

Lab Assignment 3 for ECE671

Posted: 03/16/21

Due: 04/01/21

In this assignment you are tasked to implement the functionality of a Network Address Translation Gateway in the RYU OpenFlow controller and use Mininet to confirm that it works correctly.

As a starting point for this assignment use the VM from the last assignment, which, by now should have Mininet installed.

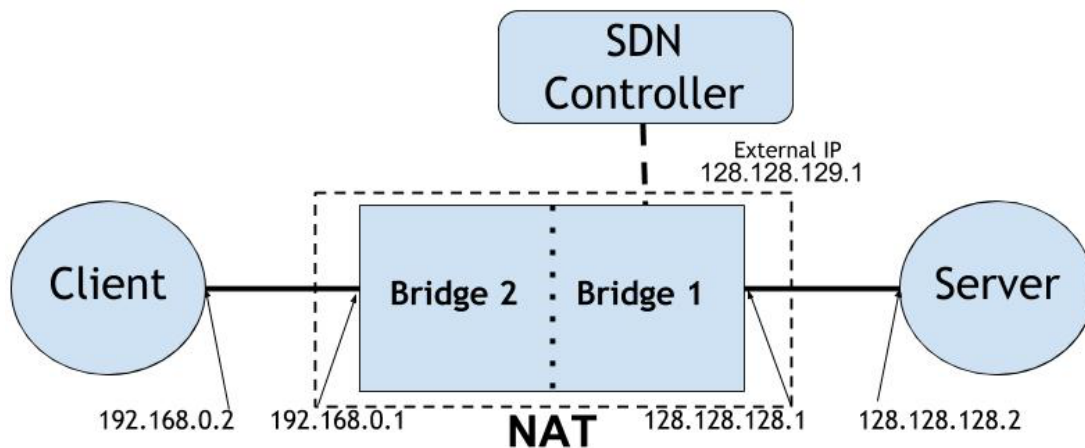


Figure 1: Topology for NAT assignment

Step 1: Mininet

Download the topology file and use it in Mininet to create the topology shown above:

```
$wget http://emmy10.casa.umass.edu/NAT_topo.py
$sudo mn --custom ~/mininet/custom/NAT_topo.py --topo
mytopo --controller remote
```

Note: **h1** is the client and **h1** is the server.

Step 2: Environment Setup

Step 2a: On the NAT

Download script to setup the switch:

```
$wget http://emmy10.casa.umass.edu/switch-setup-ryu
```

Create two bridges: OVS1 (bridge 1 in the figure), OVS2 (bridge 2 in the figure). You will see two warnings complaining deprecated use of port, ignore them.

```
$sudo bash switch-setup-ryu
```

Enable IP forwarding on localhost to allow packets to pass between two bridges.

```
$sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

If the command above is denied by permission, use this command instead.

```
$sudo bash -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
```

Step 2b: Adding static routes on the Server and the Client

You will need to setup static Linux routes for these hosts. Add static routes in the routing tables of both the server and client to ensure that these hosts know where to forward packets to. Remember that it is only the switch/NAT box that is connected to the controller and NOT the hosts.

Refer to the following link for help on Linux static routing commands:

<http://www.cyberciti.biz/tips/configuring-static-routes-in-debian-or-red-hat-linux-systems.html>

*Hints:

- 1) Create files server-setup at the server and client-setup at the client in the of-setup. It's not necessary to create these scripts but writing scripts will help repetitively executing the commands.
- 2) Also bear in mind that you will need to run these commands each time you cleanup and restart your Mininet experiment while testing your NAT controller. This is where the scripts would come on handy.

Step 3: Setup NAT

Download the template NAT file using the following command:

```
$wget http://emmy10.casa.umass.edu/nat_template.py
```

Save file with your name, i.e., **NAT_controller_yourfirst_yourlast.py**

Once you have modified this file to incorporate NAT functionality for TCP and UDP traffic, use a terminal on your Mininet VM to run the script using the following commands to start the RYU controller:

```
$cd $HOME/ryu
```

```
$/bin/ryu-manager NAT_controller_yourfirst_yourlast.py
```

Note1: You will need to start your controller before you run the Mininet command in Step 1. This ensures that the RYU controller is listening on port 6653 and waits for Bridge1 to connect to it.

Note2: If **ryu-manager** has difficulty to locate your controller module, it could be that your controller is not able to be parsed as python. Meaning you have syntax error in it.

You can use any TCP/UDP traffic to verify NAT functionality. The *netcat* (For details: <http://netcat.sourceforge.net/>) application is a good example:

On the server run:

```
$nc -l 6666
```

On the client:

```
$nc 128.128.128.2 6666
```

You should be able to type a message in the client and see it appear on the server. Similarly, you can use Iperf (<https://iperf.fr/iperf-doc.php>) to verify NAT functionality.

Submission:

1. Controller file named as **NAT_controller_yourfirst_yourlast.py**
2. The screenshots of route table of Client (h1), Server (h2), and Switch (s3).
3. The screenshot of proof of working TCP/UDP traffic between client and server, such as the use of *netcat*.

Demonstration:

Please arrange a Zoom meeting on either 04/08, 04/09, or 04/10 with Prof. Zink to demonstrate your work. No-show will result in 50% grade penalty on this lab assignment.