Mininet(1): SDN-based Switch and Hub

Week 6

Contents

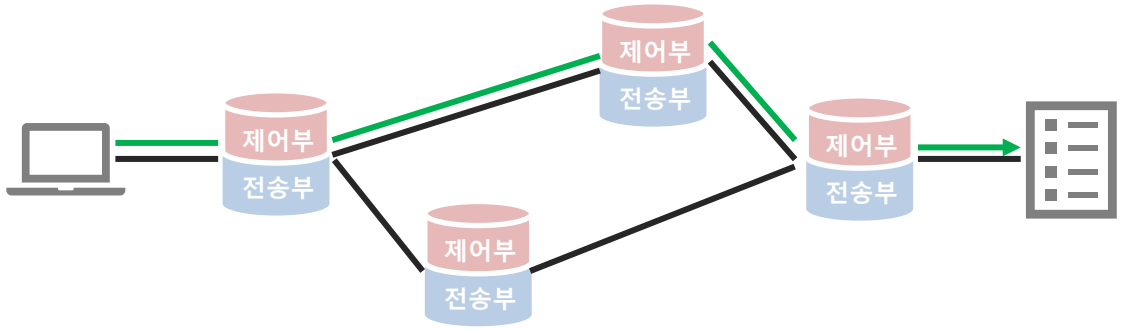
- **Getting started**
  - SDN (Software Defined Networking)
  - OpenFlow
  - Mininet and Pox controller
  - Hub and L2 Switch
- **Experiment**
  - Experiment 1: Adding Flow in Flow Table
  - Experiment 2: SDN-based Hub
  - Experiment 3: SDN-based L2 Switch
- **Result Report**
- **Pre Report**

Getting started

Getting Started

■ SDN (Software Defined Networking)

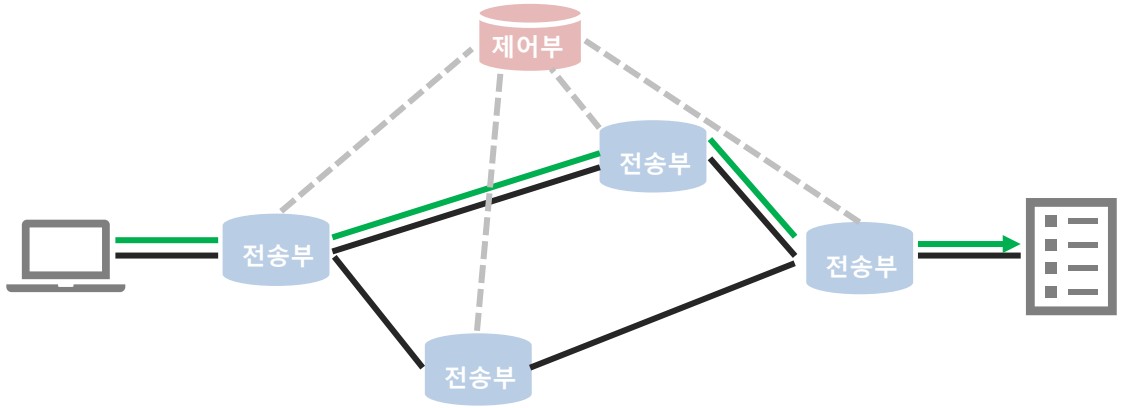
- 네트워크 장비 내의 제어부 (Control Plane)와 전송부 (Data Plane)의 분리
  - 제어부 = 네트워크 장비의 '뇌' , 전송부 = 네트워크 장비의 '손발'
    - Ex) 라우터
      - » 제어부: Routing 기능을 담당
      - » 전송부: Forwarding 기능을 담당



Legacy Network

- SDN 장점
  - 네트워크 장비 비용 절감
  - 네트워크 장비 관리 용이 (Ex: 업데이트, 기능 추가/변경 등)
  - Application Aware Routing에 용이

- SDN 구성
  - SDN 컨트롤러 (Control plane)
  - SDN 전송 장비 (Data plane)
  - SDN 컨트롤러와 SDN 전송 장비 사이의 표준 통신 규격



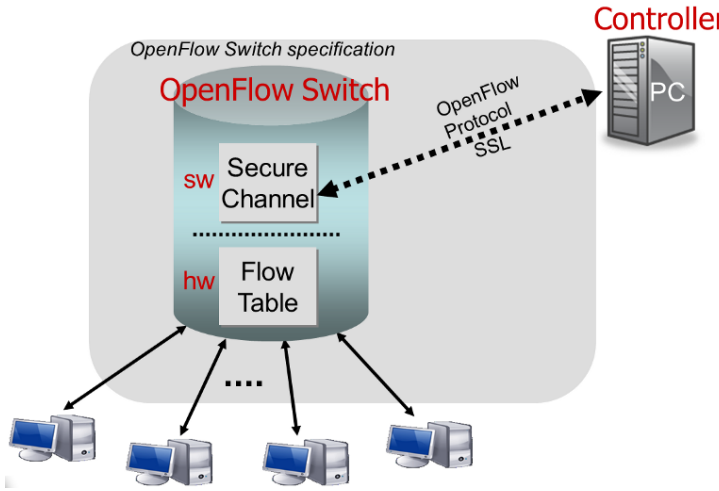
SDN Network

Getting Started

출처: Mauricio Tsugawa, “Tutorial on Software Defined Networking”

■ OpenFlow

– 가장 널리 쓰이는 SDN 컨트롤러와 SDN 전송 장비 사이의 표준 통신 규격
  - SDN 컨트롤러 → OpenFlow Controller (Controller)
  - SDN 전송장비 → OpenFlow Switch



– OpenFlow Switch
  - Secure Channel
    – Controller와 OpenFlow Switch 사이 OpenFlow를 통한 통신을 위한 통로
  - Flow Table
    – OpenFlow Switch가 패킷을 어떻게 처리해야 하는지에 대한 정보가 들어있는 테이블
    – Flow Table 구성
      » Rule: 어떻게 패킷을 처리할지를 정의하는 영역
      » Action: Rule에 의해서 정의된 패킷을 어떻게 처리할지를 정의하는 영역
      » Stats: 해당 Flow Table에 얼마나 많은 패킷이 매칭되었는지를 보여주는 영역

– Controller는 Secure Channel을 통해 OpenFlow Switch에게 Flow Table을 어떻게 채워야 한다고 알려준다.

Getting Started

■ Mininet and Pox Controller

- Mininet
  - 가상으로 OpenFlow Network을 구성할 수 있게 해주는 Tool
    - Python 기반
    - 다운로드 후 sudo mn 명령어를 이용하여 실행
    - 사용이 끝나면 exit을 입력하여 종료 후 sudo mn –c로 초기화
  - Topology
    - Controller, Switch, Host로 구성
    - 간단한 Python 코드 작성을 통해 자유로운 토폴로지 생성/수정 가능

- Pox Controller
  - Python 기반의 OpenFlow Controller
  - 사용하기 쉬워 교육용으로 자주 사용됨
  - Mininet 설치 시 패키지로 Pox Controller도 같이 설치 가능

Getting Started

■ Hub and L2 Switch

- Hub
  - 들어오는 모든 Frame을 Flooding 하는 장비
    - Flooding
      - » 수신되는 포트를 제외한 나머지 모든 포트로 Frame을 단순하게 복사 전송하는 방식
  - 어떠한 forwarding 정보도 저장되지 않는다.

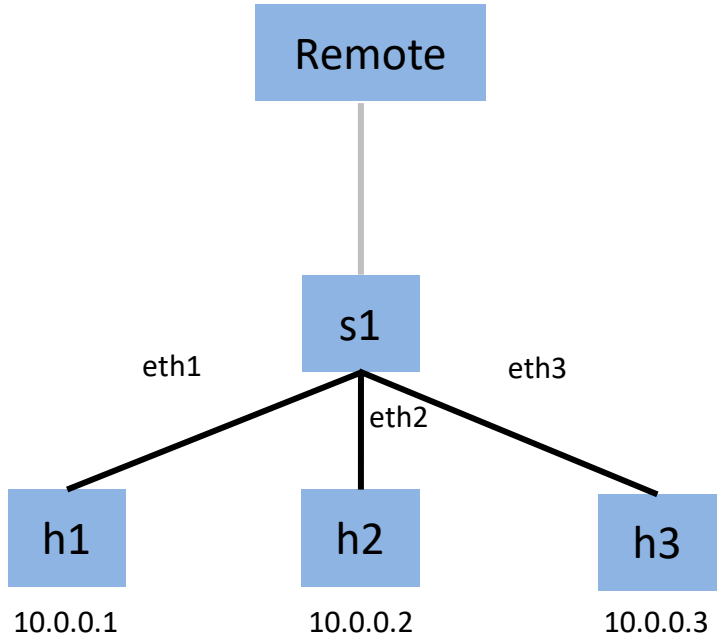
- L2 Switch
  - Frame이 들어왔을 때 그 Frame의 Destination MAC address를 보고 그 목적지로 Frame을 forwarding 해주는 장비
  - 주요 기능
    - Learning
      - » 들어오는 Frame의 Source MAC address를 MAC table에 저장하는 기능
      - » 저장된 Source MAC address는 추후 들어오는 Frame의 Destination MAC address 매칭에 사용된다.
    - Forwarding
      - » 들어오는 Frame의 Destination MAC address를 MAC table에서 찾고 매칭되는 포트로 해당 Frame을 전달하는 기능
      - » Mac table에 매칭되는 Destination MAC address가 없을 경우 Flooding을 수행한다

Experiment

Experiment

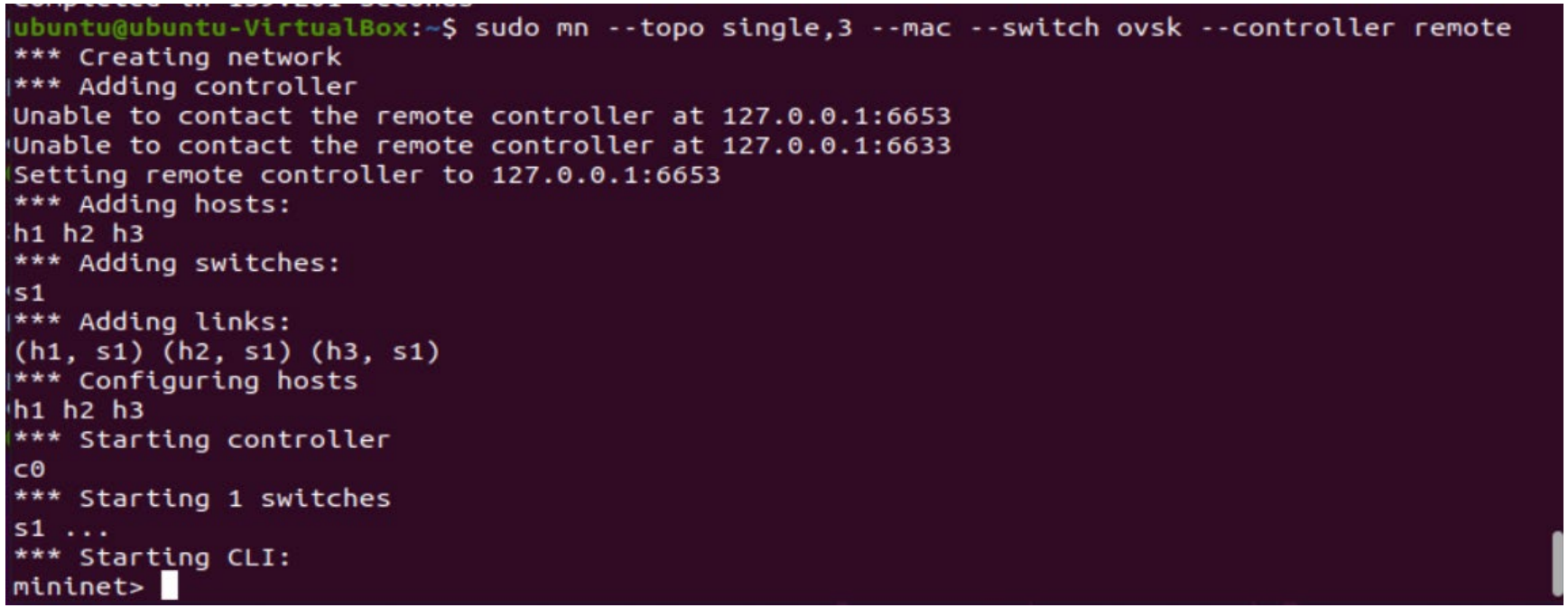
■ Experiment 1: Adding Flow in Flow Table

- h1 과 h3사이 Flow 설정하기
  - Mininet 활용하여 Star topology 구성
    - 1 Switch, 3 Host
      - » sudo mn --topo single,3 --mac --switch ovsk --controller remote
  - Flow Table 확인해보기
    - » dpctl dump-flows
  - h1 과 h3 사이 Ping Test 해보기
    - » h1 ping h3
  - h1 과 h3 사이 통신이 가능하도록 Flow 추가하기
    - » dpctl add-flow in_port=1,nw_dst=10.0.0.3,actions=output:3
    - » dpctl add-flow in_port=3,nw_dst=10.0.0.1,actions=output:1
  - Flow Table 확인하여 추가된 경로 검토하기
    - » dpctl dump-flows
  - h1 과 h3 사이 Ping Test 해보기
    - » h1 ping h3
  - h2 과 h3 사이 Ping Test 해보기
    - » h2 ping h3
  - Mininet 종료
    - » exit
  - Mininet 초기화
    - » sudo mn –c



Experiment

- **Experiment 1: Adding Flow in Flow Table**
  - h1 과 h3사이 Flow 설정하기
    - Mininet 활용하여 Star topology 구성
      - 1 Switch, 3 Host
        - » sudo mn --topo single,3 --mac --switch ovsk --controller remote



Experiment

■ Experiment 1: Adding Flow in Flow Table

- h1 과 h3사이 Flow 설정하기
  - Flow Table 확인해보기
    - » dpctl dump-flows
    - Flow Table에 추가되지 않은 것을 확인할 수 있다

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
mininet>
```

  - h1 과 h3 사이 Ping Test 해보기
    - » h1 ping h3
    - Flow Table에 Flow가 없으니 Ping이 전달되지 않는다.

```
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 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
^C
--- 10.0.0.3 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5121ms
pipe 4
```

Experiment

■ Experiment 1: Adding Flow in Flow Table

- h1 과 h3사이 Flow 설정하기
  - h1 과 h3 사이 통신이 가능하도록 Flow 추가하기
    - 1번 포트로 들어온 패킷 중 destination이 10.0.0.3 이면 3번포트로 보내라
      - » dpctl add-flow in_port=1,nw_dst=10.0.0.3,actions=output:3
    - 3번 포트로 들어온 패킷 중 destination이 10.0.0.1 이면 1번포트로 보내라
      - » dpctl add-flow in_port=3,nw_dst=10.0.0.1,actions=output:1

```
mininet> dpctl add-flow in_port=1,nw_dst=10.0.0.3,actions=output:3
*** s1 ------------------------------------------------------------------------
2021-08-25T06:31:19Z|00001|ofp_match|INFO|normalization changed ofp_match, details:
2021-08-25T06:31:19Z|00002|ofp_match|INFO| pre: in_port=1,nw_dst=10.0.0.3
2021-08-25T06:31:19Z|00003|ofp_match|INFO|post: in_port=1
mininet> dpctl add-flow in_port=3,nw_dst=10.0.0.1,actions=output:1
*** s1 ------------------------------------------------------------------------
2021-08-25T06:31:27Z|00001|ofp_match|INFO|normalization changed ofp_match, details:
2021-08-25T06:31:27Z|00002|ofp_match|INFO| pre: in_port=3,nw_dst=10.0.0.1
2021-08-25T06:31:27Z|00003|ofp_match|INFO|post: in_port=3
mininet> dpctl dump-flows
```

- Flow Table 확인하여 추가된 경로 검토하기
  - » dpctl dump-flows

```
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=14.574s, table=0, n_packets=0, n_bytes=0, in_port="s1-eth1" actions=out
put:"s1-eth3"
 cookie=0x0, duration=6.303s, table=0, n_packets=0, n_bytes=0, in_port="s1-eth3" actions=outp
ut:"s1-eth1"
```

Experiment

■ Experiment 1: Adding Flow in Flow Table

- h1 과 h3사이 Flow 설정하기
  - h1 과 h3 사이 Ping Test 해보기
    - Flow가 설정되어 Ping 주고받는게 가능해진다
      - » h1 ping h3

```
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.311 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.101 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.119 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.096 ms
```

  - h2 과 h3 사이 Ping Test 해보기
    - h2와 h3 사이에는 어떠한 Flow도 없기 때문에 Ping 주고받는게 불가능하다
      - » h2 ping h3

```
mininet> h2 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
From 10.0.0.2 icmp_seq=4 Destination Host Unreachable
```

Experiment

■ Experiment 1: Adding Flow in Flow Table

– h1 과 h3사이 Flow 설정하기

- Mininet 종료

» exit

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 16.610 seconds
ubuntu@ubuntu-VirtualBox:~$
```

- Mininet 초기화

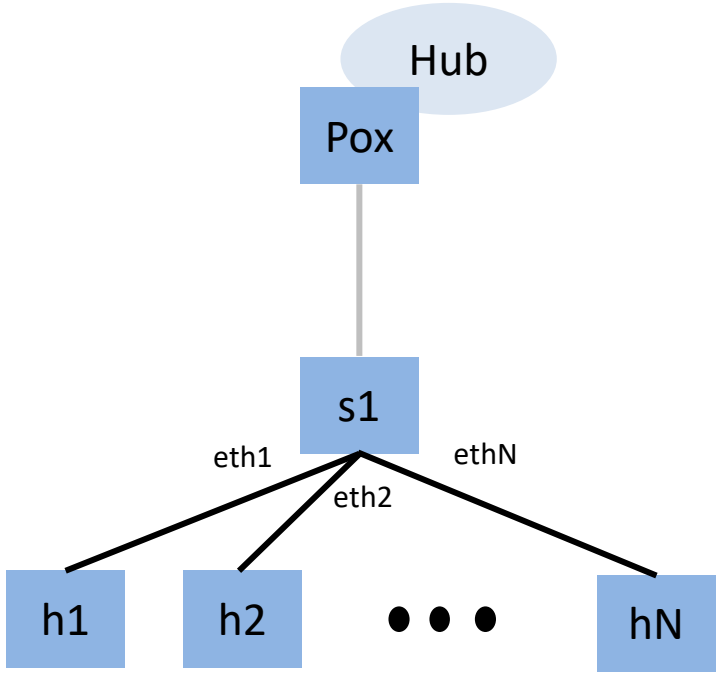
» sudo mn –c

```
ubuntu@ubuntu-VirtualBox:~$ sudo mn -c
[sudo] ubuntu의 암호:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd
ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflo
wd ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/nul
```

Experiment

■ Experiment 2: SDN-based Hub

- Pox Controller를 사용하여 Hub 구성해보기
  - Pox Controller에서 Hub.py 실행하기
    - » ./pox/pox.py forwarding.hub
  - Mininet 활용하여 Star topology 구성
    - 1 Switch, 3 Host
      - » sudo mn --topo single,3 --mac --switch ovsk --controller remote
  - Flow Table을 확인하고 분석해보기
    - » dpctl dump-flows
  - h1 과 h3 사이 Ping 2개 보내기
    - » h1 ping –c2 h3
    - 이때 xterm과 tcpdump를 사용하여 Interface Listening
      - » xterm h1 h2 h3
      - » tcpdump –XX –n –i hN–eth0
  - h2 과 h3 사이 Ping Test 해보기
    - » h2 ping h3
  - iperf를 활용하여 Host 수에 따른 Throughput 분석해보기
    - Host 수를 점점 늘려가면서 측정 (N=3,30,300)
      - » iperf h1 hN



Experiment

■ Experiment 2: SDN-based Hub

- Pox Controller를 사용하여 Hub 구성해보기
  - Pox Controller에서 Hub.py 실행하기
    - Hub.py: s1이 Hub로 동작하도록 하는 코드
      - » ./pox/pox.py forwarding.hub

```
ubuntu@ubuntu-VirtualBox:~$ ./pox/pox.py forwarding.hub
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et a
l.
INFO:forwarding.hub:Proactive hub running.
INFO:core:POX 0.3.0 (dart) is up.
```

  - Mininet 활용하여 Star topology 구성
    - 1 Switch, 3 Host
      - » sudo mn --topo single,3 --mac --switch ovsk --controller remote

```
ubuntu@ubuntu-VirtualBox:~$ sudo mn --topo single,3 --mac --s
witch ovsk --controller remote
[sudo] ubuntu의 암호:
*** 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
```

Experiment

- **Experiment 2: SDN-based Hub**
  - Pox Controller를 사용하여 Hub 구성해보기
    - Flow Table을 확인하고 분석해보기
      - » dpctl dump-flows

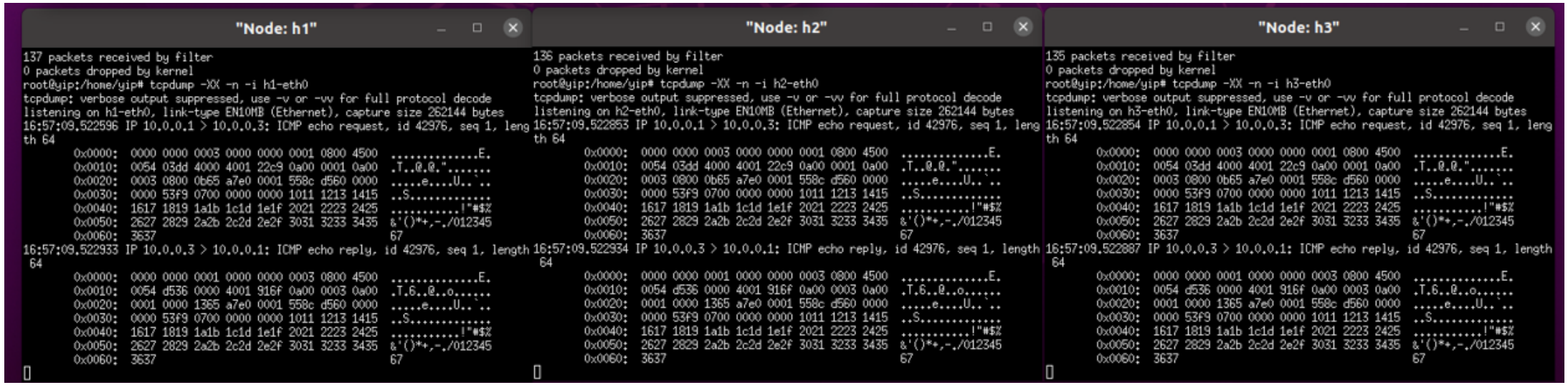
```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=32.035s, table=0, n_packets=23, n_bytes=1898, actions=FLOOD
mininet>
```

Experiment

- **Experiment 2: SDN-based Hub**
  - Pox Controller를 사용하여 Hub 구성해보기
    - h1 과 h3 사이 Ping 2개 보내기
      - » h1 ping –c2 h3
    - 이때 xterm과 tcpdump를 사용하여 Interface Listening
      - » xterm h1 h2 h3
      - » tcpdump –XX –n –i hN–eth0

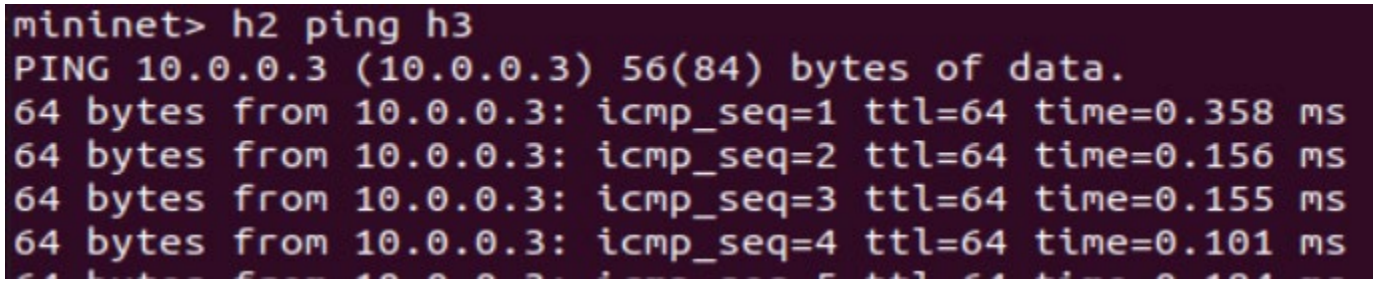
```
mininet> h1 ping -c2 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.324 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.106 ms
```

```
mininet> xterm h1 h2 h3
```



Experiment

- **Experiment 2: SDN-based Hub**
  - Pox Controller를 사용하여 Hub 구성해보기
    - h2 과 h3 사이 Ping Test 해보기
      - » h2 ping h3



Experiment

- **Experiment 2: SDN-based Hub**
  - Pox Controller를 사용하여 Hub 구성해보기
    - iperf를 활용하여 Host 수에 따른 Throughput 분석해보기
      - Host 수를 점점 늘려가면서 측정 (N=3,30,300)
        - » iperf h1 hN

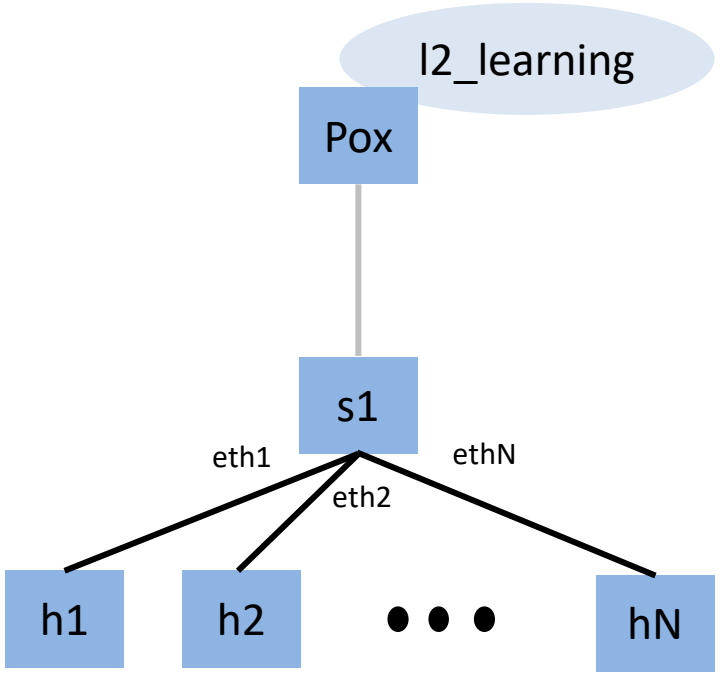
N=3

N=30

Experiment

■ Experiment 3: SDN-based L2 Switch

- Pox Controller를 사용하여 L2 Switch 구성해보기
  - Pox Controller에서 l2_learning.py 실행하기
    - ./pox/pox.py forwarding.l2_learning
  - Mininet 활용하여 Star topology 구성
    - 1 Switch, 3 Host
      - » sudo mn --topo single,3 --mac --switch ovsk --controller remote
  - Flow Table 확인해보기
    - » dpctl dump-flows
  - h1 과 h3 사이 Ping 2개 보내기
    - » h1 ping –c2 h3
    - 이때 xterm과 tcpdump를 사용하여 Interface Listening
      - » xterm h1 h2 h3
      - » tcpdump –XX –n –i hN–eth0
  - Flow Table 확인하고 분석해보기
    - » dpctl dump-flows
  - iperf를 활용하여 Host 수에 따른 Throughput 분석해보기
    - Host 수를 점점 늘려가면서 측정 (--topo single,N) (N=3,30,300)
      - » iperf h1 hN



Experiment

- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - Pox Controller에서 l2_learning.py 실행하기
      - ./pox/pox.py forwarding.l2_learning

```
ubuntu@ubuntu-VirtualBox:~$ ./pox/pox.py forwarding.l2_lear
ning
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et a
l.
INFO:core:POX 0.3.0 (dart) is up.
```

Experiment

- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - Mininet 활용하여 Star topology 구성
      - 1 Switch, 3 Host
        - » sudo mn --topo single,3 --mac --switch ovsk --controller remote

```
ubuntu@ubuntu-VirtualBox:~$ sudo mn --topo single,3 --mac --switch ovsk --controlle
r remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Experiment

- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - Flow Table 확인해보기
      - » dpctl dump-flows

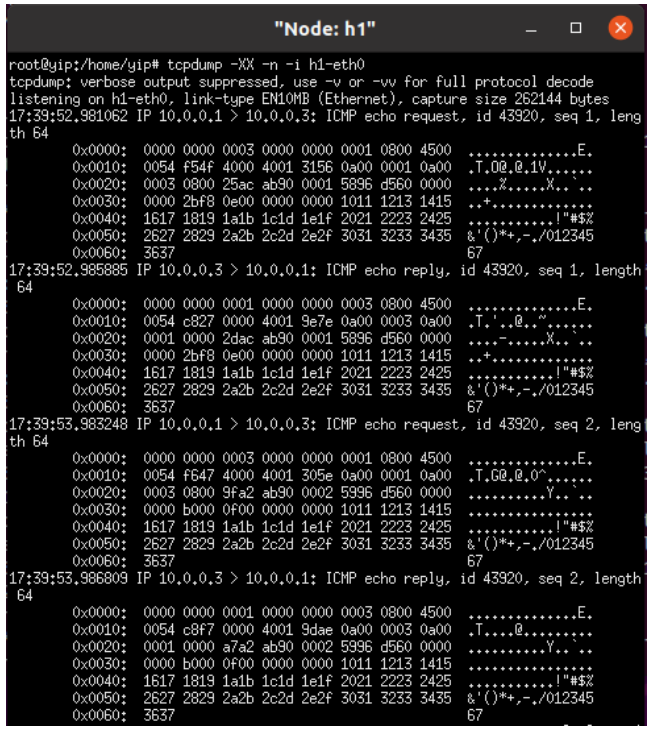
```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
mininet>
```

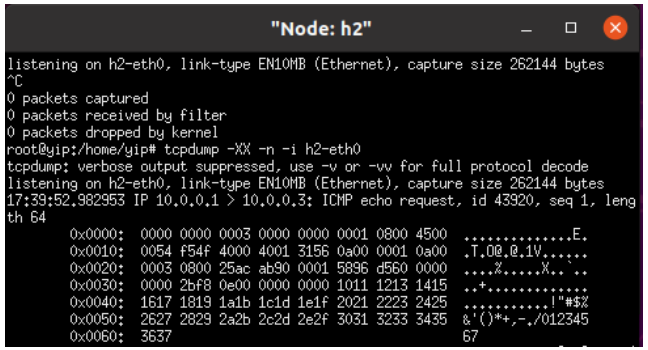
Experiment

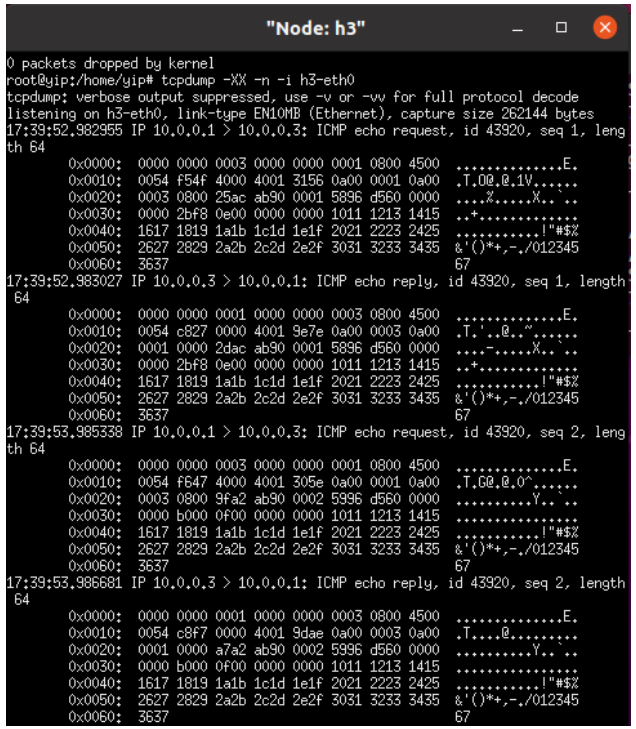
- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - h1 과 h3 사이 Ping 2개 보내기
      - » h1 ping –c2 h3
    - 이때 xterm과 tcpdump를 사용하여 Interface Listening
      - » xterm h1 h2 h3
      - » tcpdump –XX –n –i hN–eth0

```
mininet> h1 ping -c2 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=57.8 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.703 ms
```

```
mininet> xterm h1 h2 h3
```







Experiment

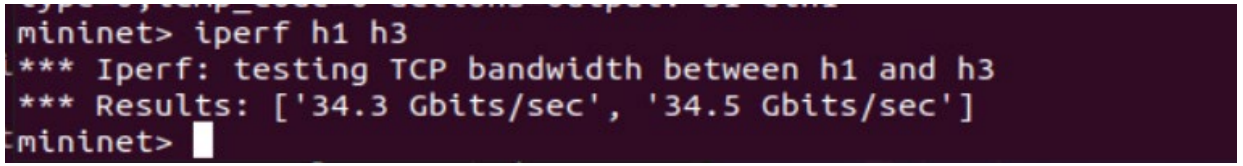
- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - Flow Table 확인하고 분석해보기
      - » dpctl dump-flows

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=2.188s, table=0, n_packets=2, n_bytes=196, idle_timeout=10, h
ard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_tci=0x0000,dl_src=00:00:
00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_
type=8,icmp_code=0 actions=output:"s1-eth3"
 cookie=0x0, duration=2.185s, table=0, n_packets=2, n_bytes=196, idle_timeout=10, h
ard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_tci=0x0000,dl_src=00:00:
00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_
type=0,icmp_code=0 actions=output:"s1-eth1"
mininet>
```

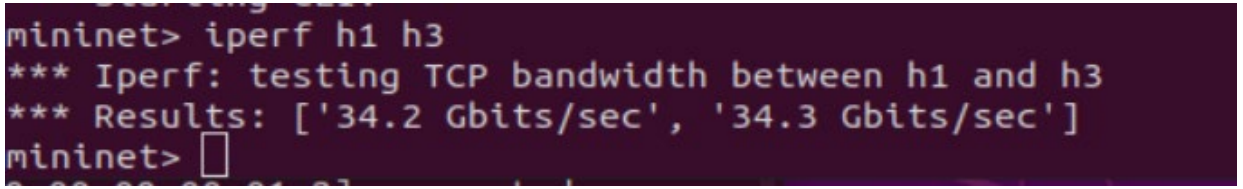
Experiment

- **Experiment 3: SDN-based L2 Switch**
  - Pox Controller를 사용하여 L2 Switch 구성해보기
    - Iperf를 활용하여 Host 수에 따른 Throughput 분석해보기
      - Host 수를 점점 늘려가면서 측정 (--topo single,N) (N=3,30,300)
        - » iperf h1 hN

N=3



N=30



Result Report

Result Report

- **Experiment 1**
  - Flow 추가 전의 Flow Table과 Ping Test 결과(H1 → H3) 캡처하여 첨부
  - Flow 추가 후의 Flow Table과 Ping Test 결과(H1 → H3, H2 → H3) 캡처하여 첨부

- **Experiment 2**
  - Flow Table 캡처하여 첨부
  - Ping Test 결과(H1 → H3, H2 → H3)결과 캡처하여 첨부
  - Interface Listening 결과(H1 → H3) 캡처하여 첨부
  - iperf 결과 (Host 수에 따른 Throughput 측정 결과) 캡처하여 첨부

- **Experiment 3**
  - Ping Test 결과(H1 → H3)와 캡처하여 첨부
  - Flow Table 캡처하여 첨부
  - Interface Listening 결과(H1 → H3) 캡처하여 첨부
  - iperf 결과 (Host 수에 따른 Throughput 측정 결과) 캡처하여 첨부

Pre Report

Pre Report

- **Switching Loop에 대해 조사해오기**
  - Switching Loop 개념
  - Switching Loop에 의한 현상
- **STP에 대해 조사해오기**
  - STP 개념
  - STP 동작
- **Static Routing에 대해 조사해오기**
  - Static Routing 개념
  - Static Routing의 장단점
- **Dynamic Routing에 대해 조사해오기**
  - Dynamic Routing 개념
  - Dynamic Routing의 장단점
- **Application-aware Routing에 대해 조사해오기**
  - Application-aware Routing 개념