



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Διαχείριση Δικτύων - Ευφυή Δίκτυα

7η Εργαστηριακή Άσκηση

Βρεττός Κωνσταντίνος

A.M: 03119856

Maria_username: netmg026

Άσκηση

- 1) Αφού εγκαταστήσαμε και εισάγαμε την εικονική μηχανή συνδεθήκαμε σε αυτή με τους κωδικούς της εκφώνησης (Username: netman, Password: netman2020) και περάσαμε με DHCP διεύθυνση ip και εκτελώντας την εντολή "ip addr" βλέπουμε ότι η ip που καταχωρήθηκε στο μηχάνημα είναι η 192.168.2.7/24, με αυτή την ip συνδεόμαστε στη συνέχεια μέσω SSH στο μηχάνημα αυτό.
- 2) Αφού συνδεθήκαμε με SSH στο μηχάνημα εκτελέσαμε τις εντολές:
 - **cd /home/netman/ryu/**
 - **python ./bin/ryu-manager ./ryu/app/simple_switch.py**

Με αυτόν τον τρόπο ξεκινήσαμε τον Openflow Controller,

```

netman@debian-ok:~$ cd /home/netman/ryu/
netman@debian-ok:~/ryu$ python ./bin/ryu-manager ./ryu/app/simple_switch.py
loading app ./ryu/app/simple_switch.py
loading app ryu.controller.ofp_handler
instantiating app ./ryu/app/simple_switch.py of SimpleSwitch
instantiating app ryu.controller.ofp_handler of OFPHandler
port added 1
port added 3
port added 2
packet in 1 00:00:00:00:00:01 33:33:00:00:00:16 1
packet in 1 00:00:00:00:00:02 33:33:ff:00:00:02 2
packet in 1 00:00:00:00:00:03 33:33:00:00:00:16 3
packet in 1 00:00:00:00:00:02 33:33:00:00:00:16 2
packet in 1 00:00:00:00:00:01 33:33:ff:00:00:01 1
packet in 1 00:00:00:00:00:03 33:33:ff:00:00:03 3
port modified 1
packet in 1 00:00:00:00:00:02 33:33:00:00:00:16 2
packet in 1 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 1 00:00:00:00:00:01 33:33:00:00:00:16 1
packet in 1 00:00:00:00:00:01 33:33:00:00:00:02 1
packet in 1 00:00:00:00:00:03 33:33:00:00:00:16 3
packet in 1 00:00:00:00:00:03 33:33:00:00:00:02 3
packet in 1 00:00:00:00:00:02 33:33:00:00:00:16 2

```

3) Στη συνέχεια συνδεθήκαμε εκ νέου με SSH στο προηγούμενο μηχάνημα και εκτελώντας την εντολή:

- **sudo mn --topo single,3 --mac --switch ovsk --controller remote,ip=127.0.0.1,port=6633**

Ξεκινάμε τον εξομοιωτή Mininet

```

netman@debian-ok:~$ sudo mn --topo single,3 --mac --switch ovsk --controller rem
ote,ip=127.0.0.1,port=6633
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:

```

4) (Α) Εκτελώντας την εντολή “s1 ovs-vsctl show” στο Mininet cli :

```

mininet> s1 ovs-vsctl show
ecf5997b-e954-4592-826a-c2fd43bf9093
    Bridge "s1"
        Controller "tcp:127.0.0.1:6633"
            is_connected: true
        Controller "ptcp:6654"
        fail_mode: secure
        Port "s1-eth3"
            Interface "s1-eth3"
        Port "s1"
            Interface "s1"
                type: internal
        Port "s1-eth2"
            Interface "s1-eth2"
        Port "s1-eth1"
            Interface "s1-eth1"
    ovs_version: "2.3.0"

```

Εδώ παρατηρούμε ότι έχουμε ένα OpenFlow switch το οποίο ακούει στην TCP πόρτα **“ptcp : 6654”** και έχει 4 ports τις :

- **“s1-eth3”**
- **“s1”**
- **“s1-eth2”**
- **“s1-eth1”**

(B) Παρατηρούμε ότι στην τοπολογία μας έχουμε: c0=Controller, 3 hosts τους h1, h2 και h3 και ένα Switch s1.

```
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s1-eth3 (OK OK)
```

Εκτελώντας τις εντολές **“h1 ifconfig -a”** για τον host h1 **“h2 ifconfig -a”** για τον host h2 και **“h3 ifconfig -a”** για τον host h3 βλέπουμε

- h1 : ip = 10.0.0.1
- h2 : ip = 10.0.0.2
- h3 : ip = 10.0.0.3

5)

(A) Εκτελώντας με τη σειρά τις εντολές **“h1 ping -c 3 h2”** ,

“s1 dpctl dump-flows tcp:127.0.0.1:6654” λαμβάνουμε τις εξής πληροφορίες :

```
mininet> s1 dpctl dump-flows tcp:127.0.0.1:6654
stats_reply (xid=0xffb2d2e1): flags=none type=1(flow)
  cookie=0, duration_sec=2770s, duration_nsec=62000000s, table_id=0, priority=32768, n_packet
s=8, n_bytes=672, idle_timeout=0,hard_timeout=0,in_port=2,d1_dst=00:00:00:00:00:01,actions=ou
tput:1
  cookie=0, duration_sec=2770s, duration_nsec=54000000s, table_id=0, priority=32768, n_packet
s=6, n_bytes=532, idle_timeout=0,hard_timeout=0,in_port=1,d1_dst=00:00:00:00:00:02,actions=ou
tput:2
```

- n_packets = 8 (Number of packets)
- N_bytes = 672 (Packet size)
- in_port = 2 (Input Switch port)
- Actions = output:1 (Matching packets are being forwarded to port #1)

- n_packets = 6 (Number of packets)
- N_bytes = 532 (Packet size)

- in_port = 1 (Input Switch port)
- Actions = output:2 (Matching packets are being forwarded to port #2)

(B) Έχοντας τερματίσει την εφαρμογή RYU Controller εκτελούμε την εντολή “**h1 ping -c 3 h3**” και αυτό που παρατηρούμε είναι ότι το ping από τον Host h1 προς τον Host h3 βγάζει μήνυμα σφάλματος “ **Destination Host Unreachable** ”, αντιθέτως κάνοντας ping στον Host h2 όπως κάναμε και στο προηγούμενο ερώτημα το ping είναι επιτυχές. Αυτό συμβαίνει καθώς όσο είχαμε ενεργοποιημένο το RYU Controller στο προηγούμενο ερώτημα ο h1 “έμαθε” την MAC address του h2, μετά αφού κλείσαμε τον RYU Controller δεν είναι εφικτό να γίνει MAC Resolution στον h3.

```
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=0.310 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.052 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.052/0.138/0.310/0.121 ms
mininet> h1 ping -c 3 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable

--- 10.0.0.3 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 1999ms
pipe 3
```

(Γ) Παρατηρούμε να γίνεται MAC Resolution μεταξύ των κόμβων που είχαν ανταλλάξει μήνυμα εώς τώρα.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

```
netman@debian-ok:~/ryu$ python ./bin/ryu-manager ./ryu/app/simple_switch.py
loading app ./ryu/app/simple_switch.py
loading app ryu.controller.ofp_handler
instantiating app ./ryu/app/simple_switch.py of SimpleSwitch
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
packet in 1 00:00:00:00:00:03 00:00:00:00:00:01 3
packet in 1 00:00:00:00:00:01 00:00:00:00:00:03 1
packet in 1 00:00:00:00:00:02 ff:ff:ff:ff:ff:ff 2
packet in 1 00:00:00:00:00:03 00:00:00:00:00:02 3
packet in 1 00:00:00:00:00:02 00:00:00:00:00:03 2
```

Στο Flow Table του Switch βλέπουμε ότι δημιουργούνται 6 κανόνες για την επικοινωνία μεταξύ των hosts, 1 κανόνας για κάθε ζεύγος από hosts.

```

mininet> sl dpctl dump-flows tcp:127.0.0.1:6654
stats_reply (xid=0x418a2874): flags=none type=1(flow)
  cookie=0, duration_sec=1463s, duration_nsec=61000000s, table_id=0, priority=32768, n_packets=3, n_bytes=238, idle_timeout=0,hard_timeout=0,in_port=3,d1_dst=00:00:00:00:00:02,actions=output:2
  cookie=0, duration_sec=7791s, duration_nsec=178000000s, table_id=0, priority=32768, n_packets=20, n_bytes=1624, idle_timeout=0,hard_timeout=0,in_port=2,d1_dst=00:00:00:00:00:01,actions=output:1
  cookie=0, duration_sec=1463s, duration_nsec=76000000s, table_id=0, priority=32768, n_packets=3, n_bytes=238, idle_timeout=0,hard_timeout=0,in_port=3,d1_dst=00:00:00:00:00:01,actions=output:1
  cookie=0, duration_sec=1463s, duration_nsec=59000000s, table_id=0, priority=32768, n_packets=2, n_bytes=140, idle_timeout=0,hard_timeout=0,in_port=2,d1_dst=00:00:00:00:00:03,actions=output:3
  cookie=0, duration_sec=1463s, duration_nsec=73000000s, table_id=0, priority=32768, n_packets=2, n_bytes=140, idle_timeout=0,hard_timeout=0,in_port=1,d1_dst=00:00:00:00:00:03,actions=output:3
  cookie=0, duration_sec=7791s, duration_nsec=170000000s, table_id=0, priority=32768, n_packets=18, n_bytes=1484, idle_timeout=0,hard_timeout=0,in_port=1,d1_dst=00:00:00:00:00:02,actions=output:2

```

6)

(Α) Εκτελώντας την εντολή **“curl -X GET http://localhost:8080/stats/flow/1 “** την οποία βρήκαμε στο URL της εκφώνησης παίρνουμε τα εξής flows στο OpenFlow Table του Switch:

```

{"1": [{"actions": ["OUTPUT:2"], "idle_timeout": 0, "cookie": 0, "packet_count": 3,
"hard_timeout": 0, "byte_count": 238, "duration_sec": 3281, "duration_nsec": 502000000,
"priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":
"00:00:00:00:00:02", "in_port": 3}},

```

```

{"actions": ["OUTPUT:1"], "idle_timeout": 0, "cookie": 0, "packet_count": 20,
"hard_timeout": 0, "byte_count": 1624, "duration_sec": 9609, "duration_nsec":
619000000, "priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":
"00:00:00:00:00:01", "in_port": 2}},

```

```

{"actions": ["OUTPUT:1"], "idle_timeout": 0, "cookie": 0, "packet_count": 3,
"hard_timeout": 0, "byte_count": 238, "duration_sec": 3281, "duration_nsec": 517000000,
"priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":
"00:00:00:00:00:01", "in_port": 3}},

```

```

{"actions": ["OUTPUT:3"], "idle_timeout": 0, "cookie": 0, "packet_count": 2,
"hard_timeout": 0, "byte_count": 140, "duration_sec": 3281, "duration_nsec": 500000000,
"priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":
"00:00:00:00:00:03", "in_port": 2}},

```

```

{"actions": ["OUTPUT:3"], "idle_timeout": 0, "cookie": 0, "packet_count": 2,
"hard_timeout": 0, "byte_count": 140, "duration_sec": 3281, "duration_nsec": 514000000,
"priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":
"00:00:00:00:00:03", "in_port": 1}},

```

```

{"actions": ["OUTPUT:2"], "idle_timeout": 0, "cookie": 0, "packet_count": 18,
"hard_timeout": 0, "byte_count": 1484, "duration_sec": 9609, "duration_nsec":

```

```
611000000, "priority": 32768, "length": 96, "flags": 1, "table_id": 0, "match": {"dl_dst":  
"00:00:00:00:00:02", "in_port": 1}}}]}
```

(B)

```
{"actions": [],  
"idle_timeout": 30 ,  
"cookie": 1 ,  
"packet_count": ,  
"hard_timeout":30 ,  
"priority": 65535 ,  
"flags": 1 ,  
"table_id": 0 ,  
"match": {  
    "dl_src": 00:00:00:00:00:01 "      #src MAC  
    "dl_dst": 00:00:00:00:00:03 ",    #dst MAC  
}  
"in_port": 1  
}  
}
```