0x0030:  0000 f1ea 0c00 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  .......... !"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
14:08:33.882688 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
        0x0000:  0000 0000 0002 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0002                 ..........
14:08:33.885666 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                 ..........

mininet@mininet-vm:~$ ▌
```

4. Cat h1.txt



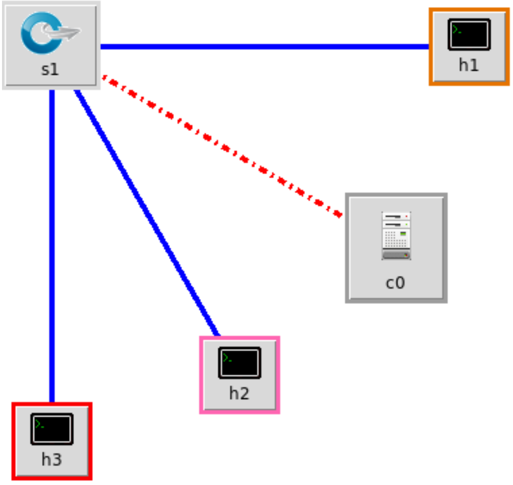
```
mininet@mininet-vm:~$ cat h1.txt
14:12:32.351883 ARP, Request who-has 10.0.0.5 tell 10.0.0.1, length 28
        0x0000:  ffff ffff ffff 0000 0000 0001 0806 0001  ................
        0x0010:  0800 0604 0001 0000 0000 0001 0a00 0001  ................
        0x0020:  0000 0000 0000 0a00 0005                 ..........
14:12:33.350810 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                 ..........
14:12:34.350907 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:~$ cat h2.txt
14:12:32.364173 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                 ..........
14:12:33.392453 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                 ..........
14:12:34.368205 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:~$ cat h3.txt
14:12:32.364171 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                 ..........
14:12:33.392451 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                 ..........
14:12:34.368204 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:~$ ▌
```

5. MiniNAM ping 10.0.0.2

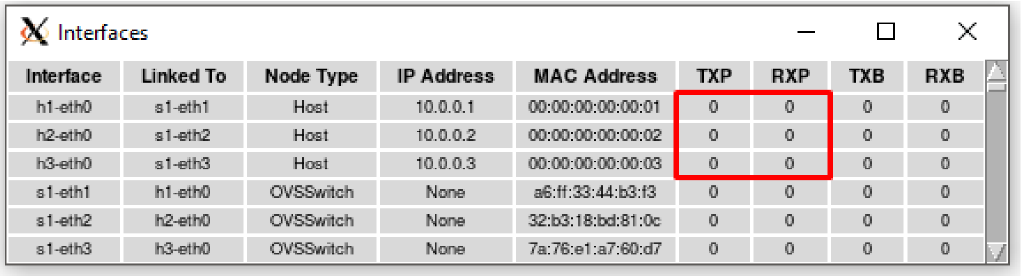


Interfaces

Interface	Linked To	Node Type	IP Address	MAC Address	TXP	RXP	TXB	RXB
h1-eth0	s1-eth1	Host	10.0.0.1	00:00:00:00:00:01	1	1	98	98
h2-eth0	s1-eth2	Host	10.0.0.2	00:00:00:00:00:02	1	1	98	98
h3-eth0	s1-eth3	Host	10.0.0.3	00:00:00:00:00:03	0	2	0	196
s1-eth1	h1-eth0	OVSSwitch	None	d2:8d:cd:ea:85:09	1	1	98	98
s1-eth2	h2-eth0	OVSSwitch	None	8e:a4:ff:ce:40:d3	1	1	98	98
s1-eth3	h3-eth0	OVSSwitch	None	be:b6:ed:d3:dc:25	2	0	196	0

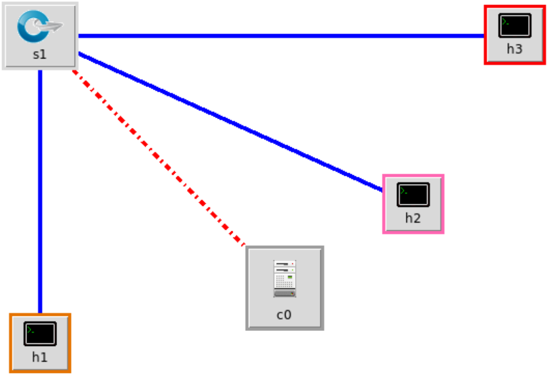


6. MiniNAM ping 10.0.0.5

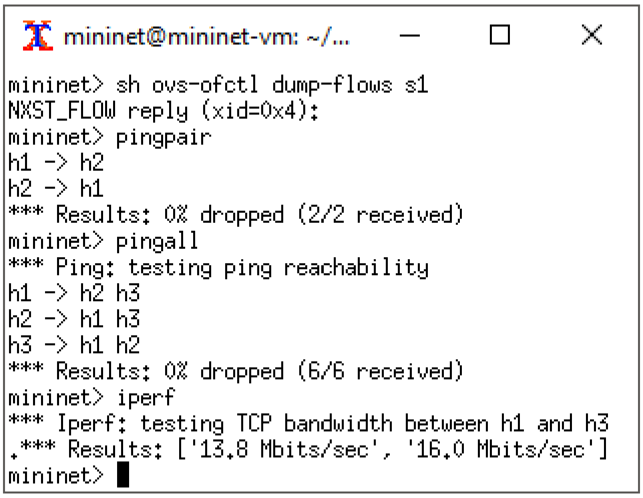


Interfaces

Interface	Linked To	Node Type	IP Address	MAC Address	TXP	RXP	TXB	RXB
h1-eth0	s1-eth1	Host	10.0.0.1	00:00:00:00:00:01	0	0	0	0
h2-eth0	s1-eth2	Host	10.0.0.2	00:00:00:00:00:02	0	0	0	0
h3-eth0	s1-eth3	Host	10.0.0.3	00:00:00:00:00:03	0	0	0	0
s1-eth1	h1-eth0	OVSSwitch	None	a6:ff:33:44:b3:f3	0	0	0	0
s1-eth2	h2-eth0	OVSSwitch	None	32:b3:18:bd:81:0c	0	0	0	0
s1-eth3	h3-eth0	OVSSwitch	None	7a:76:e1:a7:60:d7	0	0	0	0

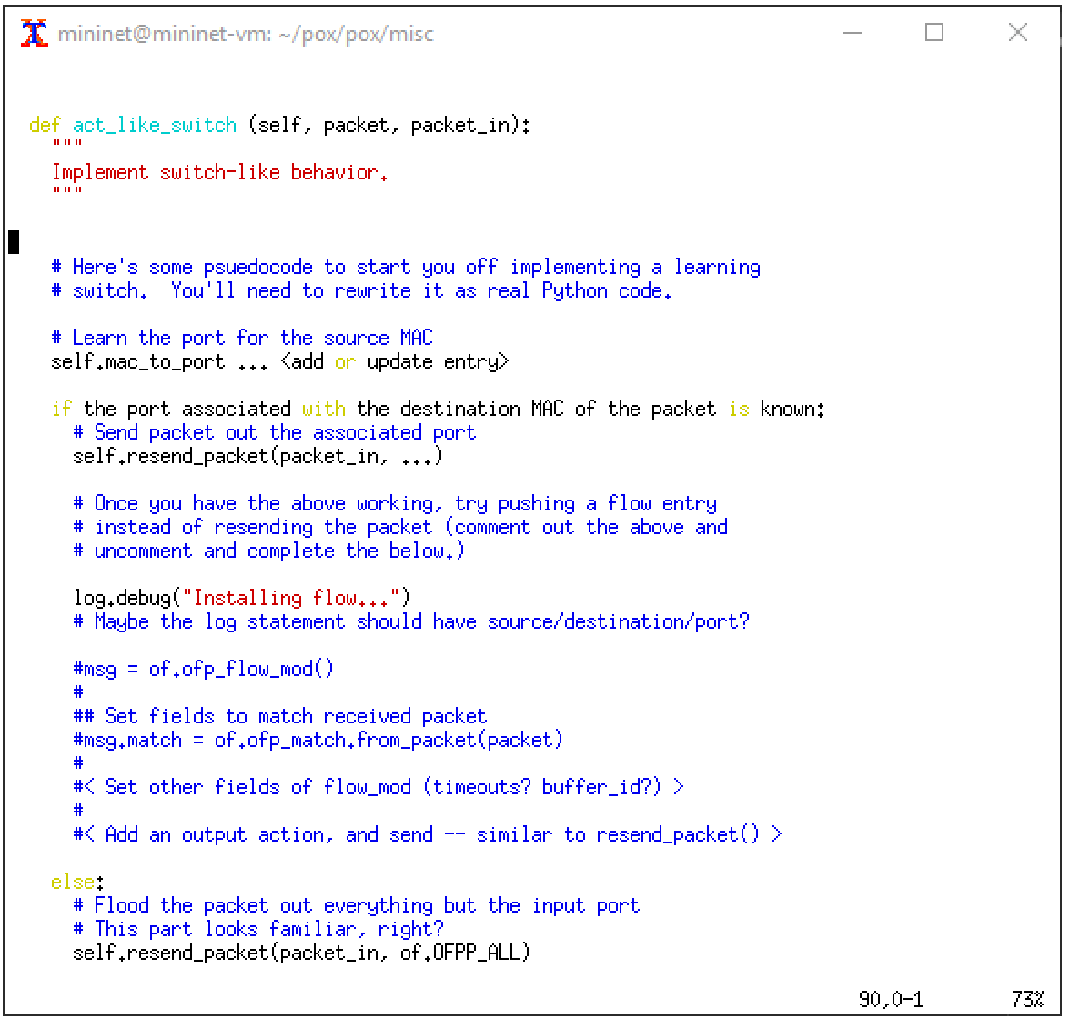


7. Manual Entry (pingpair, pingall, iperf)



```
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: ['13.8 Mbits/sec', '16.0 Mbits/sec']
mininet>
```

8. of_tutorial.py



```
 def act_like_switch (self, packet, packet_in):
   """
   Implement switch-like behavior.
   """


   # Here's some psuedocode to start you off implementing a learning
   # switch.  You'll need to rewrite it as real Python code.

   # Learn the port for the source MAC
   self.mac_to_port ... <add or update entry>

   if the port associated with the destination MAC of the packet is known:
     # Send packet out the associated port
     self.resend_packet(packet_in, ...)

     # Once you have the above working, try pushing a flow entry
     # instead of resending the packet (comment out the above and
     # uncomment and complete the below.)

     log.debug("Installing flow...")
     # Maybe the log statement should have source/destination/port?

     #msg = of.ofp_flow_mod()
     #
     ## Set fields to match received packet
     #msg.match = of.ofp_match.from_packet(packet)
     #
     #< Set other fields of flow_mod (timeouts? buffer_id?) >
     #
     #< Add an output action, and send -- similar to resend_packet() >

   else:
     # Flood the packet out everything but the input port
     # This part looks familiar, right?
     self.resend_packet(packet_in, of.OFPP_ALL)
                                                              90,0-1        73%
```

BAB III
PENUTUP

3.1 Kesimpulan

SDN memisahkan Control Plane dari suatu perangkat jaringan (switch / router) dari Data Plane, sehingga memungkinkan untuk mengontrol, memantau, dan mengelola jaringan dari pengontrol terpusat. Hal ini berbeda dengan konsep jaringan tradisional, yaitu Control Plane dan Data Plane berada dalam satu perangkat networking yang sama.

Dengan adanya pemisahaan tersebut maka diharapkan perangkat jaringan dapat di manage melalui controllernya saja, maka untuk mewujudkan tersebut dibutuhkanlah sebuah API untuk mengkoneksikan seluruh perangkat jaringan kedalam sebuah controller yang dapat di program sesuai kebutuhan perusahaan, dari sanalah paradigma Software Define Network muncul, dimana jaringan dapat diataur atau di definisikan melalui sebuah software.

3.2 Saran

Kedepannya penggunaan internet(jaringan komputer) akan semakin meningkat dan peningkatan ini tentu akan menimbulkan kompleksitas yang tinggi dalam hal managemen jaringan. Oleh karena itu, SDN hadir untuk membawa inovasi, penulis menyarankan agar dilakukan banyak penelitian dan kajian ilmiah terhadap Sofware-Define Networking.