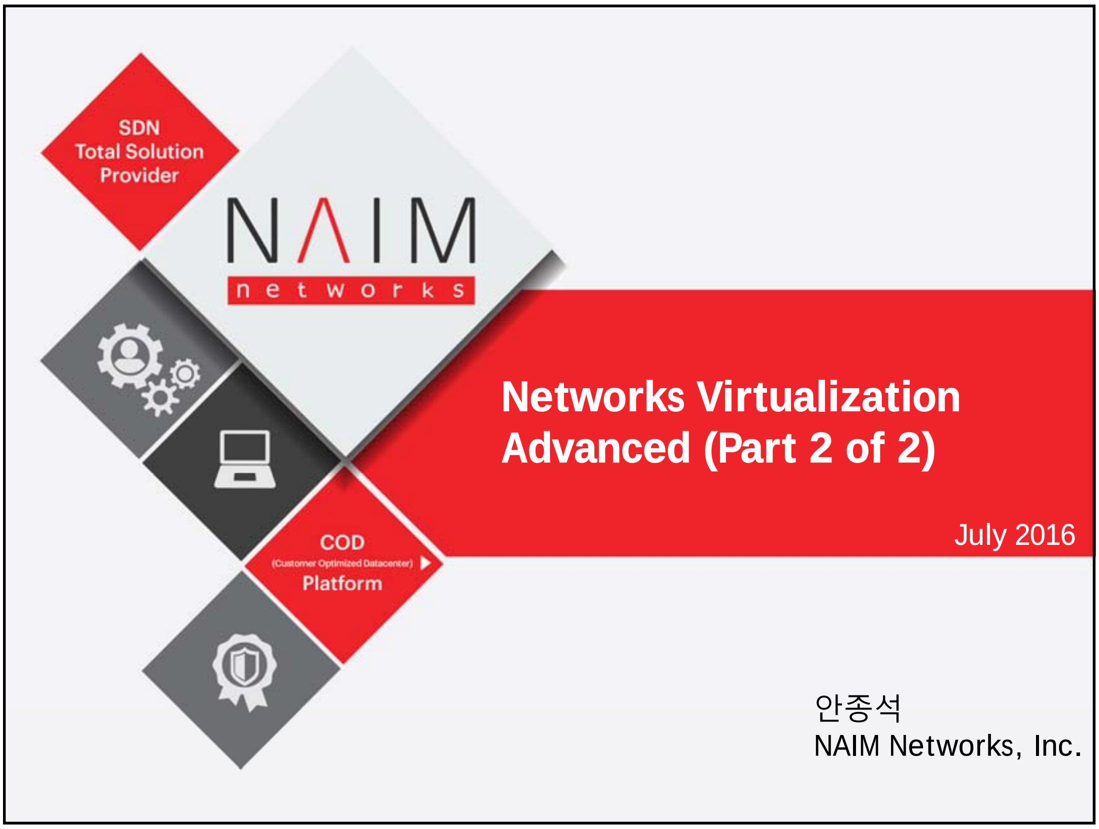
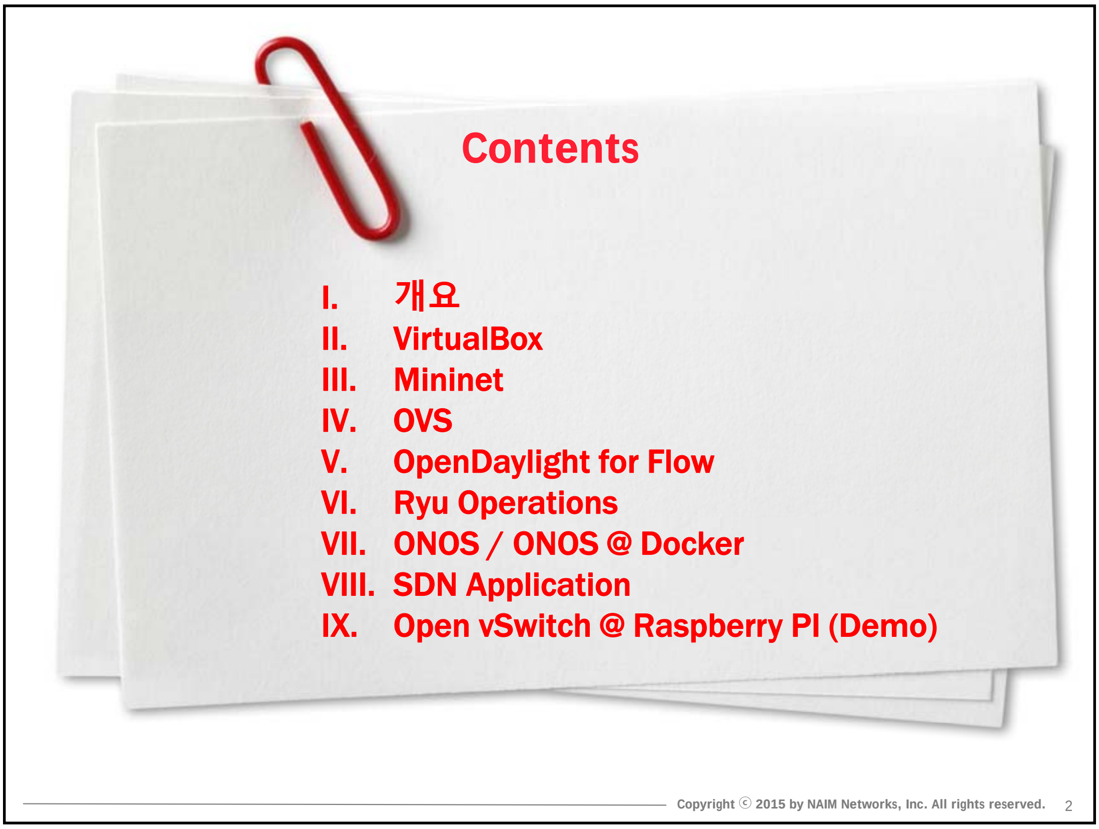


SDN
Total Solution
Provider
NAIM
networks
COD
(Customer Optimized Datacenter)
Platform

Networks Virtualization Advanced (Part 2 of 2)

July 2016

안종석
NAIM Networks, Inc.



Contents

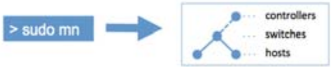
I. 개요
II. VirtualBox
III. Mininet
IV. OVS
V. OpenDaylight for Flow
VI. Ryu Operations
VII. ONOS / ONOS @ Docker
VIII. SDN Application
IX. Open vSwitch @ Raspberry PI (Demo)

I. 개요

- **내용:** SDN 기반 가상 네트워크 실습
- **환경:** M/B 가상화 지원 PC (8 GB RAM 이상, 8GB 이상 HDD, 인터넷 지원)
- **시연 / 실습:** 실습 환경 불가능 한 것은 시연만

I. 개요 - Tools

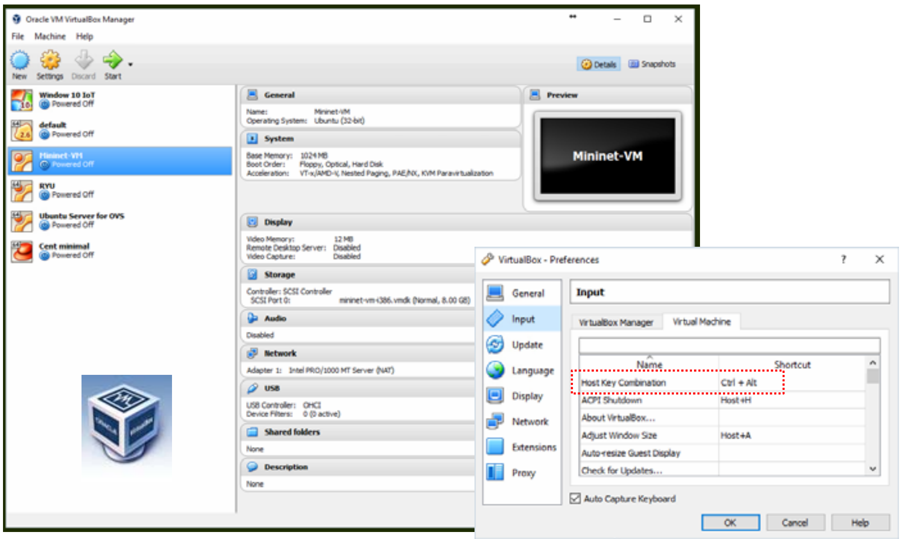
1. **VirtualBox VM** (https://www.virtualbox.org/wiki/Downloads)
2. **Mininet** (http://mininet.org/download/)



3. **ONOS Tutorial VM** (https://wiki.onosproject.org/display/ONOS/Downloads)
4. **WinSCP** (http://winscp.net/)
5. **Xming** (http://sourceforge.net/projects/xming/)
6. **Putty** (http://www.putty.org/)

II. VirtualBox

1. **PC에서 Virtualization 활성화** (Boot Option)
2. **Host key @ Preference**



Name: Mininet-VM
Operating System: Ubuntu (32-bit)

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Video Memory: 12 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Controller: SCSI Controller
SCSI Port 0: mininet-vm-i386.vmdk (Normal, 8.00 GB)

Disabled

Adapter 1: Intel PRO/1000 MT Server (NAT)

USB Controller: OHCI
Device Filters: 0 (0 active)

None

None

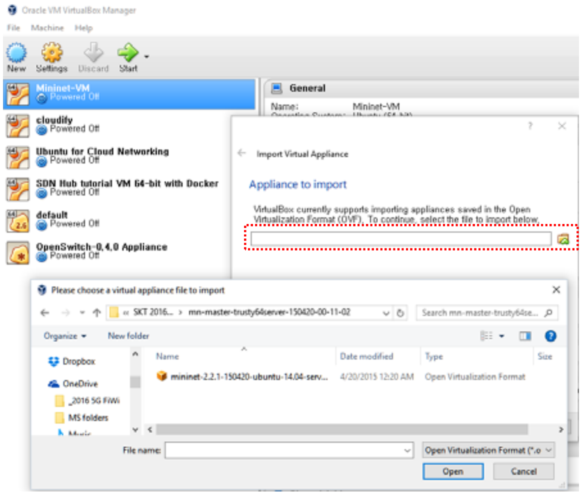
Input

Name	Shortcut
Host Key Combination	Ctrl + Alt
ACPI Shutdown	Host+H
About VirtualBox...	
Adjust Window Size	Host+A
Auto-resize Guest Display	
Check for Updates...	

☑ Auto Capture Keyboard

II. VirtualBox

1. ***.ovf 파일 사용 (Import Appliance @ File menu)**



Name: Mininet-VM

Appliance to import

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

Please choose a virtual appliance file to import

« SKT 2016... › mn-master-trusty64server-150420-00-11-02

Search mn-master-trusty64se...

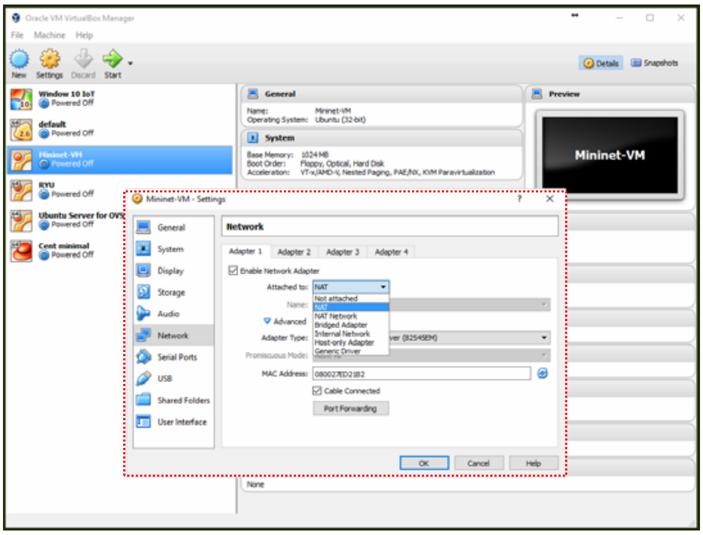
Name	Date modified	Type	Size
mininet-2.2.1-150420-ubuntu-14.04-serv...	4/20/2015 12:20 AM	Open Virtualization Format	

File name:

Open Virtualization Format (*.o

III. Mininet

1. **Network : Bridged Adapter** (Host-only or NAT)
2. **Promiscuous Mode** (Nested Hypervisor등 필요시 무작위모드)



Oracle VM VirtualBox Manager

Name: Mininet-VM
Operating System: Ubuntu (32-bit)

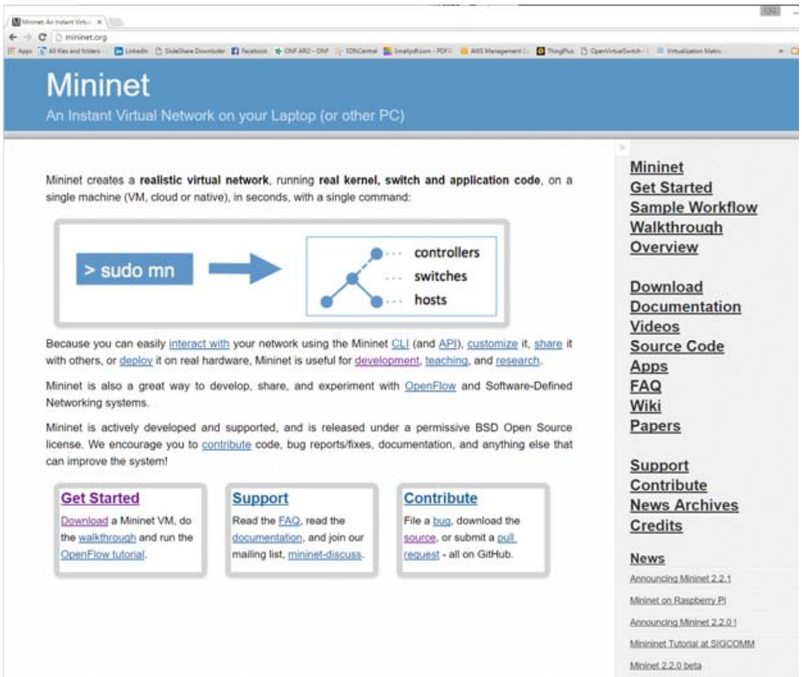
Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Mininet-VM - Settings

Network

☑ Enable Network Adapter
Attached to: NAT
MAC Address: 080027ED21B2
☑ Cable Connected

III. Mininet



mininet.org

Mininet

An Instant Virtual Network on your Laptop (or other PC)

Mininet creates a **realistic virtual network**, running **real kernel, switch and application code**, on a single machine (VM, cloud or native), in seconds, with a single command:

Because you can easily interact with your network using the Mininet CLI (and API), customize it, share it with others, or deploy it on real hardware, Mininet is useful for development, teaching, and research.

Mininet is also a great way to develop, share, and experiment with OpenFlow and Software-Defined Networking systems.

Mininet is actively developed and supported, and is released under a permissive BSD Open Source license. We encourage you to contribute code, bug reports/fixes, documentation, and anything else that can improve the system!

Get Started

Download a Mininet VM, do the walkthrough and run the OpenFlow tutorial.

Support

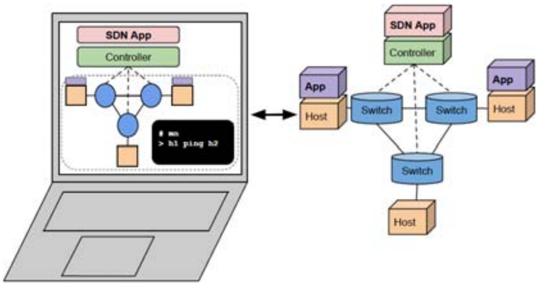
Read the FAQ, read the documentation, and join our mailing list, mininet-discuss.

Contribute

File a bug, download the source, or submit a pull request - all on GitHub.

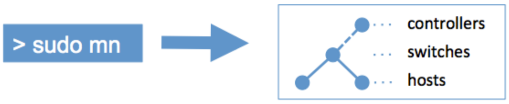
III. Mininet – What is Mininet?

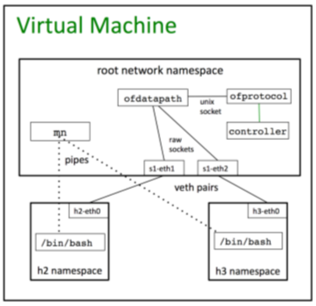
- 빠름
- 원하는 토폴로지 생성 가능 http://ramonfontes.com/vnd/#
- 실제 프로그램을 사용 (anything that can run on Linux can run on a Mininet host)
- 패킷 전송을 커스터마이즈 가능
- 프로그램 가능한 오픈플로우 스위치 사용
- 사용하기 쉬움 (Apps move seamlessly to/from hardware)
- 오픈소스

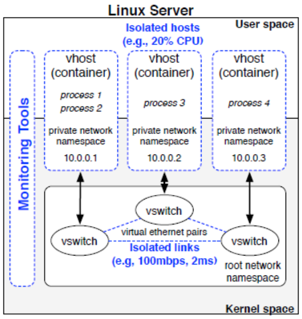


III. Mininet – Why Use Mininet?

- 1 대의 PC에서 가상 네트워크 환경을 제공
- 실제 커널을 컨테이너 기술 기반으로 스위치 애플리케이션 코드를 사용 을 사용
- 명령어, UI, 파이손(Python) 인터페이스 제공
- 오픈플로우(OpenFlow) 기능 포함



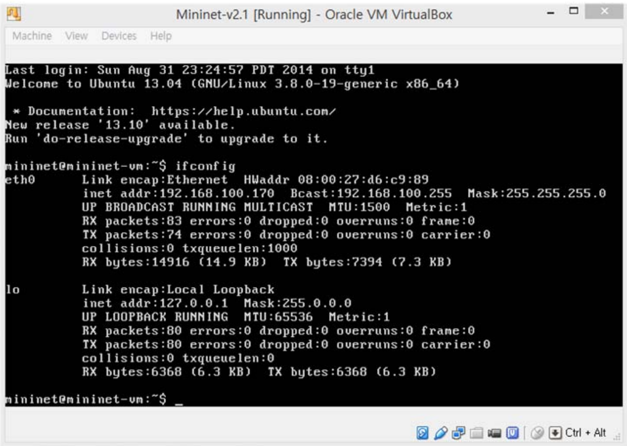




III. Mininet – Running Mininet@VM

1. Run 'Oracle VM VirtualBox Manager'
2. Start VM '**Mininet-v2.2.1**'
3. User ID/Password: **mininet/mininet**
4. ifconfig (@ VM '**Mininet-v2.2.1**')
5. $ sudo mn --topo single,3 --mac –controller=remote, ip=x.x.x.x, port=6633

6. h1 ping h2
7. mininet> exit
8. $ sudo mn –c

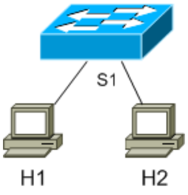


```
Last login: Sun Aug 31 23:24:57 PDT 2014 on tty1
Welcome to Ubuntu 13.04 (GNU/Linux 3.8.0-19-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '13.10' available.
Run 'do-release-upgrade' to upgrade to it.

mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d6:c9:89
          inet addr:192.168.100.170  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:83 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14916 (14.9 KB)  TX bytes:7394 (7.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6368 (6.3 KB)  TX bytes:6368 (6.3 KB)

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

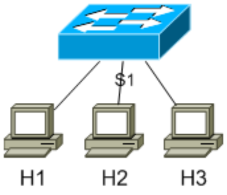
III. Mininet – configuration

sudo mn --topo minimal
mininet&gt; **net**
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0

sudo mn --topo single,3
mininet&gt; **net**
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0



Minimal

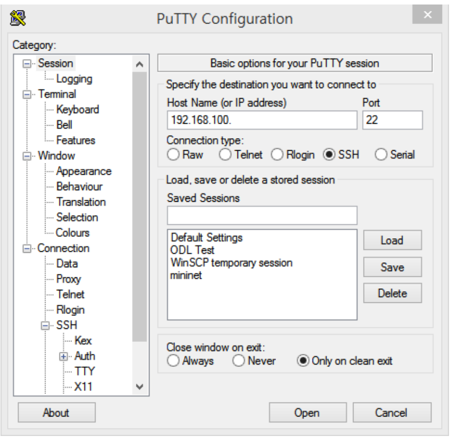


Single (3)

xterm h1 h2

III. Mininet – PuTTY w/Xming

1. **Xming Installation**
2. **PuTTY Installation**
3. **PuTTY Configuration** (w/X11)
4. NAT 선택시 Mininet 접속 되지 않는 경우 Bridge or Host Only Adaptor로 변경



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.100.

Port: 22

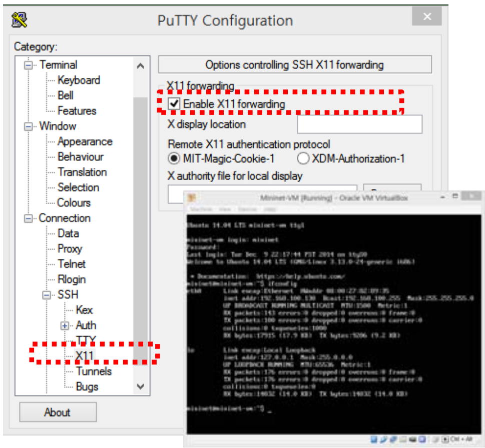
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
ODL Test
WinSCP temporary session
mininet

Close window on exit: ○ Always ○ Never ● Only on clean exit



PuTTY Configuration

Category:

Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

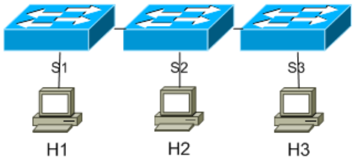
X authority file for local display

Mininet-VM [Running] - Oracle VM VirtualBox

```
Ubuntu 14.04 LTS mininet-vm tty1

mininet-vm login: mininet
Password:
Last login: Tue Dec  9 22:17:44 PST 2014 on tty[illegible]
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:02:09:35
          inet addr:192.168.100.130  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:143 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17915 (17.9 KB)  TX bytes:9206 (9.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:14032 (14.0 KB)  TX bytes:14032 (14.0 KB)

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

III. Mininet – configuration

```
sudo mn --topo linear,3
mininet&gt; net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
s1 lo:  s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo:  s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo:  s3-eth1:h3-eth0 s3-eth2:s2-eth3
```



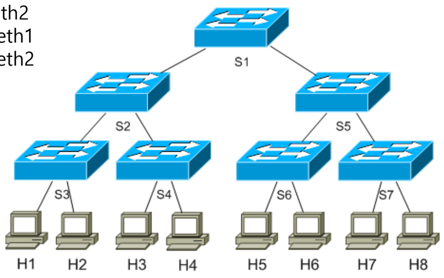
Linear (3)

xterm h1 h2 h3 @PuTTY

- ifconfig
- ping

III. Mininet – configuration

```
sudo mn --topo tree,3
mininet&gt; net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s3-eth2
h3 h3-eth0:s4-eth1
h4 h4-eth0:s4-eth2
h5 h5-eth0:s6-eth1
h6 h6-eth0:s6-eth2
h7 h7-eth0:s7-eth1
h8 h8-eth0:s7-eth2
s1 lo:  s1-eth1:s2-eth3 s1-eth2:s5-eth3
s2 lo:  s2-eth1:s3-eth3 s2-eth2:s4-eth3 s2-eth3:s1-eth1
s3 lo:  s3-eth1:h1-eth0 s3-eth2:h2-eth0 s3-eth3:s2-eth1
s4 lo:  s4-eth1:h3-eth0 s4-eth2:h4-eth0 s4-eth3:s2-eth2
s5 lo:  s5-eth1:s6-eth3 s5-eth2:s7-eth3 s5-eth3:s1-eth2
s6 lo:  s6-eth1:h5-eth0 s6-eth2:h6-eth0 s6-eth3:s5-eth1
s7 lo:  s7-eth1:h7-eth0 s7-eth2:h8-eth0 s7-eth3:s5-eth2
```

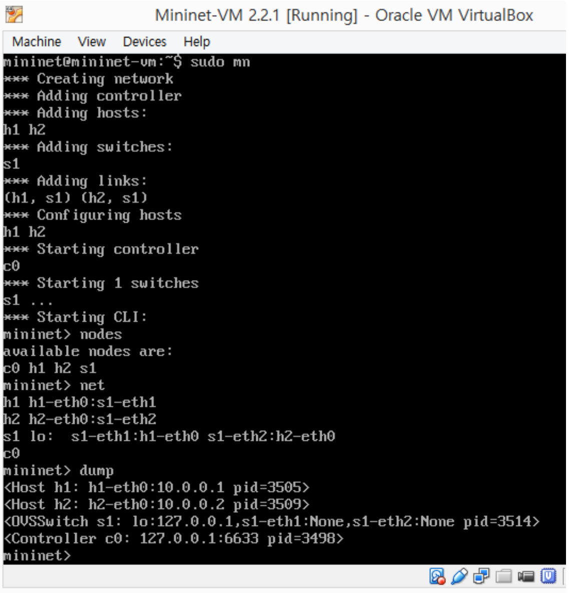


Tree (3)

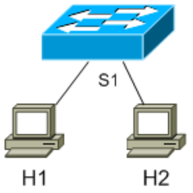
III. Mininet – Operations

1. sudo mn
2. mininet&gt; **nodes**
3. mininet&gt; **net**
4. mininet&gt; **dump**
5. mininet&gt; **xterm h1 h2 s1**

6. mininet&gt; **pingall**
7. mininet&gt; **link h1 s1 down**
8. mininet&gt; **h1 ping -c 1 h2**
9. connect: Network is unreachable



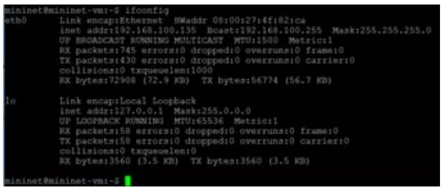
```
mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3505>
<Host h2: h2-eth0:10.0.0.2 pid=3509>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=3514>
<Controller c0: 127.0.0.1:6633 pid=3498>
mininet>
```



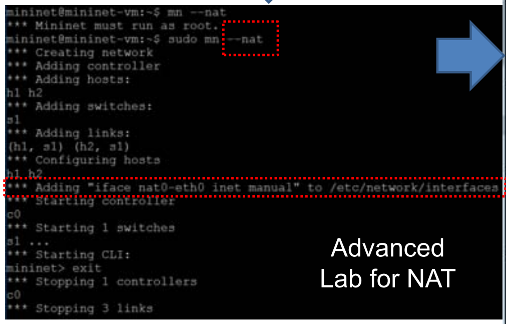
Minimal

III. Mininet – Operations ‘mn --nat’

외부 네트워크와 연결 시 사용

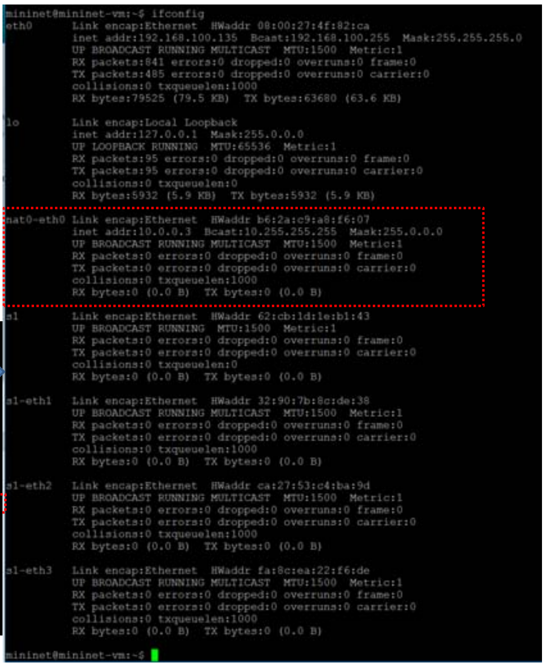


```
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4f:82:ca
          inet addr:192.168.100.135  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:745 errors:0 dropped:0 overruns:0 frame:0
          TX packets:430 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:72908 (72.9 KB)  TX bytes:56774 (56.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:58 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3560 (3.5 KB)  TX bytes:3560 (3.5 KB)

mininet@mininet-vm:~$
```



```
mininet@mininet-vm:~$ mn --nat
*** Mininet must run as root.
mininet@mininet-vm:~$ sudo mn --nat
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Adding "iface nat0-eth0 inet manual" to /etc/network/interfaces
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
```

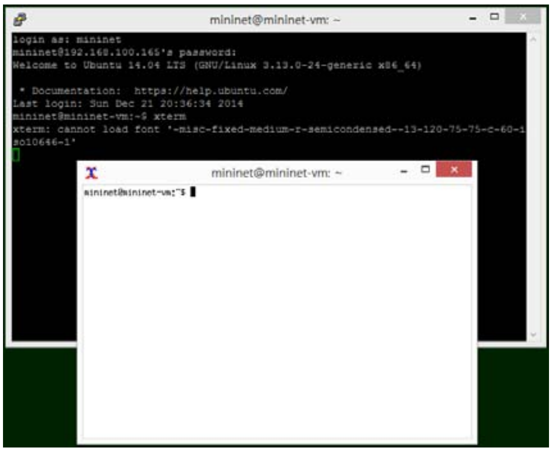
Advanced
Lab for NAT



```
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4f:82:ca
          inet addr:192.168.100.135  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:841 errors:0 dropped:0 overruns:0 frame:0
          TX packets:485 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:79525 (79.5 KB)  TX bytes:63680 (63.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:95 errors:0 dropped:0 overruns:0 frame:0
          TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5932 (5.9 KB)  TX bytes:5932 (5.9 KB)

nat0-eth0 Link encap:Ethernet  HWaddr b6:2a:c9:a8:f6:07
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1        Link encap:Ethernet  HWaddr 62:cb:1d:1e:b1:43
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1-eth1   Link encap:Ethernet  HWaddr 32:90:7b:8c:de:38
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1-eth2   Link encap:Ethernet  HWaddr ca:27:53:c4:ba:9d
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1-eth3   Link encap:Ethernet  HWaddr fa:8c:ea:22:f6:de
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mininet@mininet-vm:~$
```

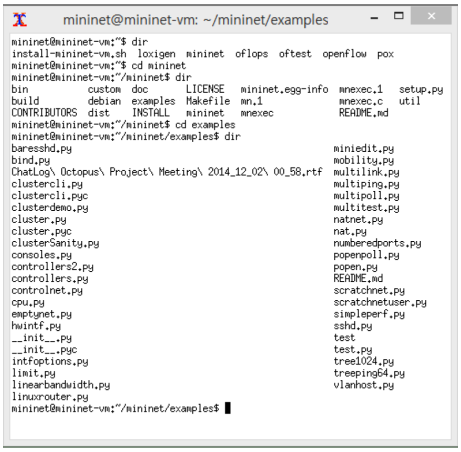
III. Mininet – MiniEdit

1. **Start and Running Mininet VM**
2. **Running Xming** (Double Click ‘Xming’ Icon)
3. **Running PuTTY** w/X11
4. **‘xtrem’** @ Mininet (PuTTY w/X11)
5. **‘sudo ./miniedit.py’** (@ mininet@mininet-vm:~mininet/examples$)



```
login as: mininet
mininet@192.168.100.165's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun Dec 21 20:36:34 2014
mininet@mininet-vm:~$ xterm
xterm: cannot load font '-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-i
so10646-1'
```

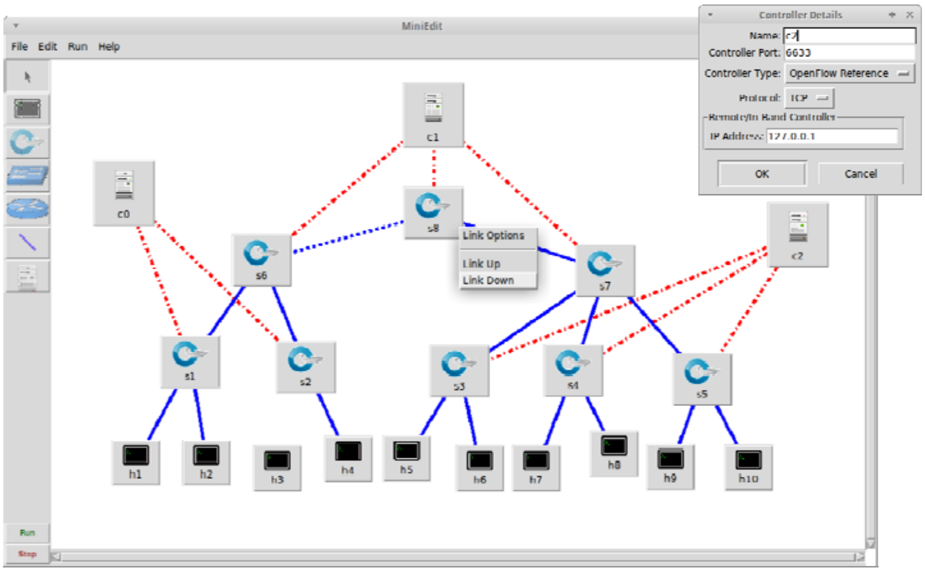
```
mininet@mininet-vm:~$
```



```
mininet@mininet-vm:~$ dir
install-mininet-vm.sh  loxigen  mininet  oflops  oftest  openflow  pox
mininet@mininet-vm:~$ cd mininet
mininet@mininet-vm:~/mininet$ dir
bin           custom  doc       LICENSE   mininet.egg-info  mnexec.1   setup.py
build         debian  examples  Makefile  mn.1              mnexec.c   util
CONTRIBUTORS  dist    INSTALL   mininet   mnexec            README.md
mininet@mininet-vm:~/mininet$ cd examples
mininet@mininet-vm:~/mininet/examples$ dir
baresshd.py                                           miniedit.py
bind.py                                               mobility.py
ChatLog\ Octopus\ Project\ Meeting\ 2014_12_02\ 00_58.rtf  multilink.py
clustercli.py                                         multiping.py
clustercli.pyc                                        multipoll.py
clusterdemo.py                                        multitest.py
cluster.py                                            natnet.py
cluster.pyc                                           nat.py
clusterSanity.py                                      numberedports.py
consoles.py                                           popenpoll.py
controllers2.py                                       popen.py
controllers.py                                        README.md
controlnet.py                                         scratchnet.py
cpu.py                                                scratchnetuser.py
emptynet.py                                           simpleperf.py
hwintf.py                                             sshd.py
__init__.py                                           test
__init__.pyc                                          test.py
intfoptions.py                                        tree1024.py
limit.py                                              treeping64.py
linearbandwidth.py                                    vlanhost.py
linuxrouter.py
mininet@mininet-vm:~/mininet/examples$
```

III. Mininet – MiniEdit

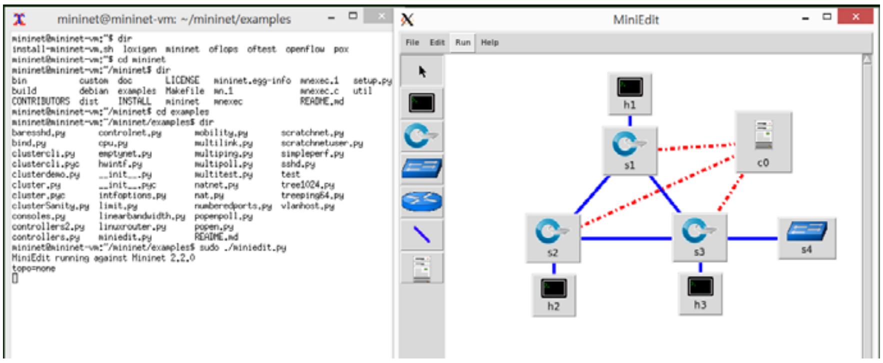
- Create a custom network topology using MiniEdit



Name: c2
Controller Port: 6633
Controller Type: OpenFlow Reference
Protocol: TCP
Remote/In Band Controller
IP Address: 127.0.0.1

III. Mininet – MiniEdit

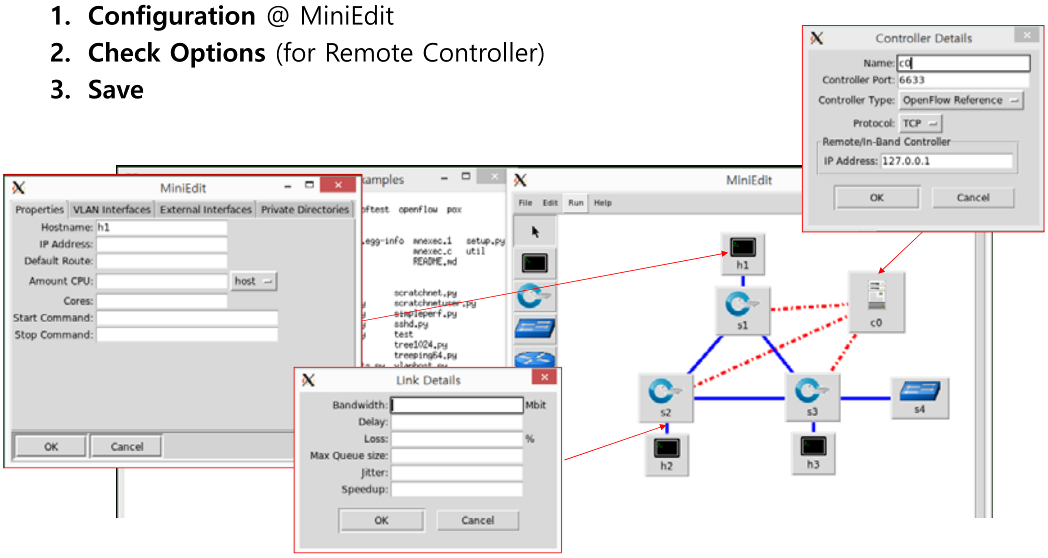
1. **Configuration** @ MiniEdit
2. **Run the Configuration** @ MiniEdit
3. **Check w/xterm**



```
mininet@mininet-vm:~$ dir
install-mininet-vm.sh  loxigen  mininet  oflops  oftest  openflow  pox
mininet@mininet-vm:~$ cd mininet
mininet@mininet-vm:~/mininet$ dir
bin           custom  doc       LICENSE   mininet.egg-info  mnexec.1   setup.py
build         debian  examples  Makefile  mn.1              mnexec.c   util
CONTRIBUTORS  dist    INSTALL   mininet   mnexec            README.md
mininet@mininet-vm:~/mininet$ cd examples
mininet@mininet-vm:~/mininet/examples$ dir
baresshd.py        controlnet.py       mobility.py         scratchnet.py
bind.py            cpu.py              multilink.py        scratchnetuser.py
clustercli.py      emptynet.py         multiping.py        simpleperf.py
clustercli.pyc     hwintf.py           multipoll.py        sshd.py
clusterdemo.py     __init__.py         multitest.py        test
cluster.py         __init__.pyc        natnet.py           tree1024.py
cluster.pyc        intfoptions.py      nat.py              treeping64.py
clusterSanity.py   limit.py            numberedports.py    vlanhost.py
consoles.py        linearbandwidth.py  popenpoll.py
controllers2.py    linuxrouter.py      popen.py
controllers.py     miniedit.py         README.md
mininet@mininet-vm:~/mininet/examples$ sudo ./miniedit.py
MiniEdit running against Mininet 2.2.0
topo=none
▯
```

III. Mininet – MiniEdit

1. **Configuration** @ MiniEdit
2. **Check Options** (for Remote Controller)
3. **Save**



Controller Details
Name: c0
Controller Port: 6633
Controller Type: OpenFlow Reference
Protocol: TCP
Remote/In-Band Controller
IP Address: 127.0.0.1

MiniEdit
Properties | VLAN Interfaces | External Interfaces | Private Directories
Hostname: h1
IP Address:
Default Route:
Amount CPU: host
Cores:
Start Command:
Stop Command:

Link Details
Bandwidth: Mbit
Delay:
Loss: %
Max Queue size:
Jitter:
Speedup:

```
oftest  openflow  pox
.egg-info  mnexec.1  setup.py
           mnexec.c  util
           README.md
scratchnet.py
scratchnetuser.py
simpleperf.py
sshd.py
test
tree1024.py
treeping64.py
vlanhost.py
```

III. Mininet – Extreme configuration

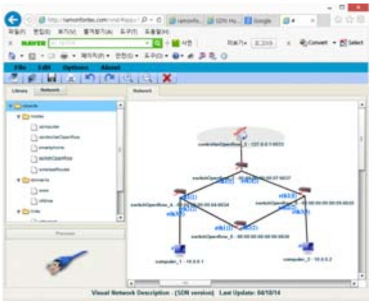
❖ **Mininet with 10X/50X/200X OpenFlow switches**

- **sudo mn --topo=linear,10 --switch=ovsk,protocols=OpenFlow13 --mac**
- **sudo mn --topo=linear,50 --switch=ovsk,protocols=OpenFlow13 –mac**
- **sudo mn --controller=remote,ip=192.168.0.211 --mac --topo=linear,200**
- **sudo mn --topo=linear,200**
- **h1 ping h200**

III. Mininet – VND (선택)

무선(OpenWRT) 테스트시 사용 가능

Visual Network Description



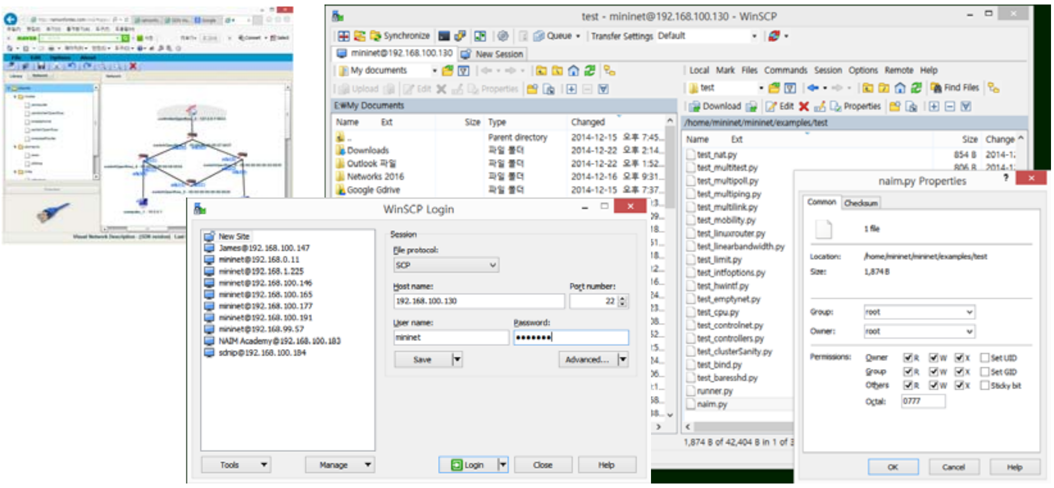
http://ramonfontes.com/vnd/#

```
#!/usr/bin/python

"""
Script created by VND – Visual Network Description (SDN version)
"""
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSKernelSwitch, OVSLegacyKernelSwitch,
UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink

def topology():
    "Create a network."
    net = Mininet( controller=RemoteController, link=TCLink, switch=OVSKernelSwitch )

    print "*** Creating nodes"
    h1 = net.addHost( 'h1', mac='00:00:00:00:00:01', ip='10.0.0.1/8' )
    h2 = net.addHost( 'h2', mac='00:00:00:00:00:02', ip='10.0.0.2/8' )
    c3 = net.addController( 'c3', controller=RemoteController, ip='127.0.0.1', port=6633 )
    s4 = net.addSwitch( 's4', listenPort=6634, mac='00:00:00:00:00:04' )
    s5 = net.addSwitch( 's5', listenPort=6635, mac='00:00:00:00:00:05' )
    s6 = net.addSwitch( 's6', listenPort=6636, mac='00:00:00:00:00:06' )
    s7 = net.addSwitch( 's7', listenPort=6637, mac='00:00:00:00:00:07' )

    print "*** Creating links"
    net.addLink(h2, s5, 0, 3)
    net.addLink(h1, s4, 0, 3)
    net.addLink(s6, s5, 2, 2)
    net.addLink(s6, s4, 1, 2)
    net.addLink(s7, s5, 2, 1)
    net.addLink(s4, s7, 1, 1)

    print "*** Starting network"
    net.start()
    s7.start( [c3] )
    c3.start()

    print "*** Running CLI"
    CLI( net )

    print "*** Stopping network"
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    topology()
```

III. Mininet – VND (선택)

1. **Installation and Running 'WinSCP'**
2. **Check property for file** (saved by VND not by MiniEdit)
3. **'sudo chmod –c 777 naim.py' at PuTTY** (Property for Execution)

http://ramonfontes.com/vnd/#



test - mininet@192.168.100.130 - WinSCP

E:₩My Documents

Name	Ext	Size	Type	Changed
..			Parent directory	2014-12-15 오후 7:45...
Downloads			파일 폴더	2014-12-22 오후 2:14...
Outlook 파일			파일 폴더	2014-12-22 오후 1:52...
Networks 2016			파일 폴더	2014-12-16 오후 9:31...
Google Gdrive			파일 폴더	2014-12-15 오후 7:37...

/home/mininet/mininet/examples/test

test_nat.py
test_multitest.py
test_multipoll.py
test_multiping.py
test_multilink.py
test_mobility.py
test_linuxrouter.py
test_linearbandwidth.py
test_limit.py
test_intfoptions.py
test_hwintf.py
test_emptynet.py
test_cpu.py
test_controlnet.py
test_controllers.py
test_clusterSanity.py
test_bind.py
test_baresshd.py
runner.py
naim.py

1,874 B of 42,404 B in 1 of 3...

WinSCP Login

New Site
James@192.168.100.147
mininet@192.168.0.11
mininet@192.168.1.225
mininet@192.168.100.146
mininet@192.168.100.165
mininet@192.168.100.177
mininet@192.168.100.191
mininet@192.168.99.57
NAIM Academy@192.168.100.183
sdnip@192.168.100.184

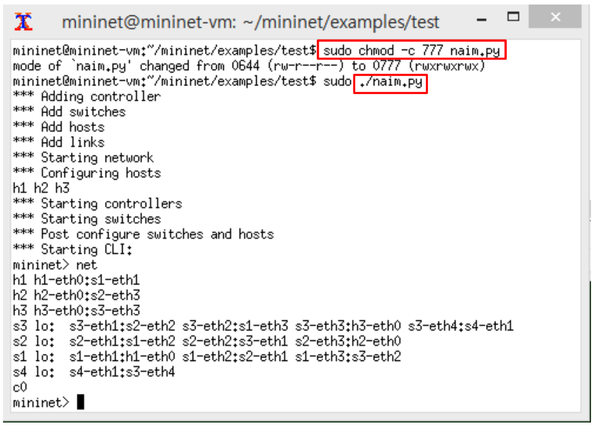
Session
File protocol: SCP
Host name: 192.168.100.130
Port number: 22
User name: mininet
Password: •••••••

naim.py Properties

1 file
Location: /home/mininet/mininet/examples/test
Size: 1,874 B
Group: root
Owner: root
Permissions: Owner ☑R ☑W ☑X □Set UID
Group ☑R ☑W ☑X □Set GID
Others ☑R ☑W ☑X □Sticky bit
Octal: 0777

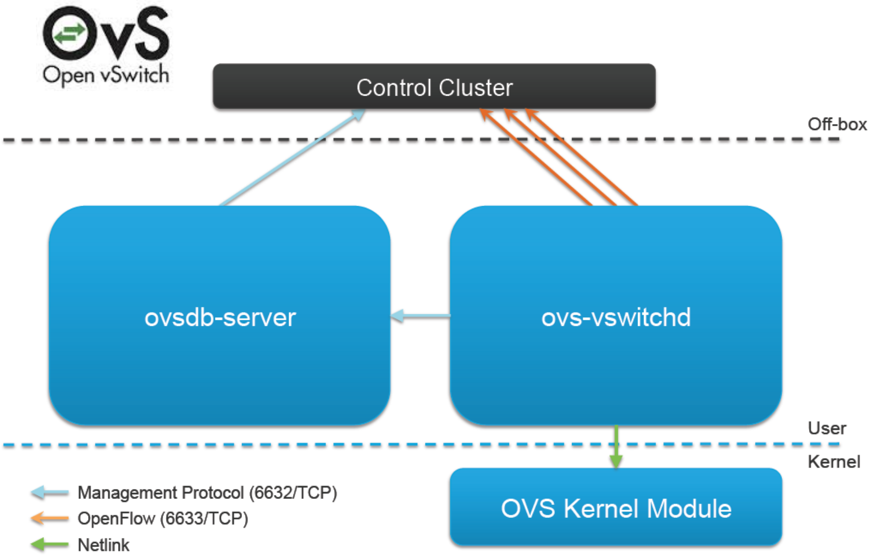
III. Mininet – VND (선택)

1. Installation and Running ‘WinSCP’
2. ‘sudo chmod –c 777 naim.py’ (Property for Execution)
3. **sudo ./naim.py**
4. **Net**
5. **Ping** (h1 ping h2) w/o Application



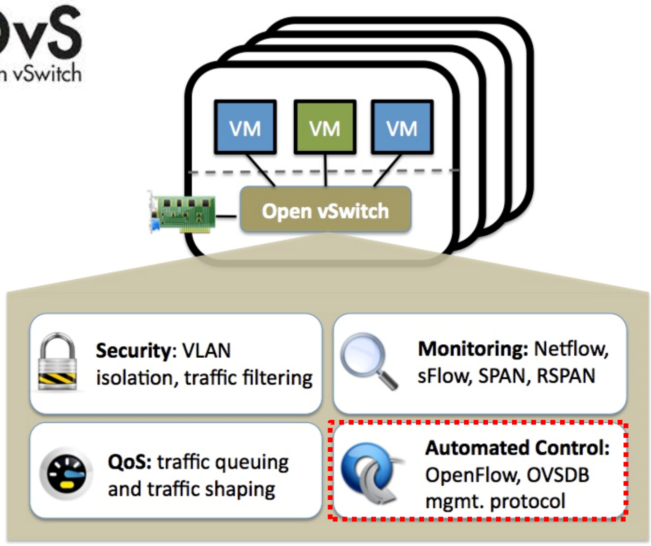
```
mininet@mininet-vm:~/mininet/examples/test$ sudo chmod -c 777 naim.py
mode of `naim.py' changed from 0644 (rw-r--r--) to 0777 (rwxrwxrwx)
mininet@mininet-vm:~/mininet/examples/test$ sudo ./naim.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h1 h2 h3
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth3
h3 h3-eth0:s3-eth3
s3 lo:  s3-eth1:s2-eth2 s3-eth2:s1-eth3 s3-eth3:h3-eth0 s3-eth4:s4-eth1
s2 lo:  s2-eth1:s1-eth2 s2-eth2:s3-eth1 s2-eth3:h2-eth0
s1 lo:  s1-eth1:h1-eth0 s1-eth2:s2-eth1 s1-eth3:s3-eth2
s4 lo:  s4-eth1:s3-eth4
c0
mininet> 
```

IV. OVS



IV. OVS

OvS Open vSwitch



Security: VLAN isolation, traffic filtering

Monitoring: Netflow, sFlow, SPAN, RSPAN

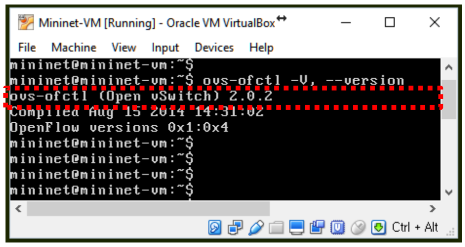
QoS: traffic queuing and traffic shaping

Automated Control: OpenFlow, OVSDB mgmt. protocol

IV. OVS – OVS v2.4.x is available

❖ **Spec.**

- Support for the Rapid Spanning Tree Protocol (IEEE 802.1D-2004).
- Support for multicast snooping (IGMPv1, IGMPv2 and IGMPv3).
- A new "conjunctive match" OpenFlow extension that allows flows to be constructed without introducing a Cartesian Product explosion.
- Add bash command-line completion for most CLI commands.
- Support for transactional flow updates through OpenFlow 1.4 Bundles.
- A number of new features from the latest OpenFlow specifications.
- A simple wrapper script, 'ovs-docker', to integrate OVS with Docker containers.
- Support for STT and basic Geneve tunneling in Linux kernels.
- Support for VXLAN Group Policy extension.
- Support for DPDK Tunneling. VXLAN, GRE, and Geneve are supported protocols.
- Support for DPDK vHost.



```
mininet@mininet-vm:~$
mininet@mininet-vm:~$ ovs-ofctl -V, --version
ovs-ofctl (Open vSwitch) 2.0.2
Compiled Aug 15 2014 14:31:02
OpenFlow versions 0x1:0x4
mininet@mininet-vm:~$
mininet@mininet-vm:~$
mininet@mininet-vm:~$
mininet@mininet-vm:~$
```

Mininet 2.2.1

IV. OVS – OVS v2.4.x is available

❖ **OpenFlow Spec.**

- OpenFlow 1.4 bundles are now supported for flow mods and portmods. For flow mods, both 'atomic' and 'ordered' bundle flags are trivially supported, as all bundled messages are executed in the order they were added and all flow table modifications are now atomic to the datapath. Port mods may not appear in atomic bundles, as port status modifications are not atomic.
- IPv6 flow label and neighbor discovery fields are now modifiable.
- OpenFlow 1.5 extended registers are now supported.
- The OpenFlow 1.5 actset_output field is now supported.
- OpenFlow 1.5 Copy-Field action is now supported.
- OpenFlow 1.5 masked Set-Field action is now supported.
- OpenFlow 1.3+ table features requests are now supported (read-only).
- Nicira extension "move" actions may now be included in action sets.
- "resubmit" actions may now be included in action sets. The resubmit is executed last, and only if the action set has no "output" or "group" action.
- OpenFlow 1.4+ flow "importance" is now maintained in the flow table.
- A new Netronome extension to OpenFlow 1.5+ allows control over the fields hashed for OpenFlow select groups. See "selection_method" and related options in ovs-ofctl(8) for details.

IV. OVS – OVS Operations

❖ **Configuration** @ MiniEdit

① **sudo ovs-vsctl show**
② **sudo ovs-ofctl dump-flows s1**
③ **sudo mn**
④ **sudo ovs-vsctl show**
⑤ **sudo ovs-ofctl dump-flows s1**
⑥ **h1 ping h2**
⑦ **sudo ovs-vsctl show**
⑧ **sudo ovs-ofctl dump-flows s1**

IV. OVS – OVS Operations

❖ Allow All

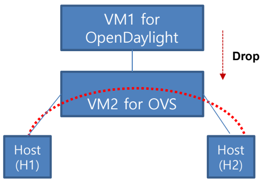
```
① sudo ovs-ofctl add-flow s1 ip,nw_src=200.200.200.11,actions=normal
② sudo ovs-ofctl add-flow s1 ip,nw_dst=200.200.200.12,actions=normal
③ ovs-ofctl dump-flows s1
```

❖ Allow TCP 80

```
① ovs-ofctl add-flow s1
  priority=37000,tcp,nw_src=200.200.200.1,tp_dst=80,actions=normal
② ovs-ofctl add-flow s1
  priority=37000,tcp,nw_dst=200.200.200.1,tp_src=80,actions=normal
③ ovs-ofctl dump-flows s1
```

V. OpenDaylight for Flow

1. Oracle VM VirtualBox Manager'
2. VM Image : ODL for base
3. WinSCP (http://winscp.net/)
4. Xming (http://sourceforge.net/projects/xming/)
5. Putty (http://www.putty.org/)



V. OpenDaylight for Flow

1. Run 'Oracle VM VirtualBox Manager'
2. Import O이 Appliance @ ' Oracle VM VirtualBox Manager '
3. Start VM
4. ID/Password = Mininet/Mininet
5. Ifconfig (for searching IP address)

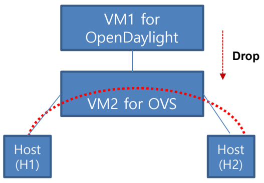
6. $ sudo mn –topo single,3 –mac –controller remote
7. $ sudo ./run.sh @ ODL base

8. http:/x.x.x.x:8080 at Web Browser
9. ID/Password = admin/admin

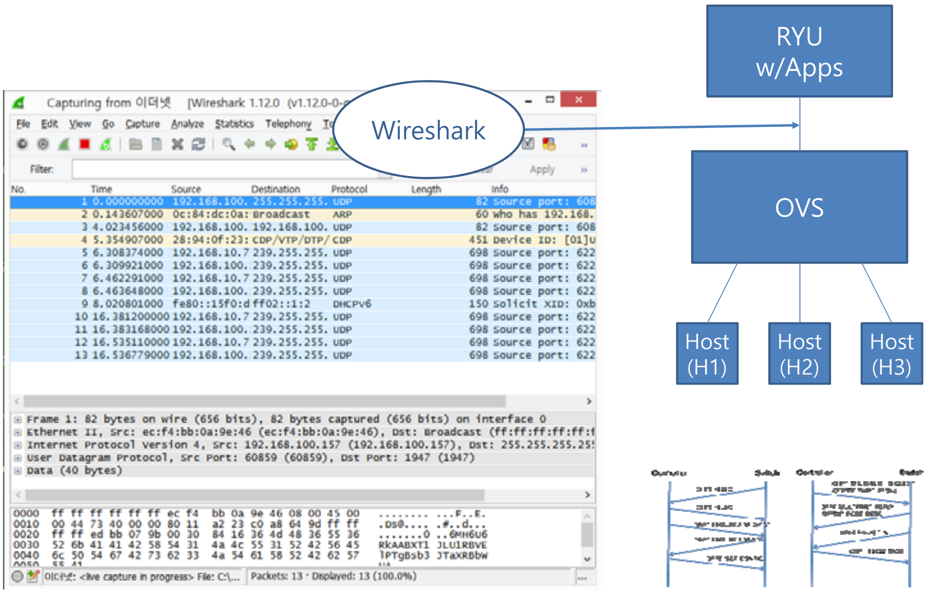
10. 'terminal' @ ODL VM or Mininet VM
11. sudo ./miniedit.py
12. Ping
13. Match Port / Action Drop

14. sudo ovs-vsctl show @ new terminal
15. sudo ovs-ofctl dump-flows s1 @ new terminal
16. OSGi

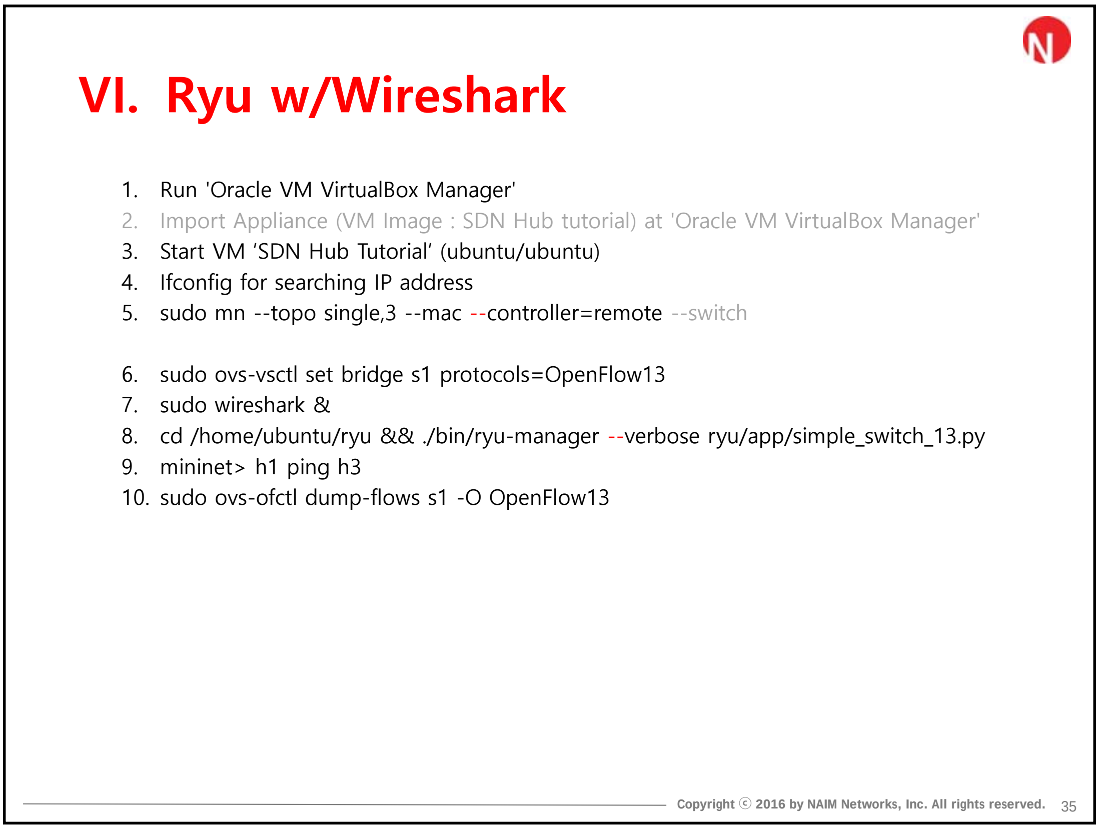
17. Mininet Operations (50X switches, loop topologies)



VI. Ryu w/Wireshark


Capturing from 이더넷 [Wireshark 1.12.0 (v1.12.0-0-
File Edit View Go Capture Analyze Statistics Telephony To
Filter:
Apply
Wireshark
RYU w/Apps
OVS
Host (H1)
Host (H2)
Host (H3)

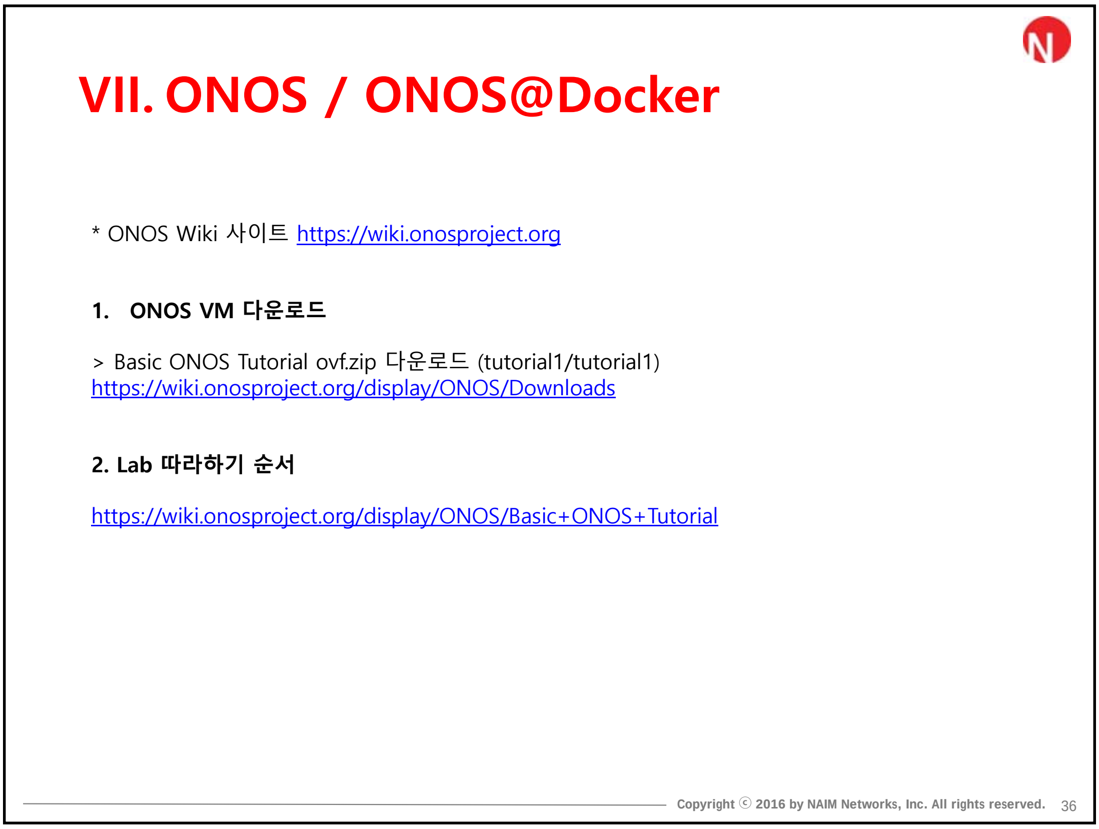
```
No.   Time          Source           Destination      Protocol  Length  Info
    1 0.000000000   192.168.100.     255.255.255.     UDP          82 Source port: 608
    2 0.143607000   0c:84:dc:0a:     Broadcast        ARP          60 Who has 192.168.
    3 4.023456000   192.168.100.     192.168.100.     UDP          82 Source port: 608
    4 5.354907000   28:94:0f:23:     CDP/VTP/DTP/     CDP         451 Device ID: [01]U
    5 6.308374000   192.168.10.7     239.255.255.     UDP         698 Source port: 622
    6 6.309921000   192.168.100.     239.255.255.     UDP         698 Source port: 622
    7 6.462291000   192.168.10.7     239.255.255.     UDP         698 Source port: 622
    8 6.463648000   192.168.100.     239.255.255.     UDP         698 Source port: 622
    9 8.020801000   fe80::15f0:d     ff02::1:2        DHCPv6      150 Solicit XID: 0xb
   10 16.381200000  192.168.10.7     239.255.255.     UDP         698 Source port: 622
   11 16.383168000  192.168.100.     239.255.255.     UDP         698 Source port: 622
   12 16.535110000  192.168.10.7     239.255.255.     UDP         698 Source port: 622
   13 16.536779000  192.168.100.     239.255.255.     UDP         698 Source port: 622

⊞ Frame 1: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
⊞ Ethernet II, Src: ec:f4:bb:0a:9e:46 (ec:f4:bb:0a:9e:46), Dst: Broadcast (ff:ff:ff:ff:ff:f
⊞ Internet Protocol Version 4, Src: 192.168.100.157 (192.168.100.157), Dst: 255.255.255.25
⊞ User Datagram Protocol, Src Port: 60859 (60859), Dst Port: 1947 (1947)
⊞ Data (40 bytes)

0000  ff ff ff ff ff ff ec f4  bb 0a 9e 46 08 00 45 00   ........ ...F..E.
0010  00 44 73 40 00 00 80 11  a2 23 c0 a8 64 9d ff ff   .Ds@.... .#..d...
0020  ff ff ed bb 07 9b 00 30  84 16 36 4d 48 36 55 36   .......0 ..6MH6U6
0030  52 6b 41 41 42 58 54 31  4a 4c 55 31 52 42 56 45   RkAABXT1 JLU1RBVE
0040  6c 50 54 67 42 73 62 33  4a 54 61 58 52 42 62 57   lPTgBsb3 JTaXRBbW
0050  55 41                                              UA

이더넷: <live capture in progress> File: C:\...   Packets: 13 · Displayed: 13 (100.0%)
```



VI. Ryu w/Wireshark

1. Run 'Oracle VM VirtualBox Manager'
2. Import Appliance (VM Image : SDN Hub tutorial) at 'Oracle VM VirtualBox Manager'
3. Start VM ‘SDN Hub Tutorial’ (ubuntu/ubuntu)
4. Ifconfig for searching IP address
5. sudo mn --topo single,3 --mac --controller=remote --switch

6. sudo ovs-vsctl set bridge s1 protocols=OpenFlow13
7. sudo wireshark &
8. cd /home/ubuntu/ryu && ./bin/ryu-manager --verbose ryu/app/simple_switch_13.py
9. mininet> h1 ping h3
10. sudo ovs-ofctl dump-flows s1 -O OpenFlow13



VII. ONOS / ONOS@Docker

* ONOS Wiki 사이트 https://wiki.onosproject.org

1. ONOS VM 다운로드

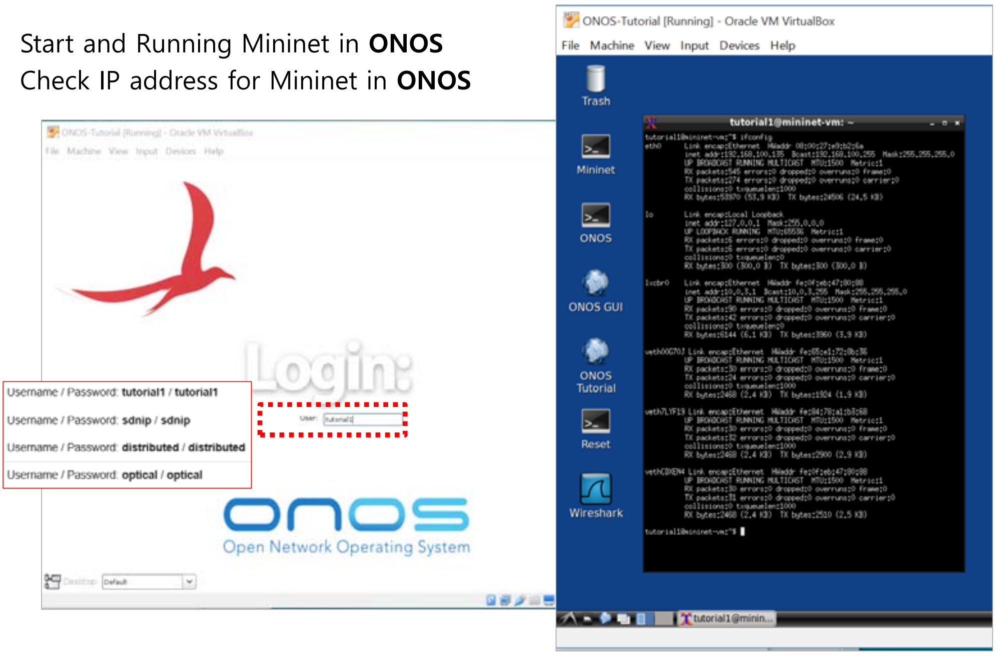
> Basic ONOS Tutorial ovf.zip 다운로드 (tutorial1/tutorial1)
https://wiki.onosproject.org/display/ONOS/Downloads

2. Lab 따라하기 순서

https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial

VII. ONOS / ONOS@Docker

1. Start and Running Mininet in **ONOS**
2. Check IP address for Mininet in **ONOS**



Username / Password: **tutorial1 / tutorial1**

Username / Password: **sdnip / sdnip**

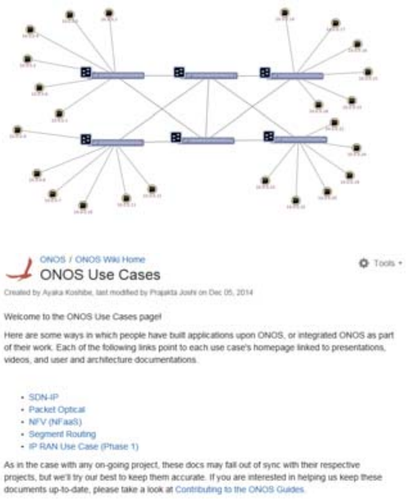
Username / Password: **distributed / distributed**

Username / Password: **optical / optical**

```
tutorial1@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e9:b2:6a
          inet addr:192.168.100.135  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:545 errors:0 dropped:0 overruns:0 frame:0
          TX packets:274 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53970 (53.9 KB)  TX bytes:24506 (24.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:300 (300.0 B)  TX bytes:300 (300.0 B)

lxcbr0    Link encap:Ethernet  HWaddr fe:0f:eb:47:80:88
          inet addr:10.0.3.1  Bcast:10.0.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:90 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6144 (6.1 KB)  TX bytes:3960 (3.9 KB)

veth0OG70J Link encap:Ethernet  HWaddr fe:05:e1:72:0b:36
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2468 (2.4 KB)  TX bytes:1924 (1.9 KB)

veth7LYF19 Link encap:Ethernet  HWaddr fe:84:78:a1:b3:68
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2468 (2.4 KB)  TX bytes:2900 (2.9 KB)

vethCDXEN4 Link encap:Ethernet  HWaddr fe:0f:eb:47:80:88
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2468 (2.4 KB)  TX bytes:2510 (2.5 KB)

tutorial1@mininet-vm:~$ 
```

VII. ONOS / ONOS@Docker

1. Start ONOS-Tutorial at VirtualBox
2. USERNAME: tutorial1
3. PASSWORD: tutorial1



ONOS / ONOS Wiki Home

ONOS Use Cases

Tools

Created by Ayaka Koshibe, last modified by Prajakta Joshi on Dec 05, 2014

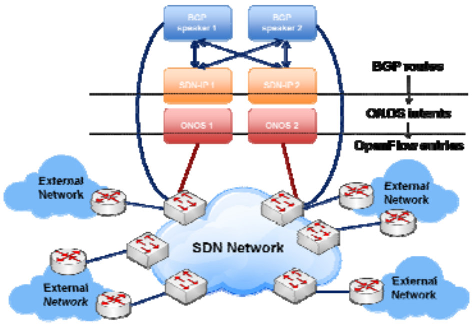
Welcome to the ONOS Use Cases page!

Here are some ways in which people have built applications upon ONOS, or integrated ONOS as part of their work. Each of the following links point to each use case's homepage linked to presentations, videos, and user and architecture documentations.

- SDN-IP
- Packet Optical
- NFV (NFaaS)
- Segment Routing
- IP RAN Use Case (Phase 1)

As in the case with any on-going project, these docs may fall out of sync with their respective projects, but we'll try our best to keep them accurate. If you are interested in helping us keep these documents up-to-date, please take a look at Contributing to the ONOS Guides.

1. Start ONOS-Tutorial at VirtualBox
2. USERNAME: sdnip
3. PASSWORD: sdnip



- Packet-Optical Tutorial
  - Username: distributed
  - Password: distributed

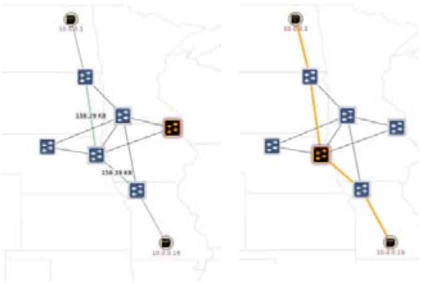
VII. ONOS / ONOS@Docker

1. Start ONOS-Tutorial at VirtualBox
2. User Name/Password: tutorial1/tutorial1
3. Resetting
4. Start Mininet
5. mininet> h11 ping h41 (Why)

6. onos> apps –a -s
7. onos> app activate org.onosproject.fwd
8. mininet> h11 ping h41 (Why)
9. onos> list
10. onos> stop org.onosproject.fwd

11. mininet> h11 ping h41 (Why)
12. onos> list
13. onos> start org.onosproject.fwd
14. mininet> h11 ping h41 (Why)
15. onos> list

VII. ONOS / ONOS@Docker



1. onos> help onos
2. onos> devices
3. onos> links
4. onos> hosts
5. mininet> h11 ping h41

6. onos> flows
7. onos> apps
8. onos> paths <TAB>
9. onos> intents
10. onos> app deactivate org.onosproject.fwd

11. onos> app activate org.onosproject.ifwd
12. onos> feature:install onos-gui
13. onos> add-host-intent 00:00:00:00:00:01/-1 00:00:00:00:00:13/-1
14. ‘A’ for Monitor all traffic
15. ‘V’ for Show all related intents

VII. ONOS / ONOS@Docker (선택)

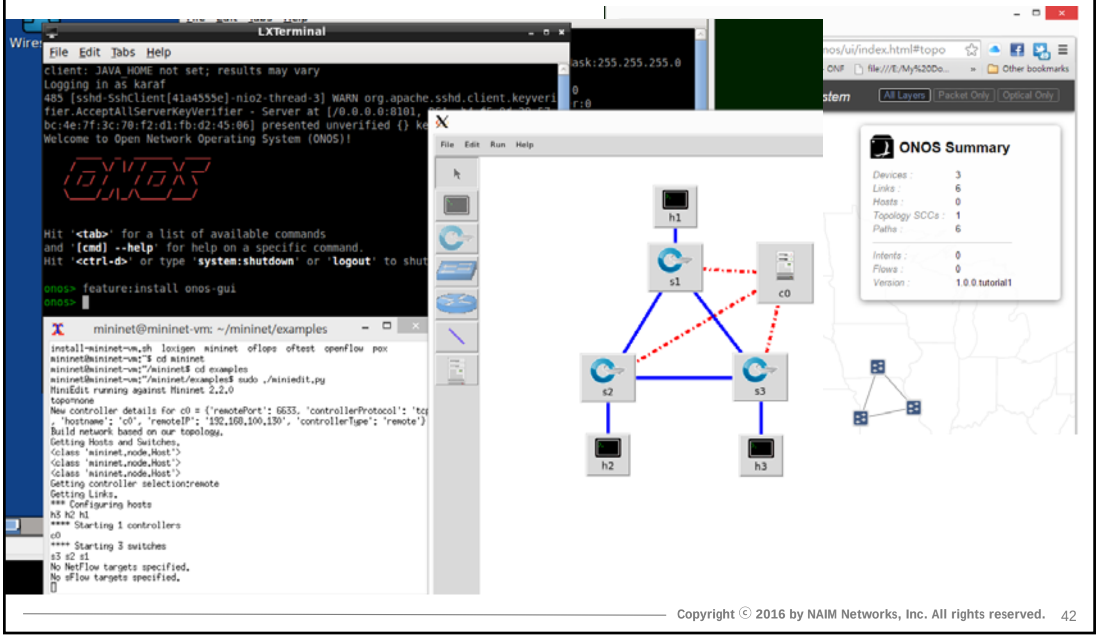
1. **Start ‘ONOS-Tutorial VM’** at VirtualBox
2. **User Name/Password: tutorial1/tutorial1**
3. **Resetting** (‘Running w/App’ and ‘sudo mn –c’ at ONOS Terminal)
4. **Start ‘Mininet VM’** at VirtualBox
5. **User Name/Password: mininet/mininet**

6. **Check IP Address** (ifconfig) : mininet / onos
7. **Running Xming (Double Click ‘Xming’ Icon)**
8. **Running PuTTY w/X11**
9. **‘xtrem’ at Mininet** (PuTTY w/X11)
10. **‘sudo ./miniedit.py’** (at mininet@mininet-vm:~mininet/examples$)

11. **Running ONOS Controller**
12. **Running External Web Browser** (http://192.168.100.***:8181/onos/ui)
13. **Update Controller IP Address at MiniEdit** (as Remote Controller)
14. **onos> feature:install onos-gui**
15. **Run MiniEdit Configuration**

VII. ONOS / ONOS@Docker

1. **Operations at MiniEdit** (Ping/LinkUp/Down/Apps)



```
client: JAVA_HOME not set; results may vary
Logging in as karaf
485 [sshd-SshClient[41a4555e]-nio2-thread-3] WARN org.apache.sshd.client.keyveri
fier.AcceptAllServerKeyVerifier - Server at [/0.0.0.0:8101,
bc:4e:7f:3c:70:f2:d1:fb:d2:45:06] presented unverified {} ke
Welcome to Open Network Operating System (ONOS)!

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shut

onos> feature:install onos-gui
onos>
```

```
install-mininet-vm.sh  loxigen  mininet  oflops  oftest  openflow  pox
mininet@mininet-vm:~$ cd mininet
mininet@mininet-vm:~/mininet$ cd examples
mininet@mininet-vm:~/mininet/examples$ sudo ./miniedit.py
MiniEdit running against Mininet 2.2.0
topo=none
New controller details for c0 = {'remotePort': 6633, 'controllerProtocol': 'tc
, 'hostname': 'c0', 'remoteIP': '192.168.100.130', 'controllerType': 'remote'}
Build network based on our topology.
Getting Hosts and Switches.
<class 'mininet.node.Host'>
<class 'mininet.node.Host'>
<class 'mininet.node.Host'>
Getting controller selection:remote
Getting Links.
*** Configuring hosts
h3 h2 h1
**** Starting 1 controllers
c0
**** Starting 3 switches
s3 s2 s1
No NetFlow targets specified.
No sFlow targets specified.
```

```
ask:255.255.255.0
0
r:0
```

ONOS Summary

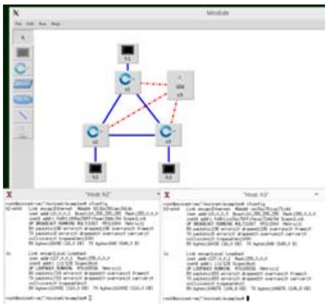
```
Devices :          3
Links :            6
Hosts :            0
Topology SCCs :    1
Paths :            6

Intents :          0
Flows :            0
Version :          1.0.0.tutorial1
```

VII. ONOS / ONOS@Docker

1. **H1/H2/H3 Terminals** (Right Click on each host H1/H2/H3)
2. **'ifconfig' for each terminal** (Right Click on each host H1/H2/H3)
3. **Ping between hosts at terminals**
4. **onos>app activate org.onosproject.fwd**

5. **Link Down/Up**

6. **onos> stop org.onosproject.fwd**
7. **onos> list**
8. **onos> start org.onosproject.fwd**
9. **onos> app activate org.onosproject.ifwd** 😉
10. **Link Down/Up**



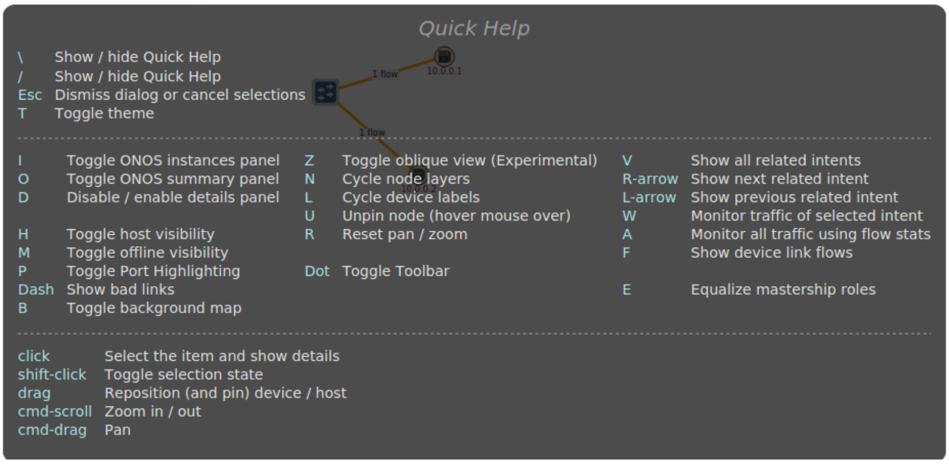
VII. ONOS / ONOS@Docker

1. **help onos**
2. **onos> list**
3. **onos> devices**
4. **onos> links**
5. **onos> hosts**

6. **onos> flows**
7. **onos> apps**
8. **onos> paths <TAB>**
9. **onos> paths of:000000000000000b of:000000000000000e**
10. **onos> intents**
    - **SUBMITTED** - The intent has been submitted and will be processed soon.
    - **COMPILING** - The intent is being compiled. This is a transient state.
    - **INSTALLING** - The intent is in the process of being installed.
    - **INSTALLED** - The intent has been installed.
    - **RECOMPILING** - The intent is being recompiled after a failure.
    - **WITHDRAWING** - The intent is being withdrawn.
    - **WITHDRAWN** - The intent has been removed.
    - **FAILED** - The intent is in a failed state because it cannot be satisfied.

VII. ONOS / ONOS@Docker

1. onos> remove-intent <TAB>
2. onos> add-host-intent 00:00:00:00:00:01/-1 00:00:00:00:00:13/-1



Quick Help

\	Show / hide Quick Help
/	Show / hide Quick Help
Esc	Dismiss dialog or cancel selections
T	Toggle theme

I	Toggle ONOS instances panel
O	Toggle ONOS summary panel
D	Disable / enable details panel
H	Toggle host visibility
M	Toggle offline visibility
P	Toggle Port Highlighting
Dash	Show bad links
B	Toggle background map

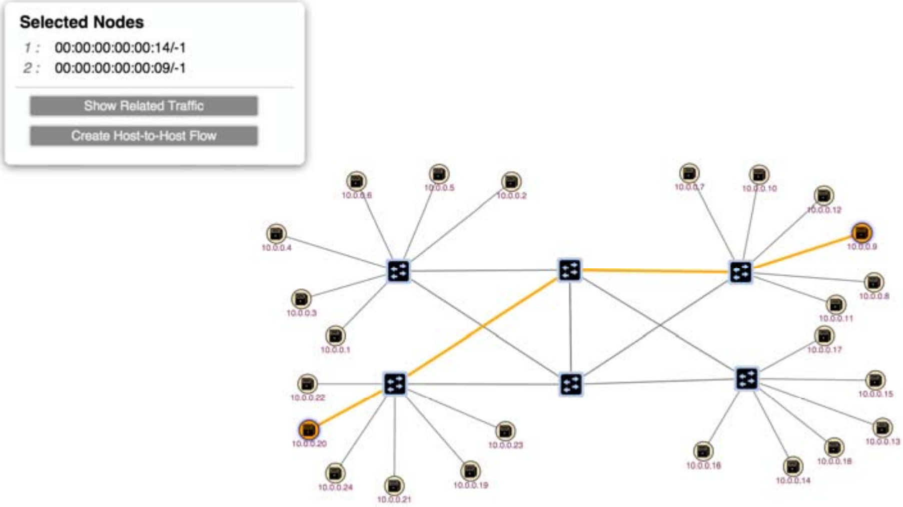
Z	Toggle oblique view (Experimental)
N	Cycle node layers
L	Cycle device labels
U	Unpin node (hover mouse over)
R	Reset pan / zoom
Dot	Toggle Toolbar

V	Show all related intents
R-arrow	Show next related intent
L-arrow	Show previous related intent
W	Monitor traffic of selected intent
A	Monitor all traffic using flow stats
F	Show device link flows
E	Equalize mastership roles

click	Select the item and show details
shift-click	Toggle selection state
drag	Reposition (and pin) device / host
cmd-scroll	Zoom in / out
cmd-drag	Pan

VII. ONOS / ONOS@Docker

1. onos> 'Create Host-to-host Flow'
2. onos> add-host-intent 00:00:00:00:00:01/-1 00:00:00:00:00:13/-1



Selected Nodes

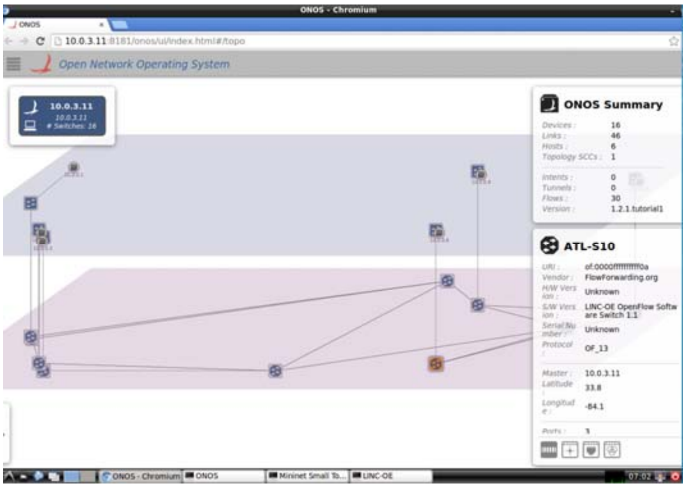
1 : 00:00:00:00:00:14/-1
2 : 00:00:00:00:00:09/-1

VII. ONOS / ONOS@Docker

1. sudo ovs-vsctl set bridge s1 protocols=OpenFlow13
2. sudo wireshark &
3. mininet> h1 ping h3
4. sudo ovs-ofctl dump-flows s1 -O OpenFlow13

VII. ONOS / ONOS@Docker

1. **Running 'Oracle VM VirtualBox Manager'**
2. **Import Appliance** (onos-tutorial-1.0.0.r161 VM or '1.2.1r2')
3. **USERNAME: optical**
4. **PASSWORD: optical**



ONOS Summary

Devices :	16
Links :	46
Hosts :	6
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	30
Version :	1.2.1.tutorial1

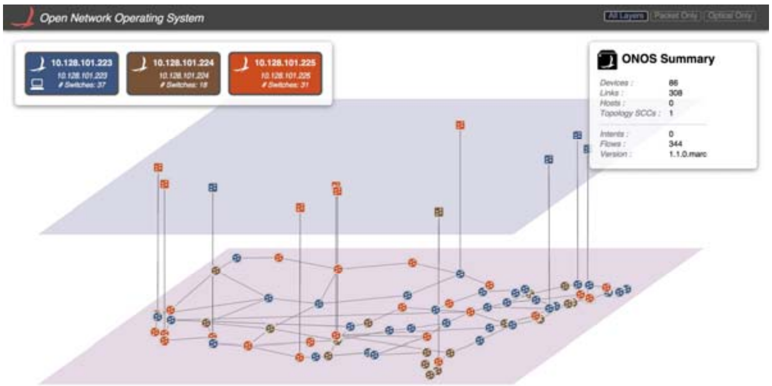
ATL-S10

URI :	of:0000ffffffffff0a
Vendor :	FlowForwarding.org
H/W Version :	Unknown
S/W Version :	LINC-OE OpenFlow Software Switch 1.1
Serial Number :	Unknown
Protocol :	OF_13
Master :	10.0.3.11
Latitude :	33.8
Longitude :	-84.1
Ports :	3

VII. ONOS / ONOS@Docker

1. Running 'Oracle VM VirtualBox Manager'
2. Import Appliance onos-tutorial-1.2.1r2 or 1.0.0.r161 VM (선택)
   or Check Network for 'onos-tutorial' VM
3. Start ONOS-Tutorial at VirtualBox
4. USERNAME: tutorial1
5. PASSWORD: tutorial1

- Building a distributed application
  - Username: distributed
  - Password: distributed



10.128.101.223 — 10.128.101.223 — # Switches: 37

10.128.101.224 — 10.128.101.224 — # Switches: 18

10.128.101.225 — 10.128.101.225 — # Switches: 31

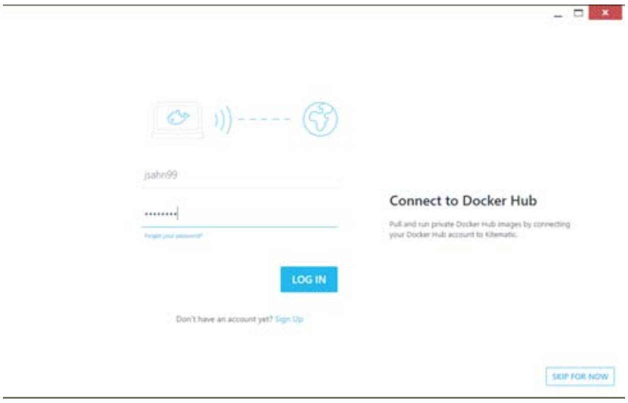
ONOS Summary

Devices : 86
Links : 308
Hosts : 0
Topology SCCs : 1
Intents : 0
Flows : 344
Version : 1.1.0.marc

VII. ONOS / ONOS@Docker

❖ Docker Tool 설치

- VirtualBox의 가상 아답터 동일한것 사용
- 리셋: Default VM 삭제 or VirtualBox 삭제후 Docker Tool로 재설치
- Host Meta App, OpenFlow App, Forwarding App 실행
- 대소문자 확인



jsahn99

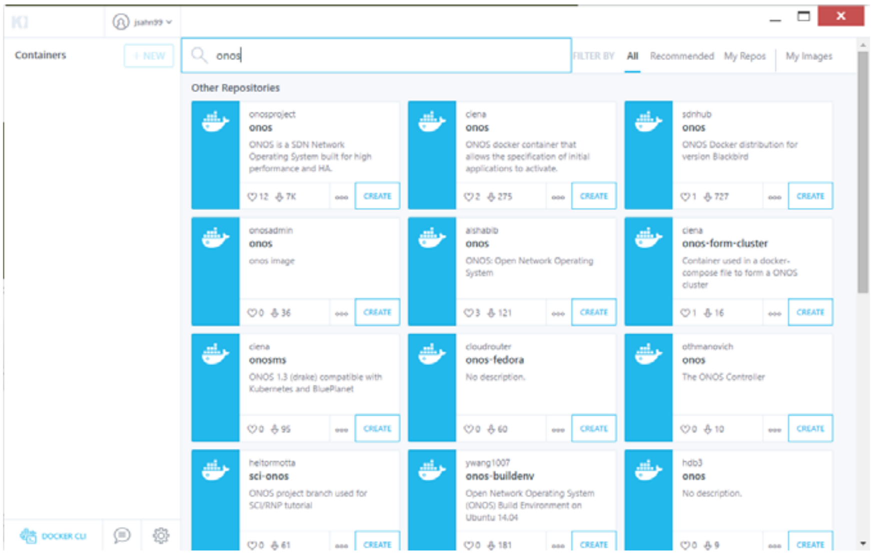
••••••••

Connect to Docker Hub

Pull and run private Docker Hub images by connecting your Docker Hub account to Kitematic.

Don't have an account yet? Sign Up

VII. ONOS / ONOS@Docker



Other Repositories

onosproject
onos
ONOS is a SDN Network Operating System built for high performance and HA.
♡12 ⤓7K CREATE

ciena
onos
ONOS docker container that allows the specification of initial applications to activate.
♡2 ⤓275 CREATE

sdnhub
onos
ONOS Docker distribution for version Blackbird
♡1 ⤓727 CREATE

onosadmin
onos
onos image
♡0 ⤓36 CREATE

alshabib
onos
ONOS: Open Network Operating System
♡3 ⤓121 CREATE

ciena
onos-form-cluster
Container used in a docker-compose file to form a ONOS cluster
♡1 ⤓16 CREATE

ciena
onosms
ONOS 1.3 (drake) compatible with Kubernetes and BluePlanet
♡0 ⤓95 CREATE

cloudrouter
onos-fedora
No description.
♡0 ⤓60 CREATE

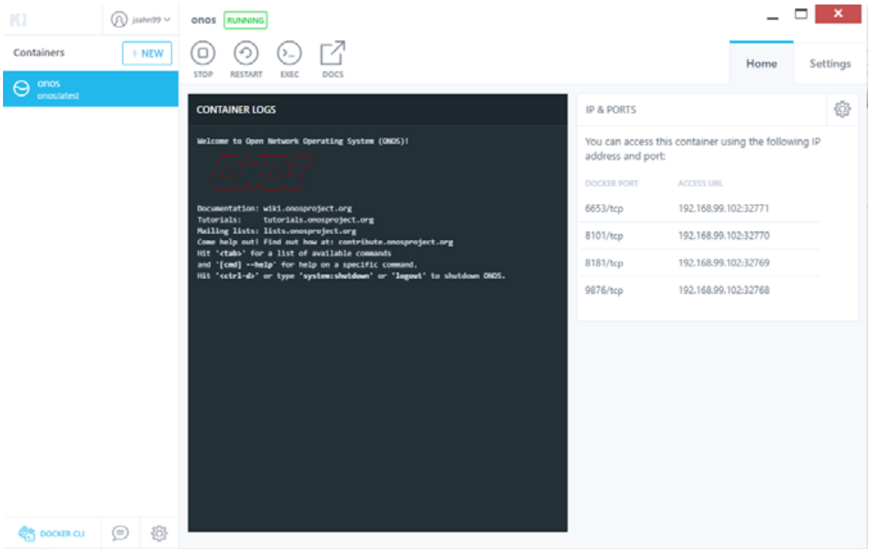
othmanovich
onos
The ONOS Controller
♡0 ⤓10 CREATE

heitormotta
sci-onos
ONOS project branch used for SCI/RNP tutorial
♡0 ⤓61 CREATE

ywang1007
onos-buildenv
Open Network Operating System (ONOS) Build Environment on Ubuntu 14.04
♡0 ⤓181 CREATE

hdb3
onos
No description.
♡0 ⤓9 CREATE

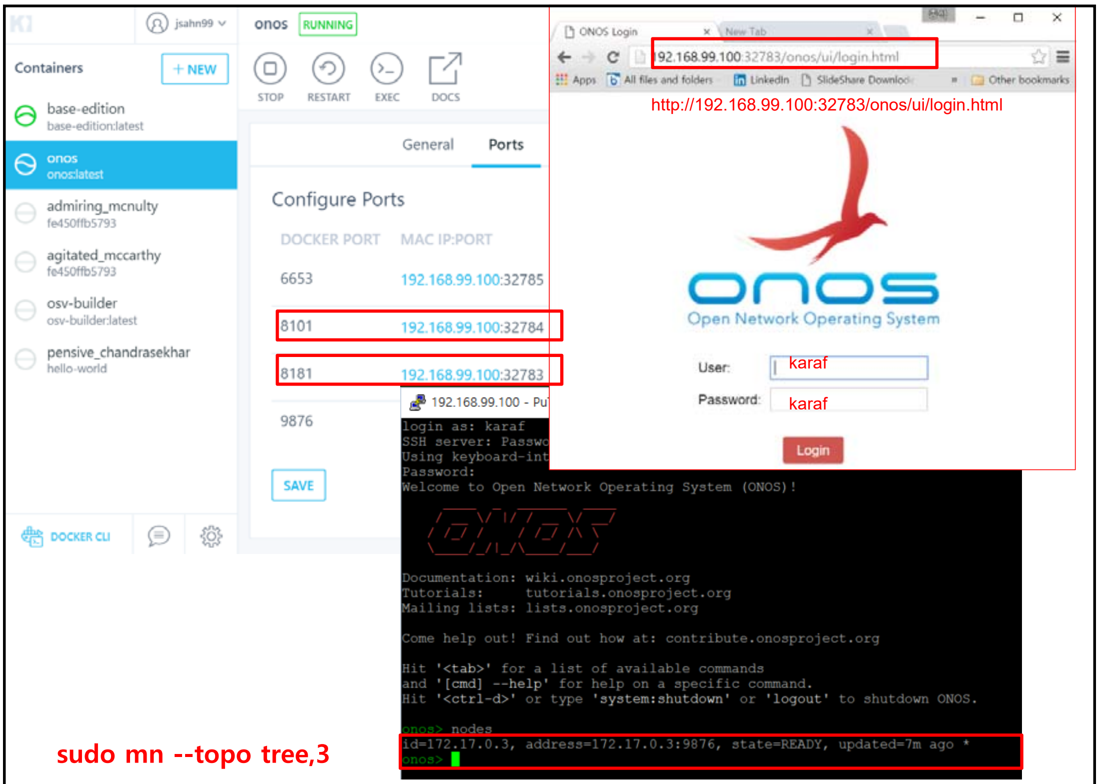
VII. ONOS / ONOS@Docker



```
Welcome to Open Network Operating System (ONOS)!

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org
Come help out! Find out how at: contribute.onosproject.org
Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.
```

You can access this container using the following IP address and port:

DOCKER PORT	ACCESS URL
6653/tcp	192.168.99.102:32771
8101/tcp	192.168.99.102:32770
8181/tcp	192.168.99.102:32769
9876/tcp	192.168.99.102:32768



Configure Ports

DOCKER PORT	MAC IP:PORT
6653	192.168.99.100:32785
8101	192.168.99.100:32784
8181	192.168.99.100:32783
9876	

192.168.99.100:32783/onos/ui/login.html

http://192.168.99.100:32783/onos/ui/login.html

User: karaf

Password: karaf

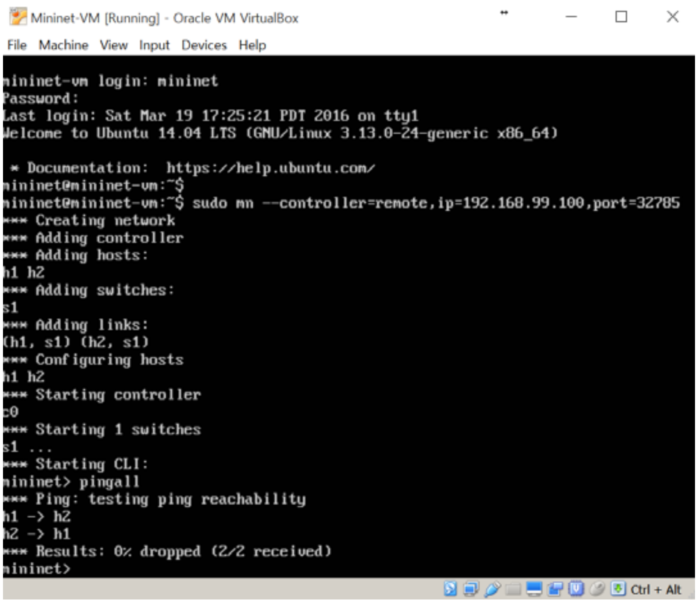
192.168.99.100 - Pu

```
login as: karaf
SSH server: Passwo
Using keyboard-int
Password:
Welcome to Open Network Operating System (ONOS)!

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos> nodes
id=172.17.0.3, address=172.17.0.3:9876, state=READY, updated=7m ago *
onos>
```

sudo mn --topo tree,3

$ sudo mn –controller=remote, ip=192.168.99.100, port=32785

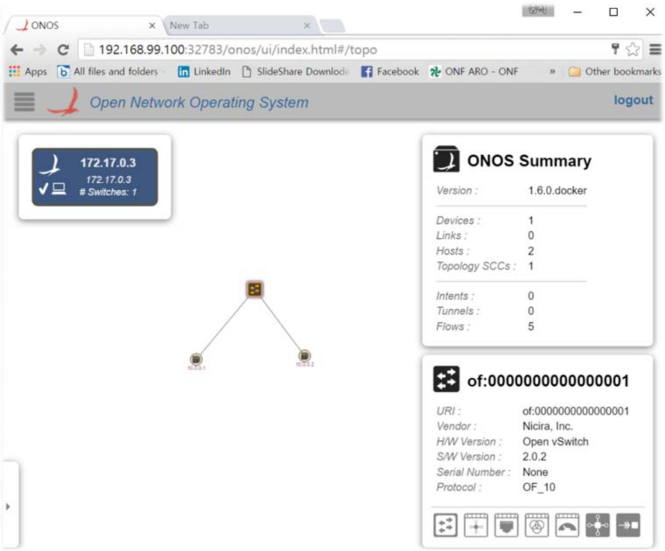
VII. ONOS / ONOS@Docker



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

```
mininet-vm login: mininet
Password:
Last login: Sat Mar 19 17:25:21 PDT 2016 on tty1
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=192.168.99.100,port=32785
*** 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> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

VII. ONOS / ONOS@Docker



ONOS Summary

Version : 1.6.0.docker

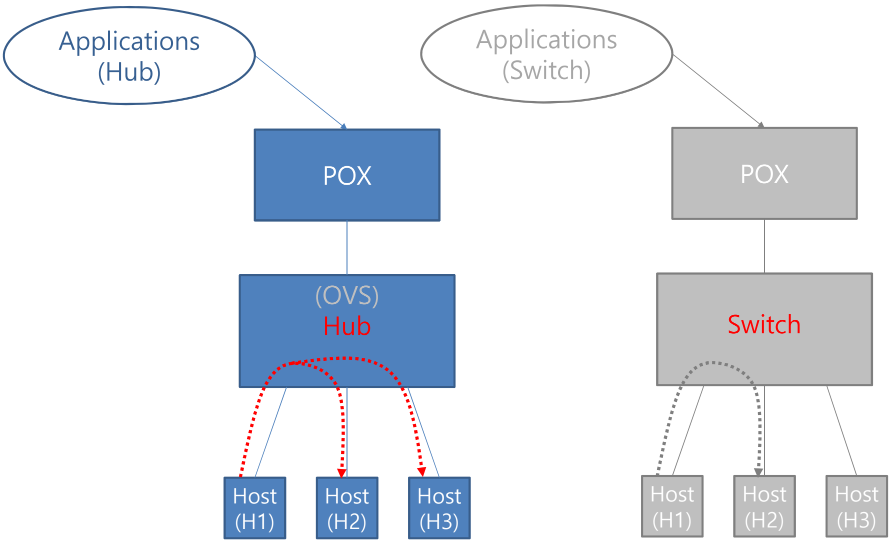
Devices : 1
Links : 0
Hosts : 2
Topology SCCs : 1

Intents : 0
Tunnels : 0
Flows : 5

of:0000000000000001

URI : of:0000000000000001
Vendor : Nicira, Inc.
H/W Version : Open vSwitch
S/W Version : 2.0.2
Serial Number : None
Protocol : OF_10

VIII.SDN Application (선택 w/POX)



VIII.SDN Application (선택 w/POX)

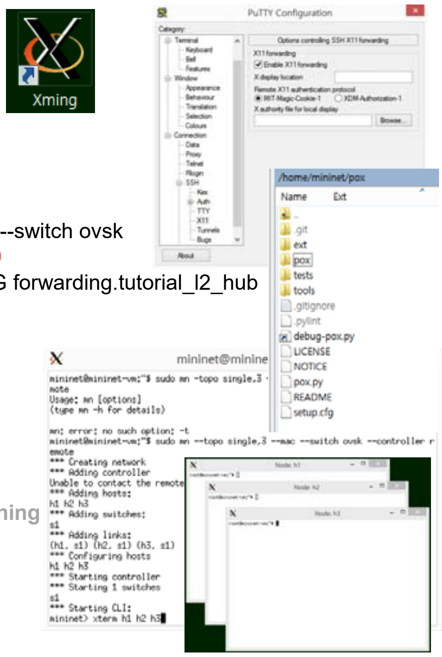
1. Start ‘SDN Hub tutorial’ at VirtualBox
2. User ID/Password : ubuntu/ubuntu
3. Ifconfig
4. Run 'Xming' at Laptop
5. putty.exe using Enable X11 forwarding

6. User ID/Password : ubuntu/ubuntu
7. xterm
8. $ sudo mn --topo single,3 --mac --controller remote --switch ovsk
9. sudo ovs-vsctl set bridge s1 protocols=OpenFlow10
10. cd /home/ubuntu/pox && ./pox.py log.level --DEBUG forwarding.tutorial_l2_hub

11. $ ./pox.py log.level --DEBUG forwarding.hub
12. mininet> xterm h1 h2 h3
13. # tcpdump -XX -n -i h2-eth0
14. # tcpdump -XX -n -i h3-eth0
15. # ping -c 1 10.0.0.2

16. # ping -c 1 10.0.0.5
17. $ ./pox.py log.level --DEBUG forwarding.l2_learning
18. mininet> xterm h1 h2 h3
19. # tcpdump -XX -n -i h2-eth0
20. # tcpdump -XX -n -i h3-eth0

21. # ping -c 1 10.0.0.2



Options controlling SSH X11 forwarding
X11 forwarding
☑ Enable X11 forwarding
X display location
Remote X11 authentication protocol
● MIT-Magic-Cookie-1 ○ XDM-Authorization-1
X authority file for local display

```
mininet@mininet-vm:~$ sudo mn -topo single,3 -
mote
Usage: mn [options]
(type mn -h for details)

mn: error: no such option: -t
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller r
emote
*** Creating network
*** Adding controller
Unable to contact the remote
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet> xterm h1 h2 h3
```

IX. OpenVSwitch @ Raspberry PI

0. sudo apt-get update
1. sudo apt-get install openvswitch-switch
2. sudo apt-get install openvswitch-common bridge-utils
3. sudo apt-get install openvswitch-<tab>

openvswitch-common openvswitch-pki openvswitch-vtep
openvswitch-dbg openvswitch-switch
openvswitch-ipsec openvswitch-test

1. sudo ovs-vsctl show (Version Check)
2. sudo ovs-vsctl add-br ovsbr0
3. sudo ovs-vsctl show
4. ifconfig

5. sudo ovs-vsctl add-port ovsbr0 eth0
6. sudo ovs-vsctl add-port ovsbr0 eth1
7. sudo ovs-vsctl show
8. **sudo ovs-ofctl dump-ports ovsbr0**
9. **sudo ovs-ofctl dump-flows ovsbr0**
10. Sudo ovs-vsctl set-controller ovsbr0 tcp:192.168.??.58:6633
11. sudo ovs-vsctl show (Check Controller Configured)

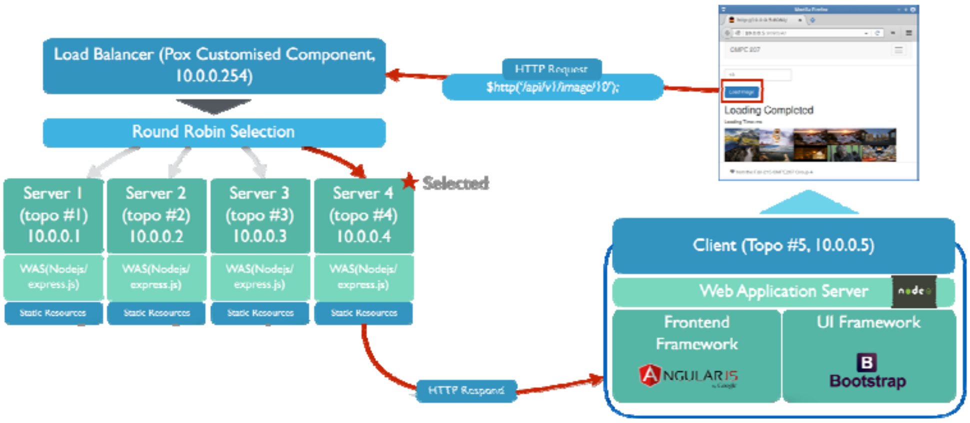
12. sudo ip route add 192.168.??.58/32 dev eth2
13. route –n
14. sudo ovs-vsctl show (Check Controller Connected)

15. iperf –s (@PC)
16. iperf -f m -c 192.168.56.120 (format Mbps) throughput test

www.naimnetworks.com
contacts@naimnetworks.com
+82-2-552-0555
5th Floor, Anam BLDG
616 Yeondongdaero, Kanamgu, Seoul,
Republic of Korea

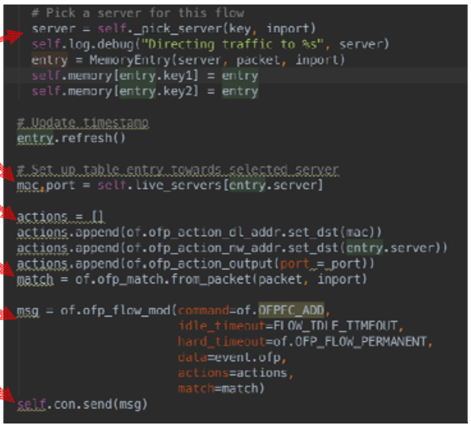
Load Balancer Algorithms

```
sudo mn --arp --topo single,5 --mac --switch ovsk --controller remote -x
```



Load Balancer Algorithms

- Select the server(self._pick_server())
- Set up the table entry towards selected server
- Set up Openflow actions using selected server's ip/mac
- Set up Openflow match based on request packet(packet, inport)
- Set up the open flow's message using customised action/match
- Send message using OpenFlow



```
    # Pick a server for this flow
    server = self._pick_server(key, inport)
    self.log.debug("Directing traffic to %s", server)
    entry = MemoryEntry(server, packet, inport)
    self.memory[entry.key1] = entry
    self.memory[entry.key2] = entry

  # Update timestamp
  entry.refresh()

  # Set up table entry towards selected server
  mac,port = self.live_servers[entry.server]

  actions = []
  actions.append(of.ofp_action_dl_addr.set_dst(mac))
  actions.append(of.ofp_action_nw_addr.set_dst(entry.server))
  actions.append(of.ofp_action_output(port = port))
  match = of.ofp_match.from_packet(packet, inport)

  msg = of.ofp_flow_mod(command=of.OFPFC_ADD,
                        idle_timeout=FLOW_IDLE_TIMEOUT,
                        hard_timeout=of.OFP_FLOW_PERMANENT,
                        data=event.ofp,
                        actions=actions,
                        match=match)
  self.con.send(msg)
```

참고자료

REPRODUCIBLE NETWORK RESEARCH WITH HIGH-FIDELITY EMULATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Brandon Heller

This dissertation is online at: http://purl.stanford.edu/zk853sv3422

ABSTRACT

In an ideal world, all research papers would be runnable: simply click to replicate the results, using the same setup as the authors. In many computational fields, like Machine Learning or Programming Languages, creating a runnable paper means packaging up the code and data in a virtual machine. However, for Network Systems, the path to a realistic, runnable paper is not so clear. This class of experiments requires many servers, network elements, and packets to run in parallel, and their
results depend on accurate timing. Current platform options either provide realism but lack flexibility (e.g., shared testbeds like Emulab [30] cannot support arbitrary topologies) or provide flexibility but lack realism (e.g., discrete-event simulators like ns-2 [57] model end-host code)……………..

Network in a Laptop: Rapid Prototyping for Software-Defined Networks

Bob Lantz
Network Innovations Lab
DOCOMO USA Labs
Palo Alto, CA, USA
rlantz@cs.stanford.edu

Brandon Heller
Dept. of Computer Science,
Stanford University
Stanford, CA, USA
brandonh@stanford.edu

Nick McKeown
Dept. of Electrical Engineering
and Computer Science,
Stanford University
Stanford, CA, USA
nickm@stanford.edu

ABSTRACT

Mininet is a system for rapidly prototyping large networks on the constrained resources of a single laptop. The lightweight approach of using OS-level virtualization features, including processes and network namespaces, allows it to scale to hundreds of nodes. Experiences with our initial implementation suggest that the ability to run, poke, and debug in real time represents a qualitative change in workflow.
We share supporting case studies culled from over 100 users, at 18 institutions, who have developed Software-Defined Networks (SDN). Ultimately, we think the greatest value of Mininet will be supporting collaborative network research, by enabling self-contained SDN prototypes which anyone with a PC can download, run, evaluate, explore, tweak, and build upon.