

```
sudo apt update
```

```
sudo apt install python3 python3-pip xterm iperf hping3 net-tools  
wireshark apache2-utils curl
```

```
sudo apt install mininet
```

```
sudo pip3 install ryu
```

```
sudo pip3 install mininet
```

```
sudo cp /usr/bin/python3 /usr/bin/python
```

```
ryu-manager --version
```

```
sudo mn -version
```

```
sudo apt-get install openswitch-switch ( meter table )
```

```
{
  "type": "GROUP_MOD",
  "dpid": 2,
  "group_id": 50,
  "command": "ADD",
  "type": "SELECT",
  "buckets": [
    {
      "actions": [
        {
          "type": "OUTPUT",
          "port": 1
        }
      ]
    },
    {
      "actions": [
        {
          "type": "OUTPUT",
          "port": 3
        }
      ]
    }
  ]
}
```

```
{
  "type": "GROUP_MOD",
  "dpid": 2,
  "group_id": 50,
  "command": "ADD",
  "type": "ALL",
  "buckets": [
    {
      "actions": [
        {
          "type": "OUTPUT",
          "port": 1
        }
      ]
    },
    {
      "actions": [
        {
          "type": "OUTPUT",
          "port": 3
        }
      ]
    }
  ]
}
```

การกำหนดค่า "type" ใน context ของ GROUP table ขึ้นอยู่กับว่าคุณต้องการกลุ่มนั้นจะทำงานกับ Flow entries ของ Switches ในเครือข่ายอย่างไร:

"type": "SELECT": หมายถึงว่ากลุ่มนี้จะทำงานร่วมกับ Flow entries ที่ถูกเลือก (selected) โดยมักจะใช้กับ Flow entries ที่ได้รับการเลือกเฉพาะ (matched) ตามเงื่อนไขที่กำหนดใน Flow Mod messages หรือ OpenFlow messages อื่นๆ ที่เกี่ยวข้อง.

"type": "ALL": หมายถึงว่ากลุ่มนี้จะทำงานร่วมกับทุก Flow entries ที่มีอยู่ใน Switch โดยไม่สนใจเงื่อนไขที่เลือกเฉพาะ.

ดังนั้น, ถ้าคุณต้องการให้กลุ่มทำงานร่วมกับ Flow entries ที่ถูกเลือก (SELECTED) เท่านั้น ให้ใช้ "type": "SELECT" แต่ถ้าคุณต้องการให้กลุ่มทำงานร่วมกับทุก Flow entries ที่มีอยู่ใน Switch โดยไม่สนใจการเลือกเฉพาะ ให้ใช้ "type": "ALL" ตามความต้องการของระบบของคุณ.

```

14 from mininet.topo import Topo
15 from mininet.net import Mininet
16 from mininet.log import setLogLevel
17 from mininet.cli import CLI
18 from mininet.node import RemoteController
19 from time import sleep
20
21
22 class SingleSwitchTopo(Topo):
23     "Single switch connected to n hosts."
24     def build(self):
25         s1 = self.addSwitch('s1')
26         s2 = self.addSwitch('s2')
27
28         h1 = self.addHost('h1', mac="00:00:00:00:11:11", ip="192.168.1.1/24")
29         h2 = self.addHost('h2', mac="00:00:00:00:11:12", ip="192.168.1.2/24")
30
31         self.addLink(h1, s1, 1, 3)
32         self.addLink(h2, s2)
33
34         self.addLink(s1, s2)
35
36
37 if __name__ == '__main__':
38     setLogLevel('info')
39     topo = SingleSwitchTopo()
40     c1 = RemoteController('c1', ip='127.0.0.1')
41     net = Mininet(topo=topo, controller=c1)
42     net.start()
43     sleep(5)
44     net.pingAll()
45
46     # get the host objects
47     h1 = net.get('h1')
48     h2 = net.get('h2')
49     h1.cmd('iperf -s &')
50     result = h2.cmd('iperf -c 192.168.1.1')
51     print(result)
52     CLI(net)
53     net.stop()
54

```

self.addLink(h1, s1, 1, 3) คือ h1 จาก port 1 ไปยัง s1 port 3

คำสั่งใน mininet

net: ดูข้อมูลทั่วไปเกี่ยวกับเครือข่าย, รวมถึงรายชื่อของโหนดทั้งหมด

nodes: แสดงรายชื่อของโหนดทั้งหมดในเครือข่าย

links: แสดงรายละเอียดเกี่ยวกับลิงค์ทั้งหมดในเครือข่าย

dump : แสดงข้อมูลสเปคตรงกับการสร้างทุกๆ โหนด ดู IP ของแต่ละอัน

pingall : ทดสอบการเชื่อมต่อระหว่างโหนดทั้งหมดในเครือข่าย

iperf : ใช้ในการทดสอบผลลัพธ์การทำงานของเครือข่าย

CLI : เปิดหน้าต่างคำสั่ง CLI สำหรับการจัดการโหนดและลิงค์ในเครือข่าย

Exit หรือ quit: ใช้เพื่อออกจากระบบ Mininet

help: แสดงรายชื่อคำสั่งทั้งหมดที่ใช้ได้

sudo mn -c ทำการเคลีย mininet

```
1  {
2      "dpid": 1,
3      "table_id": 30,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "eth_type": 2048,
9          "ipv4_dst": "10.1.1.5",
10         "ip_proto": 1
11     },
12     "actions": [
13         {
14             "type": "DROP"
15         }
16     ]
17 }
```

"ip_proto": 1 = ICMP

6 = TCP (Transmission Control Protocol)

17 = UDP (User Datagram Protocol)

2 = IGMP (Internet Group Management Protocol)

4 = IPv4 encapsulation

41 = IPv6 encapsulation

Add Group Table

```
ofctl > sniffer > {} flow1.json > ...
```

```
1  {
2    "dpid": 2,
3    "table_id": 0,
4    "idle_timeout": 0,
5    "hard_timeout": 0,
6    "priority": 100,
7    "match": {
8      "in_port": 3
9    },
10   "actions": [
11     {
12       "type": "GROUP",
13       "group_id": 50
14     }
15   ]
16 }
```

```
ofctl > sniffer > {} flow2.json > ...
```

```
1  {
2    "dpid": 2,
3    "table_id": 0,
4    "idle_timeout": 0,
5    "hard_timeout": 0,
6    "priority": 100,
7    "match": {
8      "in_port": 2
9    },
10   "actions": [
11     {
12       "type": "GROUP",
13       "group_id": 51
14     }
15   ]
16 }
```

```
ofctl > sniffer > {} group50.json > ...
```

```
1  {
2    "dpid": 2,
3    "type": "ALL",
4    "group_id": 50,
5    "buckets": [
6      {
7        "actions": [
8          {
9            "type": "OUTPUT",
10             "port": 1
11          }
12        ],
13      },
14      {
15        "actions": [
16          {
17            "type": "OUTPUT",
18            "port": 2
19          }
20        ]
21      }
22    ]
23  }
24 }
```

```
ofctl > sniffer > {} group51.json > ...
```

```
1  {
2    "dpid": 2,
3    "type": "ALL",
4    "group_id": 51,
5    "buckets": [
6      {
7        "actions": [
8          {
9            "type": "OUTPUT",
10             "port": 1
11          }
12        ],
13      },
14      {
15        "actions": [
16          {
17            "type": "OUTPUT",
18            "port": 3
19          }
20        ]
21      }
22    ]
23  }
```

Meter Table

```
ofctl > meter > {} addmeter.json > ...
1  {
2    "dpid": 1,
3    "flags": "KBPS",
4    "meter_id": 1,
5    "bands": [
6      {
7        "type": "DROP",
8        "rate": 1000
9      }
10   ]
11 }
```

```
ofctl > meter > {} switch_arp.json > ...
1  {
2    "dpid": 1,
3    "table_id": 0,
4    "idle_timeout": 0,
5    "hard_timeout": 0,
6    "priority": 100,
7    "match": {
8      "dl_dst": "ff:ff:ff:ff:ff:ff"
9    },
10   "actions": [
11     {
12       "type": "OUTPUT",
13       "port": 4294967291
14     }
15   ]
16 }
17
```

```
1  {
2    "dpid": 1,
3    "table_id": 0,
4    "idle_timeout": 0,
5    "hard_timeout": 0,
6    "priority": 100,
7    "match": {
8      "dl_dst": "00:00:00:00:00:01"
9    },
10   "actions": [
11     {
12       "type": "METER",
13       "meter_id": 1
14     },
15     {
16       "type": "OUTPUT",
17       "port": 1
18     }
19   ]
20 }
21
```

```
1  {
2    "dpid": 1,
3    "table_id": 0,
4    "idle_timeout": 0,
5    "hard_timeout": 0,
6    "priority": 100,
7    "match": {
8      "dl_dst": "00:00:00:00:00:02"
9    },
10   "actions": [
11     {
12       "type": "OUTPUT",
13       "port": 2
14     }
15   ]
16 }
17
```

Topo.py (GroupTable)

```
ofctl > lb >  topo.py > ...
```

```

1  #!/usr/bin/python
2
3
4  """Grouptable example
5
6      | | | | | | | | | | Switch2 ----switch4
7      | | | | | | | | | | /
8  h1 ---Switch1 | | | | | | | | | | Switch5-----h2
9      | | | | | | | | | | \
10     | | | | | | | | | | -----Switch3 -----
11
12
13
14  # static arp entry addition
15
16  h1 arp -s 192.168.1.2 00:00:00:00:00:02
17  h2 arp -s 192.168.1.1 00:00:00:00:00:01
18
19
20
21  ryu stuff:
22
23  ryu-manager group_table_lb.py
24
25  """
26
27  from mininet.topo import Topo
28  from mininet.net import Mininet
29  from mininet.log import setLogLevel
30  from mininet.cli import CLI
31  from mininet.node import OVSSwitch, Controller, RemoteController
32  from time import sleep
33
34

```

```

33
34
35 class SingleSwitchTopo(Topo):
36     "Single switch connected to n hosts."
37     def build(self):
38         s1 = self.addSwitch('s1', protocols='OpenFlow13')
39         s2 = self.addSwitch('s2', protocols='OpenFlow13')
40         s3 = self.addSwitch('s3', protocols='OpenFlow13')
41         s4 = self.addSwitch('s4', protocols='OpenFlow13')
42         s5 = self.addSwitch('s5', protocols='OpenFlow13')
43
44         h1 = self.addHost('h1', mac="00:00:00:00:00:01", ip="192.168.1.1/24")
45         h2 = self.addHost('h2', mac="00:00:00:00:00:02", ip="192.168.1.2/24")
46
47         self.addLink(s1,s2,1,1)
48         self.addLink(s1,s3,2,1)
49         self.addLink(s1,h1,3,1)
50
51
52         self.addLink(s3,s5,2,2)
53         self.addLink(s4,s2,1,2)
54         self.addLink(s4,s5,2,1)
55
56         self.addLink(s5,h2,3,1)
57
58 if __name__ == '__main__':
59     setLogLevel('info')
60     topo = SingleSwitchTopo()
61     c1 = RemoteController('c1', ip='127.0.0.1')
62     net = Mininet(topo=topo, controller=c1)
63     net.start()
64     #sleep(5)
65     #print("Topology is up, lets ping")
66     #net.pingAll()
67     CLI(net)
68     net.stop()
69

```

S1_flow1.json

```

ofctl > lb > {} s1_flow1.json > ...
1  {
2      "dpid": 1,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 1
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 3
14         }
15     ]
16 }

```

s2_flow2.json

```

ofctl > lb > {} s1_flow2.json > ...
1  {
2      "dpid": 1,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 2
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 3
14         }
15     ]
16 }

```

S1_flow3.json

```

ofctl > lb > {} s1_flow3.json > ...
1  {
2      "dpid": 1,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 3
9      },
10     "actions": [
11         {
12             "type": "GROUP",
13             "group_id": 50
14         }
15     ]
16 }

```

s1_group50.json

```

ofctl > lb > {} s1_group50.json > ...
1  {
2      "dpid": 1,
3      "type": "SELECT",
4      "group_id": 50,
5      "buckets": [
6          {
7              "weight": 50,
8              "actions": [
9                  {
10                     "type": "OUTPUT",
11                     "port": 1
12                 }
13             ]
14         },
15         {
16             "weight": 50,
17             "actions": [
18                 {
19                     "type": "OUTPUT",
20                     "port": 2
21                 }
22             ]
23         }
24     ]
25 }
26

```


S2_flow1.json

```

ofctl > lb > {} s2_flow1.json > ...
1  {
2      "dpid": 2,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 1
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 2
14         }
15     ]
16 }

```

S2_flow2.json

```

ofctl > lb > {} s2_flow2.json > ...
1  {
2      "dpid": 2,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 2
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 1
14         }
15     ]
16 }

```

S3_flow3.json

```

ofctl > lb > {} s3_flow2.json > ...
1  {
2      "dpid": 3,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 2
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 1
14         }
15     ]
16 }

```

S4_flow1.json

```

ofctl > lb > {} s4_flow1.json > ...
1  {
2      "dpid": 4,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 1
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 2
14         }
15     ]
16 }

```

S4_flow2.json

```

ofctl > lb > {} s4_flow2.json > ...
1  {
2      "dpid": 4,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 2
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 1
14         }
15     ]
16 }

```

s5_flow1.json

```

ofctl > lb > {} s5_flow1.json > ...
1  {
2      "dpid": 5,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 1
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 3
14         }
15     ]
16 }

```

S5_flow2.json

```

ofctl > lb > {} s5_flow2.json > ...
1  {
2      "dpid": 5,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 2
9      },
10     "actions": [
11         {
12             "type": "OUTPUT",
13             "port": 3
14         }
15     ]
16 }

```

s5_flow3.json

```

ofctl > lb > {} s5_flow3.json > ...
1  {
2      "dpid": 5,
3      "table_id": 0,
4      "idle_timeout": 0,
5      "hard_timeout": 0,
6      "priority": 100,
7      "match": {
8          "in_port": 3
9      },
10     "actions": [
11         {
12             "type": "GROUP",
13             "group_id": 51
14         }
15     ]
16 }

```

S5_group51.json

```
ofctl > lb > {-} s5_group51.json > ...
1  {
2      "dpid": 5,
3      "type": "SELECT",
4      "group_id": 51,
5      "buckets": [
6          {
7              "weight": 50,
8              "actions": [
9                  {
10                     "type": "OUTPUT",
11                     "port": 1
12                 }
13             ]
14         },
15         {
16             "weight": 50,
17             "actions": [
18                 {
19                     "type": "OUTPUT",
20                     "port": 2
21                 }
22             ]
23         }
24     ]
25 }
```

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s2_flow1.json'

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s2_flow2.json'

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s3_flow1.json'

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s3_flow2.json'

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s4_flow1.json'

curl -X POST http://localhost:8080/stats/flowentry/add -d '@s4_flow2.json'

curl -X POST http://localhost:8080/stats/groupentry/add -d '@s5_group51.json'

```
curl -X POST http://localhost:8080/stats/groupentry/add -d '@s1_group50.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s1_flow1.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s1_flow2.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s1_flow3.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s5_flow1.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s5_flow2.json'
```

```
curl -X POST http://localhost:8080/stats/flowentry/add -d '@s5_flow3.json'
```

การติดตั้ง FTP Server

```
sudo apt-get install vsftpd
```

```
sudo mv /etc/vsftpd.conf /etc/vsftpd.conf_orig
```

```
sudo nano /etc/vsftpd.conf
```

config in vsftpd.conf

```
listen=NO
```

```
listen_ipv6=YES
```

```
anonymous_enable=NO
```

```
local_enable=YES
```

```
write_enable=YES
```

```
local_umask=022
```

```
dirmessage_enable=YES
```

```
use_localtime=YES
```

```
xferlog_enable=YES
```

```
connect_from_port_20=YES
```

```
chroot_local_user=YES
```

```
secure_chroot_dir=/var/run/vsftpd/empty
```

```
pam_service_name=vsftpd
```

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
```

```
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

```
ssl_enable=NO
```

pasv_enable=Yes

pasv_min_port=10000

pasv_max_port=10100

allow_writeable_chroot=YES

sudo ufw allow from any to any port 20,21,10000:10010 proto tcp

sudo systemctl restart vsftpd

sudo systemctl status vsftpd