Project #2

Yang Cao
C14189452

1. Setup Floodlight and Test Environment

- **Prerequisites**

First of all, we need to configure the fundamental environment for this assignment. The following screenshots display some details of installation.

```
vm@vm-VirtualBox:~$ sudo apt install build-essential ant maven python-dev -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
```

```
vm@vm-VirtualBox:~$ sudo apt install software-properties-common git jq
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.20.5).
```

```
vm@vm-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java -y
gpg: keyring `/tmp/tmpr1_3m4sc/secring.gpg' created
gpg: keyring `/tmp/tmpr1_3m4sc/pubring.gpg' created
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpr1_3m4sc/trustdb.gpg: trustdb created
gpg: key EEA14886: public key "Launchpad VLC" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
OK
vm@vm-VirtualBox:~$ sudo apt update
```

```
vm@vm-VirtualBox:~$ sudo apt install oracle-java8-installer -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
vm@vm-VirtualBox:~$ sudo apt install oracle-java8-set-default -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
oracle-java8-set-default is already the newest version (8u121-1~webupd8~2).
oracle-java8-set-default set to manually installed.
The following package was automatically installed and is no longer required:
  snap-confine
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- **Install Floodlight**

```
vm@vm-VirtualBox: ~
vm@vm-VirtualBox:~$ git clone git://github.com/floodlight/floodlight.git
Cloning into 'floodlight'...
remote: Counting objects: 44354, done.
remote: Total 44354 (delta 0), reused 0 (delta 0), pack-reused 44354
Receiving objects: 100% (44354/44354), 350.09 MiB | 5.03 MiB/s, done.
Resolving deltas: 100% (27213/27213), done.
Checking connectivity... done.
vm@vm-VirtualBox:~$ cd floodlight
vm@vm-VirtualBox:~/floodlight$ git submodule init
Submodule 'src/main/resources/web' (https://github.com/floodlight/floodlight-webui) reg
istered for path 'src/main/resources/web'
vm@vm-VirtualBox:~/floodlight$ git submodule update
Cloning into 'src/main/resources/web'...
remote: Counting objects: 1314, done.
remote: Total 1314 (delta 0), reused 0 (delta 0), pack-reused 1314
Receiving objects: 100% (1314/1314), 3.70 MiB | 4.05 MiB/s, done.
Resolving deltas: 100% (353/353), done.
Checking connectivity... done.
Submodule path 'src/main/resources/web': checked out '580bf06fd86bb7ff270019447f023f9d9
8e431d9'
vm@vm-VirtualBox:~/floodlight$ ant
Buildfile: /home/vm/floodlight/build.xml
  [taskdef] Could not load definitions from resource tasks.properties. It could not be
found.

init:
    [mkdir] Created dir: /home/vm/floodlight/target/bin
    [mkdir] Created dir: /home/vm/floodlight/target/bin-test
    [mkdir] Created dir: /home/vm/floodlight/target/lib
    [mkdir] Created dir: /home/vm/floodlight/target/test

compile:
    [javac] Compiling 538 source files to /home/vm/floodlight/target/bin
    [javac] Note: Some input files use or override a deprecated API.
    [javac] Note: Recompile with -Xlint:deprecation for details.
    [javac] Note: Some input files use unchecked or unsafe operations.
```

- **Install Mininet**

```
vm@vm-VirtualBox:~/floodlight$ sudo apt-get install bridge-utils -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snap-confine
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  bridge-utils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.6 kB of archives.
After this operation, 102 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 bridge-utils amd64 1.5-9ubu
ntu1 [28.6 kB]
Fetched 28.6 kB in 0s (230 kB/s)
Selecting previously unselected package bridge-utils.
(Reading database ... 211655 files and directories currently installed.)
Preparing to unpack .../bridge-utils_1.5-9ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-9ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up bridge-utils (1.5-9ubuntu1) ...
vm@vm-VirtualBox:~/floodlight$ sudo apt-get install mininet -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Mininet is used to simulate the holistic SDN network. Floodlight is regarded as an SDN controller. The controller IP is 127.0.0.1 and the port is 6653. In this project, I distribute six hosts connected to one switch and enable the static ARP. The following screenshot shows executing Mininet command.

```
vm@vm-VirtualBox:~$ sudo mn --arp --controller=remote,ip=127.0.0.1,port=6653 --switch o
vsk,protocols=OpenFlow13 --topo single,6
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

Then we should run the Floodlight.

```
vm@vm-VirtualBox:~/floodlight$ java -jar target/floodlight.jar
2017-03-15 19:01:58.337 INFO  [n.f.c.m.FloodlightModuleLoader] Loading modules f
rom src/main/resources/floodlightdefault.properties
2017-03-15 19:01:58.514 WARN  [n.f.r.RestApiServer] HTTPS disabled; HTTPS will n
ot be used to connect to the REST API.
2017-03-15 19:01:58.514 WARN  [n.f.r.RestApiServer] HTTP enabled; Allowing unsec
ure access to REST API on port 8080.
2017-03-15 19:01:58.514 WARN  [n.f.r.RestApiServer] CORS access control allow AL
L origins: true
2017-03-15 19:01:58.723 WARN  [n.f.c.i.OFSwitchManager] SSL disabled. Using unse
cure connections between Floodlight and switches.
2017-03-15 19:01:58.723 INFO  [n.f.c.i.OFSwitchManager] Clear switch flow tables
 on initial handshake as master: TRUE
2017-03-15 19:01:58.723 INFO  [n.f.c.i.OFSwitchManager] Clear switch flow tables
 on each transition to master: TRUE
2017-03-15 19:01:58.730 INFO  [n.f.c.i.OFSwitchManager] Setting 0x1 as the defau
lt max tables to receive table-miss flow
2017-03-15 19:01:58.785 INFO  [n.f.c.i.OFSwitchManager] OpenFlow version OF_15 w
ill be advertised to switches. Supported fallback versions [OF_10, OF_11, OF_12,
 OF_13, OF_14, OF_15]
2017-03-15 19:01:58.786 INFO  [n.f.c.i.OFSwitchManager] Listening for OpenFlow s
witches on [0.0.0.0]:6653
```

We can check the running status of Floodlight and corresponding network topology diagram by visiting http://localhost:8080/ui/pages/index.html.

Controller

Active
Controller Status

00:19:22
Uptime (HH-mm-ss)

ACTIVE
Controller Role Change

1
Switches
See All

6
Hosts
See All

0
Connections (Links)
See All

0
Reserved Ports
See All

Switches

Switches Connected

Switch ID	IPv4 Address	Connected Since
00:00:00:00:00:00:00:01	/127.0.0.1:58270	Wed Mar 15 2017 23:19:19 GMT-0400 (EDT)

Showing 1 to 1 of 1 entries

Switch Roles

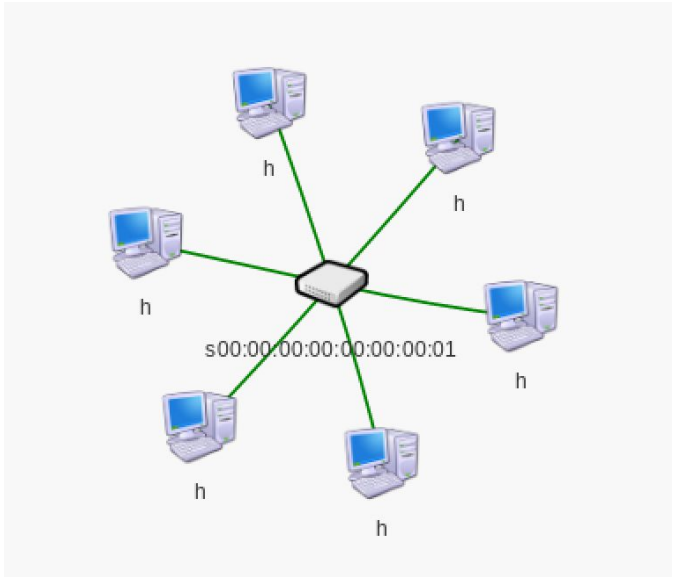
Switch MAC	Role
00:00:00:00:00:00:00:01	MASTER

Hosts

Hosts Connected

MAC	IPv4 Address	IPv6 Address	Switch	Port	Last Seen
0a:0d:3e:7b:3a:a1			00:00:00:00:00:00:00:01	2	1489624306803
26:f5:a2:dd:23:a8			00:00:00:00:00:00:00:01	1	1489624301767
92:c4:e6:0d:5f:b3			00:00:00:00:00:00:00:01	6	1489624326911
ae:8b:95:a1:61:bc			00:00:00:00:00:00:00:01	4	1489624770938
c2:12:2a:1f:b8:8d			00:00:00:00:00:00:00:01	5	1489624887432
c2:5b:78:35:ea:13			00:00:00:00:00:00:00:01	3	1489624311823

Showing 1 to 6 of 6 entries



2. Floodlight Firewall Examples

- **E.g.1**

I use “pingall” command for test before the firewall is enabled.

```
vm@vm-VirtualBox: ~
vm@vm-VirtualBox:~$ sudo apt install curl
[sudo] password for vm:
Sorry, try again.
[sudo] password for vm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snap-confine
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 139 kB of archives.
After this operation, 338 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 curl amd64 7
.47.0-1ubuntu2.2 [139 kB]
Fetched 139 kB in 0s (605 kB/s)
Selecting previously unselected package curl.
(Reading database ... 212052 files and directories currently installed.)
Preparing to unpack .../curl_7.47.0-1ubuntu2.2_amd64.deb ...
Unpacking curl (7.47.0-1ubuntu2.2) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up curl (7.47.0-1ubuntu2.2) ...
vm@vm-VirtualBox:~$ curl http://localhost:8080/wm/firewall/module/status/json
{"result" : "firewall disabled"}vm@vm-VirtualBox:~$
```

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
```

- **E.g.2**

After I enabled the firewall, it failed to run “pingall” test. The major reason is firewall denies all traffic by default unless an explicit “ALLOW” rule is added.

```
vm@vm-VirtualBox:~$ curl http://localhost:8080/wm/fire
wall/module/enable/json -X PUT -d ''
{"status" : "success", "details" : "firewall running"}
vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
*** Results: 100% dropped (0/30 received)
mininet>
```

- **E.g.3**

Then I created an ALLOW rule for all flows, so these flows can pass through switch 00:00:00:00:00:00:00:01.

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"switchid": "00:00:00:00:00:00:00:01"
}' http://localhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "-1503663245"}vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet>
```

- **E.g.4**

Before I created ALLOW rules for all flows between IP host 10.0.0.3 and host 10.0.0.6, I need to delete the previous rules.

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "1
0.0.0.6/32"}' http://localhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "998527747"}vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.6/32", "dst-ip": "1
0.0.0.3/32"}' http://localhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "1689928067"}vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X h6
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X h3 X X
*** Results: 93% dropped (2/30 received)
mininet>
```

- **E.g.5**

Before I created ALLOW rules for all flows between host mac 0a:0d:3e:7b:3a:a1 and host 26:f5:a2:dd:23:a8, I need to delete the previous rules. We can find only the two hosts can send packets to each other.

```
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "1689928067"}' http://loc
alhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "998527747"}' http://loca
lhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-mac": "0a:0d:3e:7b:3a:a1", "dst-
mac": "26:f5:a2:dd:23:a8"}' http://localhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "1106625819"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-mac": "26:f5:a2:dd:23:a8", "dst-
mac": "0a:0d:3e:7b:3a:a1"}' http://localhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "-2116617187"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X h5 X
h5 -> X X X h4 X
h6 -> X X X X X
*** Results: 93% dropped (2/30 received)
mininet>
```

- **E.g.6**

Before I created ALLOW rules for ping to work between IP hosts 10.0.0.3 and 10.0.0.6, I need to delete the previous rules. We can find only the two hosts can send ARP and ICMP packets to each other.

```
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "1106625819"}' http://loc
alhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "-2116617187"}' http://lo
calhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "1
0.0.0.6/32", "dl-type":"ARP" }'   http://localhost:8080/wm/firewall/rules/j
son
{"status" : "Rule added", "rule-id" : "1672238653"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.6/32", "dst-ip": "1
0.0.0.3/32", "dl-type":"ARP" }'   http://localhost:8080/wm/firewall/rules/j
son
{"status" : "Rule added", "rule-id" : "510847037"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "1
0.0.0.6/32", "nw-proto":"ICMP" }' http://localhost:8080/wm/firewall/rules/j
son
{"status" : "Rule added", "rule-id" : "961217820"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.6/32", "dst-ip": "1
0.0.0.3/32", "nw-proto":"ICMP" }' http://localhost:8080/wm/firewall/rules/j
son
{"status" : "Rule added", "rule-id" : "-1577142180"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X h6
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X h3 X X
*** Results: 93% dropped (2/30 received)
mininet>
```

On Mininet host, I open two XTerm to test the TCP and UDP packets transfer between the two hosts. We can find TCP or UDP packets can not be sent between the two hosts.

```
mininet> xterm h4
mininet> xterm h5
```

```
"Node: h4"
root@vm-VirtualBox:~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------

"Node: h5"
root@vm-VirtualBox:~# iperf -c 10.0.0.4
```

```
"Node: h4"
root@vm-VirtualBox:~# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------

"Node: h5"
root@vm-VirtualBox:~# iperf -u -c 10.0.0.4
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.5 port 41555 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 21] Sent 893 datagrams
[ 21] WARNING: did not receive ack of last datagram after 10 tries.
root@vm-VirtualBox:~#
```

- **E.g.7**

Before I created ALLOW rules for UDP (such as iperf) to work between IP hosts 10.0.0.4 and 10.0.0.5, and then blocking destination port 5010, I need to delete the previous rules.

```
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "1672238653"}' http://loc
alhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "510847037"}' http://loca
lhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "961217820"}' http://loca
lhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}
curl: option -X: requires parameter
curl: try 'curl --help' or 'curl --manual' for more information
vm@vm-VirtualBox:~$ curl -X DELETE -d '{"ruleid": "-1577142180"}' http://lo
calhost:8080/wm/firewall/rules/json
{"status" : "Rule deleted"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$
```

```
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "1
0.0.0.5/32", "dl-type":"ARP" }' http://localhost:8080/wm/firewall/rules/jso
n
{"status" : "Rule added", "rule-id" : "2048090813"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.5/32", "dst-ip": "1
0.0.0.4/32", "dl-type":"ARP" }' http://localhost:8080/wm/firewall/rules/jso
n
{"status" : "Rule added", "rule-id" : "1279468925"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "1
0.0.0.5/32", "nw-proto":"UDP" }' http://localhost:8080/wm/firewall/rules/js
on
{"status" : "Rule added", "rule-id" : "650662348"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.5/32", "dst-ip": "1
0.0.0.4/32", "nw-proto":"UDP" }' http://localhost:8080/wm/firewall/rules/js
on
{"status" : "Rule added", "rule-id" : "437771148"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "1
0.0.0.5/32", "nw-proto":"UDP", "tp-dst":"5010", "action":"DENY" }' http://l
ocalhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "1420166160"}vm@vm-VirtualBox:~$
vm@vm-VirtualBox:~$ curl -X POST -d '{"src-ip": "10.0.0.5/32", "dst-ip": "1
0.0.0.4/32", "nw-proto":"UDP", "tp-dst":"5010", "action":"DENY" }' http://l
ocalhost:8080/wm/firewall/rules/json
{"status" : "Rule added", "rule-id" : "651544272"}vm@vm-VirtualBox:~$
```

```
mininet> pingall 1
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
*** Results: 100% dropped (0/30 received)
mininet>
```

On Mininet host, I run the following code to open two XTerm to test the TCP and UDP packets transfer between the two hosts.

```
"Node: h4"
root@vm-VirtualBox:~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------

"Node: h5"
root@vm-VirtualBox:~# iperf -c 10.0.0.4
```

```
"Node: h4"
root@vm-VirtualBox:~# iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.4 port 5001 connected with 10.0.0.5 port 45385
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[ 21]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.089 ms    0/  893 (0%)

"Node: h5"
root@vm-VirtualBox:~# iperf -u -c 10.0.0.4
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.5 port 45385 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 21] Sent 893 datagrams
[ 21] Server Report:
[ 21]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.088 ms    0/  893 (0%)
root@vm-VirtualBox:~#
```

```
"Node: h4"
root@vm-VirtualBox:~# iperf -u -s -p 5010
------------------------------------------------------------
Server listening on UDP port 5010
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
```

```
"Node: h5"
root@vm-VirtualBox:~# iperf -u -c 10.0.0.4 -p 5010
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5010
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[ 21] local 10.0.0.5 port 55983 connected with 10.0.0.4 port 5010
[ ID] Interval       Transfer     Bandwidth
[ 21]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 21] Sent 893 datagrams
[ 21] WARNING: did not receive ack of last datagram after 10 tries.
root@vm-VirtualBox:~#
```

We find that TCP packets can not be sent between the two hosts. However, we can send UDP packet successfully, and all the UDP ACK with the 5010 destination port between the two host are dropped.