

SDN 운용

목차

I 프로젝트 소개

1. 프로젝트명
2. 개요
3. 목적
4. 구축방법
5. 측정

II 프로젝트 구축1

1. Pox-dhcp 설정하기
2. Floodlight controller 설정 및 연동하기
3. Mininet 설정 및 구동하기
4. Wireshark로 결과 측정하기

III 프로젝트 구축2

1. Floodlight controller 설정하기
2. Mininet 설정 및 구동하기
3. Wireshark로 결과 측정하기

1. 프로젝트소개

프로젝트명

- Pox-dhcp와 Floodlight를 혼용한 SDN 구성
- Miniedit과 Floodlight를 혼용한 SDN 구성

개요

- SDN 컨트롤러인 Openflow Reference, POX, Floodlight를 Mininet 상에 SDN 기반의 네트워크를 구성하고, L3 스위칭/DHCP/Loadbalancing/Firewall/ACL 시험이 가능하도록 한다.
- SDN 기반의 네트워크 망을 통해서 시험을 시행하며 데이터를 수집하여 openflow 프로토콜 분석을 시행하여 분석보고서를 작성하도록 한다.

목적

- pox-dhcp 동작 확인 및 측정
- floodlight 를 사용한 ACL/Firewall 구현
- Mininet과 floodlight를 연동하여 동작 확인 및 결과 측정

구축방법(실습1)

- 1단계 : pox Controller에서 pox-DHCP 설정 및 실행 (1번 서버)
- 2단계 : floodlight Controller 실행 (2번 서버)
- 3단계 : Miniedit에서 Pox_DHCP_Floodlight.mn 실행
- 4단계 : 각 서버의 IP와 Port 설정 및 DHCP로 IP 할당
- 5단계 : Floodlight Web에서 측정 및 확인
- 6단계 : Wireshark로 측정되는 결과 확인

(두번째 실습 또한 마찬가지로 pox없이 miniedit과 floodlight을 연동해서 진행함)

구축방법(실습2)

- 1단계 : floodlight Controller 실행 (1번 서버)
- 2단계 : Miniedit에서 miniedit.mn 실행
- 3단계 : Controller 서버와 Remote서버의 IP와 Port 설정 및 DHCP로 IP 할당
- 4단계 : Wireshark로 측정되는 결과 확인

측정

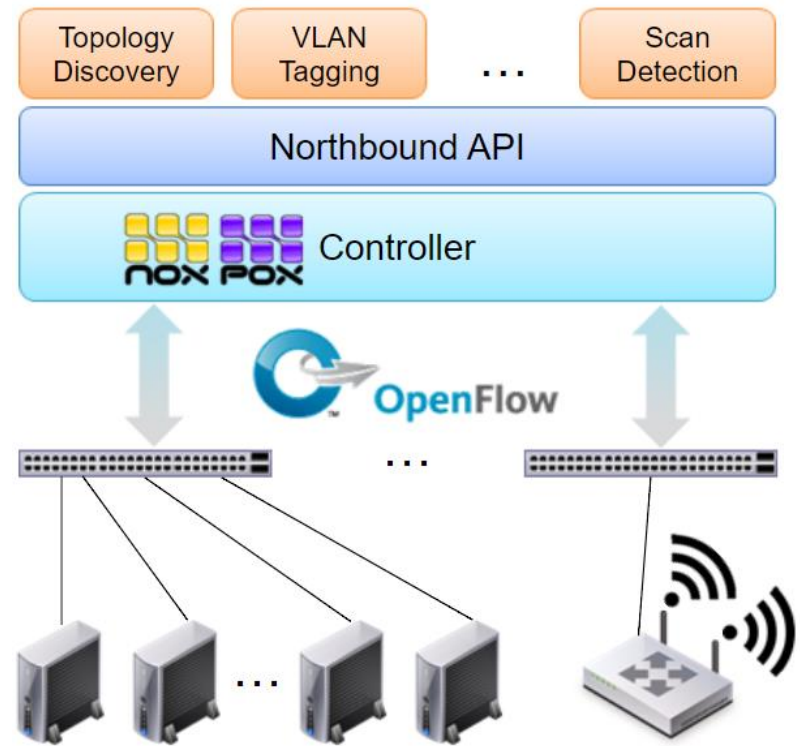
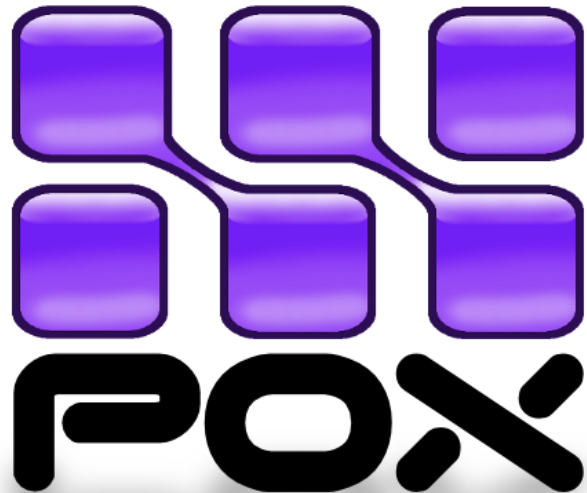
- sdn 스위치와 sdn 컨트롤러 간에 openflow 프로토콜을 수집 및 분석한 결과를 작성
- sdn 스위치와 sdn 컨트롤러 간에 Secu Channel 형성과정 측정 및 분석한 결과를 작성
- sdn 컨트롤러의 dhcp 동작 상태 측정 및 분석한 결과 작성
- sdn 컨트롤러의 flow-entry를 측정하고 분석한 결과를 작성
 - openflow 교재를 참조하여 자세히 기록

1. POX-DHCP 설정

POX Controller로 POX-DHCP 구성

POX Controller 란?

- Python 언어로 구성된 POX Controller는 학교나 연구 기관을 대상으로 하며, 리눅스, MAC, OS, 윈도우에서 설치가 가능한 네트워크 제어 플랫폼이다. 네트워크 관리 및 제어 애플리케이션을 구성할 수 있는 프로그램 인터페이스를 제공하며, 전체 네트워크에 대한 중앙 집중화된 프로그램 모델을 가능하게 한다. OpenFlow v1.3까지 지원하고 있으며 사용방법이 쉬운 것이 장점이다.
- <https://github.com/noxrepo/pox> 에서 다운로드 가능하다.

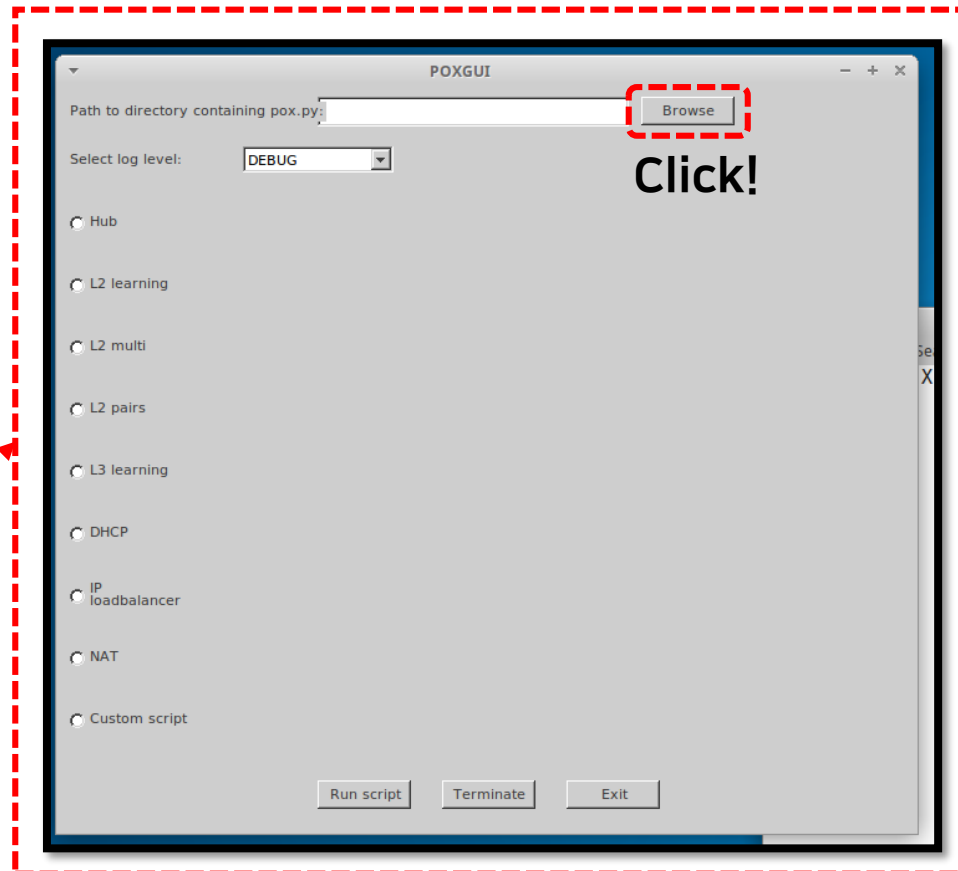
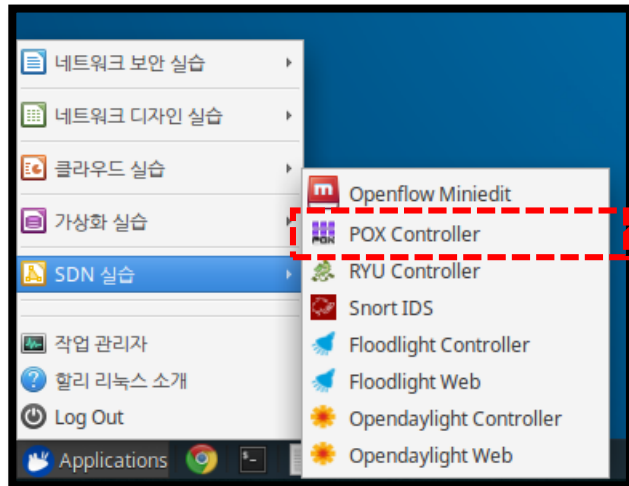


POX-DHCP 설정

1) Pox Controller 열어서 POX-DHCP 설정하기

- 미리 구성한 서버1의 hali-linux 환경에 설치된 POX Controller 열기

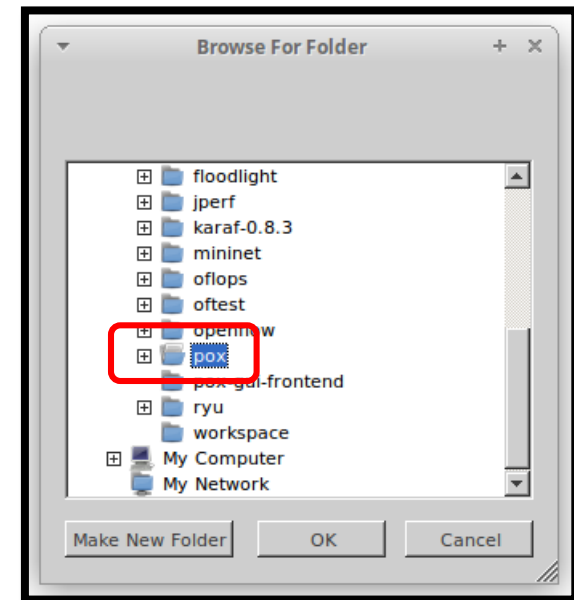
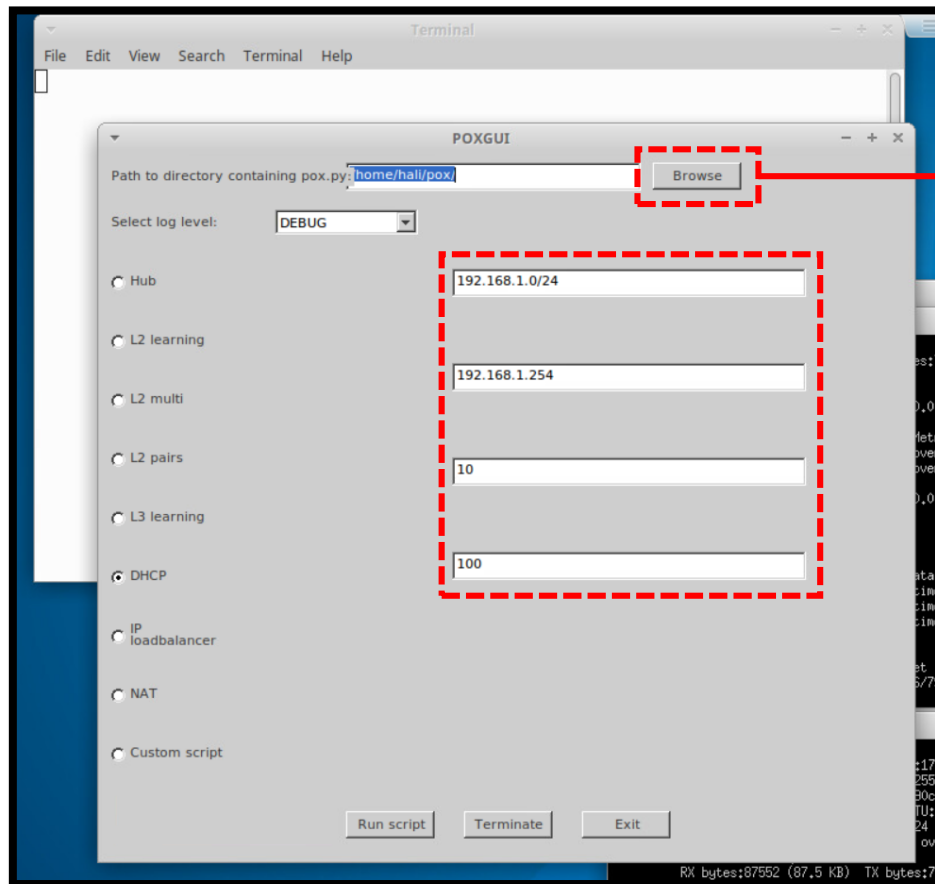
➤ 파일 위치 : Applications > SDN 실습 > POX Controller



POX-DHCP 설정

1) Pox Controller 열어서 POX-DHCP 설정하기

- POX Controller에서 경로 /home/hali/pox로 설정 후에 DHCP를 선택해서 입력하기.
 - 위에서부터 나눌 Subnet 범위/DHCP Server로 쓸 IP, 길이의 시작부터 끝을 가리키고 있음.



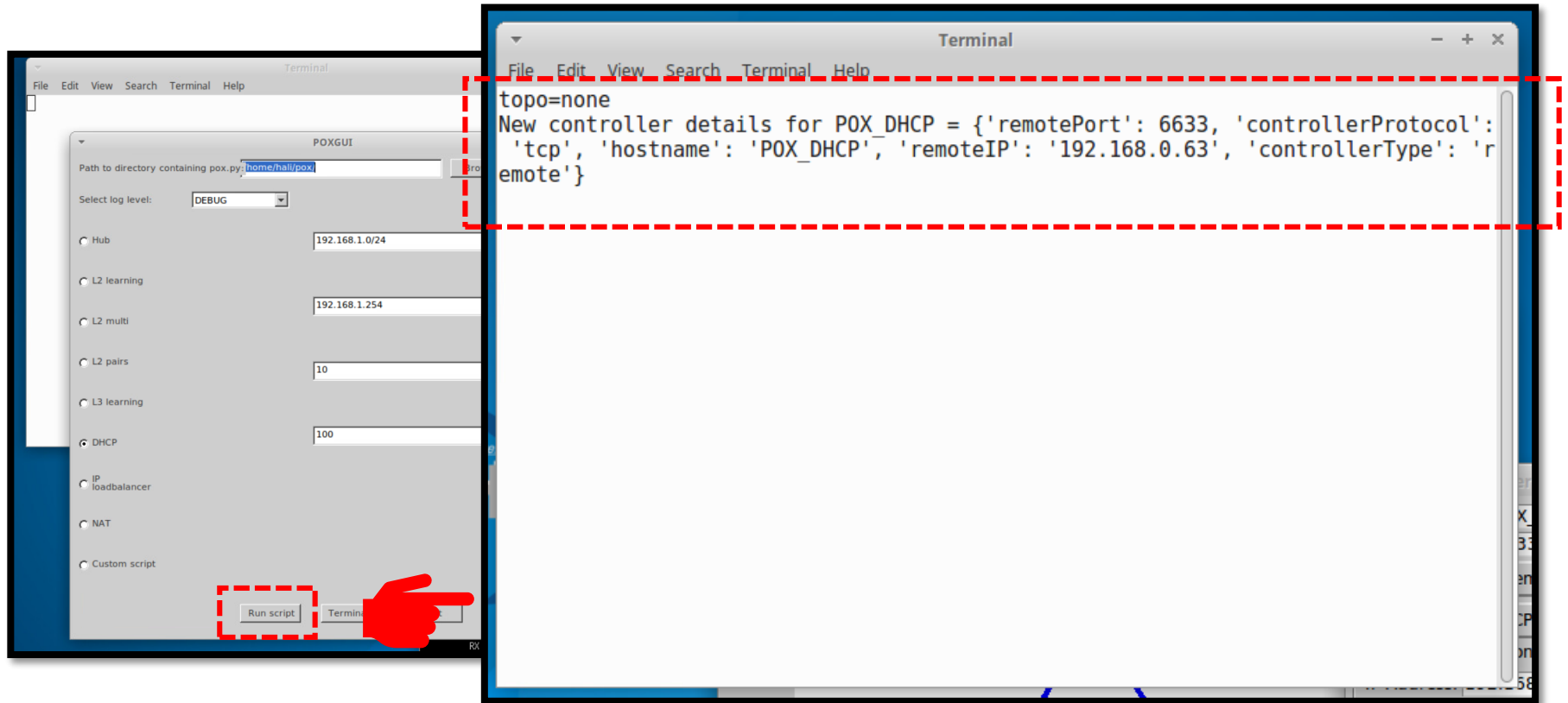
➤ Home/hali/pox

POX-DHCP 설정

2) Pox Controller에서 POX-DHCP 실행하기

- POX Controller에서 run script 눌러 실행하기

- 입력 후에 Run scrip를 누르면 pox가 구동된다.
- 새로운 POX_DHCP 라는 Controller를 인식했음을 알 수 있다.

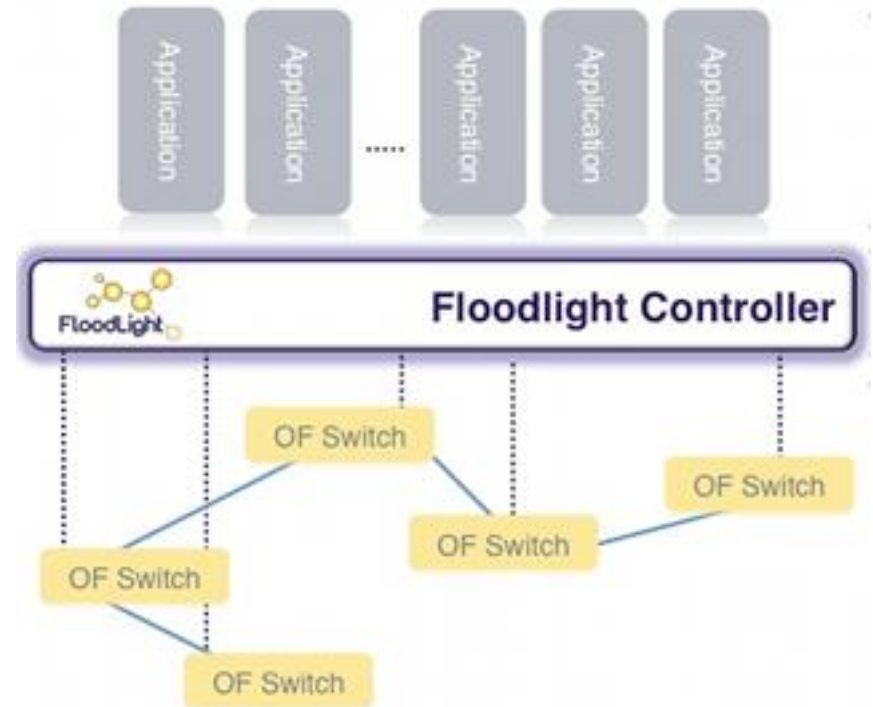


2. Floodlight 연동

Floodlight Controller로 연동시키기

Floodlight 란?

- Floodlight 컨트롤러는 자바 기반 오픈소스 컨트롤러이며 openflow v1.5까지 지원한다. 오픈 커뮤니티로 FAQ 및 지식 교류도 활발히 운영되고 있다. 스탠포드 대학교 출신들이 세운 빅스위치란 회사에서 개발한 SW이다. 본 프로젝트에서는 Floodlight 컨트롤러를 Miniedit과 연동하여 사용할 것이다.
- <https://www.projectfloodlight.org/> 에서 다운가능하다.

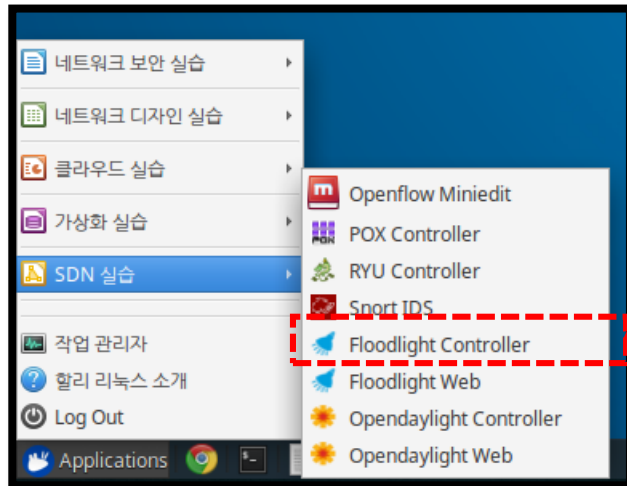


Floodlight Controller 설정

1) Floodlight Controller를 구동한다.

- 미리 구성한 서버2의 ESXi 환경에 설치된 Floodlight Controller를 연다.

- 파일 위치 : Applications > SDN 실습 > Floodlight Controller
- 열면 하단 그림과 같이 터미널 하나가 생기면서 Floodlight Controller가 켜졌음을 보여준다.



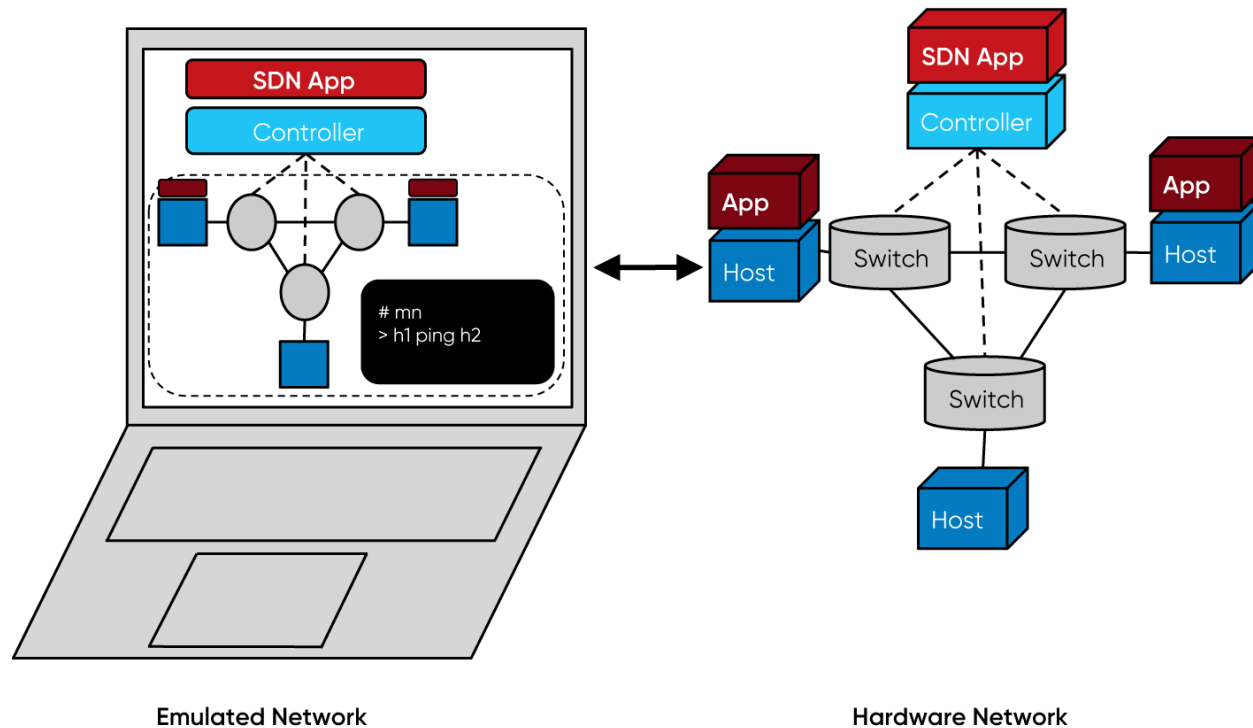
```
Terminal
File Edit View Search Terminal Help
rver on port 8080
2019-01-28 10:48:31.650 INFO [org.restlet] Starting net.floodlightcontroller.re
stserver.RestApiServer$RestApplication application
2019-01-28 10:48:31.694 INFO [n.f.c.i.OFChannelHandler] New switch connection f
rom /192.168.0.114:39704
2019-01-28 10:48:31.707 INFO [n.f.c.i.OFChannelHandler] Negotiated down to swit
ch OpenFlow version of OF_13 for /192.168.0.114:39704 using lesser hello header
algorithm.
2019-01-28 10:48:31.741 INFO [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch
DPID[00:00:00:00:00:00:10] bound to class class net.floodlightcontroller.cor
e.internal.OFSwitch, description SwitchDescription [manufacturerDescription=Nici
ra, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.4, serialNum
ber=None, datapathDescription=s1]
2019-01-28 10:48:31.919 INFO [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow t
ables of 00:00:00:00:00:00:10 on upcoming transition to MASTER.
2019-01-28 10:48:31.985 ERROR [n.f.c.i.OFSwitchHandshakeHandler] OFGroupModFaile
dErrorMsgVer13(xid=12, code=INVALID GROUP, data=OFGroupDeleteVer13(xid=12, group
Type=INDIRECT, group=all, buckets=[])) from switch OFSwitch DPID[00:00:00:00:
00:00:10] in state net.floodlightcontroller.core.internal.OFSwitchHandshakeHandl
er$MasterState@32644e23
2019-01-28 10:48:32.78 INFO [n.f.t.TopologyManager] Recomputing topology due to
: link-discovery-updates
2019-01-28 10:48:32.997 INFO [n.f.j.JythonServer] Starting DebugServer on :6655
```

3. Miniedit 구동

Miniedit 구동시키기

Miniedit (Mininet Edit GUI Tool) 란?

- Miniedit 는 mininet에서 제공하는 에디터이다. CLI 방식은 불편하기에 Miniedit을 이용하여 간단하고 쉽게 토폴로지를 구성하고 각종 Controller를 연결하여 SDN 실습을 진행할 수 있다. 본 프로젝트에서는 이 Miniedit을 통해 Floodlight Controller와 연동, 혹은 Pox controlle와 연동하여 그 결과를 측정할 것이다.

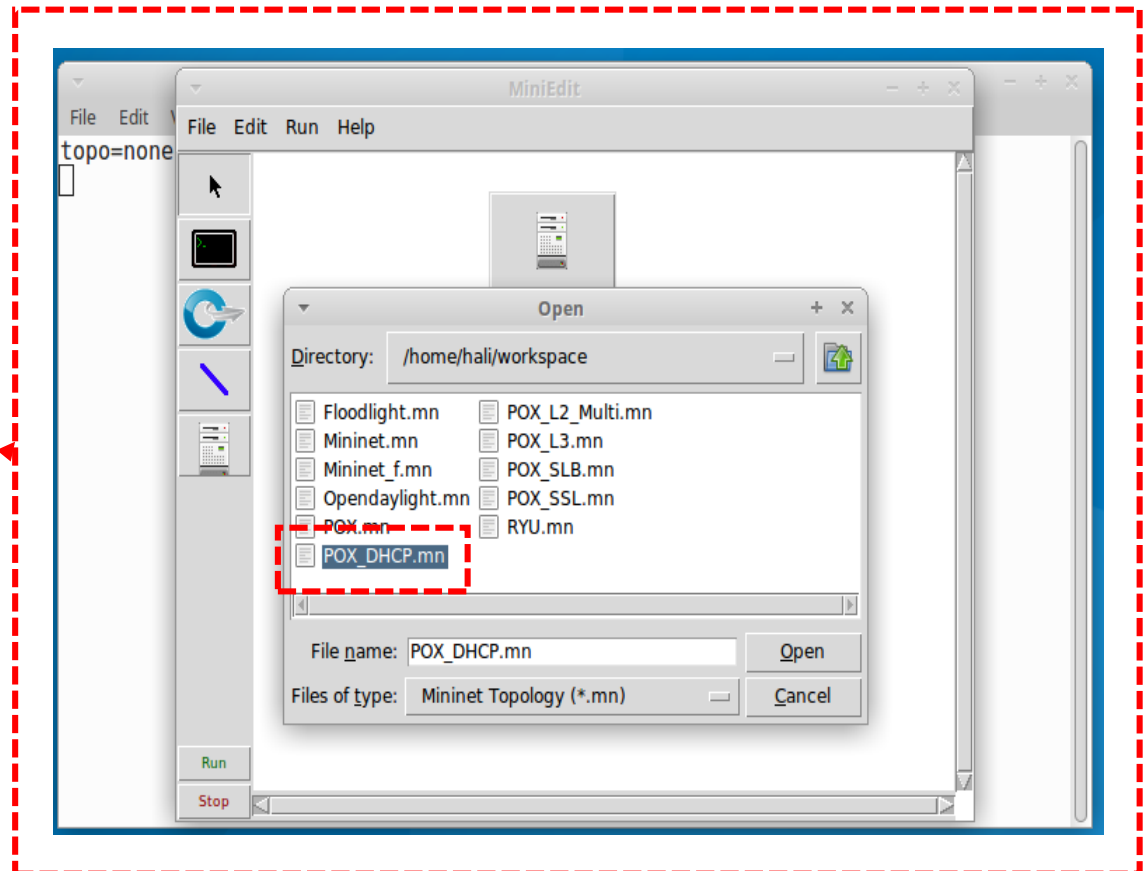
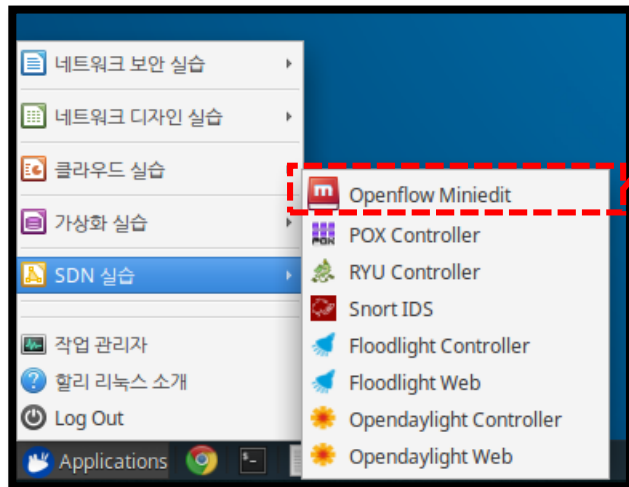


Miniedit 구동 및 Controller 연동하기

1) Pox Controller 와 Miniedit 연동하기

- 미리 구성한 서버1의 hali-linux에 설치된 Openflow Miniedit을 연다. (단, POX Controller가 먼저 켜져야 함)

- 파일 위치 : Applications > SDN 실습 > Openflow Miniedit
- File > Open > POX_DHCP.mn을 불러온다. (기본 틀이 되는 예시를 가져와서 쓸 것)

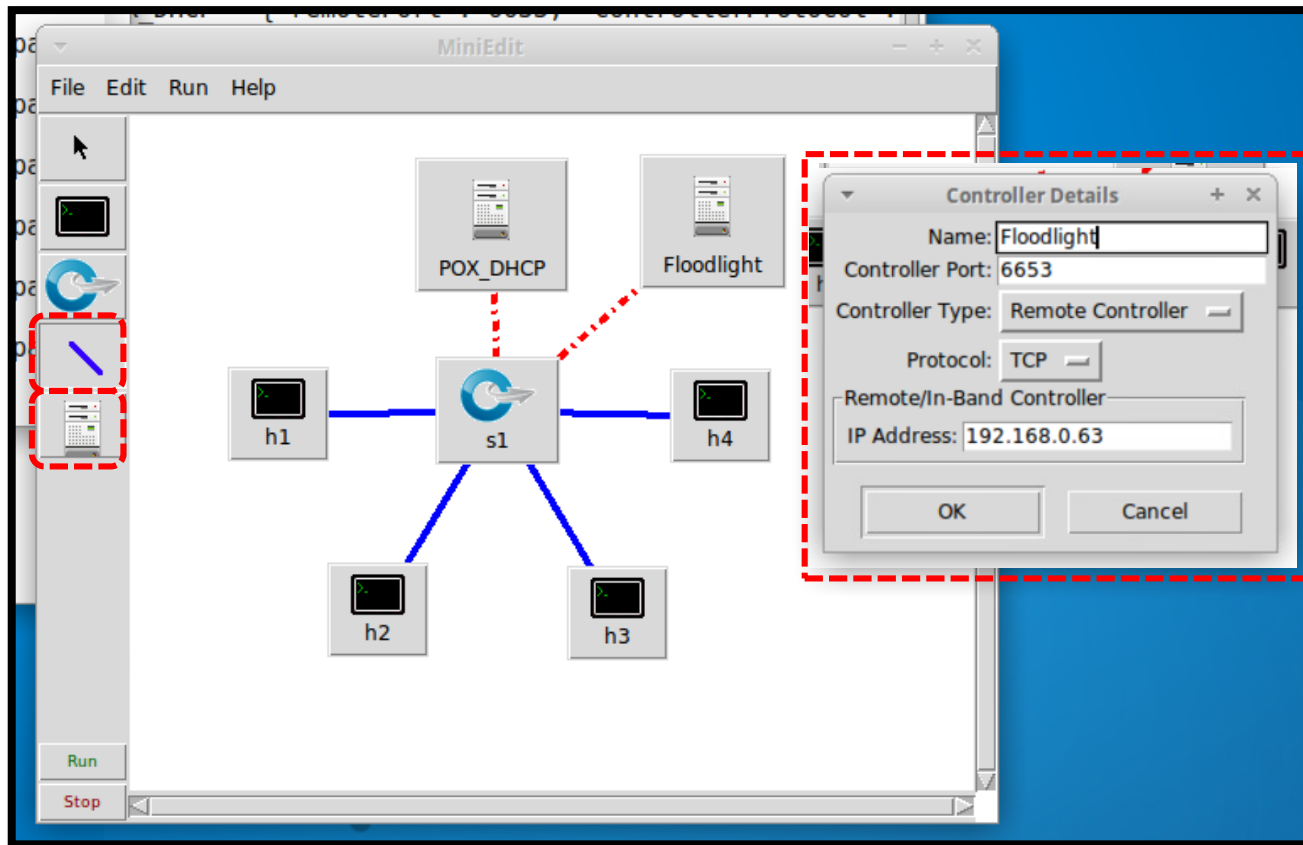


Miniedit 구동 및 Controller 연동하기

1) Pox Controller 와 Miniedit 연동하기

- POX_DHCP.mn 기본틀에서 Floodlight Controller 하나를 더 추가해준다.

- Controller Name : Floodlight / Controller Port : 6653 / IP : 192.168.0.63 / Controller Type: Remote Controller
- Remote Controller : 서버2가 되므로 서버2의 ip를 써준다.

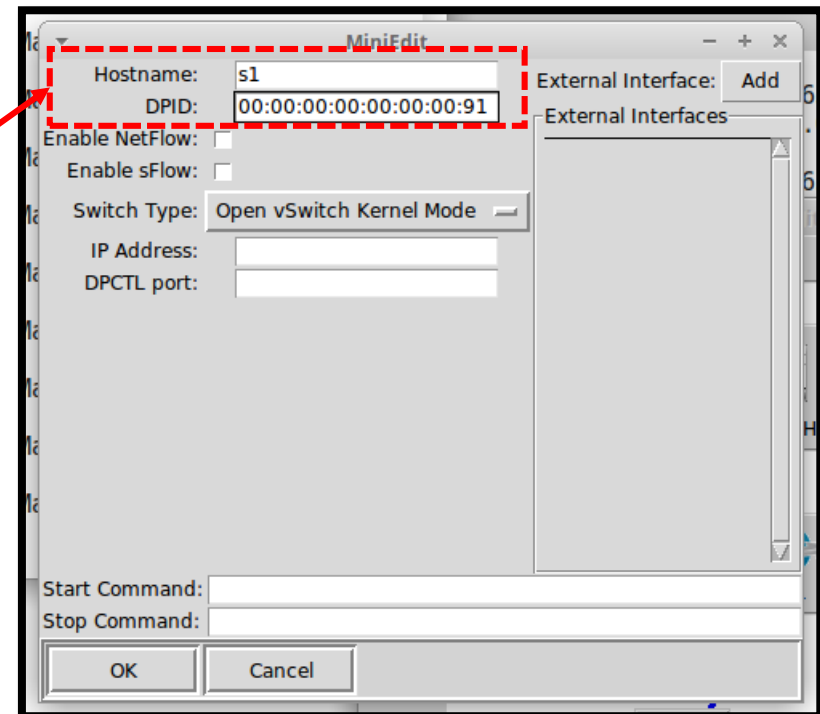
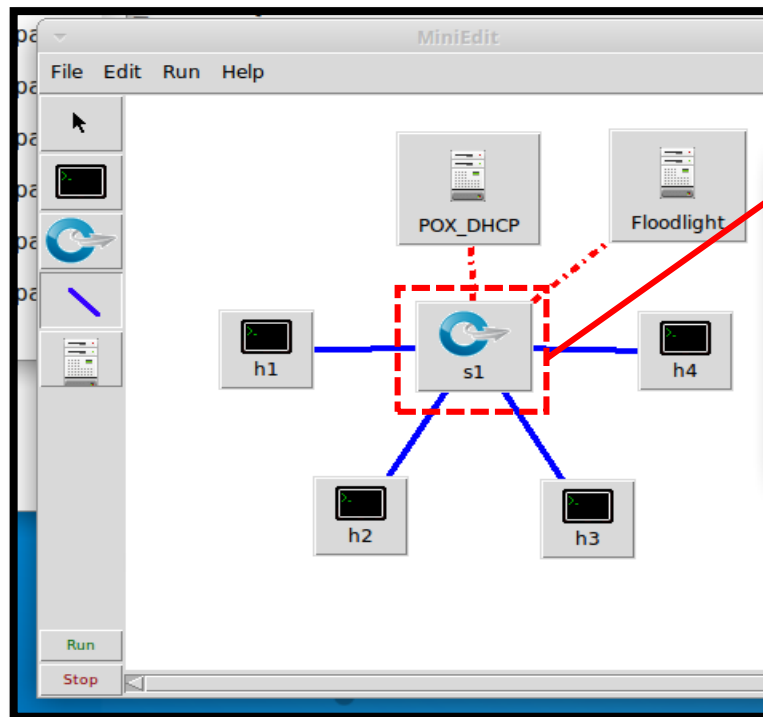


Miniedit 구동 및 Controller 연동하기

1) Pox Controller 와 Miniedit 연동하기

- POX_DHCP.mn s1 (스위치)설정을 서버2의 DPID와 겹치지 않도록 임시로 지정한다. (91로 지정함)

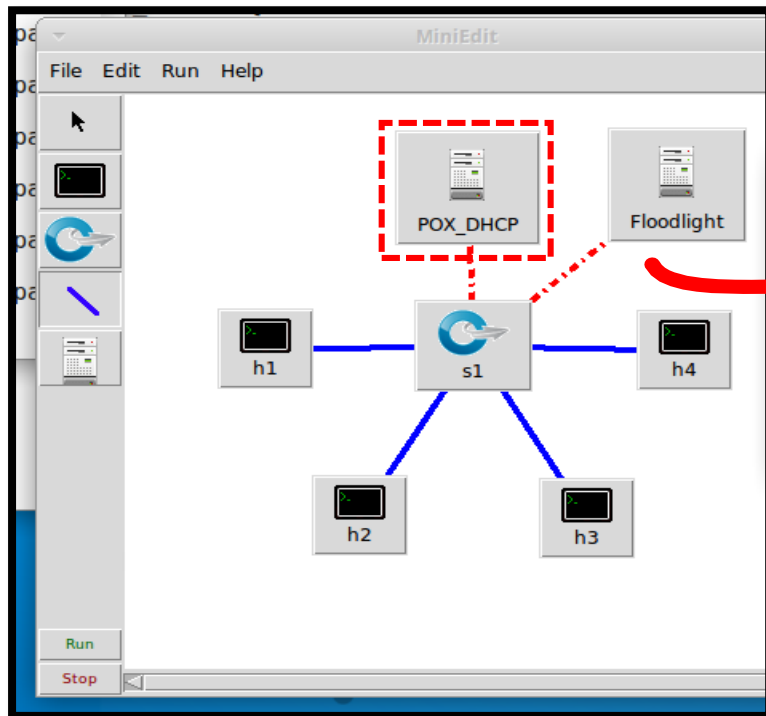
➤ 서버1의 s1 DPID -> 00:00:00:00:00:00:91



Miniedit 구동 및 Controller 연동하기

1) Pox Controller 와 Miniedit 연동하기

- POX_DHCP.mn 기본틀에서 POX_DHCP Controller 의 설정을 바꿔주고서 POX_DHCP_floodlight.mn으로 저장한다.
 - Controller Name : POX_DHCP / Controller Port : 6633 / IP : 127.0.0.1 / Controller Type: Remote Controller
 - Remote Controller : 자기자신이 되므로 자신의 로컬IP를 써준다.



The 'Controller Details' dialog box is shown with the following configuration:

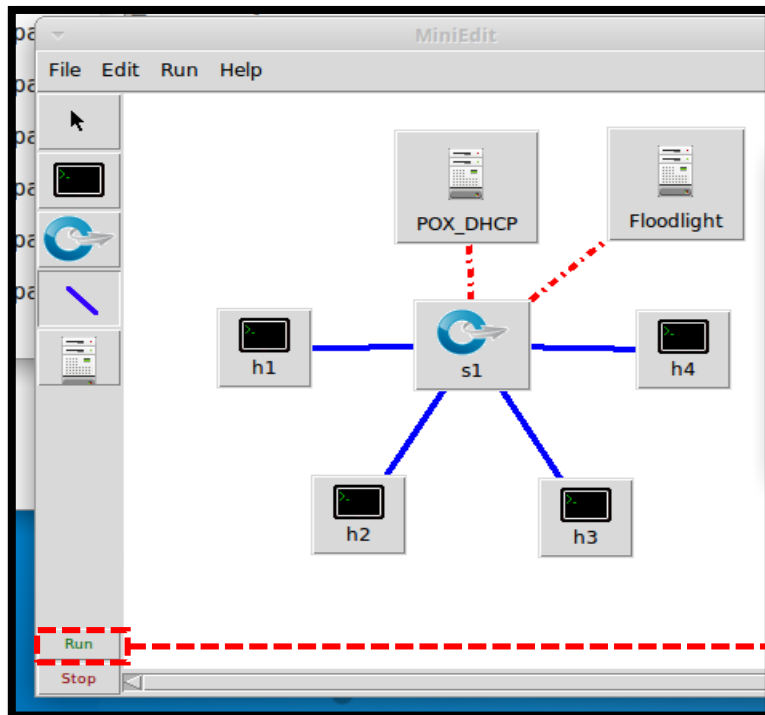
- Name: POX_DHCP
- Controller Port: 6633
- Controller Type: Remote Controller
- Protocol: TCP
- Remote/In-Band Controller: (checked)
- IP Address: 127.0.0.1

Buttons: OK, Cancel

Miniedit 구동 및 Controller 연동하기

1) Pox Controller 와 Miniedit 구동하기

- 왼쪽 하단에 있는 Run 버튼을 누르면 Mininet이 실행되어 Terminal에 실행되고 있음을 알리는 확인창이 뜬다.
 - 해당 터미널에서 어떤 Controller가 어떤 IP, PORT로 연결되었는지 다 볼 수 있으니 확인 차 보는 것도 좋다.
 - 올바르게 mininet이 구동되었다면 CLI 명령어를 칠 수 있는 란이 나온다.



```
File Edit View Search Terminal Help
'tcp', 'hostname': 'POX_DHCP', 'remoteIP': '192.168.0.63', 'controllerType': 'r
emote'}
New controller details for POX_DHCP = {'remotePort': 6633, 'controllerProtocol':
'tcp', 'hostname': 'POX_DHCP', 'remoteIP': '192.168.0.63', 'controllerType': 'r
emote'}
Getting Hosts and Switches.
Getting controller selection:remote
Getting Links.
*** Configuring hosts
h1 h3 h2 h4
**** Starting 1 controllers
POX_DHCP
**** Starting 1 switches
s1
No NetFlow targets specified.
No sFlow targets specified.

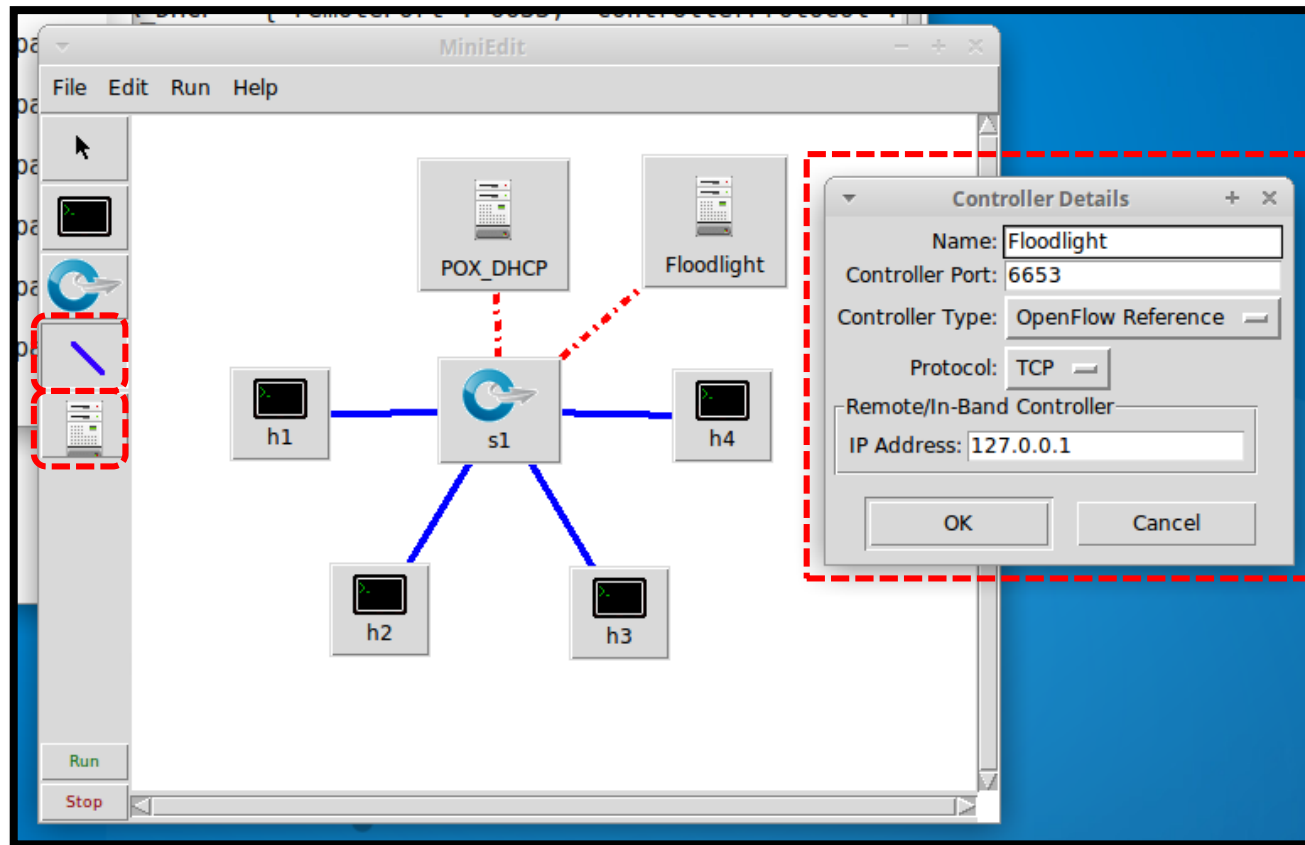
NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exi
ting will prevent MiniEdit from quitting and will prevent you from starting the
network again during this session.

*** Starting CLI:
mininet>
```

Miniedit 구동 및 Controller 연동하기

2) Floodlight Controller 와 Miniedit 연동하기

- 1)과정이 비슷하나 로컬이 다른 부분을 수정해서 설정해준다. (단, 서버2는 Floodlight Controller가 먼저 켜져야 함)
 - Controller Name : Floodlight / Controller Port : 6653 / IP : 127.0.0.1 / Controller Type: Remote Controller
 - Remote Controller : 자기 자신이므로 로컬 IP를 써준다.

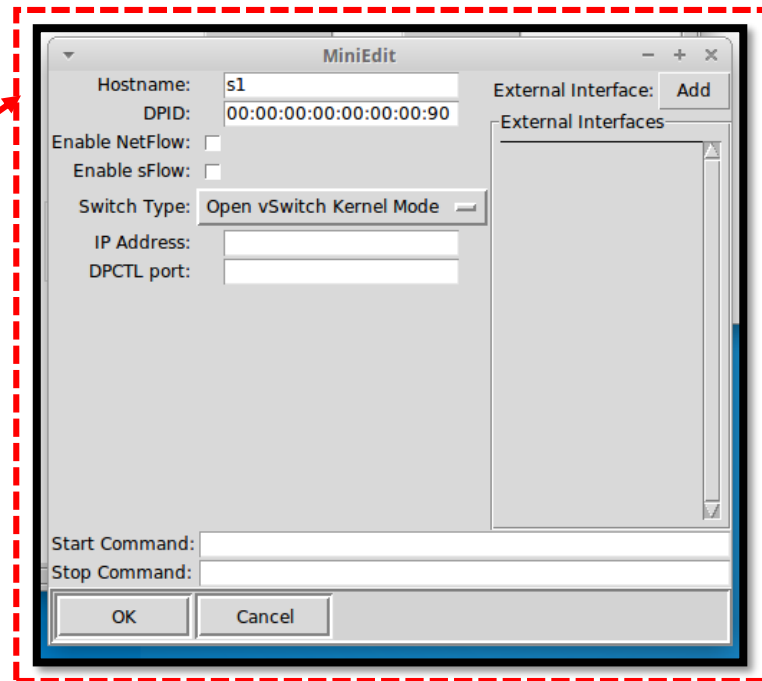
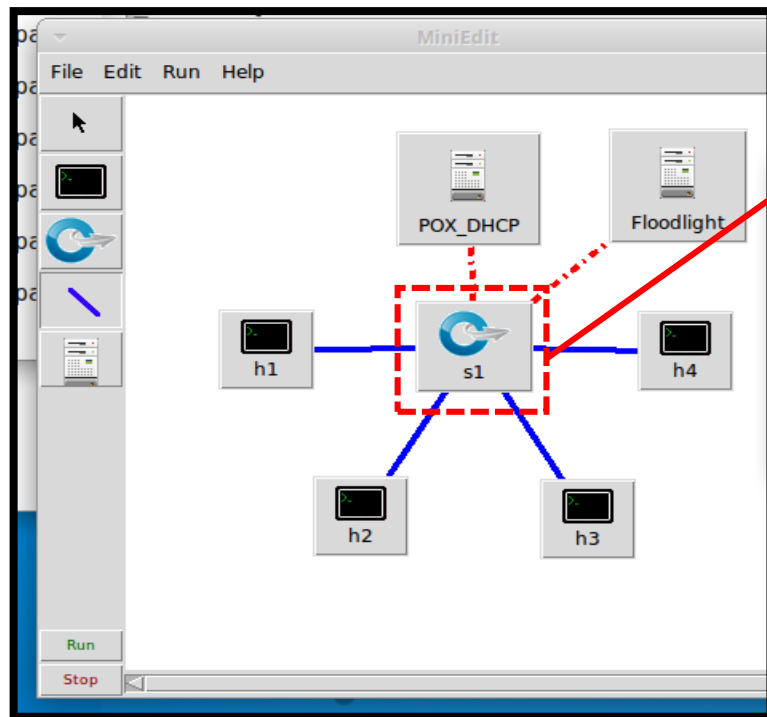


Miniedit 구동 및 Controller 연동하기

2) Floodlight Controller 와 Miniedit 연동하기

- POX_DHCP.mn s1 (스위치) 설정을 서버1의 DPID와 겹치지 않도록 임시로 지정한다. (90로 지정함)

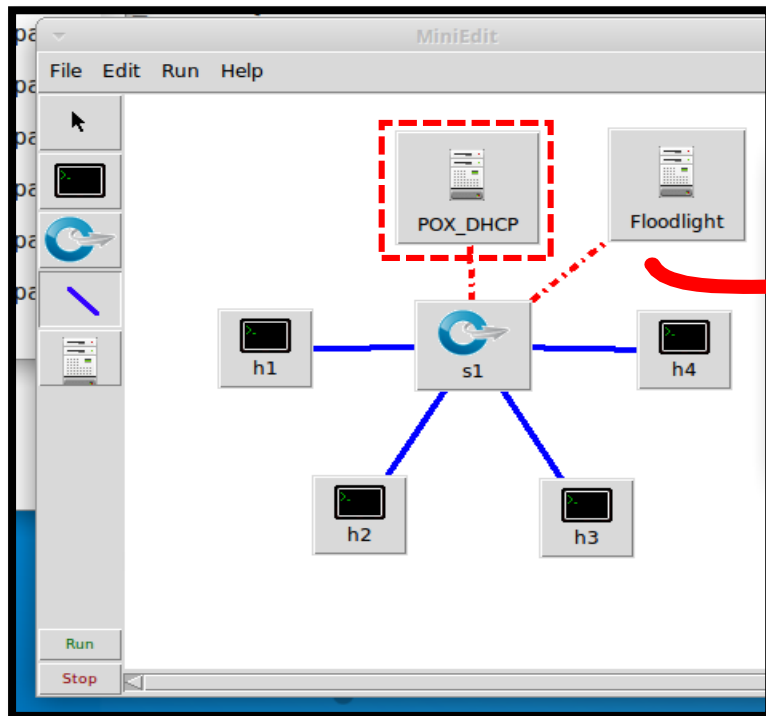
➤ 서버2의 s1 DPID -> 00:00:00:00:00:00:90 (서버1은 91이었음)



Miniedit 구동 및 Controller 연동하기

2) Floodlight Controller 와 Miniedit 연동하기

- POX_DHCP.mn 기본틀에서 POX_DHCP Controller 의 설정을 바꿔주고서 POX_DHCP_floodlight.mn으로 저장한다.
 - Controller Name : POX_DHCP / Controller Port : 6633 / IP : 192.168.0.185 / Controller Type: Remote Controller
 - Remote Controller : 서버1의 IP를 적어준다.



The 'Controller Details' dialog box is shown with the following configuration:

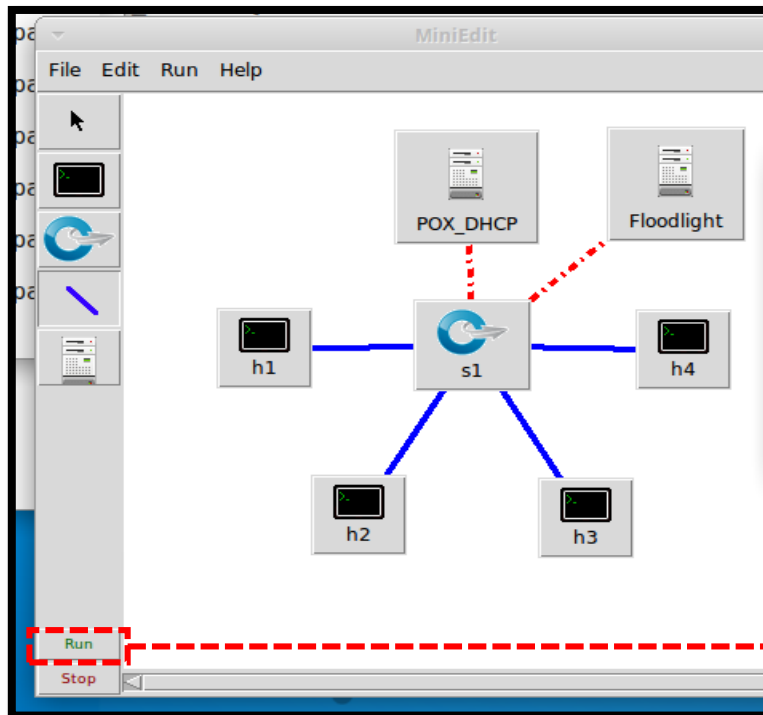
- Name: POX_DHCP
- Controller Port: 6633
- Controller Type: Remote Controller
- Protocol: TCP
- Remote/In-Band Controller: IP Address: 192.168.0.185

Buttons: OK, Cancel

Miniedit 구동 및 Controller 연동하기

2) Floodlight Controller 와 Miniedit 연동하기

- 왼쪽 하단에 있는 Run 버튼을 누르면 Mininet이 실행되어 Terminal에 실행되고 있음을 알리는 확인창이 뜬다.
 - 해당 터미널에서 어떤 Controller가 어떤 IP, PORT로 연결되었는지 다 볼 수 있으니 확인 차 보는 것도 좋다.
 - 올바르게 mininet이 구동되었다면 CLI 명령어를 칠 수 있는 란이 나온다.



```
File Edit View Search Terminal Help
topo=None
New controller details for POX_DHCP = {'remotePort': 6633, 'controllerProtocol':
'tcp', 'hostname': 'POX_DHCP', 'remoteIP': '192.168.0.185', 'controllerType': '
remote'}
New controller details for floodlight = {'remotePort': 6653, 'controllerProtocol
': 'tcp', 'hostname': 'floodlight', 'remoteIP': '127.0.0.1', 'controllerType': '
remote'}
Getting Hosts and Switches.
Getting controller selection:remote
Getting controller selection:remote
Getting Links.
*** Configuring hosts
h2 h1 h4 h3
**** Starting 2 controllers
POX_DHCP floodlight
**** Starting 1 switches
s1
No NetFlow targets specified.
No sFlow targets specified.

NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exi
ting will prevent MiniEdit from quitting and will prevent you from starting the
network again during this session.

*** Starting CLI:
```

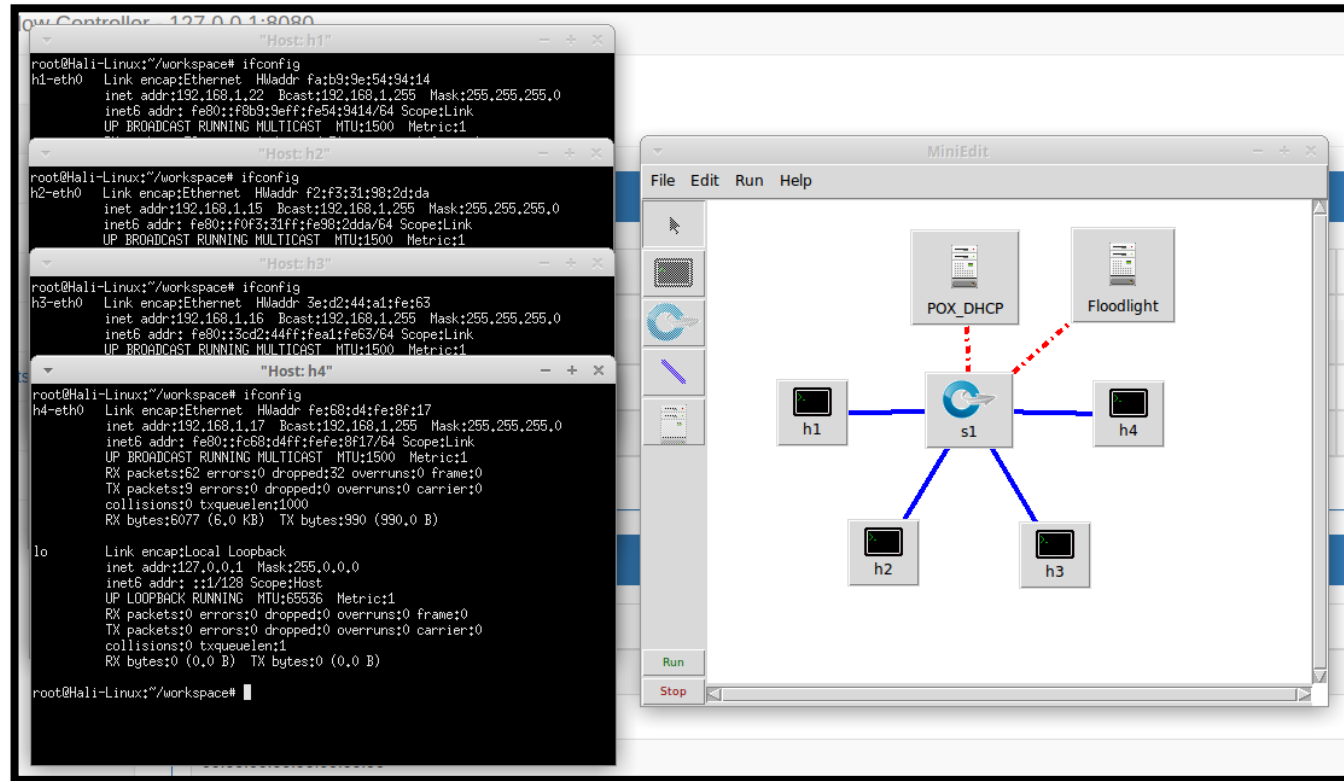

Miniedit 구동 및 Controller 연동하기

3) Miniedit Terminal에서 DHCP로 각각 IP 할당해주기

- POX와 연동된 Mininet CLI 창에서 dhclient 명령어를 통해 h1,h2,h3,h4 에 각각 ip를 할당해주기

➤ h1 : 192.168.1.22 / h2 : 192.168.1.15 / h3 : 192.168.1.16 / h4 : 192.168.1.17

```
*** Starting CLI:  
mininet> h1 dhclient  
mininet> h2 dhclient  
mininet> h3 dhclient  
mininet> h4 dhclient
```



Miniedit 구동 및 Controller 연동하기

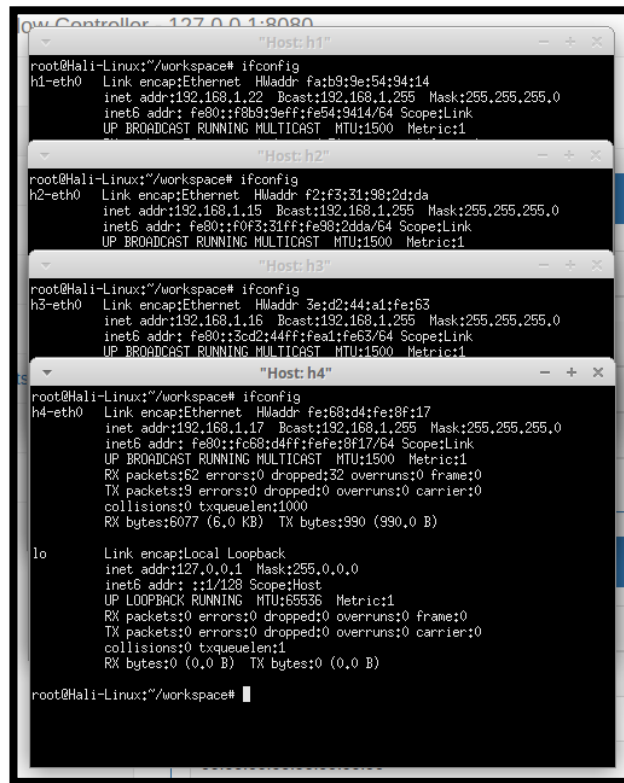
3) Miniedit Terminal에서 DHCP로 각각 IP 할당해주기

- 각각 연동된 Mininet CLI 창에서 dhclient 명령어를 통해 h1,h2,h3,h4 에 각각 ip를 할당해주기

➢ [POX_서버1] h1 : 192.168.1.22 / h2 : 192.168.1.15 / h3 : 192.168.1.16 / h4 : 192.168.1.17

➢ [Flood_서버2] h1 : 192.168.1.30 / h2: 192.168.1.29 / h3 : 192.168.1.28 / h4 : 192.168.1.27

```
*** Starting CLI:
mininet> h1 dhclient
mininet> h2 dhclient
mininet> h3 dhclient
mininet> h4 dhclient
```



```
POX_Controller 127.0.0.1:8080
root@Hali-Linux:~/workspace# ifconfig
h1-eth0 Link encap:Ethernet HWaddr fa:b9:9e:54:94:14
        inet addr:192.168.1.22 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::f8b3:9eff:fe54:9414/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

root@Hali-Linux:~/workspace# ifconfig
h2-eth0 Link encap:Ethernet HWaddr f2:f3:31:98:2d:da
        inet addr:192.168.1.15 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::f0f3:31ff:fe98:2dda/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

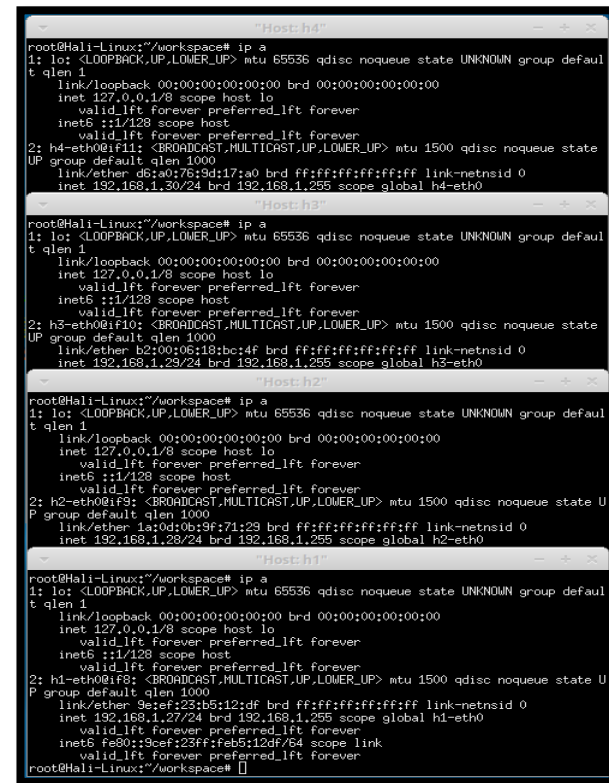
root@Hali-Linux:~/workspace# ifconfig
h3-eth0 Link encap:Ethernet HWaddr Seid2:44:a1:fe:63
        inet addr:192.168.1.16 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::3cd2:44ff:feal:fe63/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

root@Hali-Linux:~/workspace# ifconfig
h4-eth0 Link encap:Ethernet HWaddr fe:68:d4:fe:8f:17
        inet addr:192.168.1.17 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::fc68:d4ff:fe8f:17/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:62 errors:0 dropped:32 overruns:0 frame:0
        TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:6077 (6.0 KB)  TX bytes:990 (990.0 B)

lo Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  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:1
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@Hali-Linux:~/workspace#
```

POX_서버1



```
root@Hali-Linux:~/workspace# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: h4-eth0:f11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    group default qlen 1000
    link/ether d8:a0:76:9d:17:a0 brd ffff:ffff:ffff:ffff link-netnsid 0
    inet 192.168.1.30/24 brd 192.168.1.255 scope global h4-eth0

root@Hali-Linux:~/workspace# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: h3-eth0:f10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    group default qlen 1000
    link/ether b2:00:06:18:bc:4f brd ffff:ffff:ffff:ffff link-netnsid 0
    inet 192.168.1.29/24 brd 192.168.1.255 scope global h3-eth0

root@Hali-Linux:~/workspace# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: h2-eth0:f8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    group default qlen 1000
    link/ether 1a:0d:0b:9f:71:29 brd ffff:ffff:ffff:ffff link-netnsid 0
    inet 192.168.1.28/24 brd 192.168.1.255 scope global h2-eth0

root@Hali-Linux:~/workspace# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: h1-eth0:f8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    group default qlen 1000
    link/ether 9e:ef:23:b5:12:df brd ffff:ffff:ffff:ffff link-netnsid 0
    inet 192.168.1.27/24 brd 192.168.1.255 scope global h1-eth0
    inet6 fe80::9eef:23bf:feb5:12df/64 scope link
        valid_lft forever preferred_lft forever

root@Hali-Linux:~/workspace#
```

Floodlight_서버2

Miniedit 구동 및 Controller 연동하기

4) 각 Host 별로 웹 서버를 하나 만들어서 통신이 가능한지 확인하기

- 각각 연동된 Mininet CLI 창에서 dhclient 명령어를 통해 h1,h2,h3,h4 에 각각 ip를 할당해주기

➤ [POX_서버1] h1 :

4. Floodlight 결과 측정

Floodlight Web에서 결과 측정하기

1) Floodlight Web을 켜서 Dashboard 들어가기

- Floodlight Controller를 실행하고 있는 서버2에서 진행해야 함. Floodlight Web을 들어가면 아래와 같은 대시보드가 뜬
 - 서버1과 서버2의 호스트, 스위치가 모두 다 나와야 정상임 (현재 POX와 Floodlight controller 둘 다 연동되어 있기 때문)

The image shows a Windows taskbar on the left with a menu open, highlighting 'Floodlight Web'. The main part of the image is a screenshot of the Floodlight Web dashboard in a web browser. The dashboard displays the following information:

- Controller Status:** Active (Green checkmark icon)
- Uptime:** 00:09:44 (Orange clock icon)
- Controller Role:** ACTIVE (Blue user icon)
- Switches:** 2 (Red box with double arrows icon)
- Hosts:** 8 (Red box with laptop icon)
- Connections (Links):** 0 (Red box with double arrows icon)
- Reserved Ports:** 0 (Red box with checkmark icon)
- JVM Memory Bloat:** 128.13 MB (Green gauge icon)
- Consumption Detail:**

Total:	369.62 MB
Used:	128.65 MB
Free:	240.97 MB
- Storage Tables:**

controller_controller
controller_controllerinterface
controller_switchconfig
controller_forwardingconfig
controller_staticentrytable
controller_topologyconfig
controller_link
controller_firewallrules

Floodlight Web에서 결과 측정하기

2) Floodlight Web에서 스위치 확인하기.

- Switches 란에 들어가면 앞서서 설정해둔 서버1인 91과 서버2인 90이 현재 연동되었음을 확인가능하다.

➤ 해당 스위치에 들어가면 더 자세한 정보를 볼 수 있다.

Floodlight OpenFlow Controller - 127.0.0.1:8080

Controller (Home)
Switches
Hosts
Links
Topology
Firewall
Access Control Lists
Statistics
Change Controllers

Switches

Switches Connected

Switch ID	IPv4 Address	Connected Since
00:00:00:00:00:00:90	/127.0.0.1:38128	Mon Jan 28 2019 02:36:19 GMT+0900 (Korean Standard Time)
00:00:00:00:00:00:91	/192.168.0.185:50618	Mon Jan 28 2019 02:38:12 GMT+0900 (Korean Standard Time)

Showing 1 to 2 of 2 entries

Switch Roles

Switch MAC
00:00:00:00:00:00:91
00:00:00:00:00:00:90

Switch Detail

Switch Detail

MAC	: 00:00:00:00:00:00:90
Version	: OF_13
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.5.4
Serial Number	: None
Datapath	: s1

Flow Summary

Flow Count	: 1
Packet Count	: 58
Byte	: 5164
Flag	:
Buffer	: 256
Table Count	: 254

Role Info

☒ MASTER
☐ SLAVE
☐ EQUAL

Change

Port Table

Show 10 entries Search:

No	R. Packets	Tran. Packets	R. Bytes	Tran. Bytes	R. Dropped	Tran. Dropped	Coll.	Duration(s)
local	0	0	0	0	35	0	0	590

Showing 1 to 1 of 1 entries.

Floodlight Web에서 결과 측정하기

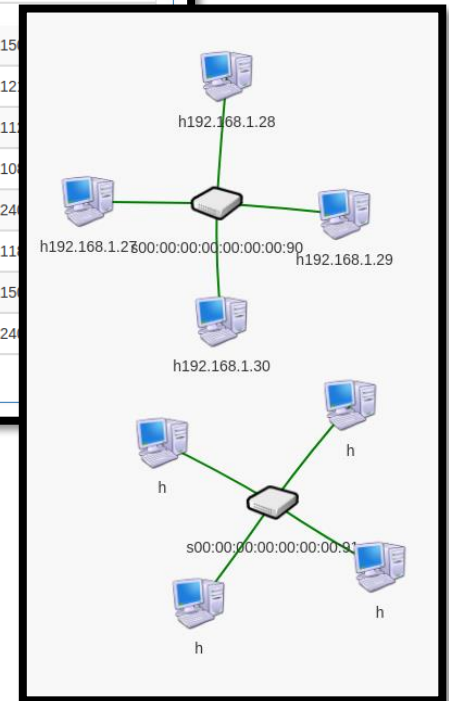
2) Floodlight Web에서 Host와 네트워크 토폴로지 확인하기.

- Hosts, Topology 란에 들어가면 앞서서 설정해둔 서버1인 91과 서버2인 90이 현재 연동되었음을 확인 가능하다.

➢ 그러나 어디까지나 해당 로컬인 90의 host들만 나타나고, 91의 host ip는 나타나지 않는다. (그러나 실제로 연동되긴 했음)

Hosts						
Hosts Connected						
MAC	IPv4 Address	IPv6 Address	Switch	Port	Last Seen	
1a:0d:0b:9f:71:29	192.168.1.28	fe80::180d:bff:fe9f:7129	00:00:00:00:00:00:90	2	154864315	
36:4e:00:07:90:12			00:00:00:00:00:00:91	4	154864312	
6a:cc:ad:55:72:27			00:00:00:00:00:00:91	2	154864311	
82:53:96:34:41:73			00:00:00:00:00:00:91	1	154864310	
9e:ef:23:b5:12:df	192.168.1.27	fe80::9cef:23ff:feb5:12df	00:00:00:00:00:00:90	1	154864324	
a2:f7:14:1c:d2:d1		fe80::a0f7:14ff:fe1c:d2d1	00:00:00:00:00:00:91	3	154864311	
b2:00:06:18:bc:4f	192.168.1.29	fe80::b000:6ff:fe18:bc4f	00:00:00:00:00:00:90	3	154864315	
d6:a0:76:9d:17:a0	192.168.1.30	fe80::d4a0:76ff:fe9d:17a0	00:00:00:00:00:00:90	4	154864324	

Showing 1 to 8 of 8 entries



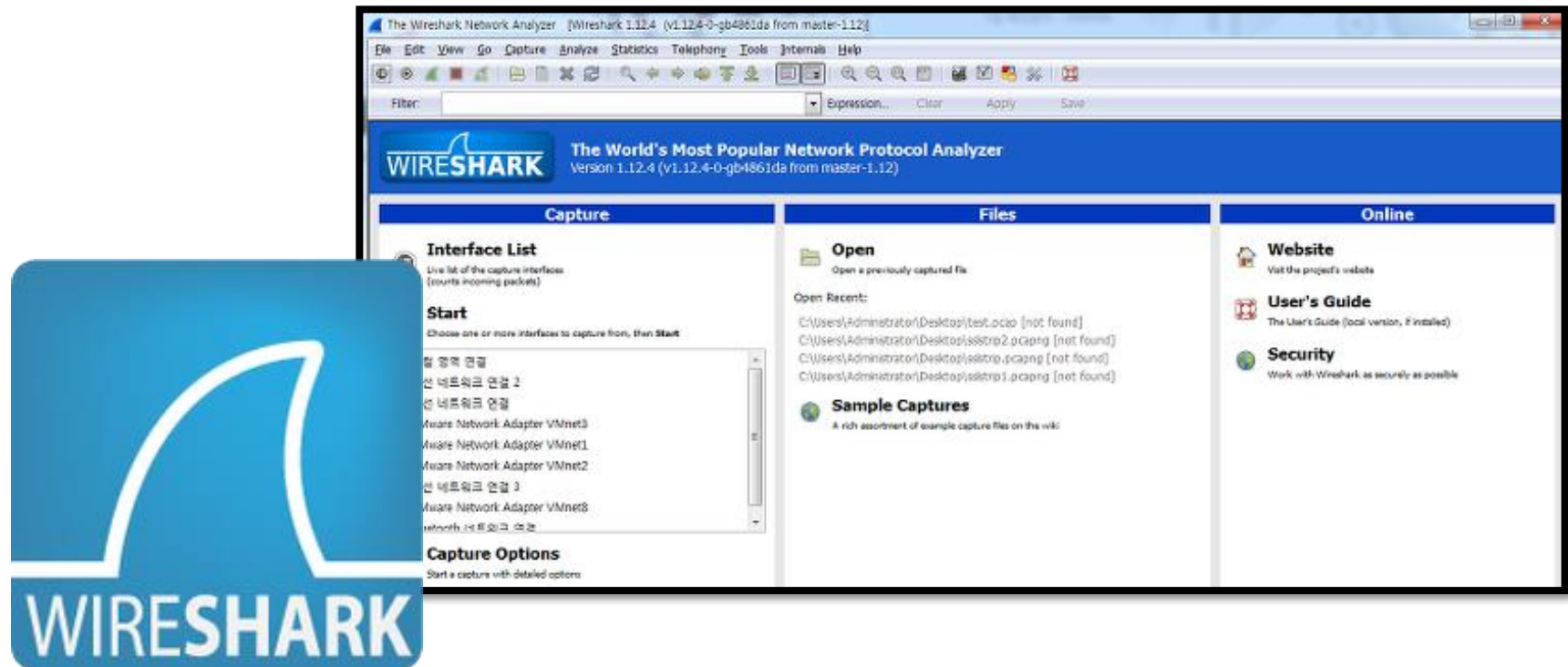
➢ 토폴로지에서 91(서버1의 스위치)의 호스트 IP는 나오지 않는다.

5. WireShark 결과 측정 및 분석

Wireshark로 패킷 측정하기

Wireshark 란?

- Wireshark란 네트워크를 분석하는데 사용되는 공개된 패킷 스나핑(packet sniffing) 프로그램이다. 제럴드 콤즈라는 사람에 의해 만들어졌으며 Ehtereal 프로젝트에서 Wireshark로 이름이 바뀌었다. 세계에서 가장 널리 쓰이는 네트워크 분석 프로그램으로 네트워크 상에서 캡처한 데이터에 대한 네트워크 / 상위 레이어 프로토콜의 정보를 제공해준다. 패킷캡처를 위해 pcap 네트워크 라이브러리를 이용한다.

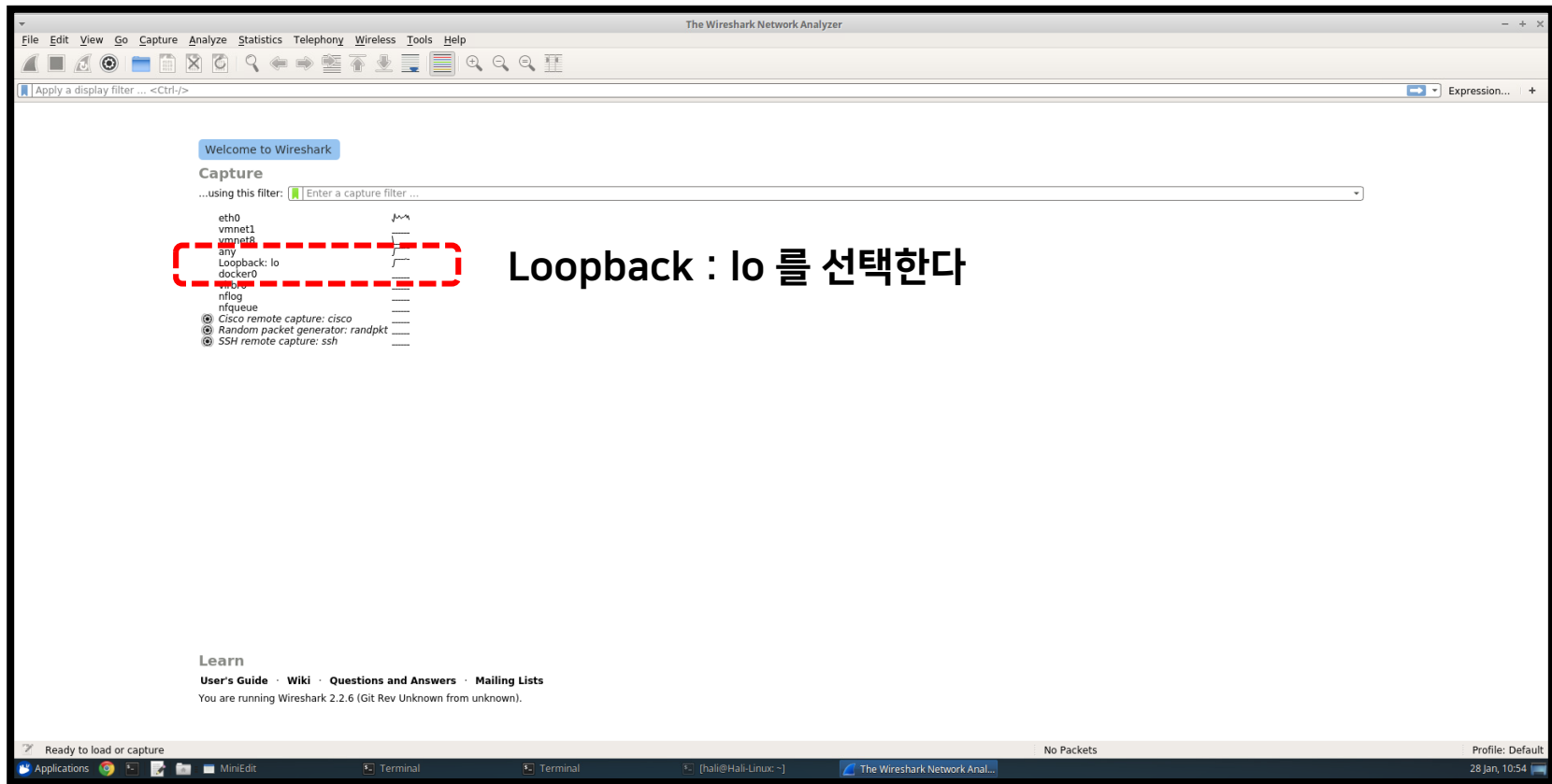


Wireshark로 패킷 측정 및 분석하기

1) Wireshark를 먼저 들어가서 켜다

- Hosts, Topology 란에 들어가면 앞서서 설정해둔 서버1인 91과 서버2인 90이 현재 연동되었음을 확인 가능하다.

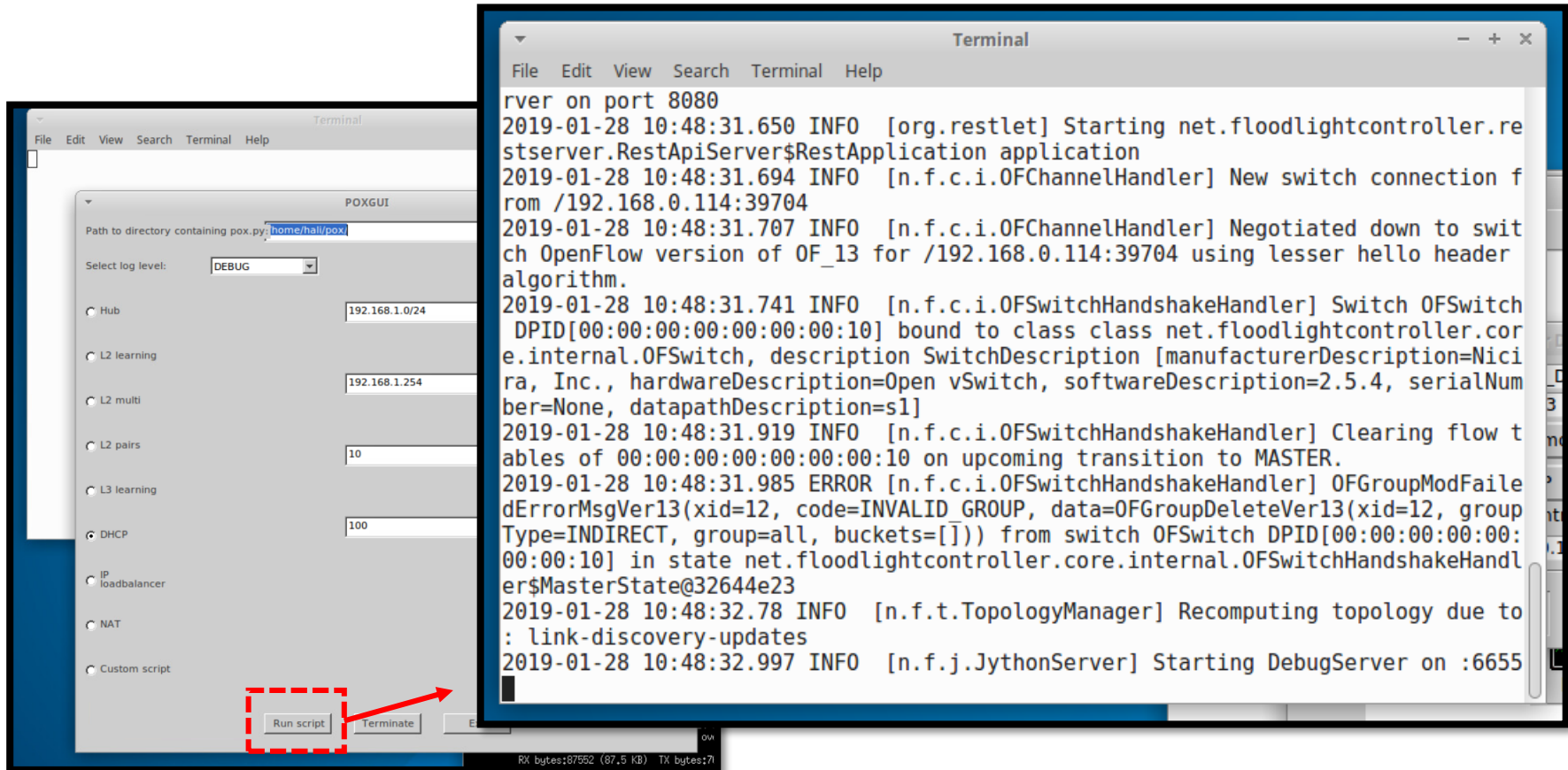
➤ 그러나 어디까지나 해당 로컬인 90의 host들만 나타나고, 91의 host ip는 나타나지 않는다. (그러나 실제로 연동되긴 했음)



POX-DHCP 설정

1) Pox Controller 열어서 POX-DHCP 설정하기

- POX Controller에서 경로 /home/hali/pox로 설정 후에 DHCP를 선택해서 입력하기.
 - 입력 후에 Run scrip를 누르면 같이 뜬 Terminal 창에서 아래와 같이 pox가 구동되었음을 보여주고 있음.



1. Floodlight 구동

2. Miniedit 구동

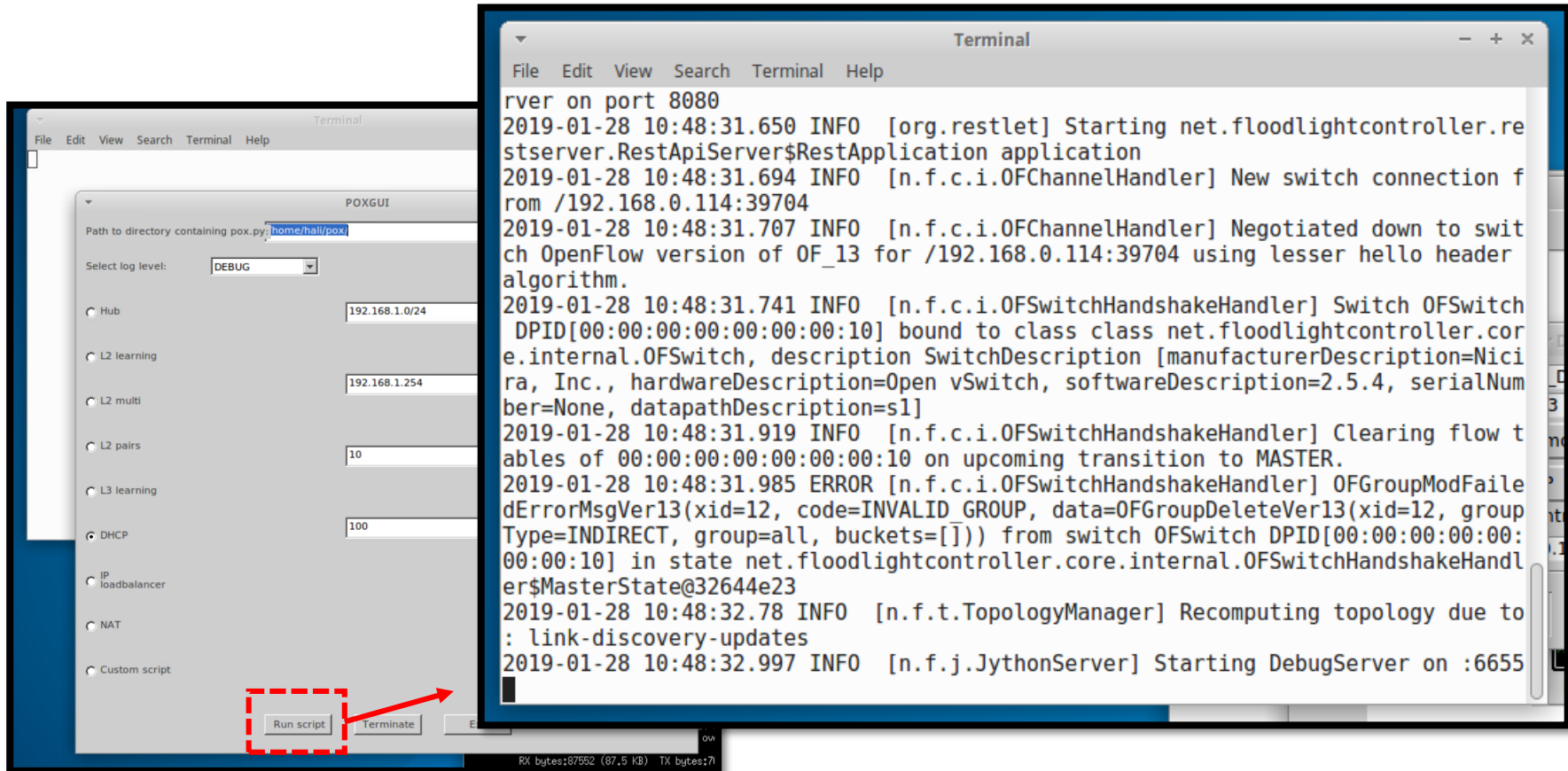
3. WireShark 결과 측정 및 분석

POX-DHCP 설정

2) Pox Controller에서 POX-DHCP 실행하기

- POX Controller에서 run script 눌러 실행하기

➤ 입력 후에 Run scrip를 누르면 같이 뜬 Terminal 창에서 아래와 같이 pox가 구동되었음을 보여주고 있음.



감사합니다