Mininet을 이용한 SDN 구현 및 네트워크 관리 시스템

모바일 시스템 융합 및 실습

B689007

김덕용

목차

01 SDN

SDN(Software Define Network)이란?

-소프트웨어 프로그래밍을 통해 네트워크를 제어하는 차세대 네트워킹 기술

-OVS(가상 스위치) 가상화 기술로 기존의 하드웨어로 연결되어 있던 스위치를 소프트웨어로 구현함.

01 SDN

SDN 탄생 이유

1. 트래픽 패턴, 통신 환경의 급격한 변화

2. 네트워크 확장에 대한 어려움

3. 네트워크 복잡도

4. 벤더(Vender) 의존성

->좀 더 유연한 네트워크 아키텍처 필요!!

01 SDN

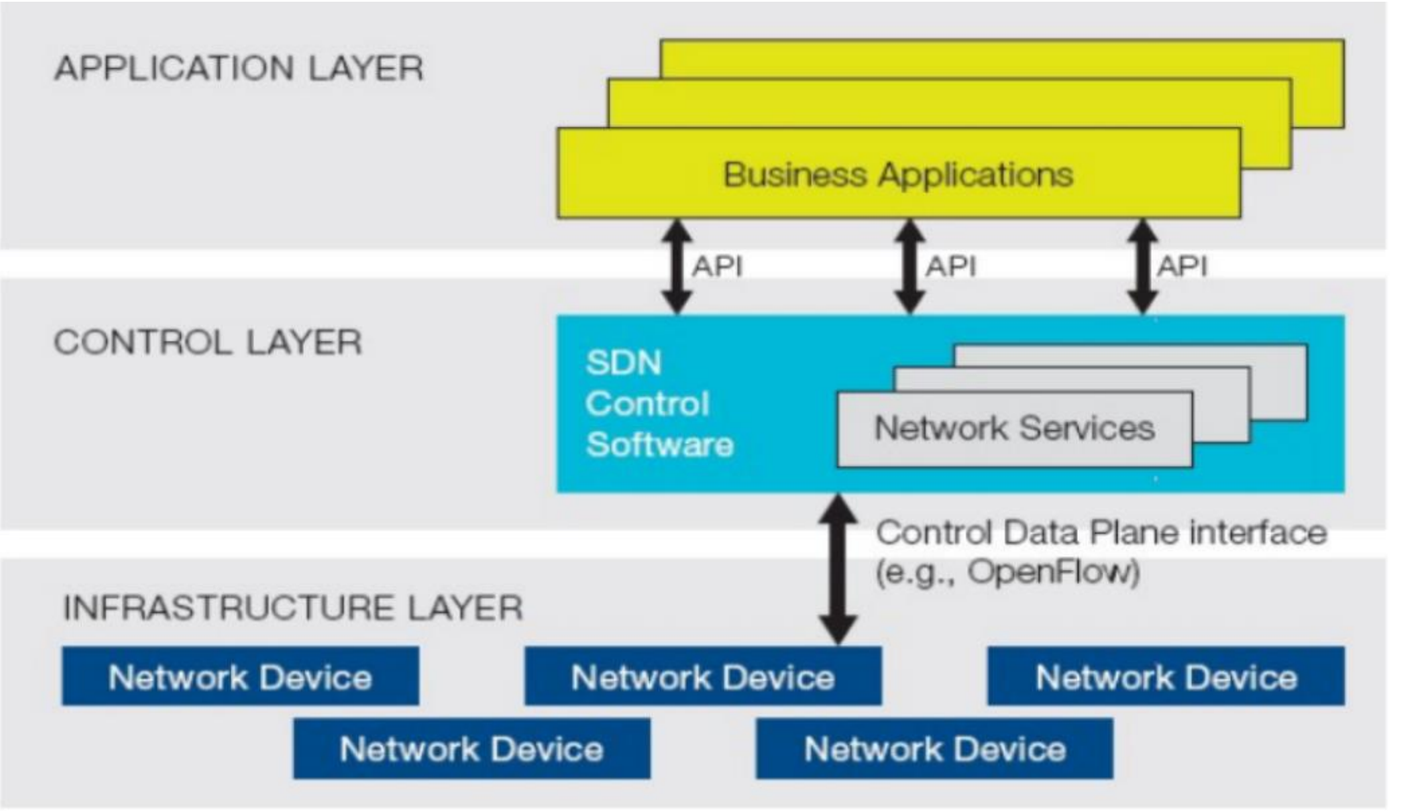
OpenFlow

-SDN을 구성하는 하나의 요소로, 외부에 있는 소프트웨어와 하드웨어 장비인 라우터,스위치 간 통신을 하게 해주는 표준 인터페이스

-소프트웨어와 네트워크 장비가 원활히 네트워킹 언어를 주고받을 수 있도록 도와주는 인터페이스

01 SDN

SDN 구조



네트워크 장비에서 하드웨어 기능과 소프트웨어 기능을 분리

02 SDN Network 통신 방식

01 Legacy Network

02. SDN Network

Legacy Network

1. Legacy network에서는 자신이 가지고 있지 않은 ARP 정보를 얻기 위해 기본적으로 Broadcast형태로 패킷 전송

2. Broadcast를 받은 스위치는 네트워크 환경 내에 있는 자신을 제외한 나머지 포트 전체에 패킷을 전송

문제점

-스위치가 많아지면 관리가 복잡해짐(Broadcast 방식으로 인해)

-네트워크의 고질적 문제인 loop가 발생할 가능성 매우 높음

02 SDN Network 통신 방식

01. Legacy Network

02 SDN Network

SDN Network

1. switch에 ARP Request Packet들어옴
2. (1) OpenFlow Protocol Matching Rule에 의해 match되는 패킷은 Match Rule대로 처리
   (2) Match Rule에 포함되어 있지 않은 패킷은 Controller에게 Packet In Message로 질의
3. (1) Controller가 상대방 Mac 주소를 모르고 있는 경우
   - Controller는 switch에게 Packet Out Message를 보냄 (broadcast 형식)
   -> 이때, Legacy Network와는 다르게 output action을 지정해 패킷을 어디로 보낼지 지정
   (loop가 발생하지 않도록 해줌)

   (2)Controller가 상대방 Mac 주소를 알고 있는 경우
   - Controller는 switch에게 Packet Out, Flow Modification Message를 보냄(Unicast 형식)
   -> 이미 상대방의 Mac주소를 알고 있기 때문
4. Flow Modification Message를 받은 switch는 Flow Rule을 만들어 Rule에 맞게 패킷을 전송

03 프로그램 구현

활용 기술

언어

- python
- bash shell script programming
- Rest API
- XML programming

사용 툴

- VirtualBox
- Vim 편집기
- Visual Studio Code
- Opendaylight DLUX
- curl

운영체제

- Linux Ubuntu 20.04.2 LTS

03 프로그램 구현

01 활용 기술

02. ODL Controller

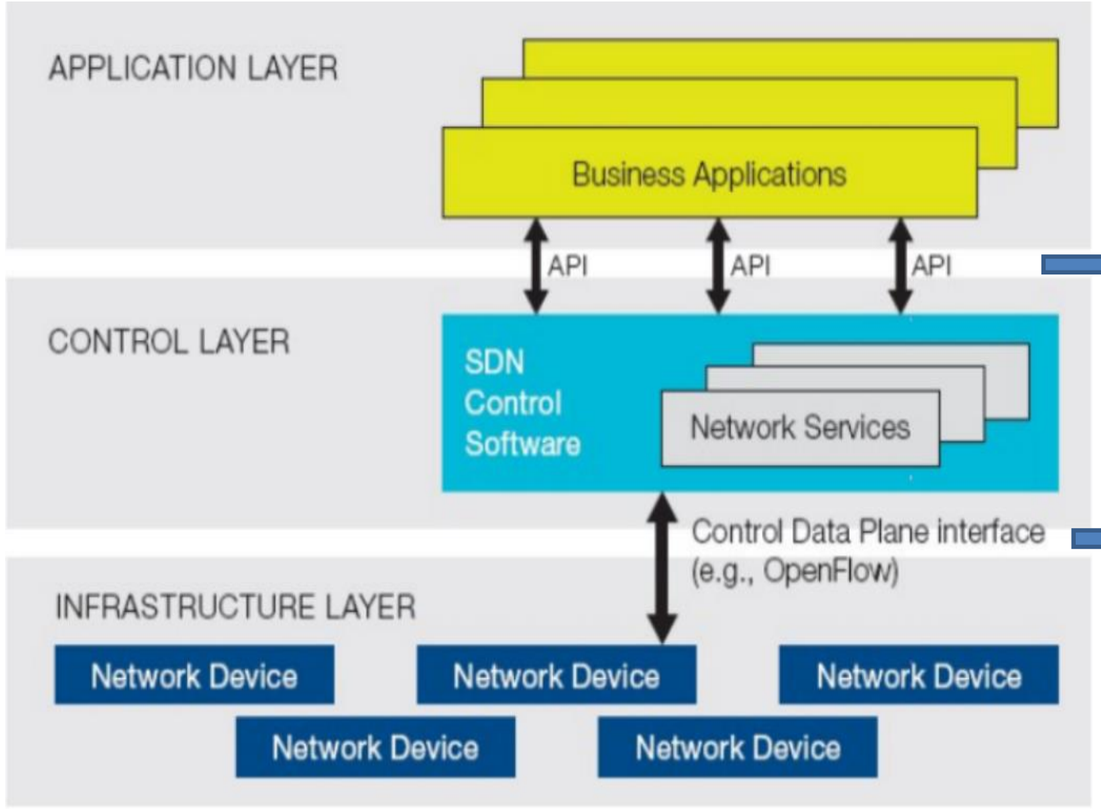
03. 네트워크 topology

04. 시나리오

05. 코드

06. GUI

활용 기술



Northbound API (Ex : REST API)

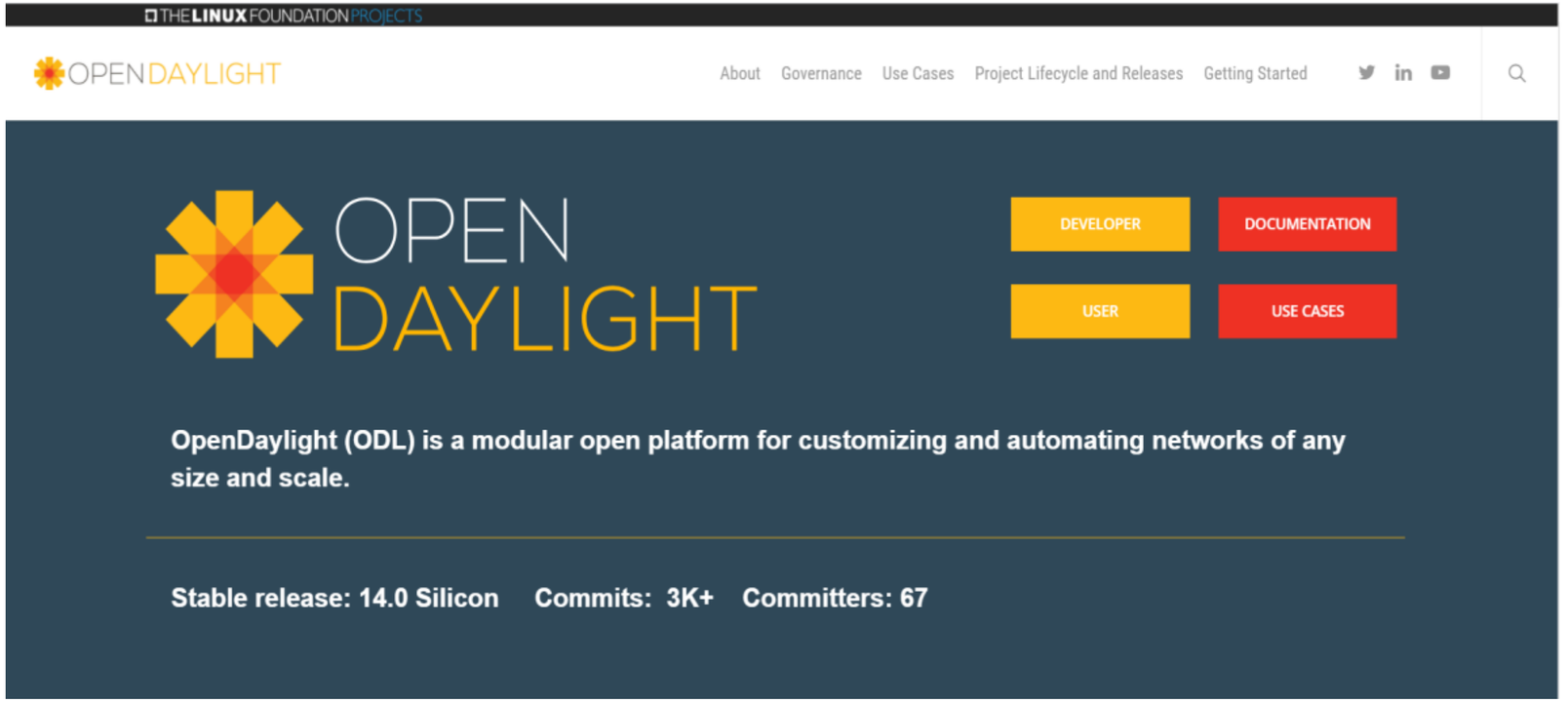
Southbound API (Ex : OpenFlow)

03 프로그램 구현

ODL Controller

SDN Controller : Opendaylight Controller 사용

Opendaylight release version : Oxygen



03 프로그램 구현

ODL Controller

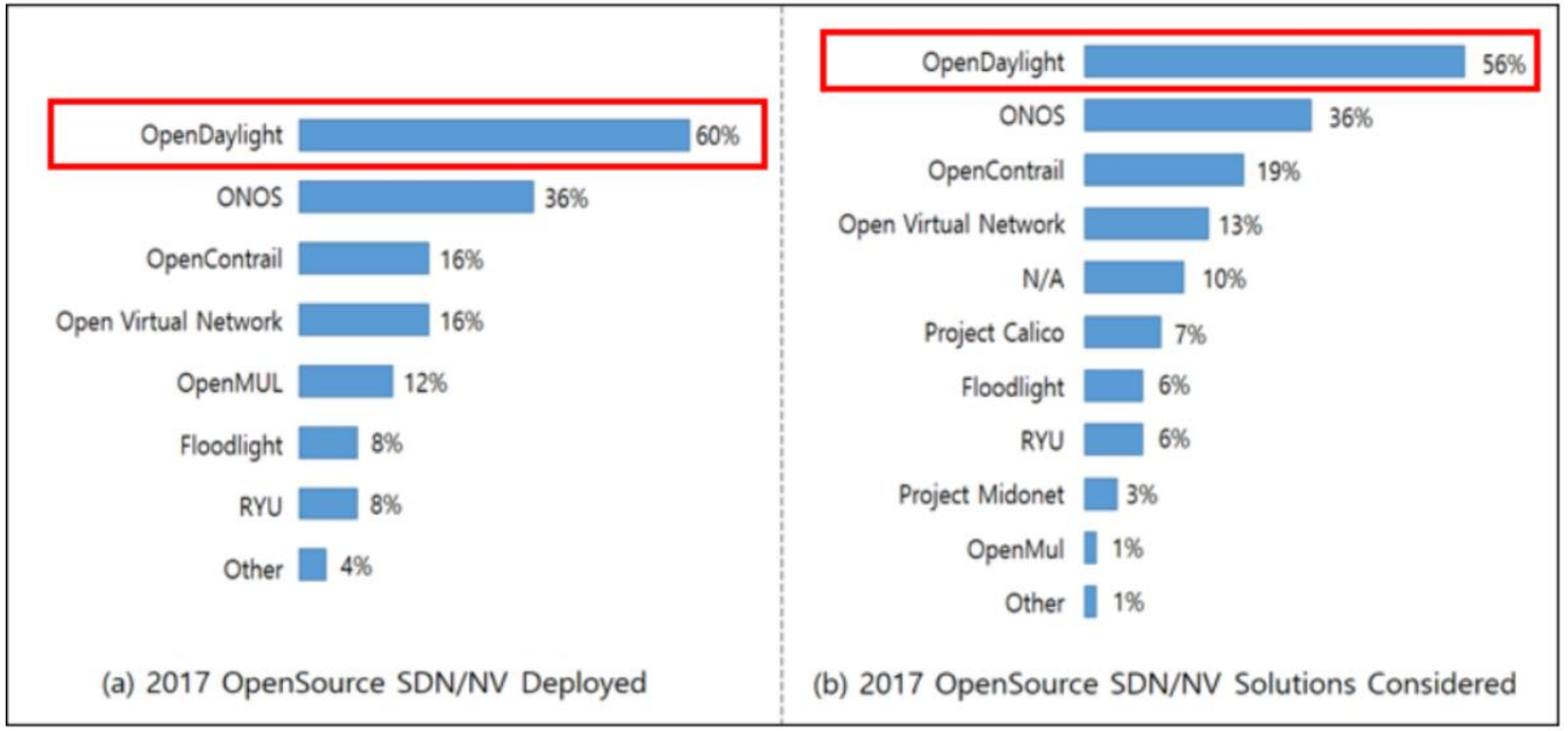
Opendaylight 선택 이유

- 프로젝트 멤버 대부분이 상용 네트워크 솔루션 제공업체인 만큼 SDN제어기 중 상용화를 통한 수요가 가장 많음

03 프로그램 구현

ODL Controller

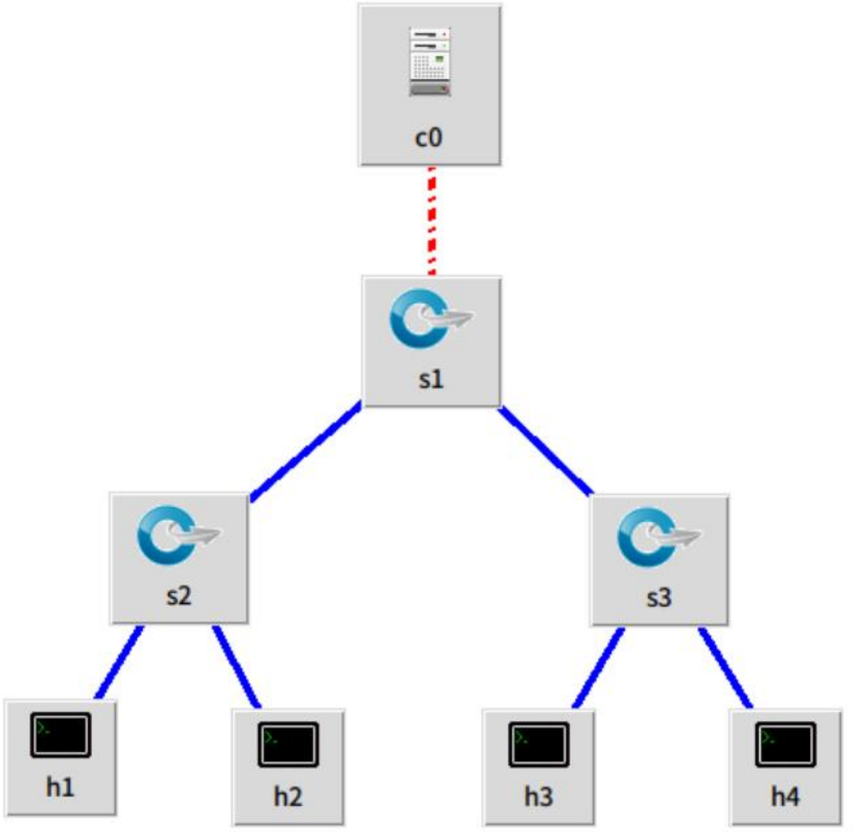
Opendaylight 시장 동향



03 프로그램 구현

네트워크 topology

Tree 형태 Topology



03 프로그램 구현

네트워크 topology

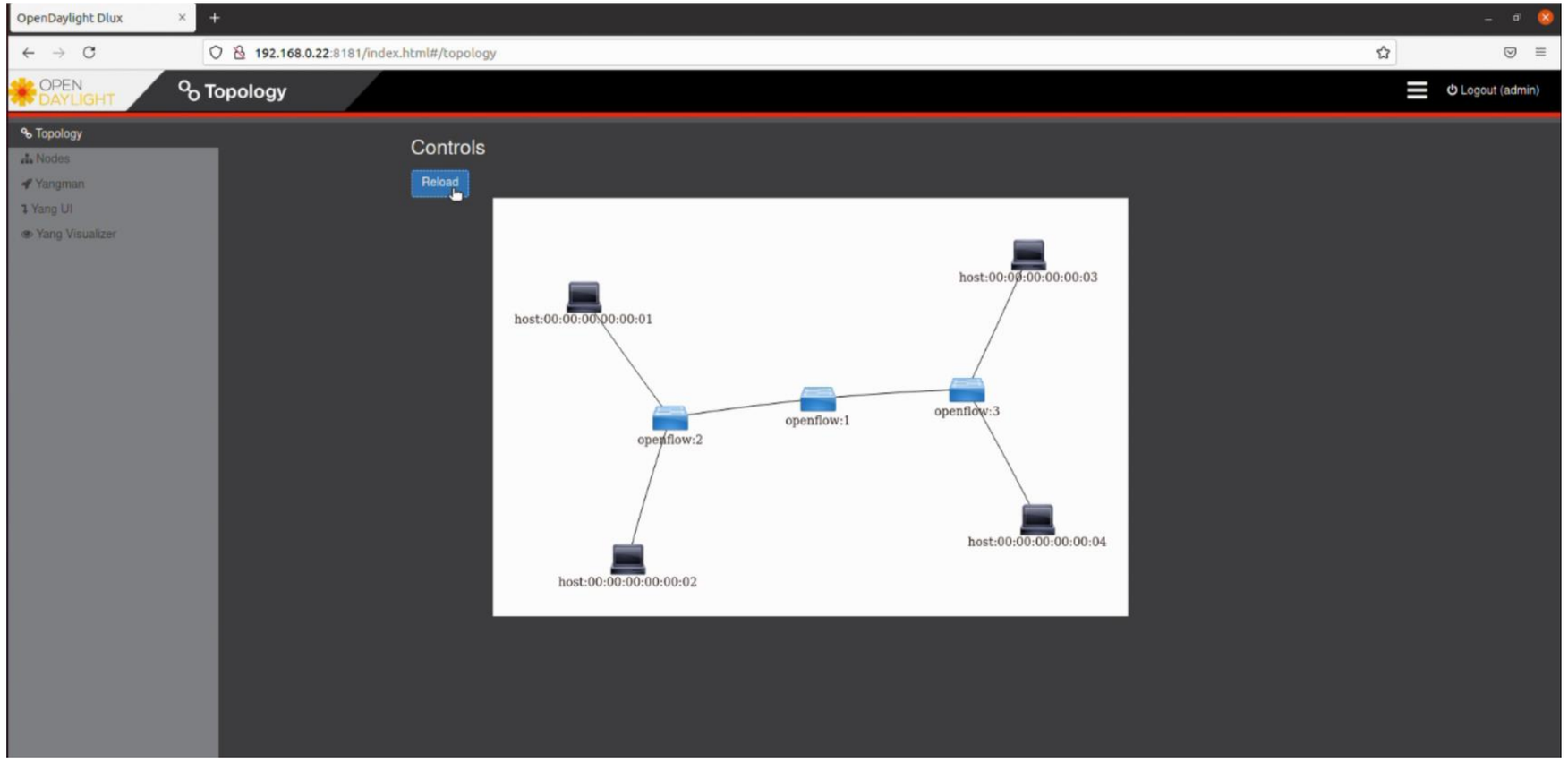
구현코드

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel, info

class TreeTopo(Topo):
    def __init__(self, **opts):
        super(TreeTopo, self).__init__(**opts)

        net = Mininet(controller=RemoteController)
        net.addController('co')

        switch1 = net.addSwitch('s1')
        switch2 = net.addSwitch('s2')
        switch3 = net.addSwitch('s3')

        h1 = net.addHost( 'h1', ip='10.0.0.1', mac='00:00:00:00:00:01' )
        h2 = net.addHost( 'h2', ip='10.0.0.2', mac='00:00:00:00:00:02' )
        h3 = net.addHost( 'h3', ip='10.0.0.3', mac='00:00:00:00:00:03' )
        h4 = net.addHost( 'h4', ip='10.0.0.4', mac='00:00:00:00:00:04' )

        net.addLink(switch1, switch2 )
        net.addLink(switch1, switch3)
        net.addLink(h1,switch2)
        net.addLink(h2,switch2)
        net.addLink(h3,switch3)
        net.addLink(h4,switch3)

        net.start()
        CLI(net)

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

03 프로그램 구현

네트워크 topology

OpendayLight DLUX



03 프로그램 구현

Flow Rule

switch,host의 연결 인터페이스 확인

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3153>
<Host h2: h2-eth0:10.0.0.2 pid=3155>
<Host h3: h3-eth0:10.0.0.3 pid=3157>
<Host h4: h4-eth0:10.0.0.4 pid=3159>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=3142>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=3145>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=3148>
<RemoteController co: 127.0.0.1:6653 pid=3134>
mininet>
```

```
mininet> links
s1-eth1<->s2-eth1 (OK OK)
s1-eth2<->s3-eth1 (OK OK)
h1-eth0<->s2-eth2 (OK OK)
h2-eth0<->s2-eth3 (OK OK)
h3-eth0<->s3-eth2 (OK OK)
h4-eth0<->s3-eth3 (OK OK)
```

03 프로그램 구현

Flow Rule

같은 스위치끼리만 연결(h1<->h2 / h3<->h4)

flow_kdy22.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>1000</priority>
    <flow-name>0</flow-name>
    <id>flow_kdy22</id>
    <table_id>0</table_id>
    <match>
        <in-port>2</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>3</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

flow_kdy33.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>1000</priority>
    <flow-name>2</flow-name>
    <id>flow_kdy33</id>
    <table_id>0</table_id>
    <match>
        <in-port>3</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>2</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

OVS 스위치2 Flow Rule 지정 (s2-eth2 <-> s2-eth3)

OVS 스위치3 Flow Rule 지정 (s3-eth2 <-> s3-eth3)

03 프로그램 구현

Flow Rule

Flow Rule 확인

```
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=26.051s, table=0, n_packets=1412, n_bytes=120105, priority=1000,in_port="s1-eth1" actions=output:"s1-eth2"
 cookie=0x0, duration=25.995s, table=0, n_packets=1412, n_bytes=120105, priority=1000,in_port="s1-eth2" actions=output:"s1-eth1"
 cookie=0xa, duration=6961.310s, table=0, n_packets=2, n_bytes=170, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=27.394s, table=0, n_packets=9, n_bytes=630, priority=1000,in_port="s2-eth2" actions=output:"s2-eth3"
 cookie=0x0, duration=27.335s, table=0, n_packets=9, n_bytes=630, priority=1000,in_port="s2-eth3" actions=output:"s2-eth2"
 cookie=0xa, duration=6962.752s, table=0, n_packets=2826, n_bytes=240365, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=28.395s, table=0, n_packets=9, n_bytes=630, priority=1000,in_port="s3-eth2" actions=output:"s3-eth3"
 cookie=0x0, duration=28.322s, table=0, n_packets=10, n_bytes=700, priority=1000,in_port="s3-eth3" actions=output:"s3-eth2"
 cookie=0xa, duration=6963.882s, table=0, n_packets=2826, n_bytes=240365, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$
```

03 프로그램 구현

Flow Rule

host2 <-> host4 연결(OVS 인터페이스 연결)

flow_kdy1.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>1000</priority>
    <flow-name>0</flow-name>
    <id>flow_kdy1</id>
    <table_id>0</table_id>
    <match>
        <in-port>1</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>2</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

flow_kdy2.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>1000</priority>
    <flow-name>2</flow-name>
    <id>flow_kdy2</id>
    <table_id>0</table_id>
    <match>
        <in-port>2</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>1</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

OVS 스위치1 Flow Rule 지정(s1-eth1 <-> s1-eth2)

03 프로그램 구현

Flow Rule

host2 <-> host4 연결(OVS 인터페이스 연결)

linear11.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>200</priority>
    <flow-name>0</flow-name>
    <id>linear11</id>
    <table_id>0</table_id>
    <match>
        <in-port>1</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>3</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

linear33.xml (~/ODL_flow_test/B689007) - VIM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
    <priority>200</priority>
    <flow-name>2</flow-name>
    <id>linear33</id>
    <table_id>0</table_id>
    <match>
        <in-port>3</in-port>
    </match>
    <instructions>
        <instruction>
        <order>0</order>
        <apply-actions>
            <action>
                <order>0</order>
                <output-action>
                    <output-node-connector>1</output-node-connector>
                </output-action>
            </action>
        </apply-actions>
        </instruction>
    </instructions>
</flow>
```

OVS 스위치2 Flow Rule 지정(s2-eth1 <-> s2-eth3)

OVS 스위치3 Flow Rule 추가(s3-eth1 <-> s3-eth3)

03 프로그램 구현

Flow Rule

Flow Rule 확인

```
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=12.324s, table=0, n_packets=2, n_bytes=170, priority=1000,in_port="s1-eth1" actions=output:"s1-eth2"
 cookie=0x0, duration=11.822s, table=0, n_packets=2, n_bytes=170, priority=1000,in_port="s1-eth2" actions=output:"s1-eth1"
 cookie=0xa, duration=8190.522s, table=0, n_packets=460, n_bytes=39114, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=13.597s, table=0, n_packets=6, n_bytes=510, priority=200,in_port="s2-eth1" actions=output:"s2-eth3"
 cookie=0x0, duration=13.019s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s2-eth3" actions=output:"s2-eth1"
 cookie=0xa, duration=8192.055s, table=0, n_packets=3088, n_bytes=262627, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=14.632s, table=0, n_packets=6, n_bytes=510, priority=200,in_port="s3-eth1" actions=output:"s3-eth3"
 cookie=0x0, duration=14.357s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s3-eth3" actions=output:"s3-eth1"
 cookie=0xa, duration=8193.309s, table=0, n_packets=3088, n_bytes=262627, priority=0 actions=CONTROLLER:65535
kimdeokyong@kimdeokyong:~/ODL_flow_test/B689007$
```

03 프로그램 구현

Flow Rule

REST API로 Flow Rule 전송(xml파일) - 같은 스위치끼리만 연결

flow_same_switch.sh (~/ODL_flow_test/B689007) - VIM

```
ODL_IP=192.168.0.22
ODL_PORT=8181

echo "----Push FLow Rule s1(port 1 <-> 2)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy1.xml' http://
$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy1
 -u admin:admin -v

echo "----Push FLow Rule s1(port 2 <-> 1)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy2.xml' http://
$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy2
 -u admin:admin -v

echo "----Push FLow Rule s2(port 2 <-> 3)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy22.xml' http:/
/$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/flow_kdy
22 -u admin:admin -v

echo "----Push FLow Rule s2(port 3 <-> 2)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy33.xml' http:/
/$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/flow_kdy
33 -u admin:admin -v

echo "----Push FLow Rule s3(port 2 <-> 3)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy22.xml' http:/
/$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/flow_kdy
22 -u admin:admin -v

echo "----Push FLow Rule s3(port 3 <-> 2)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy33.xml' http:/
/$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/flow_kdy
33 -u admin:admin -v
```

Flow Rule

URL

http://<controller-IP>:8181/restconf/config/opendaylight-inventory:nodes/node/<스위치명>/table/<table_id>/flow/<id> -u admin:admin

03 프로그램 구현

Flow Rule

REST API로 Flow Rule 전송(xml파일) - h2 <-> h4 연결

flow_h2_h4.sh (~/ODL_flow_test/B689007) - VIM

```
ODL_IP=192.168.0.22
ODL_PORT=8181

echo "----Push FLow Rule s1(port 1 <-> 2)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy1.xml' http://
$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy1
 -u admin:admin -v

echo "----Push FLow Rule s1(port 2 <-> 1)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'flow_kdy2.xml' http://
$ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy2
 -u admin:admin -v

echo "----Push FLow Rule s2(port 1 <-> 3)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'linear11.xml' http://$
ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/linear11 -
u admin:admin -v

echo "----Push FLow Rule s2(port 3 <-> 1)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'linear33.xml' http://$
ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/linear33 -
u admin:admin -v

echo "----Push FLow Rule s3(port 1 <-> 3)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'linear11.xml' http://$
ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/linear11 -
u admin:admin -v

echo "----Push FLow Rule s3(port 3 <-> 1)----"
curl -X PUT -H 'Content-Type:application/xml' -H 'Accept:application/xml' -d @'linear33.xml' http://$
ODL_IP:$ODL_PORT/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/linear33 -
u admin:admin -v
```

03 프로그램 구현

Flow Rule

REST API로 Flow Rule 전송(xml파일) - 같은 스위치 Flow Rule 삭제

del_flow_same_switch.sh (~/ODL_flow_test/B689007) - VIM

```
ODL_IP=192.168.0.22
ODL_PORT=8181

echo "----Delete FLow Rule s1(port 1 <-> 2)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy1 -u admin:admin -
v

echo "----Delete FLow Rule s1(port 2 <-> 1)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy2 -u admin:admin -
v



echo "----Delete FLow Rule s2(port2 <-> port3)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/flow_kdy22 -u admin:admin
-v

echo "----Delete FLow Rule s2(port3 <-> port2)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/flow_kdy33 -u admin:admin
-v

echo "----Push FLow Rule s3(port 2 <-> 3)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/flow_kdy22 -u admin:admin
-v

echo "----Push FLow Rule s3(port 3 <-> 2)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/flow_kdy33 -u admin:admin
-v
```

03 프로그램 구현

Flow Rule

REST API로 Flow Rule 전송(xml파일) - h2 <-> h4 Flow Rule 삭제

del_flow_h2_h4.sh (~/ODL_flow_test/B689007) - VIM

```
ODL_IP=192.168.0.22
ODL_PORT=8181

echo "----DROP FLow Rule s1(port 1 <-> 2)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy1 -u admin:admin -
v

echo "----DROP FLow Rule s1(port 2 <-> 1)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/flow_kdy2 -u admin:admin -
v

echo "----DROP FLow Rule s2(port 1 <-> 3)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/linear11 -u admin:admin -v

echo "----DROP FLow Rule s2(port 3 <-> 1)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:2/table/0/flow/linear33 -u admin:admin -v

echo "----DROP FLow Rule s3(port 1 <-> 3)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/linear11 -u admin:admin -v

echo "----DROP FLow Rule s3(port 3 <-> 1)----"
curl -X DELETE -H 'Content-Type:application/xml' -H 'Accept:application/xml' http://$ODL_IP:$ODL_PORT
/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/linear33 -u admin:admin -v
```

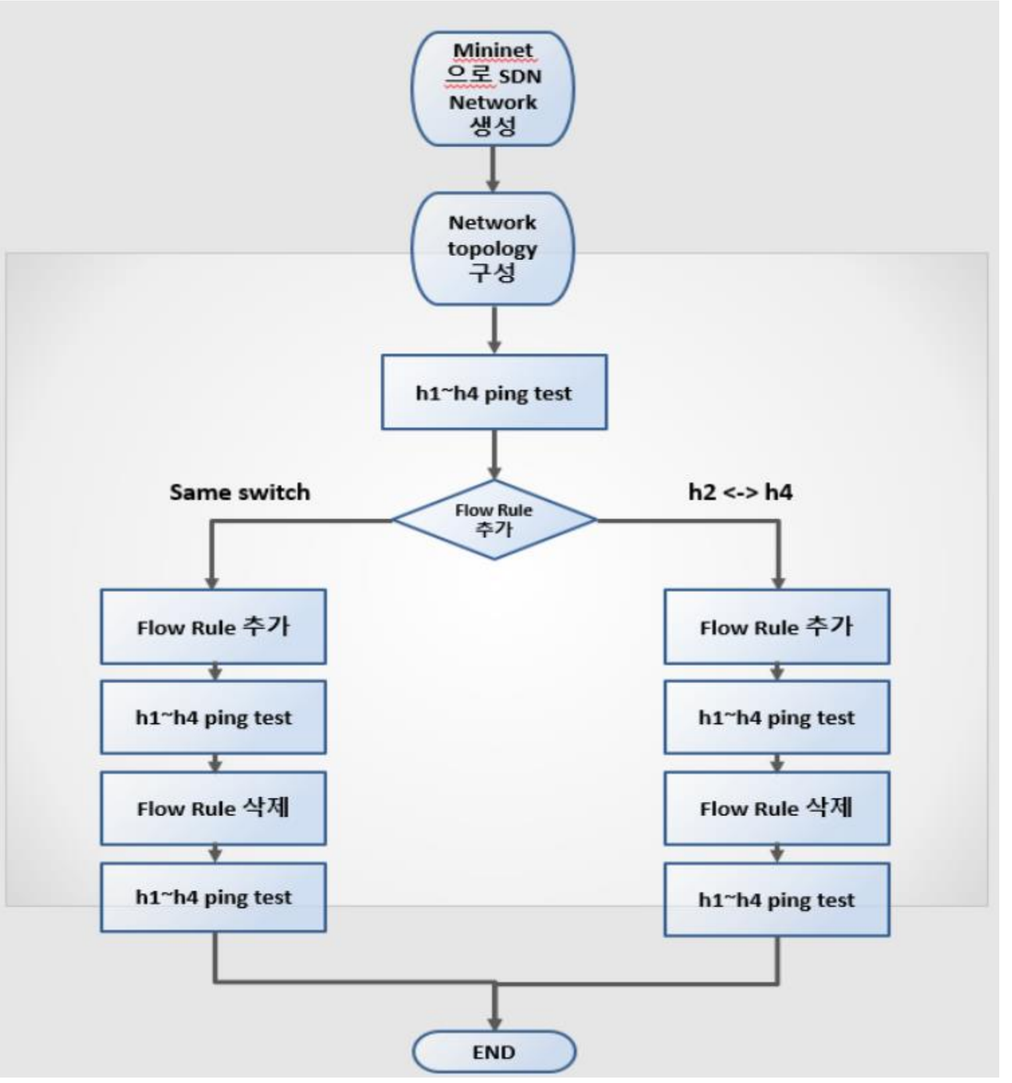
03 프로그램 구현

시나리오

1. Mininet으로 SDN Network 생성
2. SDN 제어기(Opendaylight) 연결해 Network Control
3. Network topology 구성
3. h1 ~ h4 ping test
4. Flow Rule(S2 & S3) 지정
5. Flow Rule 삭제
6. Flow Rule(S1 & S2 & S3) 지정
7. Flow Rule 삭제

03 프로그램 구현

시나리오



03 프로그램 구현

GUI

코드

```
from tkinter import *
import os
from os import *

window = Tk()
window.title("Network Control")
window.geometry("450x217")

def create():
    print("create Network Topology(Mininet)")
    os.system("gnome-terminal -e 'sudo python3 tree.py'")


def same_switch_Flow():
    print("Flow Rule Update")
    os.system("gnome-terminal -e './flow_same_switch.sh'")
def drop_same_switch_Flow():
    print("Drop Flow Rule")
    os.system("gnome-terminal -e './del_flow_same_switch.sh'")

def h2_h4_Flow():
    print("Flow Rule Update")
    os.system("gnome-terminal -e './flow_h2_h4.sh'")
def drop_h2_h4_Flow():
    print("Drop Flow Rule")
    os.system("gnome-terminal -e './del_flow_h2_h4.sh'")


title = Label(window,text="Network Management System",font=("맑은 고딕",17,"bold"))


topology = Label(window,text="Network Topology : ", font=("맑은 고딕", 13,"bold"))
flowName1 = Label(window,text="Same Switch        : ", font=("맑은 고딕", 13,"bold"))
flowName2 = Label(window,text="host2 <-> host4    : ", font=("맑은 고딕", 13,"bold"))
```

03 프로그램 구현

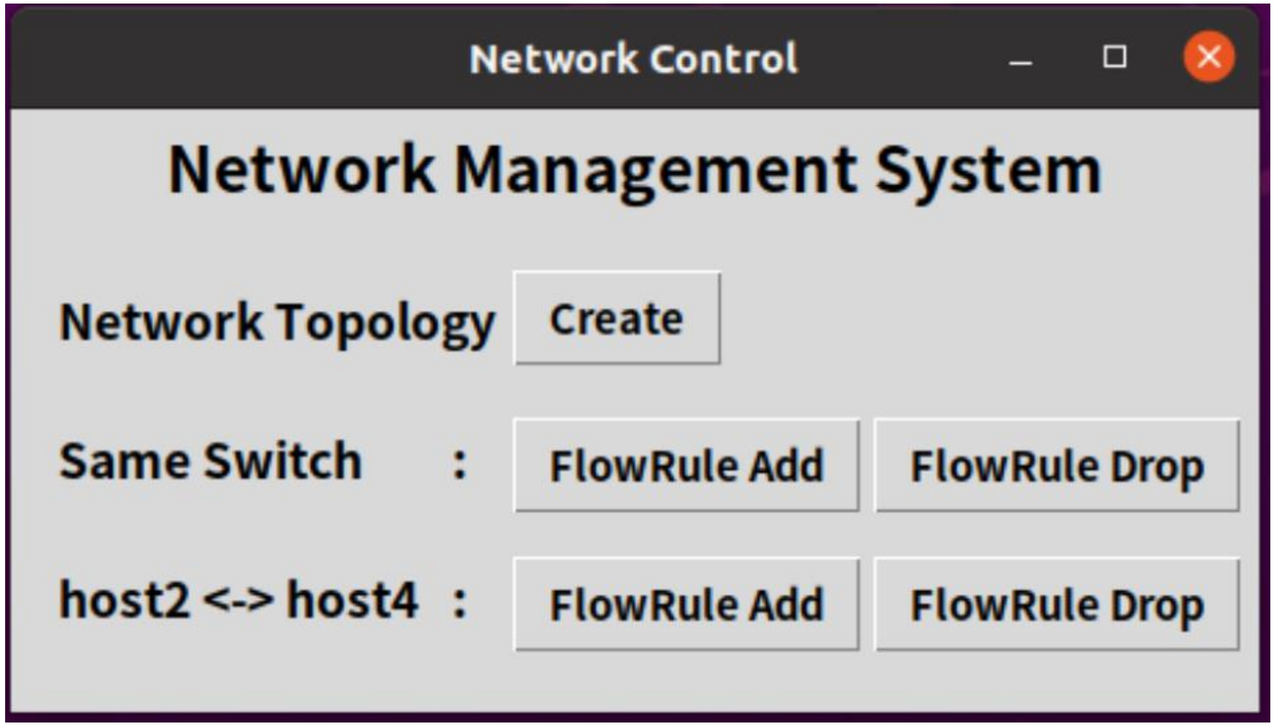
GUI

```
topology_create = Button(window,text="Create",font=("맑은 고딕",11,"bold"),command=create)

flowName1_add = Button(window,text="FlowRule Add",font=("맑은 고딕",11,"bold"),command=same_switch_Flow)
flowName1_drop = Button(window,text="FlowRule Drop",font=("맑은 고딕",11,"bold"),command=drop_same_switch_Flow)

flowName2_add = Button(window,text="FlowRule Add",font=("맑은 고딕",11,"bold"),command=h2_h4_Flow)
flowName2_drop = Button(window,text="FlowRule Drop",font=("맑은 고딕",11,"bold"),command=drop_h2_h4_Flow)


topology_create.place(x=180,y=57)
flowName1_add.place(x=180,y=110)
flowName1_drop.place(x=310,y=110)
flowName2_add.place(x=180,y=160)
flowName2_drop.place(x=310,y=160)

title.place(relx=0.5,y=20,anchor="center")
topology.place(x=15,y=60)
flowName1.place(x=15,y=110)
flowName2.place(x=15,y=160)

window.mainloop()
```

03 프로그램 구현

GUI

GUI를 이용한 SDN Network Control



결론

수업 때 배운 python과 shell script 언어를 활용해 배웠던 내용을 확실히 습득할 수 있었고, 추가로 네트워크 구현하고 직접 관리하는 것을 만들면서 학교 다니면서 배웠던 네트워크의 구조나 동작 형태를 전체적으로 다시 정리할 수 있었다.

네트워크를 구성하면서 Rest API와 XML을 사용하면서 네트워크에서 웹을 통해 데이터를 전달할 수 있는 것을 공부하게 됐다.

이번 프로젝트를 통해 실력이 많이 향상한 것 같다.

04 Q&A

Q&A

감사합니다

B689007 김덕용