Computer Comm & Networks - ITCS 6166/8166

Assignment - 4 **Due: 05/02/2019 11:59pm**

Goal: In this assignment, you will build an SDN application for implementing shortest path routing using RYU SDN controller. You'll also learn how to verify the network performance using ping and iperf.

Discussion Topic: Shortest Path -RYU SDN controller: Post any questions and collaborate with your peers.

Overview:

You should familiarize yourself with shortest path routing algorithms. Understand types of OpenFlow messages exchanged between the switch and controller. In particular, you should be able identify message type and the action that needs to be taken by the controller for the message and how you can utilize them to write your routing application.

Some key message types are OF-PacketIn, OF-PacketOut, OF-PacketMod:

• OF-PacketIn – Message sent by the switch to controller for requesting forwarding decision (flow rule)

• OF-PacketOut – Message sent by the controller to switch to forward the packet using the port on the switch

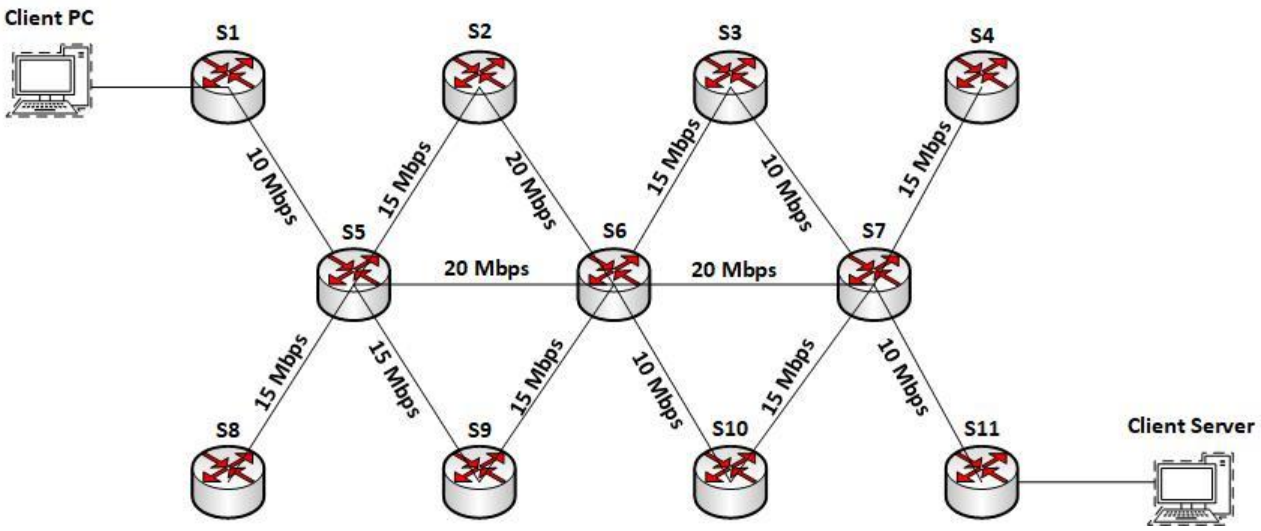
• OF-FlowMod – Message sent by the controller to switch with instruction to ADD/DELETE/MODIFY the existing forwarding decision (flow rule) on the flow table.

The switch will use OF-PacketIn messages to inform the controller whenever it receives a unknown destination MAC address.

Your RYU forwarding application should be able to install forwarding rule (flow rule) to the switches to enable communication between all the hosts. You should make use of OF-PacketIn, OF-PacketOut and OF-FlowMod to write this forwarding application.

Steps:

- Fire up your mininet python script which contains your network topology
- Start your RYU controller with `--verbose` option to enable logging



You are working as a Networking Administrator for one of the leading ISP. One of your client have requested you to route their traffic that provides maximum bandwidth/least delay between their Client server and Client PC. Luckily, it's the same ISP that provides connectivity to locations of Client Server and Client PC.

Since you can route the traffic within your own network, you have been asked to configure shortest path routing between Client PC and Client Server. Considering the end to end link bandwidth, write a shortest path routing application using RYU SDN framework.

Client PC IP - 10.0.0.1 / 255.255.255.0
Client Server IP - 10.0.0.254 / 255.255.255.0

Verification:

1. Use pingall command in mininet terminal and ensure all hosts are reachable with 0% dropped.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> _
```

2. Use iperf command to measure the end to end bandwidth

Example: To measure the end-to-end bandwidth between Client PC and Client Server do the following start iperf server on Client Server use the following command in terminal:
`iperf -s`

To start the iperf client on Client PC use the following command in terminal: iperf -c 10.0.0.254

Now, you will should be able to see the end-to-end network bandwidth

3. You should be able to identify the path by using counters in you OpenFlow table.

Submitting your Code

You should turn in your python script on CANVAS course site assignment section with filename named in the following format:

Grading Criteria:

1. Submission of Topology.py and RYU spr.py files
2. Screenshot of flow table output from all switches
3. Screenshot of output showing pingall command with 0% dropped and iperf command verifying link performance.
4. Description about how you implemented the shortest path routing and draw the path your algorithm chose for routing.

Put all together in zip file and give file name in this format - shortestpath_yourgroupname.zip

References:

1. https://bradfieldcs.com/algos/graphs/dijkstras-algorithm/
2. http://www.gilles-bertrand.com/2014/03/dijkstra-algorithm-python-example- source-code-shortest-path.html
3. https://github.com/mininet/mininet/wiki/Documentation
4. https://www.tutorialspoint.com/python/
5. https://ryu.readthedocs.io/en/latest/writing_ryu_app.html

Instructions:

- Understand the reference document and how the simple network topologies have been build using mininet, you must be able to build the above topology using your own logic.
- Not only focusing on the solution by achieving the expected output, you should equally focus on the efficiency and best practices [code comments] while implementing the solution.
- The most you would learn is from troubleshooting your own code. So, please spend time on troubleshooting the code.
- It is not expected to post simple code implementation errors/issues on the discussion board.