COL334

Assignment - 3

Part - 1: Controller Hub and Learning Switch

1. pingall and flows

    a. Controller Hub

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)

mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=10.496s, table=0, n_packets=61, n_bytes=4861, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=10.512s, table=0, n_packets=61, n_bytes=4861, priority=0 actions=CONTROLLER:65535
```

    b. Learning Switch

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)

mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=3.026s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth2",dl_src=4e:09:13:af:d8:20,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s1-eth1"
 cookie=0x0, duration=3.024s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=0a:93:cd:c7:7b:3b,dl_dst=4e:09:13:af:d8:20 actions=output:"s1-eth2"
 cookie=0x0, duration=3.015s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s1-eth1"
 cookie=0x0, duration=3.013s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=0a:93:cd:c7:7b:3b,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s1-eth3"
 cookie=0x0, duration=3.003s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=32:0e:e5:da:bc:be,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s1-eth1"
 cookie=0x0, duration=3.002s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=0a:93:cd:c7:7b:3b,dl_dst=32:0e:e5:da:bc:be actions=output:"s1-eth4"
 cookie=0x0, duration=2.990s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=4e:0e:8c:97:46:72,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s1-eth1"
 cookie=0x0, duration=2.988s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth1",dl_src=0a:93:cd:c7:7b:3b,dl_dst=4e:0e:8c:97:46:72 actions=output:"s1-eth4"
 cookie=0x0, duration=2.971s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=4e:09:13:af:d8:20 actions=output:"s1-eth2"
 cookie=0x0, duration=2.969s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth2",dl_src=4e:09:13:af:d8:20,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s1-eth3"
 cookie=0x0, duration=2.955s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=32:0e:e5:da:bc:be,dl_dst=4e:09:13:af:d8:20 actions=output:"s1-eth2"
 cookie=0x0, duration=2.953s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth2",dl_src=4e:09:13:af:d8:20,dl_dst=32:0e:e5:da:bc:be actions=output:"s1-eth4"
 cookie=0x0, duration=2.939s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=4e:0e:8c:97:46:72,dl_dst=4e:09:13:af:d8:20 actions=output:"s1-eth2"
 cookie=0x0, duration=2.937s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth2",dl_src=4e:09:13:af:d8:20,dl_dst=4e:0e:8c:97:46:72 actions=output:"s1-eth4"
 cookie=0x0, duration=2.918s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=32:0e:e5:da:bc:be,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s1-eth3"
 cookie=0x0, duration=2.915s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=32:0e:e5:da:bc:be actions=output:"s1-eth4"
 cookie=0x0, duration=2.902s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s1-eth4",dl_src=4e:0e:8c:97:46:72,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s1-eth3"
 cookie=0x0, duration=2.901s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s1-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=4e:0e:8c:97:46:72 actions=output:"s1-eth4"
 cookie=0x0, duration=12.984s, table=0, n_packets=30, n_bytes=1820, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=3.021s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth1",dl_src=32:0e:e5:da:bc:be,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s2-eth3"
 cookie=0x0, duration=3.017s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=0a:93:cd:c7:7b:3b,dl_dst=32:0e:e5:da:bc:be actions=output:"s2-eth1"
 cookie=0x0, duration=3.008s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth2",dl_src=4e:0e:8c:97:46:72,dl_dst=0a:93:cd:c7:7b:3b actions=output:"s2-eth3"
 cookie=0x0, duration=3.003s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=0a:93:cd:c7:7b:3b,dl_dst=4e:0e:8c:97:46:72 actions=output:"s2-eth2"
 cookie=0x0, duration=2.973s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth1",dl_src=32:0e:e5:da:bc:be,dl_dst=4e:09:13:af:d8:20 actions=output:"s2-eth3"
 cookie=0x0, duration=2.966s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=4e:09:13:af:d8:20,dl_dst=32:0e:e5:da:bc:be actions=output:"s2-eth1"
 cookie=0x0, duration=2.956s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth2",dl_src=4e:0e:8c:97:46:72,dl_dst=4e:09:13:af:d8:20 actions=output:"s2-eth3"
 cookie=0x0, duration=2.952s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=4e:09:13:af:d8:20,dl_dst=4e:0e:8c:97:46:72 actions=output:"s2-eth2"
 cookie=0x0, duration=2.936s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth1",dl_src=32:0e:e5:da:bc:be,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s2-eth3"
 cookie=0x0, duration=2.930s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=32:0e:e5:da:bc:be actions=output:"s2-eth1"
 cookie=0x0, duration=2.920s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth2",dl_src=4e:0e:8c:97:46:72,dl_dst=76:8f:cf:9a:a7:f3 actions=output:"s2-eth3"
 cookie=0x0, duration=2.915s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth3",dl_src=76:8f:cf:9a:a7:f3,dl_dst=4e:0e:8c:97:46:72 actions=output:"s2-eth2"
 cookie=0x0, duration=2.892s, table=0, n_packets=2, n_bytes=196, priority=1,in_port="s2-eth2",dl_src=4e:0e:8c:97:46:72,dl_dst=32:0e:e5:da:bc:be actions=output:"s2-eth1"
 cookie=0x0, duration=2.890s, table=0, n_packets=1, n_bytes=98, priority=1,in_port="s2-eth1",dl_src=32:0e:e5:da:bc:be,dl_dst=4e:0e:8c:97:46:72 actions=output:"s2-eth2"
 cookie=0x0, duration=13s, table=0, n_packets=27, n_bytes=1610, priority=0 actions=CONTROLLER:65535
```

Observations: Both switches (s1 and s2) have a single flow entry each, with a priority of 0 and an action of CONTROLLER:65535. For the *Controller Hub*, all packets arriving at any switch port are sent to the controller for processing. This results in no specific flow rules being installed for individual hosts or flows. The controller is responsible for all traffic forwarding. For the *Learning Switch*, flow rules show specific matches based on MAC addresses (dl_src, dl_dst) and in-port, with corresponding actions that direct traffic to the correct output port. The Learning Switch learns the paths over time and installs flow rules to forward packets directly between hosts, optimizing the network by avoiding unnecessary controller involvement. The flow rules have a priority of 1, meaning they take precedence over the default controller action.

2. iperf h1 h5
    a. Controller Hub

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['28.0 Mbits/sec', '27.9 Mbits/sec']
```

    b. Learning Switch

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['37.7 Gbits/sec', '37.6 Gbits/sec']
```

Observations: The *Controller Hub* broadcasts packets to all devices, resulting in wasted bandwidth and much lower throughput while the *Learning Switch* forwards packets directly to the destination, allowing for efficient use of bandwidth and much higher throughput.

Part - 2: Spanning Tree

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```

Approach: The implementation uses a simple Depth-First Search (DFS) algorithm to construct the spanning tree. The root bridge is selected as the switch with the lowest datapath ID (dpid). All the non-tree ports are blocked by installing flow rules that drop packets.

Assumptions: I have assumed here that the network once set up, will not go through any changes. For example, no new switches being added or removed, no new links are established and no link goes down, etc.

Part –3: Shortest Route Path

Approach: The implementation here first uses `get_switch` and `get_link` function provided by the ryu library to find the links and switches local to the current switch. Then uses lldp packets to find the link delays between the links. Maintaining a Graph class where the weights are updated, which are later used in the djikstra's function to calculate the optimal path between links

Assumptions: Here we assume the links to be stable, and the link delays to be invariant, and also assume no buffer delays, causing no congestion.