Nishant Shah (B00659531)

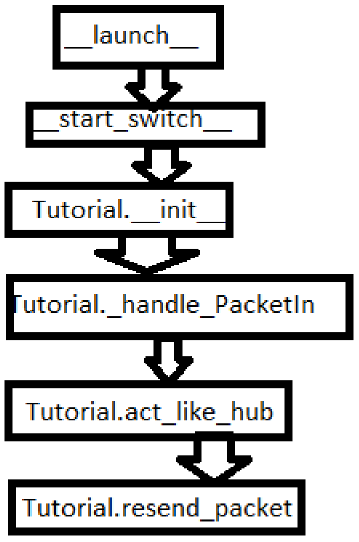
Assignment – 2: Exploring Software-Defined Networking Techniques

TASK 1:

Please find 'binary_tree.py' file in this same package

TASK 2:

- Q1 – Function call graph of the 'of_tutorial' controller:



- Q2 – Average for pinging 100 times:
    - h1 -> h2: 26.522 ms
    - h1 -> h8: 30.379 ms
    - It takes little longer to ping in the second case. This is because h1 and h2 share the same direct switch through which they are connected. So data goes from h1 to s1 to h2. While h1 and h8, they are not connected diectly to single switch. Data has to travel through multiple switches to reach from h1 to h8
- Q3 – IPERF:
    - IPERF is used to test the TCP bandwidth between two ports.
    - iperf h1 h2: '8.87 Mbits/sec' , '10.8 Mbits/sec'
    - iperf h1 h8: '4.18 Mbits/sec' , '5.08 Mbits/sec'
    - Bandwidth for h1 to h2 is approximately double the bandwidth of h1 and h8.
    - This is because h1 and h2 are directly connected to a single switch and no traffic. Hence, have high bandwidth as they can exchange data much faster. While on the other hand for h1 and h8, they are not directly connected to single switch. Data has to go through multiple switches (observe traffic) and hence they have a lower bandwidth.

- Q4 – Observing traffic:
    - Switches s5, s6, and s7 observe traffic with s7 observing most traffic. This is bacause of the number of ports they are connected to. As s7 is connected to all ports, it behaves as 'hub' and observes the most traffic.
    - s5 acts as mini hub for any data sharing between h1 to h4 and s6 acts as mini hub for data sharing between h4 to h8.

TASK 3:

- Please find the 'of_tutorial.py' controller code in this same package.
- Q1 – Code working:
    - We are taking an example of h1 ping h2
    - Here, h1 is src so for very first time it will not be in the mac_to_port so the system learns that this packet is attached to the src
    - For the first time h2 will also be not known so the packet is flooded to each and every port except the src port and all ports except src are sent as dst to the resend_packet
    - in resend_packet, it adds a connection, checks with all the ports and sends via controller to the port h2.
    - Once this is done, src and dst both are known by the system. So whenever the next time h1 ping h2 comes, src is already in mac_to_port, dst is in mac_to_port; hence hence both src and dst are sent to resend_packet so it forwards the packet.
- Q2 – Average for pinging 100 times:
    - h1 -> h2: 23.157 ms
    - h1 -> h8: 32.075 ms
    - There is no significant difference from TASK 2. It drops approx 3 ms for first case and increase 2 ms for 2nd case
- Q3 – IPERF:
    - iperf h1 h2: '16.5 Mbits/sec' , '17.5 Mbits/sec'
    - iperf h1 h8: '7.95 Mbits/sec' , '9.60 Mbits/sec'
    - The performance here increases about 50% more compared to TASK 2

TASK 4:

- Please find the controller code attached in this same package. I renamed the controller code to 'of_controller_task-4.py' for submission purposes to avoid confusion with controller code for the previous task.
- Q1 – Average for pinging 100 times:
    - h1 -> h2: 0.075 ms
    - h1 -> h8: 0.072 ms
    - This time is reduced significantly and in both the cases takes around same time. Also for h1 -> h2, 2 % packet loss is observed and for h1 -> h8, 9% packet loss is observed.

- Q2 – IPERF:
  - iperf h1 h2: '35.7 Mbits/sec' , '35.7 Mbits/sec'
  - iperf h1 h8: '30.4 Mbits/sec' , '30.4 Mbits/sec'
  - The performance increases significantly and is very high as compared to TASK 3.
- Q3 – Eplaining above results:
  - We saw that the results here are improved significantly. Average ping time is very low with very high bandwidth.
  - This is because we imposed rules on the switches.
  - For the first time, it may drop a packet or take more time to connect with different ports as it may not have the flow entry.
  - But once it knows about the ports and it's flow entry, data transfer takes place very fast and also the bandwidth becomes very high.
- Q4 – Here's the output of pingall two consecutive times:

```
mininet> pingall

*** Ping: testing ping reachability

h1 -> h2 X X X X X h8

h2 -> h1 X X X X X h8

h3 -> X h2 X X X X X

h4 -> h1 h2 h3 X X X h8

h5 -> X h2 X h4 X X X

h6 -> X h2 h3 h4 h5 X h8

h7 -> X h2 h3 h4 h5 h6 h8

h8 -> h1 h2 h3 h4 h5 h6 h7

*** Results: 48% dropped (29/56 received)

mininet> pingall

*** Ping: testing ping reachability

h1 -> h2 h3 h4 h5 h6 h7 h8

h2 -> h1 h3 h4 h5 h6 h7 h8

h3 -> h1 h2 h4 h5 h6 h7 h8

h4 -> h1 h2 h3 h5 h6 h7 h8

h5 -> h1 h2 h3 h4 h6 h7 h8

h6 -> h1 h2 h3 h4 h5 h7 h8

h7 -> h1 h2 h3 h4 h5 h6 h8

h8 -> h1 h2 h3 h4 h5 h6 h7

*** Results: 0% dropped (56/56 received)
```

- Q5 – Using 'ovs-ofctl dump-flows':
    - Here's the output for each switch:

```
root@200985a79052:~# ovs-ofctl dump-flows s1

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=591.505s, table=0, n_packets=244958, n_bytes=16173036, idle_age=73, dl_dst=46:16:6e:3f:af:86
actions=output:1

 cookie=0x0, duration=590.406s, table=0, n_packets=457556, n_bytes=24407889880, idle_age=73, dl_dst=e2:83:67:a8:58:77
actions=output:2

 cookie=0x0, duration=476.814s, table=0, n_packets=353830, n_bytes=18873606580, idle_age=78, dl_dst=e6:ff:47:de:d4:58
actions=output:3

 cookie=0x0, duration=335.239s, table=0, n_packets=15, n_bytes=1022, idle_age=73, dl_dst=12:8a:f4:b8:5c:ca actions=output:3

 cookie=0x0, duration=326.261s, table=0, n_packets=15, n_bytes=1022, idle_age=73, dl_dst=7e:92:95:a0:2b:d0
actions=output:3

 cookie=0x0, duration=315.237s, table=0, n_packets=17, n_bytes=1050, idle_age=73, dl_dst=56:a0:4b:85:22:e7
actions=output:3

 cookie=0x0, duration=306.229s, table=0, n_packets=12, n_bytes=896, idle_age=73, dl_dst=46:bf:4c:01:ab:f1 actions=output:3

 cookie=0x0, duration=296.226s, table=0, n_packets=11, n_bytes=854, idle_age=78, dl_dst=4e:35:22:99:2a:f9 actions=output:3

root@200985a79052:~# ovs-ofctl dump-flows s2

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=352.542s, table=0, n_packets=16, n_bytes=1008, idle_age=89, dl_dst=46:16:6e:3f:af:86 actions=output:3

 cookie=0x0, duration=302.506s, table=0, n_packets=17, n_bytes=1050, idle_age=89, dl_dst=e2:83:67:a8:58:77
actions=output:3

 cookie=0x0, duration=259.445s, table=0, n_packets=45, n_bytes=3010, idle_age=89, dl_dst=12:8a:f4:b8:5c:ca actions=output:1

 cookie=0x0, duration=249.395s, table=0, n_packets=47, n_bytes=3206, idle_age=89, dl_dst=7e:92:95:a0:2b:d0
actions=output:2

 cookie=0x0, duration=236.327s, table=0, n_packets=11, n_bytes=854, idle_age=94, dl_dst=46:bf:4c:01:ab:f1 actions=output:3

 cookie=0x0, duration=225.354s, table=0, n_packets=11, n_bytes=854, idle_age=94, dl_dst=4e:35:22:99:2a:f9 actions=output:3

 cookie=0x0, duration=216.317s, table=0, n_packets=12, n_bytes=896, idle_age=94, dl_dst=e6:ff:47:de:d4:58 actions=output:3

 cookie=0x0, duration=173.170s, table=0, n_packets=11, n_bytes=742, idle_age=89, dl_dst=56:a0:4b:85:22:e7 actions=output:3

root@200985a79052:~# ovs-ofctl dump-flows s3

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=339.750s, table=0, n_packets=15, n_bytes=966, idle_age=96, dl_dst=46:16:6e:3f:af:86 actions=output:3

 cookie=0x0, duration=289.733s, table=0, n_packets=15, n_bytes=966, idle_age=96, dl_dst=e2:83:67:a8:58:77 actions=output:3

 cookie=0x0, duration=246.598s, table=0, n_packets=15, n_bytes=966, idle_age=96, dl_dst=12:8a:f4:b8:5c:ca actions=output:3

 cookie=0x0, duration=213.521s, table=0, n_packets=15, n_bytes=966, idle_age=96, dl_dst=7e:92:95:a0:2b:d0 actions=output:3
 cookie=0x0, duration=183.228s, table=0, n_packets=39, n_bytes=2702, idle_age=96, dl_dst=56:a0:4b:85:22:e7
actions=output:1

 cookie=0x0, duration=170.371s, table=0, n_packets=36, n_bytes=2688, idle_age=96, dl_dst=46:bf:4c:01:ab:f1 actions=output:2

 cookie=0x0, duration=159.343s, table=0, n_packets=10, n_bytes=812, idle_age=102, dl_dst=4e:35:22:99:2a:f9
actions=output:3

 cookie=0x0, duration=150.355s, table=0, n_packets=10, n_bytes=812, idle_age=102, dl_dst=e6:ff:47:de:d4:58 actions=output:3

root@200985a79052:~# ovs-ofctl dump-flows s4

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=508.354s, table=0, n_packets=106954, n_bytes=7061732, idle_age=105, dl_dst=46:16:6e:3f:af:86
actions=output:3

 cookie=0x0, duration=500.137s, table=0, n_packets=353853, n_bytes=18873608330, idle_age=105, dl_dst=e6:ff:47:de:d4:58
actions=output:2

 cookie=0x0, duration=280.560s, table=0, n_packets=14, n_bytes=980, idle_age=105, dl_dst=e2:83:67:a8:58:77
actions=output:3

 cookie=0x0, duration=237.425s, table=0, n_packets=12, n_bytes=840, idle_age=105, dl_dst=12:8a:f4:b8:5c:ca actions=output:3

 cookie=0x0, duration=204.349s, table=0, n_packets=14, n_bytes=980, idle_age=105, dl_dst=7e:92:95:a0:2b:d0
actions=output:3

 cookie=0x0, duration=164.255s, table=0, n_packets=12, n_bytes=840, idle_age=105, dl_dst=56:a0:4b:85:22:e7
actions=output:3

 cookie=0x0, duration=134.226s, table=0, n_packets=11, n_bytes=854, idle_age=105, dl_dst=46:bf:4c:01:ab:f1 actions=output:3

 cookie=0x0, duration=124.211s, table=0, n_packets=34, n_bytes=2604, idle_age=105, dl_dst=4e:35:22:99:2a:f9
actions=output:1

root@200985a79052:~# ovs-ofctl dump-flows s5

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=523.838s, table=0, n_packets=106982, n_bytes=7063580, idle_age=119, dl_dst=46:16:6e:3f:af:86
actions=output:1

 cookie=0x0, duration=521.782s, table=0, n_packets=353841, n_bytes=18873607378, idle_age=124, dl_dst=e6:ff:47:de:d4:58
actions=output:3

 cookie=0x0, duration=331.148s, table=0, n_packets=43, n_bytes=2870, idle_age=119, dl_dst=e2:83:67:a8:58:77
actions=output:1

 cookie=0x0, duration=330.128s, table=0, n_packets=39, n_bytes=2646, idle_age=119, dl_dst=12:8a:f4:b8:5c:ca
actions=output:2

 cookie=0x0, duration=322.189s, table=0, n_packets=41, n_bytes=2786, idle_age=119, dl_dst=7e:92:95:a0:2b:d0
actions=output:2

 cookie=0x0, duration=309.122s, table=0, n_packets=22, n_bytes=1652, idle_age=119, dl_dst=46:bf:4c:01:ab:f1
actions=output:3

 cookie=0x0, duration=298.099s, table=0, n_packets=21, n_bytes=1610, idle_age=124, dl_dst=4e:35:22:99:2a:f9
actions=output:3
```

```
 cookie=0x0, duration=212.898s, table=0, n_packets=27, n_bytes=1694, idle_age=119, dl_dst=56:a0:4b:85:22:e7
actions=output:3

root@200985a79052:~# ovs-ofctl dump-flows s6

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=536.889s, table=0, n_packets=106968, n_bytes=7062656, idle_age=130, dl_dst=46:16:6e:3f:af:86
actions=output:3

 cookie=0x0, duration=530.727s, table=0, n_packets=353849, n_bytes=18873607994, idle_age=135, dl_dst=e6:ff:47:de:d4:58
actions=output:2

 cookie=0x0, duration=322.160s, table=0, n_packets=28, n_bytes=1904, idle_age=130, dl_dst=e2:83:67:a8:58:77
actions=output:3

 cookie=0x0, duration=279.074s, table=0, n_packets=26, n_bytes=1764, idle_age=130, dl_dst=12:8a:f4:b8:5c:ca
actions=output:3

 cookie=0x0, duration=245.947s, table=0, n_packets=28, n_bytes=1904, idle_age=130, dl_dst=7e:92:95:a0:2b:d0
actions=output:3

 cookie=0x0, duration=217.692s, table=0, n_packets=35, n_bytes=2366, idle_age=130, dl_dst=56:a0:4b:85:22:e7
actions=output:1

 cookie=0x0, duration=173.783s, table=0, n_packets=31, n_bytes=2310, idle_age=130, dl_dst=46:bf:4c:01:ab:f1
actions=output:1

 cookie=0x0, duration=162.760s, table=0, n_packets=29, n_bytes=2226, idle_age=135, dl_dst=4e:35:22:99:2a:f9
actions=output:2

root@200985a79052:~# ovs-ofctl dump-flows s7

NXST_FLOW reply (xid=0x4):

 cookie=0x0, duration=545.928s, table=0, n_packets=106967, n_bytes=7062614, idle_age=140, dl_dst=46:16:6e:3f:af:86
actions=output:1

 cookie=0x0, duration=541.819s, table=0, n_packets=353840, n_bytes=18873607280, idle_age=145, dl_dst=e6:ff:47:de:d4:58
actions=output:2

 cookie=0x0, duration=331.217s, table=0, n_packets=27, n_bytes=1862, idle_age=140, dl_dst=e2:83:67:a8:58:77
actions=output:1

 cookie=0x0, duration=288.080s, table=0, n_packets=25, n_bytes=1722, idle_age=140, dl_dst=12:8a:f4:b8:5c:ca
actions=output:1

 cookie=0x0, duration=255.004s, table=0, n_packets=27, n_bytes=1862, idle_age=140, dl_dst=7e:92:95:a0:2b:d0
actions=output:1

 cookie=0x0, duration=253.982s, table=0, n_packets=21, n_bytes=1554, idle_age=140, dl_dst=46:bf:4c:01:ab:f1
actions=output:2

 cookie=0x0, duration=243.009s, table=0, n_packets=20, n_bytes=1512, idle_age=145, dl_dst=4e:35:22:99:2a:f9
actions=output:2

 cookie=0x0, duration=228.786s, table=0, n_packets=26, n_bytes=1652, idle_age=140, dl_dst=56:a0:4b:85:22:-e7
actions=output:2
```

- There are ‘eight’ rules for each OpenFlow switch. For each switch there is rule for each port and in total there are 8 ports so eight rules.
- Each flow entry contains:
    - cookies for that flow
    - duration since that flow is there
    - which table that particular flow is for
    - number of packets through that flow
    - number of bytes of data passed through that flow
    - for how much time is it idle
    - destination address
    - what action that flow performs

TASK 5:

- We can do it by ovs-ovctl commands.
- Add a flow rule to switch seven matching the IP addresses
- ovs-ofctl add_flow s7 dl_src<IP of s1 to s4>
- ovs-ofctl add_flow s7 dl_dst<IP of s5 to s8
- Hence, each time data passes through s7, it will pass through our IP matching rules.