

Simulation of SDN/OpenFlow Operations

EstiNet Technologies, Inc.

Outline

- Controller : Ryu
- Curl Commands in Ryu
- Add, Show, Modify and Delete Flow Entry
- Add, Show, Modify and Delete Meter
- Add, Show, Modify and Delete Group

Controller : Ryu

Install Ryu in Fedora 20

Perform the following commands in terminal with **root** privilege.

1. yum update
2. yum install python-pip
3. yum install python-devel
4. yum install python-eventlet
5. yum install python-routes
6. yum install python-webob
7. yum install python-paramiko
8. pip install ryu
9. pip install six --upgrade

Verify the Installation

```
ryu@ubuntu:~$ ryu-manager
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
```

Setup Ryu for EstiNet

1

1. Find ryu-manager binary in your system

- whereis ryu-manager

```
[root@localhost ~]# whereis ryu
ryu: /usr/bin/ryu /usr/etc/ryu
```

2. Copy ryu-manager binary to EstiNet workdir

- `cp /usr/bin/ryu-manager /usr/local/estinet/tools`

3. Find Ryu workdir in your system

- ryu-manager
- Ctrl + C

```
[root@localhost ~]# ryu-manager
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
^CTraceback (most recent call last):
  File "/bin/ryu-manager", line 9, in <module>
    load_entry_point('ryu==3.22', 'console_scripts', 'ryu-manager')()
  File "/usr/lib/python2.7/site-packages/ryu/cmd/manager.py", line 99, in main
    hub.joinall(services)
```

Setup Ryu for EstiNet

2

1. Move to Ryu workdir

- `cd /usr/lib/python2.7/site-packages/ryu`

```
[root@localhost ~]# cd /usr/lib/python2.7/site-packages/ryu
[root@localhost ryu]# ls
app      cfg.pyc  controller  flags.py  hooks.pyc  lib      ofproto  topology
base    cmd      exception.py  flags.pyc  __init__.py  log.py  services  utils.py
cfg.py  contrib  exception.pyc  hooks.py  __init__.pyc  log.pyc  tests    utils.pyc
```

2. Move to Ryu APPs folder

- `cd /app`

Setup Ryu for EstiNet

3

1. Copy **simple_switch_13.py** and **ofctl_rest.py** to EstiNet workdir

- `cp simple_switch_13.py /usr/local/estinet/tools/`
- `cp ofctl_rest.py /usr/local/estinet/tools/`

```
[root@localhost app]# ls
bmpstation.py          ofctl          rest_qos.py         simple_switch_12.pyc
bmpstation.pyc        ofctl_rest.py  rest_qos.pyc       simple_switch_13.pyc
cbench.py             ofctl_rest.pyc rest_quantum.py    simple_switch_13.pyc
cbench.pyc            quantum_adapter.py rest_quantum.pyc  simple_switch_14.py
client.py             quantum_adapter.pyc rest_router.py    simple_switch_14.pyc
client.pyc            rest_conf_switch.py rest_router.pyc   simple_switch_igmp.py
conf_switch_key.py   rest_conf_switch.pyc rest_topology.py  simple_switch_igmp.pyc
conf_switch_key.pyc  rest_firewall.py rest_topology.pyc simple_switch_lacp.py
gre_tunnel.py        rest_firewall.pyc rest_tunnel.py    simple_switch_lacp.pyc
gre_tunnel.pyc       rest_nw_id.py  rest_tunnel.pyc  simple_switch.py
gui_topology         rest_nw_id.pyc simple_isolation.py simple_switch.pyc
__init__.py         rest.py       simple_isolation.pyc simple_switch_snort.py
__init__.pyc        rest.pyc      simple_switch_12.py simple_switch_snort.pyc
```

Setup Ryu for EstiNet

4

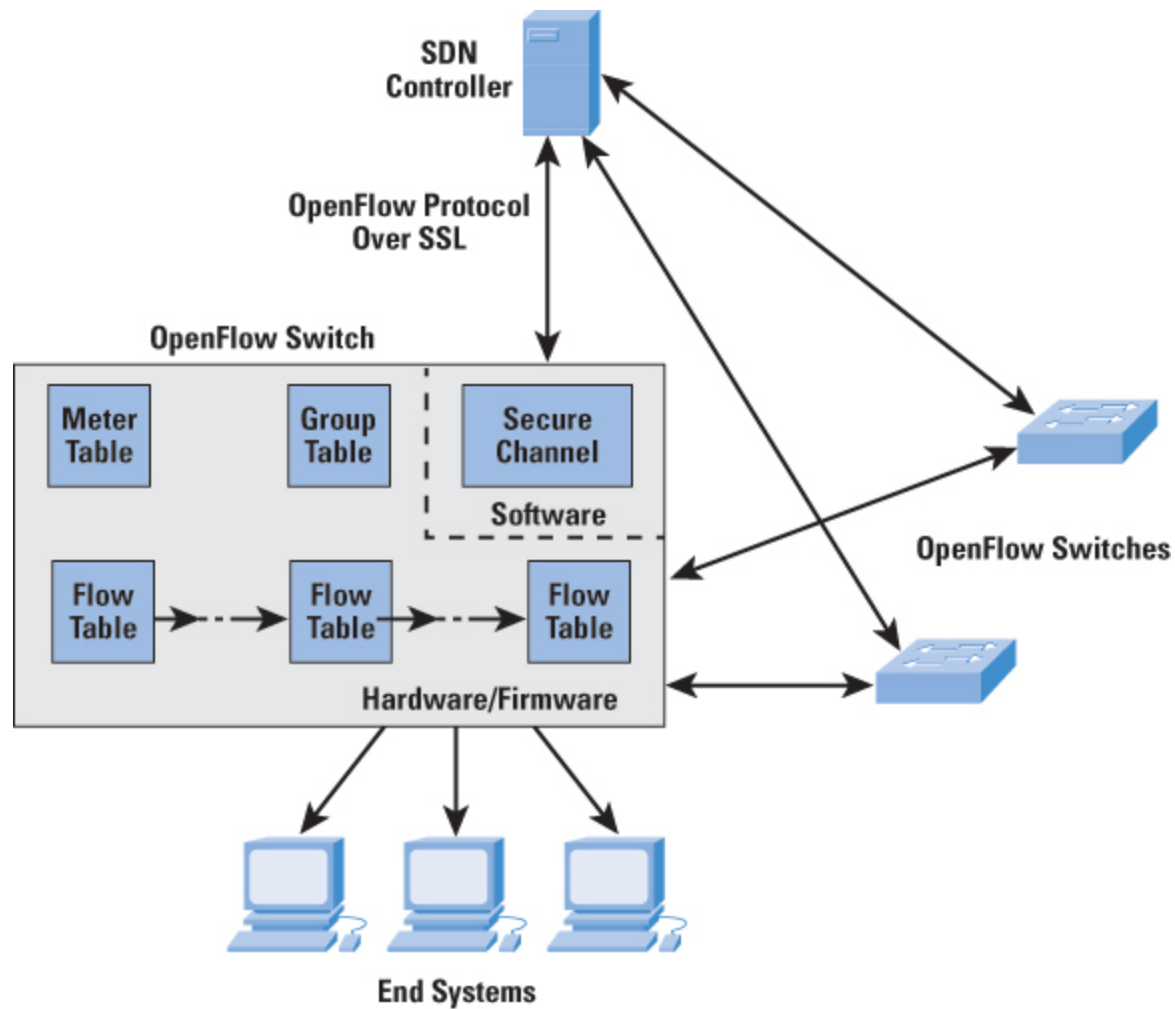
1. Move to EstiNet workdir
 - `cd /usr/local/estinet/tools/`
2. Verify `simple_switch_13.py`
 - `./ryu-manager simple_switch_13.py`

```
[root@localhost tools]# ./ryu-manager simple_switch_13.py
loading app simple_switch_13.py
loading app ryu.controller.ofp_handler
instantiating app simple_switch_13.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
```

3. Verify `ofctl_rest.py`
 - `./ryu-manager ofctl_rest.py`

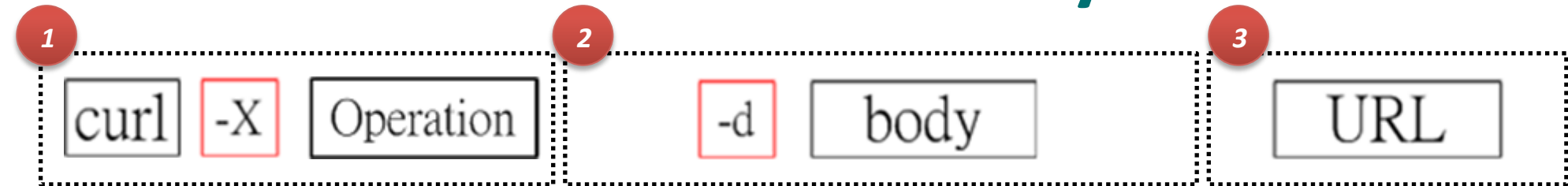
```
[root@localhost tools]# ./ryu-manager ofctl_rest.py
loading app ofctl_rest.py
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ofctl_rest.py of RestStatsApi
(16863) wsgi starting up on http://0.0.0.0:8080/
```


Flow, Meter, Group Tables



Curl Commands in Ryu

Curl command in Ryu



```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":2}, "actions":[{"type":"OUTPUT", "port":5}]}' http://1.0.1.1:8080/stats/flowentry/add
```

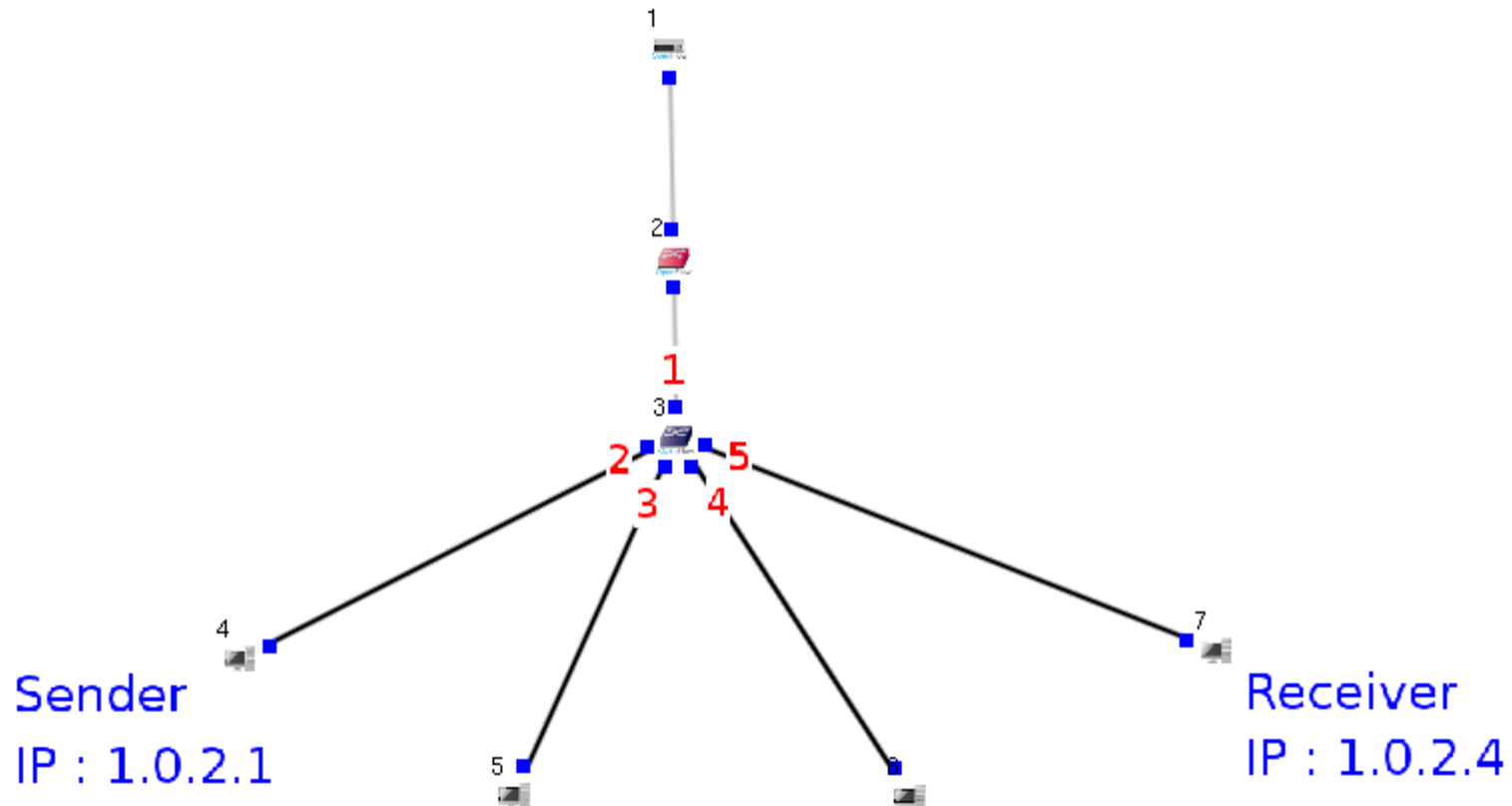
Syntax :

- Check /usr/lib/python2.7/site-packages/ryu/app/ofctl_rest.py
- Check /usr/lib/python2.7/site-packages/ryu/lib/ofctl_v1_3.py

Add, Show, Modify and Delete Flow Entry

Topology

Controller : Ryu
IP : 1.0.1.1



Get Switch's DPID

Curl Command :

```
curl http://1.0.1.1:8080/stats/switches
```

```
[root@localhost ~]# curl http://1.0.1.1:8080/stats/switches  
3 [root@localhost ~]# █
```

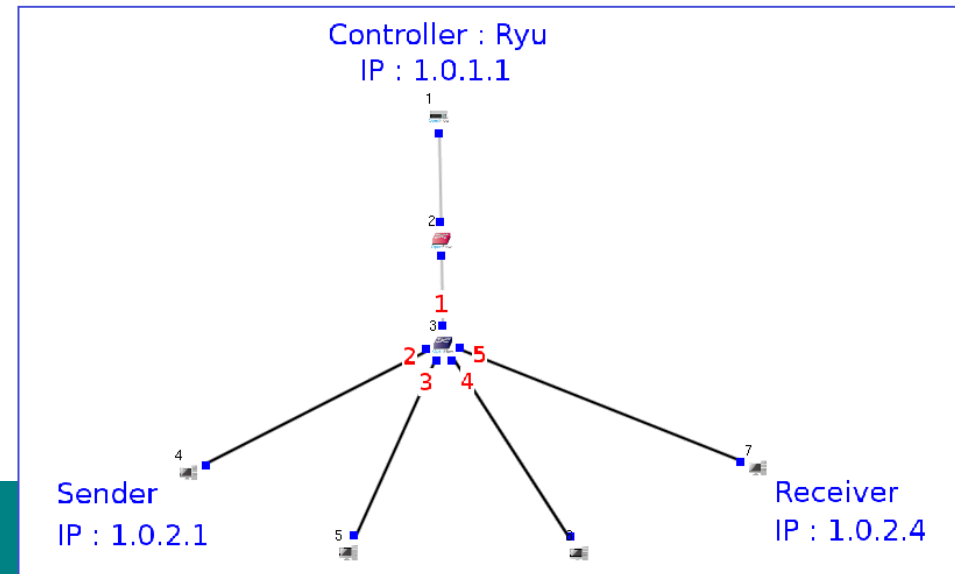
Add Flow Entries

Flow entry 1 : Forward packets from port 2 to port 5

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":2}, "actions":[{"type":"OUTPUT", "port":5}]}' http://1.0.1.1:8080/stats/flowentry/add
```

Flow entry 2 : Forward packets from port 5 to port 2

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":5}, "actions":[{"type":"OUTPUT", "port":2}]}' http://1.0.1.1:8080/stats/flowentry/add
```



Show Flow Entries

Curl Command :

<Format>

```
curl http://Contoller IP:8080/stats/flow/DPID
```

<Example>

```
curl http://1.0.1.1:8080/stats/flow/3
```

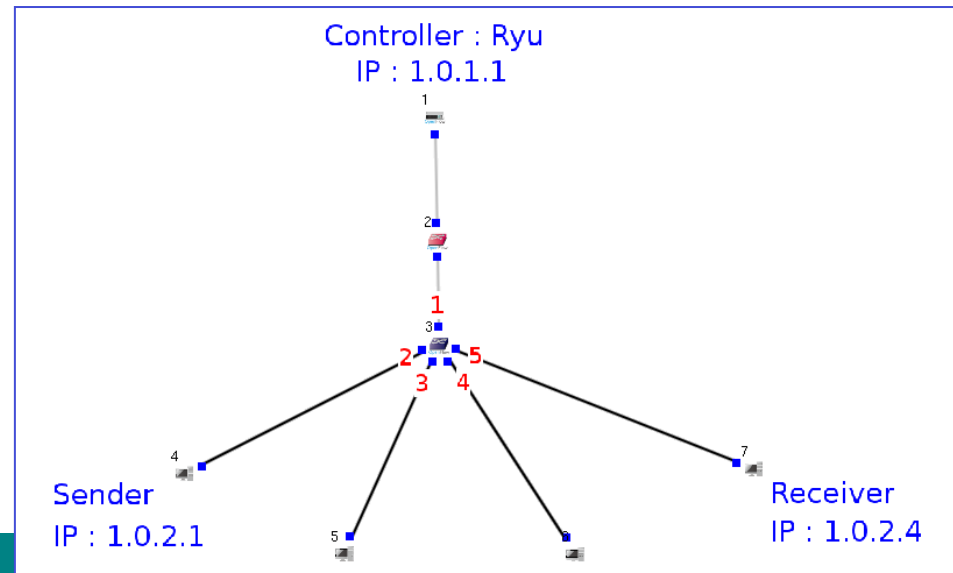

Modify Flow Entry

Modify Flow entry 2 : Drop the packets from port 5

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":5}, "actions":[]}' http://1.0.1.1:8080/stats/flowentry/modify
```

Recover Flow entry 2 :

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":5}, "actions":[{"type":"OUTPUT", "port":2}]}' http://1.0.1.1:8080/stats/flowentry/modify
```



Delete Flow Entries

Curl Command :

<Format>

```
curl -X DELETE http://Controller IP:8080/stats/flowentry/clear/DPID
```

<Example>

```
curl -X DELETE http://1.0.1.1:8080/stats/flowentry/clear/3
```

Add, Show, Modify and Delete Meter

Meter Type

Meter Identifier	Meter Bands	Counters
------------------	-------------	----------

Type 1 : DROP

Type 2 : DSCP_REMARK

Add Meter Entries

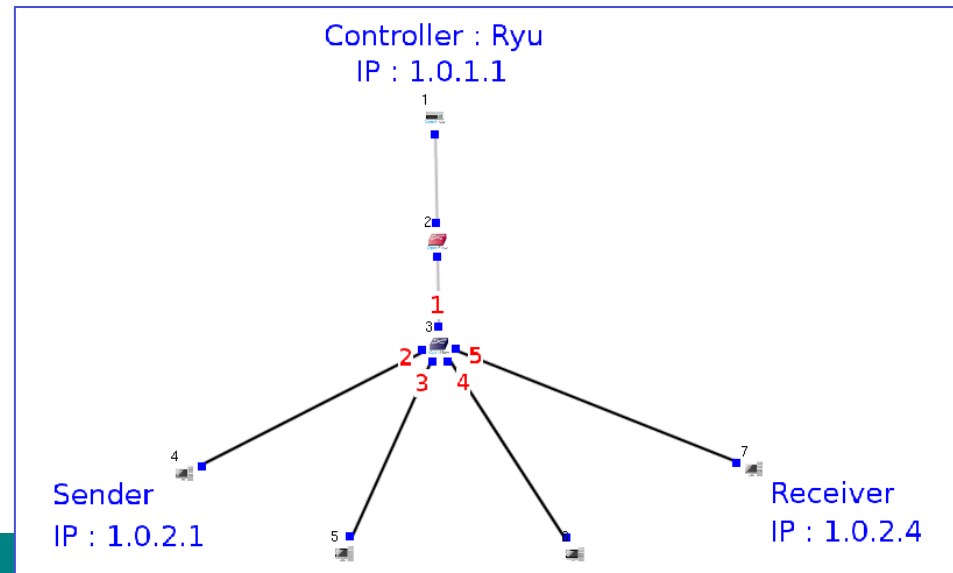
Type 1 : DROP

Meter entry 1 : 6 Mbps limit

```
curl -X POST -d '{"dpid": "3", "meter_id":1, "flags":"KBPS", "bands": [{"type":"DROP", "rate":6000}]}' http://1.0.1.1:8080/stats/meterentry/add
```

Meter entry 2 : 2 Mbps limit

```
curl -X POST -d '{"dpid": "3", "meter_id":2, "flags":"KBPS", "bands": [{"type":"DROP", "rate":2000}]}' http://1.0.1.1:8080/stats/meterentry/add
```



Show Meter Entries

Curl Command :

<Format>

```
curl http://Controller IP:8080/stats/meter/DPID
```

<Example>

```
curl http://1.0.1.1:8080/stats/meter/3
```

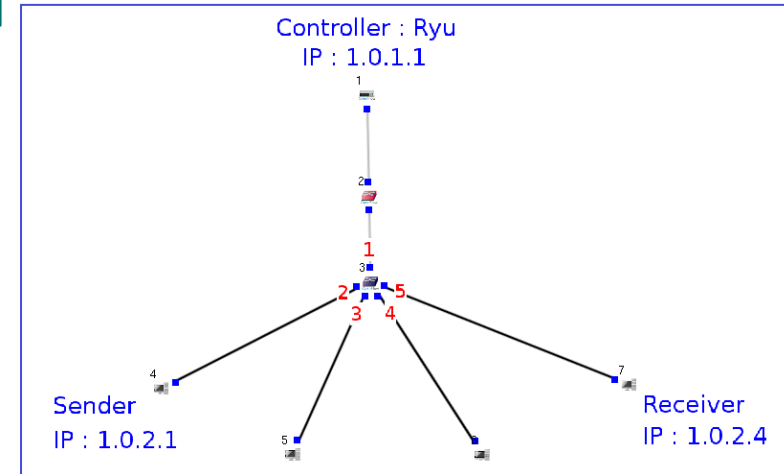
Add Flow Entries with Meter

Flow entry 1 :
Forward packets from port 2 to port 5
& limit throughput to 6 Mbps

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":2},  
"actions":[{"type":"METER", "meter_id":1}, {"type":"OUTPUT", "port":5}]}' http://  
1.0.1.1:8080/stats/flowentry/add
```

Flow entry 2 :
Forward packets from port 5 to port 2
& limit throughput to 2 Mbps

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":5},  
"actions":[{"type":"METER", "meter_id":2}, {"type":"OUTPUT", "port":2}]}' http://  
1.0.1.1:8080/stats/flowentry/add
```



Modify Meter Entry

Modify Meter entry 2 :

Change meter band from 6 Mbps to 2 Mbps

```
curl -X POST -d '{"dpid": "3", "meter_id":1, "flags":"KBPS", "bands": [{"type":"DROP", "rate":2000}]}' http://1.0.1.1:8080/stats/meterentry/modify
```


Delete Meter Entries

Delete Meter entry 1 :

```
curl -X POST -d '{"dpid": "3", "meter_id":1, "flags":"KBPS", "bands": [{"type":"DROP", "rate":2000}]}' http://1.0.1.1:8080/stats/meterentry/delete
```

Delete Meter entry 2

```
curl -X POST -d '{"dpid": "3", "meter_id":2, "flags":"KBPS", "bands": [{"type":"DROP", "rate":2000}]}' http://1.0.1.1:8080/stats/meterentry/delete
```

DSCP Remark

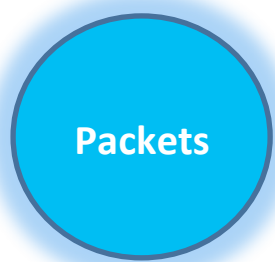


Table 0

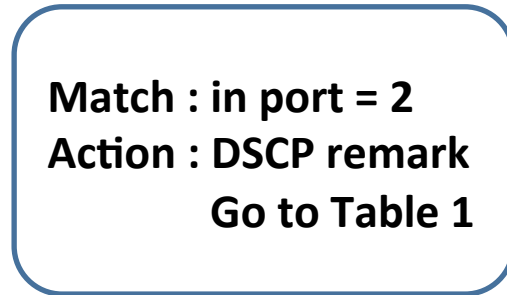
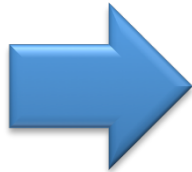
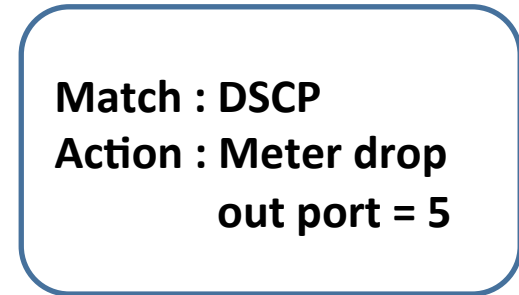


Table 1



Send UDP

Sender :

host

Host ID Name

Application Down time Emulation

Start time(s) ▲	Stop time(s)	Command	Input f	
1	1000	#stcp 1.0.2.4		Add
1	1000	ttcp -t -u -s 1.0.2.4		Modify
				Delete
				App. Usage

Node editor Command console OK

C.P.A.N.S.T. Cancel

Receiver :

host

Host ID Name

Application Down time Emulation

Start time(s) ▲	Stop time(s)	Command	
1	1000	#rtcp	Add
1	1000	tcpdump -U -i eth1 -w host	Modify
1	1000	ttcp -r -u -s	Delete
			App. Usage

Node editor Command console OK

C.P.A.N.S.T. Cancel

Add Meter Entries

Type 2 : DSCP_REMARK

Meter entry 1 : Remark DSCP to 1 if throughput reaches 8 Mbps

```
curl -X POST -d '{"dpid": "3", "meter_id":1, "flags":"KBPS", "bands":  
[{"type":"DSCP_REMARK", "rate":8000, "prec_level":1}]}' http://1.0.1.1:8080/stats/  
meterentry/add
```

Meter entry 2 : 6 Mbps limit

```
curl -X POST -d '{"dpid": "3", "meter_id":2, "flags":"KBPS", "bands":  
[{"type":"DROP", "rate":6000}]}' http://1.0.1.1:8080/stats/meterentry/add
```

Add Flow Entries with Meter

Flow entry 1 on Table 0:

**Assign packets from port 2 to go to Table 1
& Remark DSCP to 1 if throughput reaches 8 Mbps**

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "match":{"in_port":2},  
"actions":[{"type":"METER", "meter_id":1}, {"type":"GOTO_TABLE", "table_id":1}]}'  
http://1.0.1.1:8080/stats/flowentry/add
```

Flow entry 2 on Table 1:

**Assign packets from Table 0 to go out to port 5
& limit throughput to 6 Mbps if DSCP is 1**

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"0", "table_id":1, "match":  
{"ip_dscp":1}, "actions":[{"type":"METER", "meter_id":2}, {"type":"OUTPUT", "port":  
5}]}' http://1.0.1.1:8080/stats/flowentry/add
```

Add Flow Entries for ARP

Flow entry 3 on Table 0:

Assign ARP packets from port 2 to go out to port 5

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"1", "match":{"in_port":2, "arp_tpa":"1.0.2.4"}, "actions":[{"type":"OUTPUT", "port":5}]}' http://1.0.1.1:8080/stats/flowentry/add
```

Flow entry 4 on Table 0:

Assign ARP packets from port 5 to go out to port 2

```
curl -X POST -d '{"dpid": "3", "cookie":1, "priority":"1", "match":{"in_port":5, "arp_tpa":"1.0.2.1"}, "actions":[{"type":"OUTPUT", "port":2}]}' http://1.0.1.1:8080/stats/flowentry/add
```

Add, Show, Modify and Delete Group

Group Type

Group Identifier	Group Type	Counters	Action Buckets
------------------	------------	----------	----------------

Type 1 : ALL

Type 2 : SELECT

Type 3 : INDIRECT

Type 4 : Fast Failover

ARP Mode : Build ARP Table in Advance

Module Edit

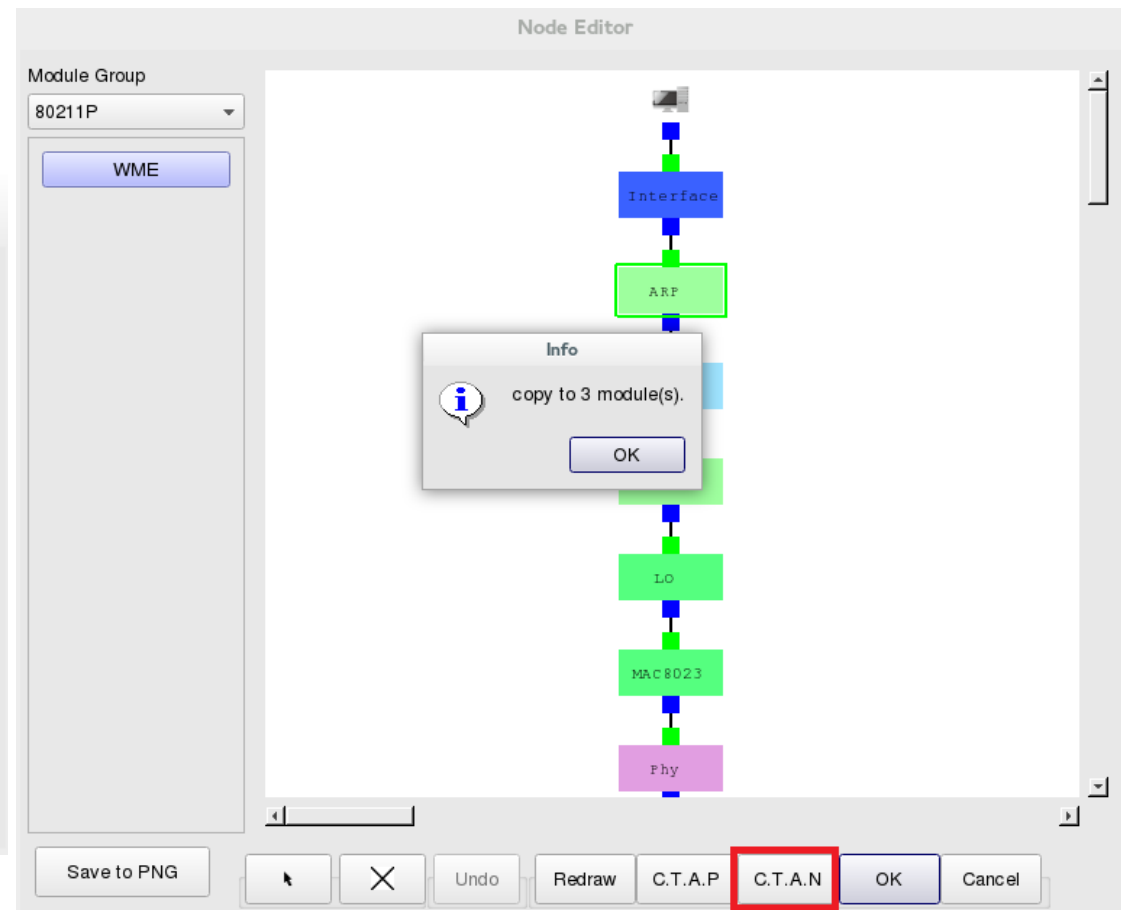
Parameters Setting

ARP Mode

Run ARP Protocol
Flush Time Interval (ms)

Build ARP Table In Advance
File Name

Run Time Query



Add Group Entry

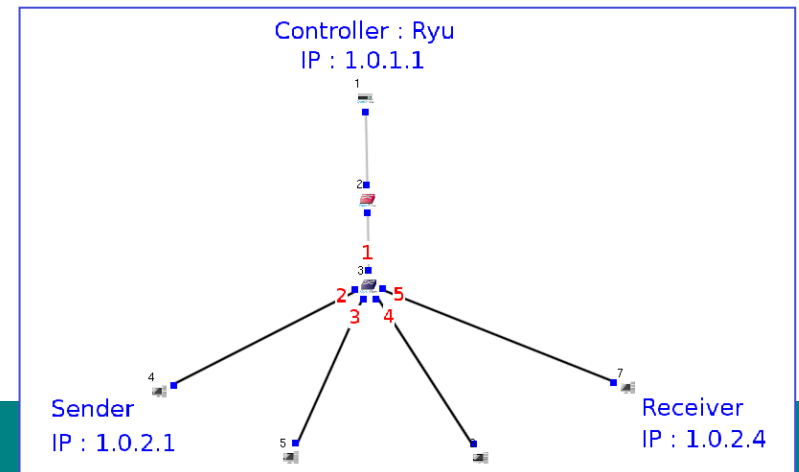
Type 1 : ALL

Group entry 1 : Output to port 3, Output to port 4

```
curl -X POST -d '{"dpid": "3", "type": "ALL", "group_id": 1, "buckets": [{"actions": [{"type": "OUTPUT", "port": 3}], "actions": [{"type": "OUTPUT", "port": 4}]}' http://1.0.1.1:8080/stats/groupentry/add
```

Flow entry 1 : Output to port 5 & Go to Group table

```
curl -X POST -d '{"dpid": "3", "cookie": 1, "priority": "0", "match": {"in_port": 2}, "actions": [{"type": "OUTPUT", "port": 5}, {"type": "GROUP", "group_id": 1}]}' http://1.0.1.1:8080/stats/flowentry/add
```



Show Group Entry

Curl Command :

<Format>

```
curl http://Controller IP:8080/stats/group/DPID
```

<Example>

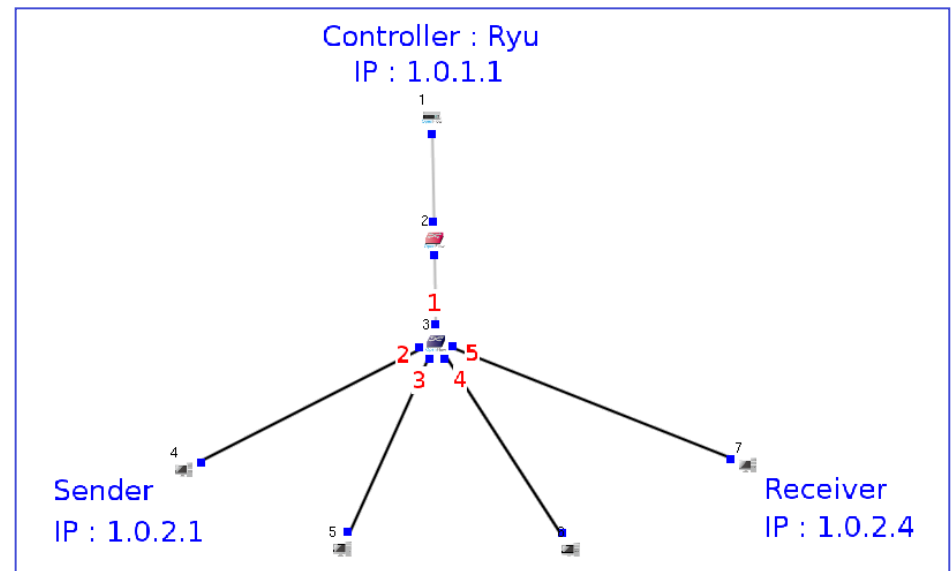
```
curl http://1.0.1.1:8080/stats/group/3
```

Modify Group Entry

Type 3 : INDIRECT

Group entry 1 : Define one bucket only

```
curl -X POST -d '{"dpid": "3", "type":"INDIRECT", "group_id":1, "buckets": [{"actions":[{"type":"OUTPUT", "port":4}]}]}' http://1.0.1.1:8080/stats/groupentry/modify
```



Delete Group Entry

Delete Group entry 1 :

```
curl -X POST -d '{"dpid": "3", "type":"INDIRECT", "group_id":1, "buckets": [{"actions":[{"type":"OUTPUT", "port":4}]}]}' http://1.0.1.1:8080/stats/groupentry/delete
```

ADD Group Entry

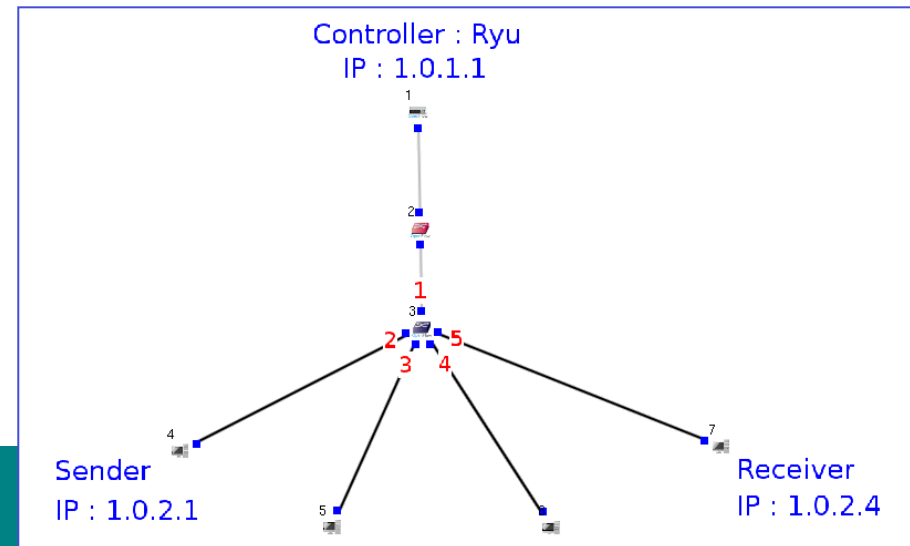
Type 2 : SELECT

Group entry 1 : Round Robin output to port 3 and port 4

```
curl -X POST -d '{"dpid": "3", "type": "SELECT", "group_id": 1, "buckets": [{"weight": 1, "actions": [{"type": "OUTPUT", "port": 3}], {"weight": 1, "actions": [{"type": "OUTPUT", "port": 4}]}]' http://1.0.1.1:8080/stats/groupentry/add
```

Flow entry 1 : Output to port 5 & Go to Group table

```
curl -X POST -d '{"dpid": "3", "cookie": 1, "priority": "0", "match": {"in_port": 2}, "actions": [{"type": "OUTPUT", "port": 5}, {"type": "GROUP", "group_id": 1}]}' http://1.0.1.1:8080/stats/flowentry/add
```



Modify Group Entry

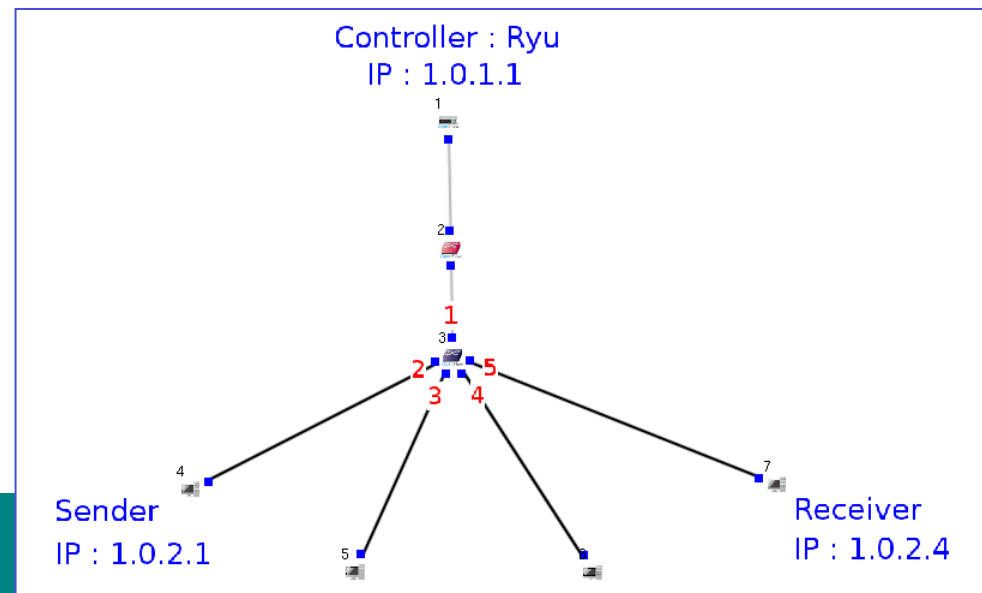
Type 4 : Fast Failover

Group entry 1 : Fast failover

```
curl -X POST -d '{"dpid": "3", "type": "FF", "group_id": 1, "buckets": [{"watch_port": 3, "actions": [{"type": "OUTPUT", "port": 3}], {"watch_port": 4, "actions": [{"type": "OUTPUT", "port": 4}]}]}' http://1.0.1.1:8080/stats/groupentry/add
```

Flow entry 1 : Output to port 5 & Go to Group table

```
curl -X POST -d '{"dpid": "3", "cookie": 1, "priority": "0", "match": {"in_port": 2}, "actions": [{"type": "OUTPUT", "port": 5}, {"type": "GROUP", "group_id": 1}]}' http://1.0.1.1:8080/stats/flowentry/add
```



Thank you for using EstiNet !