Assignment 3: An Introduction to the World of SDN

COL 334/672, Diwali'24

September 23, 2024
Deadline: October 6, 2024

Goal: The goal of this assignment is to give you hands-on experience with software-defined networking. In particular, you will learn how to implement key network policies using OpenFlow-like APIs.

1 Part1: Controller Hub and Learning Switch (25%)

You need to compare the performance of a Hub Controller and a Learning Switch.

- A **Hub Controller** redirects all the traffic on a switch to itself and then forwards it to all switch ports except the incoming port.
- A **Learning Switch** installs flow rules on the switches based on the MAC to Port mappings it learns from incoming traffic.

We have also discussed both these controller types in the class. Begin by implementing a Controller Hub and Learning Switch. You are given a network topology file implementing the topology showing in Figure 1. Run the controller against the given network topology file and answer the following questions:

- Run pingall in both cases and report the installed rules in switches. Explain your observations briefly.
- Run a throughput test between h1 and h5 in both cases using iperf. Report the observed values. Explain the differences in the speed values between the Hub Controller and Learning Switch.

Hint: You can use the example code in the Ryu codebase. This part is relatively easy. You should use this to understand the working of Ryu controller.

2 Part 2: Spanning Tree (40%)

Recall that having loops in an L2-topology can lead to packets getting stuck in loops and constructing spanning trees are one efficient mechanism to prevent flooding. Consider the learning

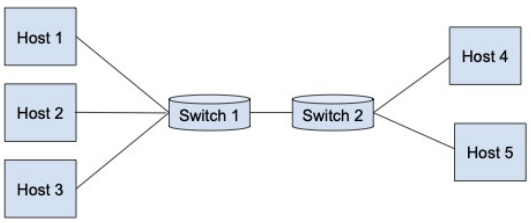

Figure 1: Network Topology

switch implemented in Section 1. You should realize that *broadcast* packets will cause issues if the underlying network topology has cycles.

In this part, you should modify the learning switch app to handle cycles in the topology. The app should handle cycles as follows:

- It should construct a spanning tree given a network topology. You can assume all link weights are 1. Feel free to use any algorithm to construct spanning tree. Note that you don't need to implement the Spanning Tree Protocol we studied in class as that was to construct a spanning tree in a distributed manner.
- For any broadcast packet, the switch should forward it to only the open ports in the spanning tree and any hosts that are attached to the switch.

You are given a network topology file implementing four-node cycle topology. Run the controller against the given network topology file to ensure that the app is able to gracefully handle loops in the network. You can run `pingall` to test this. Ultimately your code should run for any network topology. *Briefly* mention your approach and assumptions you made in the report.

3 Part 3: Shortest Path Routing (35%)

In this part, you will write a controller app that implements shortest path routing for a given network topology, while still performing L2 routing. The weights of the links should be set based on their capacities. You can test your code using a network topology that is provided with the assignment. It is similar to the topology in Section 2 with minor variations to the link capacity. In your report, please *briefly* describe your approach and any assumptions you made.

4 Part 4 (Bonus): Congestion-aware Shortest Path Routing (20%)

The shortest path described in Section 3 uses static link weights. In a real-world network, operators route traffic based on the current link load. Modify the shortest path routing app above to create dynamic routes based on the link utilization. Feel free to make assumptions whether you are routing at flow-level or packet-level. Just state them in the report along with your implementation logic. A proportion of the marks will be awarded based on the observed throughput for a given network topology.

Logistics

Housekeeping

- Any instance of cheating will receive strict penalty.
- You are allowed to do this assignment in a pair.
- **Piazza protocol 1**: Each *new* question should be asked in a separate thread with a clear subject line. A question is counted as *new* if a related question has not already been asked before. [I understand this definition is vague but please use common sense to make a judgement ☺.].
- **Piazza protocol 2**: *Please ask questions in advance.* Not all questions that are asked within 2 days of the deadline may be answered. No questions will be answered after the deadline.

Setup and Resources

You will need to install Mininet and Ryu for this assignment.

- **Installing Mininet**: Mininet works on Linux. If you are using Windows, you will have to either use a virtualization system or you can also request a VM on Baadal. If needed, please direct the request to me (user ID: `tmangla`). Installation instructions for mininet can be found here: `https://mininet.org/download/`.
- **Installing Ryu**: The installation instructions and tutorial for Ryu can be found here: `https://ryu.readthedocs.io/en/latest/getting_started.html`

Some useful commands:

- Ping between hosts h1 and h2: `h1 ping h2`
- Any command you want to send to a host, say h1: h1 cmd
- To open a new terminal for host h1: xterm h1. If you are using a VM on Baadal, make sure you set up x11 forwardning
- Print the rules currently installed on switches: dpctl dump-flows
- Running a ryu app: ryu-manager app.py

Submission Instructions

Your submission should contain a single PDF (other formats will not be graded) called `report.pdf`. In addition, you should submit the following:

- **Part 1**: Attach the screenshots of the rules as well as the ping/iperf results in the PDF. In addition submit two controller applications, naming them `p1_learning.py` and `p1_hub.py`.
- **Part 2**: Submit the controller application named `p2_spanning_tree.py`. Write the approach and assumptions in the report.
- **Part 3**:Submit the controller application named `p3_spr.py`.
- **Part 4**: Submit the controller application named `p4_ca_spr.py`.

You should submit a single zipped folder containing all the code as well as the report. Name it `<entry_no_1>_<entry_no_2>.zip`, i.e., the names of your and your partner's entry number. *Please note: Only one submission per group is required.*