Lab III Simulating SDN with Mininet

Group 13 B07902066 B07902062 B07902068 B07902080

1. Environment: OS: Ubuntu 18.04
   VM Platform: QEMU-KVM on CSIE workstation linux13
   Language: Python

2. Pros: Loops in the network enable rerouting when congestion happens, which improves the network performance. That is, we can change the path of flow to balance the loadings on each link, which can not be done in topologies without loops, since there's only a single path between the source and destination.
   Cons: Loops cause broadcast storms, which paralyzed the network by flooding the network with the same packet, so the loops need to be handled carefully.

3. A broadcast storm is when repeated rebroadcast messages flood the network, and eventually causes the network system to fail. It occurs when a device in the network system sends a broadcast message, and another device rebroadcasts the message back, they will repeatedly rebroadcast the packet until the network system is flooded.
   We handle the loops with the spanning tree protocol (STP), which can prevent physical loops like the switching loops. When multiple links are between two network nodes, STP will cut off the links until there's one left, and eventually form a spanning tree in the network topology, eliminating the loops. However, this method gives away the advantage of looping, i.e., rerouting when congestion happens. A better way may be choosing the routes dynamically, and create flows that don't contain loops within. This may be done by performing shortest path algorithms (according to the paths loading) that generate shortest-path trees, for example, the Dijkstra algorithm or the Bellman-Ford algorithm. Everytime we want to create a flow, we perform the algorithm on the source node, and generate the flow to the destination node according to the spanning tree that we generated. In this way, we not only prevent broadcast storms, but also benefit the network by preventing congestion by creating the most efficient flows everytime.