Content

Explanation:

Environment: windows or linux
Compiler: python 2.7+
Pox：pox-betta

Chapter 1 OPENFLOW Configuration (match)

1.1 ./pox.py openflow.of_01 --address=x.x.x.x –port=yy py

Command：**./pox.py openflow.of_01 --address=x.x.x.x –port=yy py**
Function：The controller enables the corresponding address monitoring.
Parameters：**address，port，py**
Default：None.
Command Mode：The commands including the path of pox.py.
Usage Guide：**./pox.py openflow.of_01 --address=x.x.x.x –port=yy py**。
Example：
Controller terminal:
Enable the pox address monitoring. The port of 6633 will be used to enter into the interactive mode as default.

```
root@long-Aspire-4733Z:/home/long/usr/pox# ./pox.py openflow.of_01 --address=6.6.6.6
py
POX 0.1.0 (betta) / Copyright 2011-2013 James McCauley, et al.
INFO:core:POX 0.1.0 (betta) is up.
This program comes with ABSOLUTELY NO WARRANTY.  This program is free
software,
and you are welcome to redistribute it under certain conditions.
Type 'help(pox.license)' for details.
Ready.
POX>
```

Pc terminal:

```
SW1(config)#openflow mode
SW1(config-openflow)#openflow tcp 6.6.6.6 6633
```

Controller terminal:
The switch connection prompt

```
POX> INFO:openflow.of_01:[00-03-0f-27-5e-91 1] connected
```

1.2 Import pox.openflow.libopenflow_01 as of

Command：**import pox.openflow.libopenflow_01 as of**
Function：Export the core module and name it as of.
Parameters：None.
Default：None.

Command Mode：Interactive mode, pox>
Usage Guide：**import pox.openflow.libopenflow_01 as of**. Export the core module and name it as of.
Example：
Controller terminal:
POX> **import pox.openflow.libopenflow_01 as of**

1.3 core.openflow.connections.keys()

Command：**core.openflow.connections.keys()**
Function：Get the key of the “of switch” connected to the controller.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**core.openflow.connections.keys().** Send the msg by the key after got it.
Example：
Controller terminal:
POX> **core.openflow.connections.keys()**

1.4 core.openflow.connections[key].send(msg)

Command：**core.openflow.connections[key].send(msg)**
Function：Send the msg through the specific key.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**core.openflow.connections[key].send(msg).** The **key** is the switch code which was got by the last command.
Example：
Controller terminal:
POX> **core.openflow.connections[key].send(msg)**

1.5 msg=of.ofp_flow_mod()

Command：**msg=of.ofp_flow_mod()**
Function：Compile the message and the type of message is flow mod.
Parameters：**command**：0 means to ADD (add flow). 1 means to MODIFY, 2 means to MODIFY_STRICT, 3 means to DELETE (delete all the flow rules) and 4 means to DELETE_STRICT (delete the flow rules according to the mask and priority)
Default：command：0
Command Mode：Interactive mode, pox>
Usage Guide：**msg=of.ofp_flow_mod().** The type of the msg message is flow mod.
Example：

Controller terminal:

POX> **msg=of.ofp_flow_mod()**

1.6 msg.priority=x

Command：**msg.priority=x**

Function：Configure the priority of the rules.

Parameters：None.

Default：None.

Command Mode：Interactive mode, pox>

Usage Guide：**msg.priority=x.** The priority must be initialized and the rage is from 1 to5.

Example：

Controller terminal:

POX> **msg.priority=5**

1.7 msg.match.in_port=y

Command：**msg.match.in_port=y**

Function：Match the access port.

Parameters：None.

Default：None.

Command Mode：Interactive mode, pox>

Usage Guide：msg.match.in_port=y. y is the value of the field.

Example：

Controller terminal:

Appoint the rule to match the access port of 1.

POX> **msg.match.in_port=1**

1.8 msg.match.dl_src=EthAddr(“”)

Command：**msg.match.dl_src=EthAddr(“”)**

Function：Match the so urce mac.

Parameters：None.

Default：None.

Command Mode：Interactive mode, pox>

Usage Guide：msg.match.dl_src=EthAddr(“”)

Example：

Controller terminal:

Appoint the rule to match the source mac of 00:00:00:00:00:11.

POX> **msg.match.dl_src=EthAddr(“00:00:00:00:00:11”)**

1.9 msg.match.dl_dst=EthAddr(“”)

Command：**msg.match.dl_dst=EthAddr(“”)**
Function：Match the destination mac.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.dl_dst=EthAddr(“”)
Example：
Controller terminal:
Appoint the rule to match the destination mac of 00:00:00:00:00:11.
POX> **msg.match.dl_dst=EthAddr(“00:00:00:00:00:11”)**

1.10 msg.match.dl_type=x

Command：**msg.match.dl_type=x**
Function：Match the type of ethernet.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.dl_type=
Example：
Controller terminal:
Appoint the rule to match the packets of the ip type.
POX> **msg.match.dl_type=0x800**

1.11 msg.match.dl_vlan=x

Command：**msg.match.dl_vlan=x**
Function：Match the vlan id.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.dl_vlan=
Example：
Controller terminal:
Appoint the rule to match the vlan id.
POX> **msg.match.dl_vlan=3**
Explanation: dl_vlan must be “of vlan”.

1.12 msg.match.dl_vlan_pcp=x

Command：**msg.match.dl_vlan_pcp=x**
Function：Match the tos value.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.dl_vlan_pcp=
Example：
Controller terminal:
Appoint the rule to match the cos value.
POX> **msg.match.dl_vlan_pcp=3**
Explanation: dl_vlan_pcp must be from 0 to 7.

1.13 msg.match.nw_src=

Command：**msg.match.nw_src=**
Function：Match the source ip address.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.nw_src=
Example：
Controller terminal:
Appoint the rule to match the source IP.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_src="192.168.2.133/24"**
Explanation: The type of ethernet must be appointed and the mask of ip can be appointed with "/".

1.14 msg.match.nw_dst=

Command：**msg.match.nw_dst=**
Function：Match the destination ip address.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.nw_dst=
Example：
Controller terminal:
Appoint the rule to match the destination IP address.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_dst="192.168.2.133/24"**

Explanation: The type of ethernet must be appointed and the mask of ip can be appointed with "/".

1.15 msg.match.nw_proto=x

Command: **msg.match.nw_proto=x**
Function: Match the protocol type.
Parameters: None.
Default: None.
Command Mode: Interactive mode, pox>
Usage Guide: msg.match.nw_proto=
Example:
Controller terminal:
Appoint the rule to match the packet of IP type.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_proto=6**
Explanation: The type of ethernet must be appointed and then match the ip protocol.

1.16 msg.match.nw_tos=x

Command: **msg.match.nw_tos=x**
Function: Match the tos.
Parameters: None.
Default: None.
Command Mode: Interactive mode, pox>
Usage Guide: msg.match.nw_tos=
Example:
Controller terminal:
Appoint the rule to match the ip protocol.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_tos=64**
Explanation: The type of ethernet must be appointed and then match the tos value.

1.17 msg.match.tp_src=x

Command: **msg.match.tp_src=**
Function: Match the tcp source port.
Parameters: None.
Default: None.
Command Mode: Interactive mode, pox>
Usage Guide: msg.match.tp_src=
Example:

Controller terminal:
Appoint the rule to match the tcp source port.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_proto=6**
POX> **msg.match.tp_src=179**
Explanation: The type of ethernet must be appointed, then match the ip protocol and match the tcp port at last.

1.18 msg.match.tp_dst=x

Command：msg.match.tp_dst=
Function：Match the tcp destination port.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.match.tp_dst=
Example：
Controller terminal:
Appoint the rule to match the tcp destination port.
POX> **msg.match.dl_type=0x800**
POX> **msg.match.nw_proto=6**
POX> **msg.match.tp_dst=179**
Explanation: The type of ethernet must be appointed, then match the ip protocol and match the tcp port at last.

1.19 msg.idle_timeout=x

Command：msg.idle_timeout=
Function：In the interval of idle, if there is no packet triggering this action, this rule will be deleted.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.idle_timeout=30
Example：
Controller terminal:
Appoint the idle time of the rule as 30s.
POX> **msg.idle_timeout=30**
Explanation: None.

1.20 msg.hard_timeout=x

Command：msg.hard_timeout=

Function：This rule will be deleted anyway before achieving the time of hard.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：msg.hard_timeout=30
Example：
Controller terminal:
Appoint the hard time of the rule as 30s.
POX> **msg.hard_timeout=30**
Explanation: None.

Chapter 2 OPENFLOW Configuration (action)

Explanation: If there is no action in rules, it means to drop as default; the egress port needs to be added after the corresponding action if there is no display in the rules and the egress port was configured.

2.1 msg.actions.append(of.ofp_action_output(port=x))

Command：**msg.actions.append(of.ofp_action_output(port=))**
Function：Appoint the egress port action.
Parameters：None.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_output(port=))**
Example：
Controller terminal: Appoint the egress port of the packet.
POX> **msg.actions.append(of.ofp_action_output(port=20))**
Explanation: The port number is the port in the "of vlan".

2.2 msg.actions.append(of.ofp_action_output(port=x))

Command：**msg.actions.append(of.ofp_action_output(port=))**
Function：Forward the appointed port type.
Parameters：IN_PORT = 0xfff8：send packets from the access port; FLOOD= 0xfffb：all the ports except the access ports and the ports which are not allowed by stp; ALL = 0xfffc：other ports except the access ports; CONTROLLER = 0xfffd： send to the controller; NONE = 0xffff：unrelated to the physical port.
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_output(port=))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_output(port=all))**
Explanation: send a packet to all the ports except the access port.

2.3 msg.actions.append(of.ofp_action_enqueue(port= x， queue_id=y))

Command：**msg.actions.append(of.ofp_action_enqueue(port=x,queue_id=y))**
Function：Forward the appointed port and queue.
Parameters：port，queue_id (queue number)
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_enqueue(port=x，enqueue_id=y))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_enqueue(port=13，queue_id=4))**

2.4 msg.actions.append(of.ofp_action_dl_addr.set_dst (“mac”))

Command：**msg.actions.append(of.ofp_action_dl_addr.set_dst(“”))**
Function：Change the destination mac to be the appointed mac.
Parameters：dst mac
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_dl_addr.set_dst(“11:11:11:11:11:11”))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_dl_addr.set_dst(“11:11:11:11:11:11”))**

2.5 msg.actions.append(of.ofp_action_dl_addr.set_src (“mac”))

Command：**msg.actions.append(of.ofp_action_dl_addr.set_src(“”))**
Function：Change the source mac to be the appointed mac.
Parameters：src mac
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_dl_addr.set_src(“00:03:11:11:11:11”))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_dl_addr.set_src(“00:03:11:11:11:11”))**

2.6 msg.actions.append(of.ofp_action_nw_tos(nw_tos=x))

Command：**msg.actions.append(of.ofp_action_nw_tos(nw_tos=))**
Function：Configure the tos value.
Parameters：nw_tos
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_nw_tos(nw_tos=56))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_nw_tos(nw_tos=56))**

2.7 msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=x))

Command：**msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=))**
Function：Configure the vlan value.
Parameters：vlan_vid
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=3))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=3))**

2.8 msg.actions.append(of.ofp_action_vlan_pcp(vlan_pcp=x))

Command：**msg.actions.append(of.ofp_action_vlan_pcp(vlan_pcp=))**
Function：Configure the vlan cos value.
Parameters：vlan cos
Default：None.
Command Mode：Interactive mode, pox>
Usage Guide：**msg.actions.append(of.ofp_action_vlan_pcp(vlan_pcp=3))**
Example：
Controller terminal:
POX> **msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=3))**
POX> **msg.actions.append(of.ofp_action_vlan_pcp(vlan_pcp=4))**
Explanation: the vlan id must be configured first before configure the cos value.

Chapter 3 OPENFLOW Configuration (examples)

3.1 add flow-match the access port, the action is the egress port

```
POX>import pox.openflow.libopenflow_01 as of
POX>msg2=of.ofp_flow_mod()
POX>msg2.priority=3
POX>msg2.match.in_port=193
POX>msg2.actions.append(of.ofp_action_output(port=194))
POX>core.openflow.connections[13136560386L].send(msg2)
```

3.2 add flow-match the destination mac, the action is the egress port

```
POX>import pox.openflow.libopenflow_01 as of
POX>msg2=of.ofp_flow_mod()
POX>msg2.priority=3
POX>msg2.match.dl_src=EthAddr("ff:ff:ff:ff:ff:ff")
POX>msg2.actions.append(of.ofp_action_output(port=194))
POX>core.openflow.connections[13136560386L].send(msg2)
```

3.3 add flow-match the type of Ethernet, the action is the egress port and queue

```
POX>msg=of.ofp_flow_mod()
POX>msg.priority=5
POX>msg.match.dl_type=0x800
POX>msg.actions.append(of.ofp_action_enqueue(queue_id=5,port=194))
POX>core.openflow.connections[13136560386L].send(msg)
```

3.4 add flow-match the source mac, the action is to configure the vlan and appoint the egress port

```
POX>msg=of.ofp_flow_mod()
POX>msg.priority=5
POX>msg.match.dl_src=EthAddr(“00:03:0f:01:12:43”)
POX>msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=3))
POX>msg.actions.append(of.ofp_action_output(port=194))
POX>core.openflow.connections[13136560386L].send(msg)
```

3.5 add flow-match the access port, the action is to configure the vlan and cos and appoint the egress port

```
POX>msg=of.ofp_flow_mod()
POX>msg.priority=5
POX>msg.match.in_port=193
POX>msg.actions.append(of.ofp_action_vlan_vid(vlan_vid=4))
POX>msg.actions.append(of.ofp_action_vlan_pcp(vlan_pcp=5))
POX>msg.actions.append(of.ofp_action_output(port=194))
POX>core.openflow.connections[13136560386L].send(msg)
```

3.6 del flow

```
POX>msg=of.ofp_flow_mod(command=3)
POX>core.openflow.connections[13136560386L].send(msg)
```

3.7 del flow-strict

```
POX>msg=of.ofp_flow_mod(command=4)
POX>msg.wildcards= 4194302
POX>msg.priority=5
POX>core.openflow.connections[13136560386L].send(msg)
```