

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

User Manual

AmLight - Learning OpenFlow/SDN Network Research and
Experimentation

Team Members: Cindy Perez, Jose Mercardo, Kenia Chang He, and Yuyang Leng

Product Owner: Vasilka Chergarova

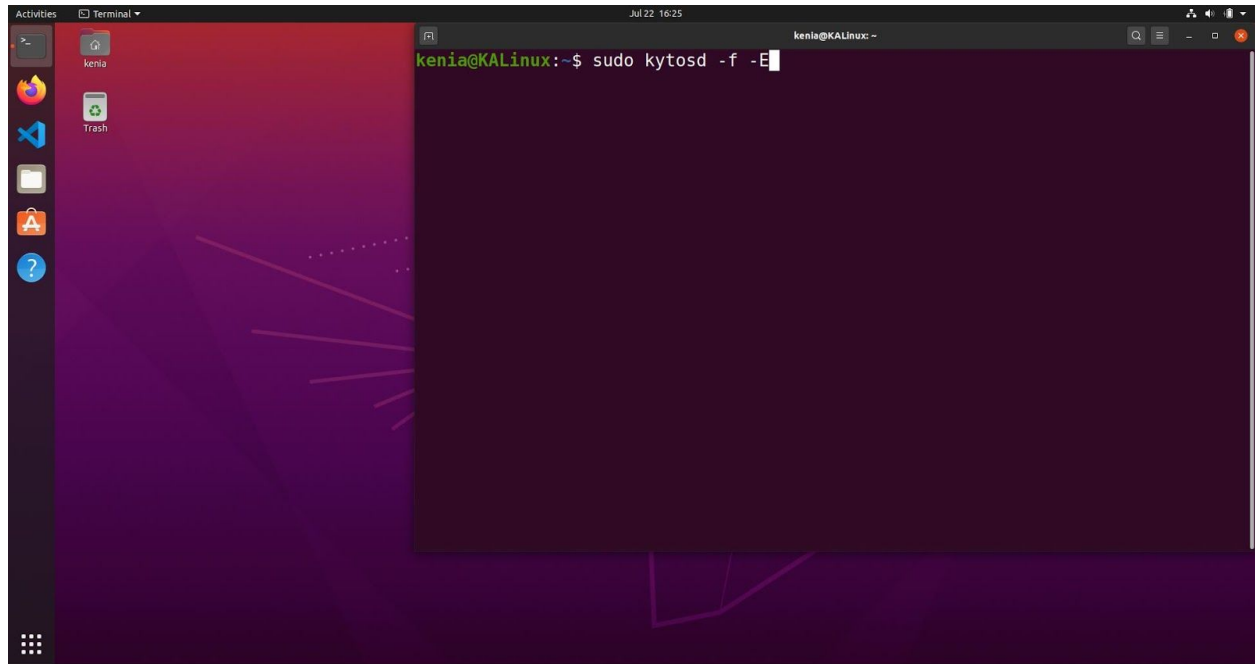
Mentor: Jeronimo Bezerra

Instructor: Masoud Sadjadi

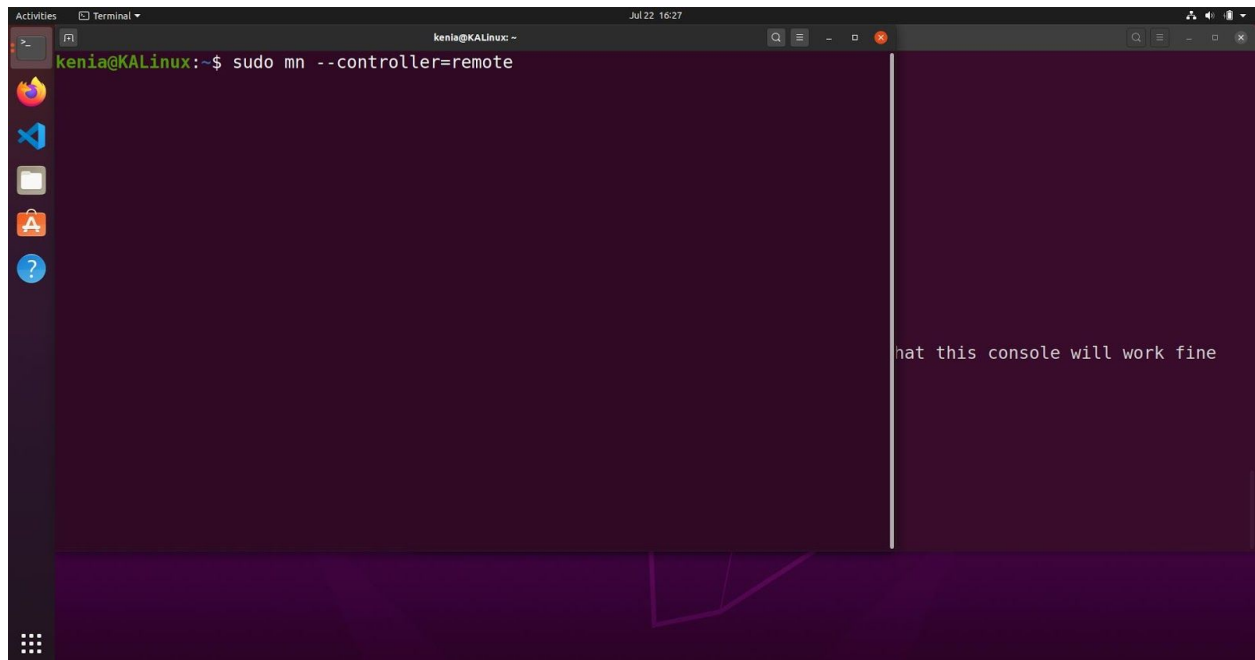
This section contains all the necessary steps in order to successfully test the match fields with masking and the IPv6 packet structure implemented for Kytos.

Match Fields

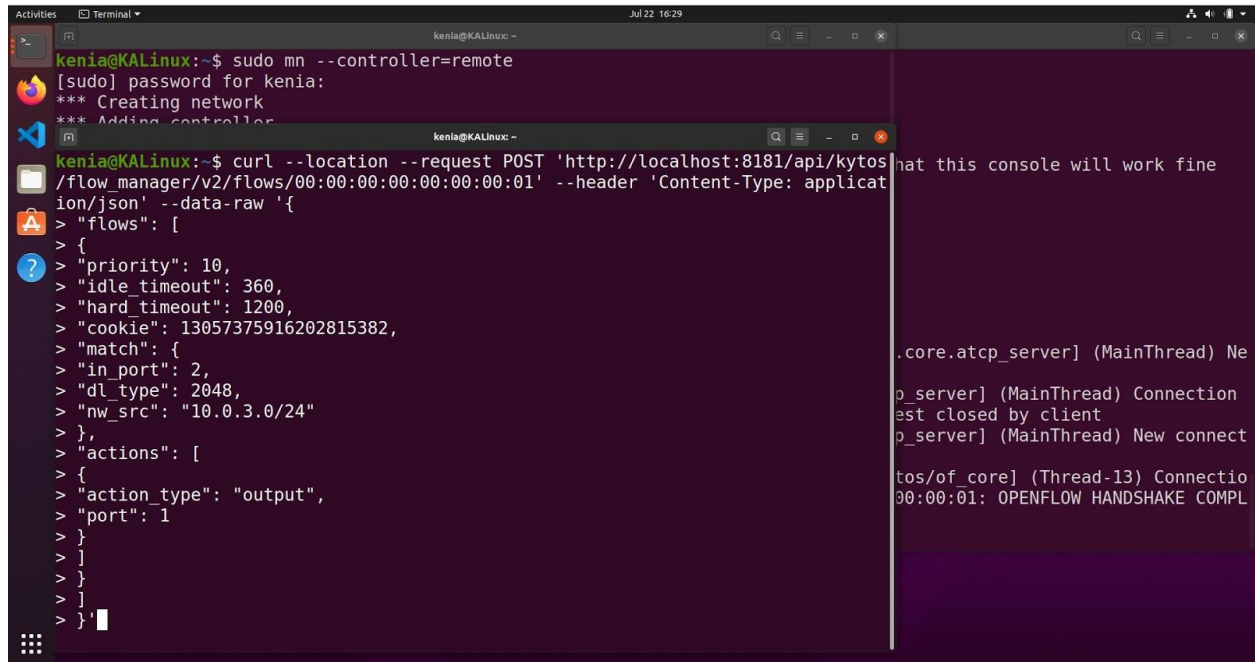
Step 1: Enter the command in the terminal to run Kytos in foreground mode.



Step 2: Enter the command in the terminal to run Mininet.



Step 3: Enter the command in the terminal to send the flow to the Kytos server and it is processed by the flow_manager NApp.



The terminal window shows the following commands and output:

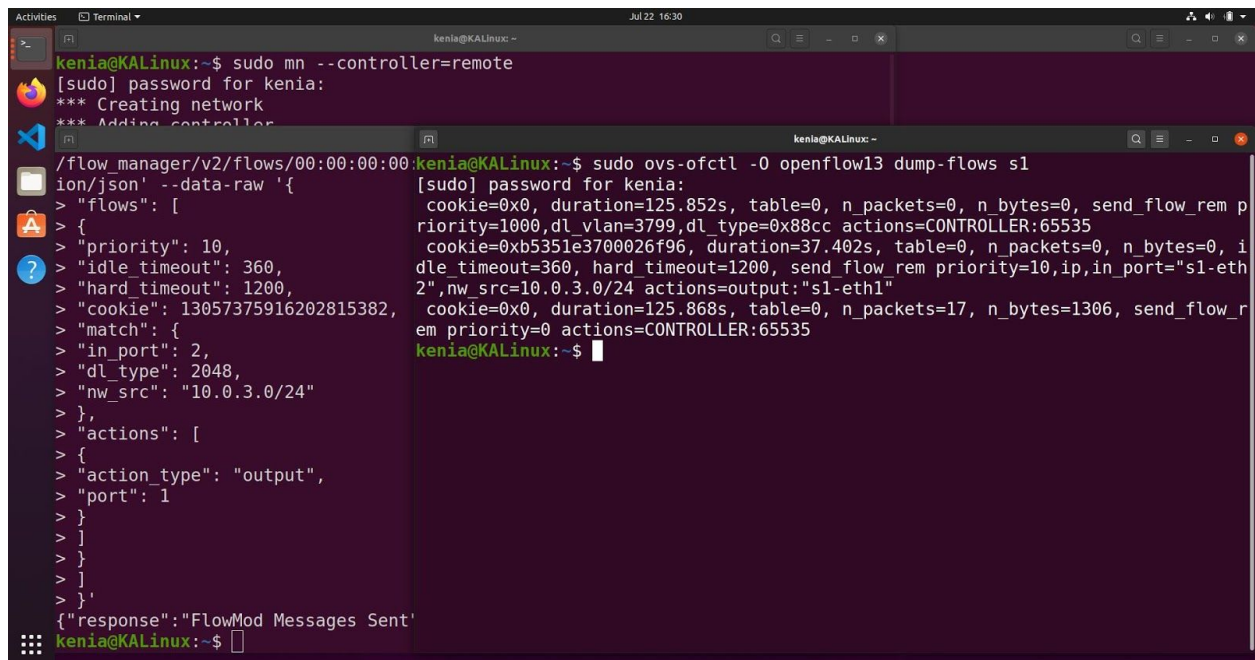
```
kenia@KALinux:~$ sudo mn --controller=remote
[sudo] password for kenia:
*** Creating network
*** Adding controller

kenia@KALinux:~$ curl --location --request POST 'http://localhost:8181/api/kytos/flow_manager/v2/flows/00:00:00:00:00:00:01' --header 'Content-Type: application/json' --data-raw '{
> "flows": [
> {
>   "priority": 10,
>   "idle_timeout": 360,
>   "hard_timeout": 1200,
>   "cookie": 13057375916202815382,
>   "match": {
>     "in_port": 2,
>     "dl_type": 2048,
>     "nw_src": "10.0.3.0/24"
>   },
>   "actions": [
>     {
>       "action_type": "output",
>       "port": 1
>     }
>   ]
> }
> ]
> }'
```

The log window on the right shows the following messages:

```
...core.atcp_server] (MainThread) Ne
...p_server] (MainThread) Connection
...est closed by client
...p_server] (MainThread) New connect
...tos/of_core] (Thread-13) Connectio
00:00:01: OPENFLOW HANDSHAKE COMPL
```

Step 4: Enter the command in the terminal to check the flow installed on switch 1.



The terminal window shows the following commands and output:

```
kenia@KALinux:~$ sudo mn --controller=remote
[sudo] password for kenia:
*** Creating network
*** Adding controller

kenia@KALinux:~$ sudo ovs-ofctl -O openflow13 dump-flows s1
[sudo] password for kenia:
cookie=0x0, duration=125.852s, table=0, n_packets=0, n_bytes=0, send_flow_re p
priority=1000,dl_vlan=3799,dl_type=0x88cc actions=CONTROLLER:65535
cookie=0xb5351e3700026f96, duration=37.402s, table=0, n_packets=0, n bytes=0, i
dle_timeout=360, hard_timeout=1200, send_flow_re priority=10,ip,in_port="s1-eth
2",nw_src=10.0.3.0/24 actions=output:"s1-eth1"
cookie=0x0, duration=125.868s, table=0, n_packets=17, n_bytes=1306, send_flow_r
em priority=0 actions=CONTROLLER:65535
kenia@KALinux:~$
```

The log window on the right shows the following messages:

```
...{"response":"FlowMod Messages Sent"
```

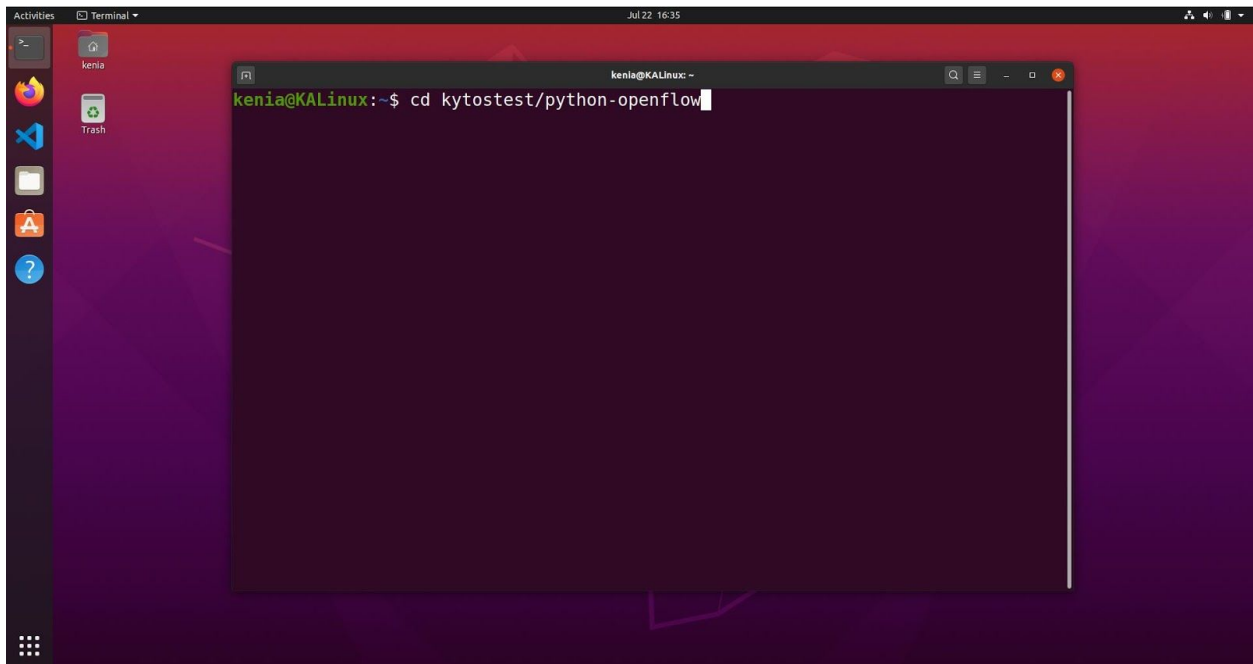
Step 5: Enter the command in the terminal to check the flow processed by the flow_manager NApp.

```
kenia@KALinux:~$ sudo mn --controller=remote
[sudo] password for kenia:
*** Creating network
*** Adding controller

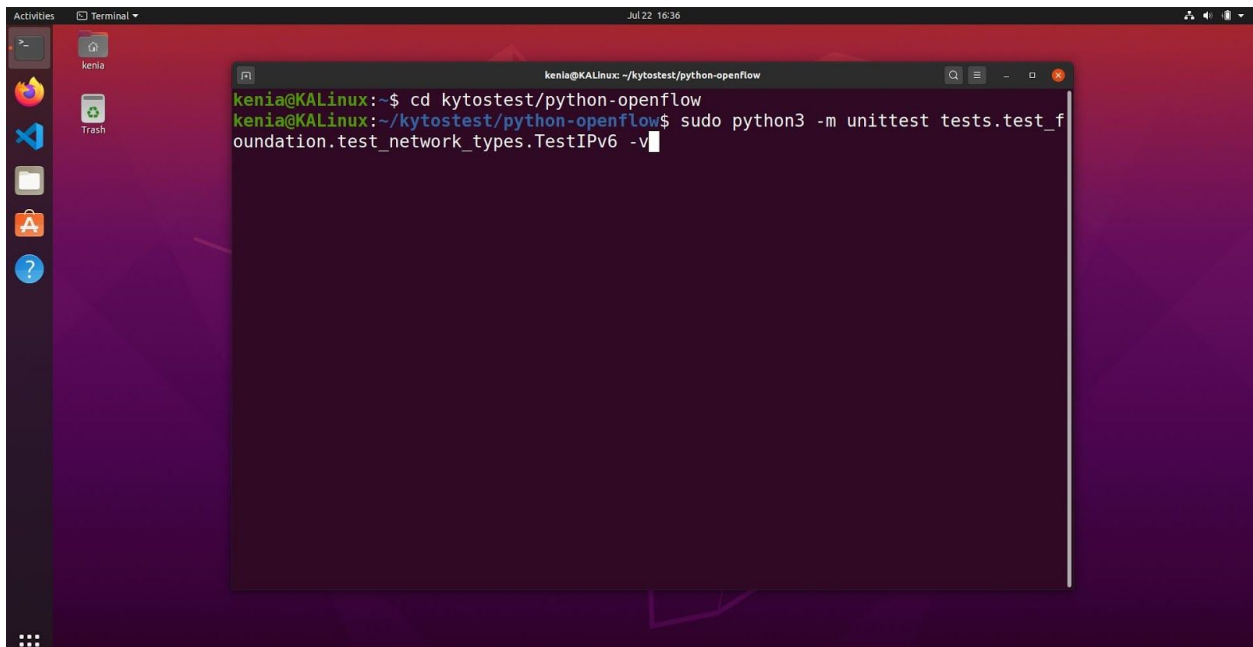
/flow_manager/vkenia@KALinux:~$ curl --location --request GET 'http://localhost:8181/api/kytos/
ion/json' --datflow_manager/v2/flows/00:00:00:00:00:00:00:01'
> "flows": [ {"00:00:00:00:00:00:00:01":{"flows":[{"actions":[{"action_type":"output","port":
> { 4294967293},"cookie":0,"hard_timeout":0,"id":"a09156410a0edcc48ff0b1080d877e2b" 5535
> "priority": 1,"idle_timeout":0,"match":{"dl_type":35020,"dl_vlan":3799},"priority":1000,"stat
> "idle_timeouts":{"byte_count":0,"duration_nsec":97000000,"duration_sec":168,"packet_count":0}
> "hard_timeout":{"switch":"00:00:00:00:00:00:00:01"},"table_id":0},"actions":[{"action_type":"ou
> "cookie": 130tput"},"port":1},"cookie":13057375916202815382,"hard_timeout":1200,"id":"3b693fc
> "match":{" 5c8e4b60f6c3fc3f5b3d78dc7","idle_timeout":360,"match":{"dl_type":2048,"in_port":
> "in_port": 2,2,"nw_src":"10.0.3.0/24"},"priority":10,"stats":{"byte_count":0,"duration_nsec":
> "dl_type": 20647000000,"duration_sec":79,"packet_count":0},"switch":"00:00:00:00:00:00:00:01"
> "nw_src": "10","table_id":0},"actions":[{"action_type":"output","port":4294967293},"cookie":
> , 0,"hard_timeout":0,"id":"bf73e38c0451803cbc5954ec73c58af2","idle_timeout":0,"mat
> "actions": [ ch":{"priority":0,"stats":{"byte_count":1376,"duration_nsec":113000000,"durati
> { "on_sec":168,"packet_count":18},"switch":"00:00:00:00:00:00:00:01"},"table_id":0}}]
> "action_type"}}
> "port": 1 kenia@KALinux:~$
>
> ]
> ]
> ]
> ]
> '
{"response":"Fl
kenia@KALinux:~$
```

IPv6 Packet

Step 1: Go to where the python-openflow repository is on the machine.



Step 2: Enter the command in the terminal to run the unit test for IPv6 packet.



Here is the result of testing for the IPv6 packet.

```
kenia@KALinux: ~$ cd kytostest/python-openflow
kenia@KALinux:~/kytostest/python-openflow$ sudo python3 -m unittest tests.test_foundation.test_network_types.TestIPv6 -v
[sudo] password for kenia:
test_IPv6_pack (tests.test_foundation.test_network_types.TestIPv6)
Test pack/unpack of IPv6 class. ... ok
test_IPv6_unpack (tests.test_foundation.test_network_types.TestIPv6)
Test unpack of IPv6 binary packet. ... ok

-----
Ran 2 tests in 0.001s

OK
kenia@KALinux:~/kytostest/python-openflow$
```