

# Lab 9 (Traffic Measurement)

In this lab, the number of packets or bytes for IP or ARP will be recorded in Table 0. Then IP traffic will be forwarded to Table 5 for further classification. The number of packets or bytes for ICMP, TCP, or UDP will be recorded. Finally, all packets will be forwarded to table 10 for flooding.

Put this file (measure\_traffic.py) under /pox/ext

```
from pox.core import core
from pox.lib.addresses import EthAddr
import pox.openflow.libopenflow_01 as of
import pox.openflow.nicira as nx
from pox.lib.revent import EventRemove
from pox.lib.packet.arp import arp
from pox.lib.packet.ipv4 import ipv4
import pox.lib.packet as pkt

# Even a simple usage of the logger is much nicer than print!
log = core.getLogger()

def _handle_ConnectionUp (event):
    print "_handle_ConnectionUP"

    # Set up this switch.
    # After setting up, we send a barrier and wait for the response
    # before starting to listen to packet_ins for this switch -- before
    # the switch is set up, the packet_ins may not be what we expect,
    # and our responses may not work!

    # Turn on Nicira packet_ins
    msg = nx.nx_packet_in_format()
    event.connection.send(msg)

    # Turn on ability to specify table in flow_mods
    msg = nx.nx_flow_mod_table_id()
    event.connection.send(msg)
```

```
# Fallthrough rule for table 0: flood and send to controller
msg = nx.nx_flow_mod()
msg.priority = 1 # Low priority
msg.actions.append(of.ofp_action_output(port = of.OFPP_CONTROLLER))
msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 10))
event.connection.send(msg)
```

```
# Fallthrough rule for table 1: flood
msg = nx.nx_flow_mod()
msg.table_id = 10
msg.priority = 1 # Low priority
msg.actions.append(of.ofp_action_output(port = of.OFPP_FLOOD))
event.connection.send(msg)
```

```
def ready (event):
    if event.ofp.xid != 0x80000000:
        # Not the right barrier
        return
    log.info("%s ready", event.connection)
    event.connection.addListenerByName("PacketIn", _handle_PacketIn)
    return EventRemove
```

```
event.connection.send(of.ofp_barrier_request(xid=0x80000000))
event.connection.addListenerByName("BarrierIn", ready)
```

```
def _handle_PacketIn(event):
    print "_handle_PacketIn"
    packet = event.parsed

    if event.port > of.OFPP_MAX:
        log.debug("Ignoring special port %s", event.port)
        return

    print packet.src, "-->", packet.dst

    a=packet.find('arp')
    b=packet.find('ipv4')
    print a, b

    #print "ICMP Packet"
    msg = nx.nx_flow_mod()
    msg.table_id = 5
    msg.match.of_eth_type = pkt.ethernet.IP_TYPE
    msg.match.of_ip_proto = 1
    msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 10))
    event.connection.send(msg)

    #print "TCP Packet"
    msg = nx.nx_flow_mod()
    msg.table_id = 5
    msg.match.of_eth_type = pkt.ethernet.IP_TYPE
    msg.match.of_ip_proto = 6
    msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 10))
    event.connection.send(msg)
```

```
#print "UDP Packet"
msg = nx.nx_flow_mod()
msg.table_id = 5
msg.match.of_eth_type = pkt.ethernet.IP_TYPE
msg.match.of_ip_proto = 17
msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 10))
event.connection.send(msg)
```

```
#print "ARP Packet"
msg = of.ofp_flow_mod()
msg.priority = 100
msg.match.dl_type = 0x0806
msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 10))
event.connection.send(msg)
```

```
#print "IP Packet"
msg = of.ofp_flow_mod()
msg.priority = 100
msg.match.dl_type = 0x0800
msg.actions.append(nx.nx_action_resubmit.resubmit_table(table = 5))
event.connection.send(msg)
```

```
def launch ():
    assert core.NX, "Nicira extensions required"
    assert core.NX.convert_packet_in, "PacketIn conversion required"
    core.openflow.addListenerByName("ConnectionUp", _handle_ConnectionUp)

    log.info("Simple NX switch running.")
```

```
ubuntu@sdnhubvm:~/pox$ ./pox.py openflow.nicira --convert-packet-in measure_traffic
POX 0.1.0 (beta) / Copyright 2011-2013 James McCauley, et al.
INFO:measure_traffic:Simple NX switch running.
INFO:core:POX 0.1.0 (beta) is up.
```

H1----switch ---H2

The image shows two terminal windows. The top window, titled 'Terminal - ubuntu@sdnhubvm: ~/pox', displays a series of network traffic logs. Each log entry consists of a source MAC address, a destination MAC address, and the text 'None None' followed by a function call '\_handle\_PacketIn'. The source MAC addresses are 00:00:00:00:00:01 and 00:00:00:00:00:02, and the destination MAC addresses are 33:33:00:00:00:16 and 33:33:00:00:00:02. The bottom window, titled 'Terminal - ubuntu@sdnhubvm: ~', shows the execution of the 'mn' command to create a network topology. The output includes: '\*\*\* Creating network', '\*\*\* Adding controller', '\*\*\* Adding hosts: h1 h2', '\*\*\* Adding switches: s1', '\*\*\* Adding links: (h1, s1) (h2, s1)', '\*\*\* Configuring hosts h1 h2', '\*\*\* Starting controller', '\*\*\* Starting 1 switches s1', and '\*\*\* Starting CLI: mininet>'. The prompt 'mininet>' is visible at the end of the output.

```
Terminal - ubuntu@sdnhubvm: ~/pox
File Edit View Terminal Tabs Help
00:00:00:00:00:01 --> 33:33:00:00:00:16
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:02 --> 33:33:00:00:00:16
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:16
None None
_handle_PacketIn
00:00:00:00:00:02 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:02 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:02
None None
_handle_PacketIn
None None

Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$ sudo mn --topo single,2 --mac --controller=remote
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet>
```

Use dpctl command to directly check the information in switch

```
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0xe86f14b5): flags=none type=1(flow)
  cookie=0, duration_sec=95s, duration_nsec=734000000s, table_id=0, priority=1,
  n_packets=14, n_bytes=1116, idle_timeout=0,hard_timeout=0,actions=CONTROLLER:655
  35,vendor action:0x2320
  cookie=0, duration_sec=86s, duration_nsec=654000000s, table_id=0, priority=100
  , n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,ip,actions=vendor action
  :0x2320
  cookie=0, duration_sec=86s, duration_nsec=655000000s, table_id=0, priority=100
  , n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,arp,actions=vendor actio
  n:0x2320
  cookie=0, duration_sec=86s, duration_nsec=655000000s, table_id=5, priority=327
  68, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,icmp,actions=vendor ac
  tion:0x2320
  cookie=0, duration_sec=86s, duration_nsec=655000000s, table_id=5, priority=327
  68, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,udp,actions=vendor act
  ion:0x2320
  cookie=0, duration_sec=86s, duration_nsec=655000000s, table_id=5, priority=327
  68, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,tcp,actions=vendor act
  ion:0x2320
  cookie=0, duration_sec=95s, duration_nsec=734000000s, table_id=10, priority=1,
  n_packets=14, n_bytes=1116, idle_timeout=0,hard_timeout=0,actions=FLOOD
mininet>
```

Initially, no ip, arp, icmp, udp, and tcp traffic.

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.110 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.093 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.093/0.421/1.062/0.453 ms
```

After executing ping from h1 to h2,

```
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0x8de5d838): flags=none type=1(flow)
  cookie=0, duration_sec=284s, duration_nsec=383000000s, table_id=0, priority=1,
  n_packets=14, n_bytes=1116, idle_timeout=0,hard_timeout=0,actions=CONTROLLER:65
535,vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=303000000s, table_id=0, priority=10
0, n_packets=6, n_bytes=588, idle_timeout=0,hard_timeout=0,ip,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=0, priority=10
0, n_packets=4, n_bytes=168, idle_timeout=0,hard_timeout=0,arp,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=32
768, n_packets=6, n_bytes=588, idle_timeout=0,hard_timeout=0,icmp,actions=vendor
action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=32
768, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,udp,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=32
768, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,tcp,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=284s, duration_nsec=383000000s, table_id=10, priority=1
, n_packets=24, n_bytes=1872, idle_timeout=0,hard_timeout=0,actions=FLLOOD
mininet>
```

IP packets: 6 packets (3 ping request packets and 3 ping response packets.)

ARP packet: 4 packets

ICMP packets: 6 packets (3 ping request packets and 3 ping response packets.)

No UDP nor TCP traffic



Open terminals for h1 and h2

```
Terminal - ubuntu@sdn...
File Edit View Terminal Tabs Help
00:00:00:00:00:01 --> 33:33:00:00:00:16
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:02 --> 33:33:00:00:00:00
None None
_handle_PacketIn

Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0x8de5d838): flags=none type=1(flow)
  cookie=0, duration_sec=284s, duration_nsec=383000000s, table_id=0, priority=0, n_packets=14, n_bytes=1116, idle_timeout=0,hard_timeout=0,actions=CONTROLLED, vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=303000000s, table_id=0, priority=0, n_packets=6, n_bytes=588, idle_timeout=0,hard_timeout=0,ip,actions=vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=0, priority=0, n_packets=4, n_bytes=168, idle_timeout=0,hard_timeout=0,arp,actions=vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=768, n_packets=6, n_bytes=588, idle_timeout=0,hard_timeout=0,icmp,actions=vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=768, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,udp,actions=vendor action:0x2320
  cookie=0, duration_sec=275s, duration_nsec=304000000s, table_id=5, priority=768, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,tcp,actions=vendor action:0x2320
  cookie=0, duration_sec=284s, duration_nsec=383000000s, table_id=10, priority=0, n_packets=24, n_bytes=1872, idle_timeout=0,hard_timeout=0,actions=FLOOD
mininet> xterm h1 h2
mininet>
```

h1: iperf -c 10.0.0.2

h2: iperf -s

TCP traffic

The image shows three terminal windows. The top-left window is a standard Ubuntu terminal showing network interface statistics for eth0. The top-right window, titled 'Node: h1', shows the output of 'iperf -c 10.0.0.2', including connection details and a performance summary table. The bottom window, titled 'Terminal - ubuntu@sdnhubvm: ~', shows the output of 'iperf -s' and a 'mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634' command. The flow dump output lists several flows with their respective statistics, including 'n\_packets=1811636' and 'n\_bytes=43279097440' for a flow to IP 10.0.0.2.

```
Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
00:00:00:00:00:01 --> 33:33:00:00:00:16
None None
_handle_PacketIn
00:00:00:00:00:01 --> 33:33:00:00:00:02
None None
_handle_PacketIn
00:00:00:00:00:02 --> 33:33:00:00:00:16
None None
_handle_PacketIn

Node: h1
root@sdnhubvm:~# iperf -c 10.0.0.2
-----
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 10.0.0.1 port 56468 connected with 10.0.0.2 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 4]  0.0-10.0 sec  40.2 GBytes   34.5 Gbits/sec
root@sdnhubvm:~#

Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0x97a35ec6): flags=none type=1(flow)
  cookie=0, duration_sec=665s, duration_nsec=742000000s, table_id=0, priority=1,
  n_packets=14, n_bytes=1116, idle_timeout=0,hard_timeout=0,actions=CONTROLLER:65
535,vendor action:0x2320
  cookie=0, duration_sec=656s, duration_nsec=662000000s, table_id=0, priority=10
0, n_packets=1811636, n_bytes=43279097440, idle_timeout=0,hard_timeout=0,ip,acti
ons=vendor action:0x2320
  cookie=0, duration_sec=656s, duration_nsec=663000000s, table_id=0, priority=10
0, n_packets=6, n_bytes=252, idle_timeout=0,hard_timeout=0,arp,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=656s, duration_nsec=663000000s, table_id=5, priority=32
768, n_packets=6, n_bytes=588, idle_timeout=0,hard_timeout=0,icmp,actions=vendor
action:0x2320
  cookie=0, duration_sec=656s, duration_nsec=663000000s, table_id=5, priority=32
768, n_packets=0, n_bytes=0, idle_timeout=0,hard_timeout=0,udp,actions=vendor ac
tion:0x2320
  cookie=0, duration_sec=656s, duration_nsec=663000000s, table_id=5, priority=32
768, n_packets=1811630, n_bytes=43279096852, idle_timeout=0,hard_timeout=0,tcp,a
ctions=vendor action:0x2320
  cookie=0, duration_sec=665s, duration_nsec=742000000s, table_id=10, priority=1
, n_packets=1811656, n_bytes=43279098808, idle_timeout=0,hard_timeout=0,actions=
FLOOD
mininet>
```

IP: 1811636 (TCP:1811630 packets + ICMP: 6 packets)

TCP: 1811630 packets

h1: iperf -c 10.0.0.2 -u

h2: iperf -s -u

UDP traffic

The image shows three terminal windows. The top-left window is a standard Ubuntu terminal with a menu bar (File, Edit, View, Terminal, Tabs, Help) and shows the execution of 'iperf -c 10.0.0.2 -u'. The top-right window, titled 'Node: h1', shows the server-side output of 'iperf -s -u', including 'Client connecting to 10.0.0.2, UDP port 5001' and 'Sending 1470 byte datagrams'. The bottom window, titled 'Terminal - ubuntu@sdnhubvm: ~', shows the execution of 'mininet> s1 dpctl dump-flows tcp:127.0.0.1:6634', which outputs a list of network flows. Several flows are highlighted with red boxes, showing 'n\_packets=1812531' and 'n\_packets=895'.

IP: 1812531 packets(UDP: 895 TCP:1811630 packets + ICMP: 6 packets)

UDP: 895 packets

## References

- POX Wiki, Nicira/Open vSwitch Extensions

<https://openflow.stanford.edu/display/ONL/POX+Wiki>

Dr. Chih-Heng Ke

<http://csie.nqu.edu.tw/smallko>

SDN website: <http://csie.nqu.edu.tw/smallko/sdn/sdn.htm>