# Programming Assignment 3: SDN and Ryu

*Ranjul Bandyopadhyay – JCS232541*

## Part 1:

**Q.) a.)** Conducting 3 pings for each case. Report the latency values.

**Output:**

**For Learning_Switch:**

```
mininet> h1 ping -c 3 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=16.7 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.271 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.083 ms

--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2022ms
rtt min/avg/max/mdev = 0.083/5.668/16.650/7.765 ms
```

```
mininet> h2 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=20.0 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.292 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.051 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2023ms
rtt min/avg/max/mdev = 0.051/6.795/20.043/9.368 ms
```

```
mininet> h3 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=16.8 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.236 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.049 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.049/5.700/16.817/7.860 ms
```

**For Controller_Hub:**

```
mininet> h1 ping -c 3 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=3.66 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=4.01 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=3.78 ms

--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.657/3.813/4.005/0.144 ms
```

```
mininet> h2 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=3.41 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=3.90 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=3.97 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 3.411/3.759/3.968/0.247 ms
```

```
mininet> h3 ping -c 3 h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=3.90 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=4.00 ms

--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 3.499/3.799/3.999/0.216 ms
```

**b.)** Explain the observed latency differences between the Hub Controller and Learning Switch

**Solution:** For the controller hub, it forwards all packets to all its ports. This means that it must first receive the entire packet before it can forward it, which introduces latency. For the learning switch it learns the MAC addresses of the devices on its network and builds a forwarding table. This forwarding table allows the learning switch to forward packets directly to the destination port without having to receive the entire packet first. This significantly reduces the latency.

**c.)** Explain differences (if any) observed between h2 and h5 for both controller types.

**Solution:** The main difference between h2 and h5 in both the controllers can be observed in the throughput and in the latency. The reason is due to the features of the hub and switch as explained in the above solutions.

**Q.) a.)** Run a throughput test between h1 and h5. Report the observed values.

**Output:**

**For Learning_Switch:**

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['20.6 Gbits/sec', '20.7 Gbits/sec']
```

**For Controller_Hub:**

```
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['9.99 Mbits/sec', '10.7 Mbits/sec']
```

**b.)** Explain the differences between the Hub Controller and Learning Switch.

**Solution:** The main difference in throughput between a hub controller and a learning switch is due to the way they forward traffic.

A hub controller forwards all the packets to all its ports, regardless of the destination MAC address. This leads to a lot of unnecessary traffic on the network, which reduces the throughput.

A learning switch, on the other hand, learns the MAC addresses of the devices on its network and builds a forwarding table. This forwarding table allows the learning switch to forward packets directly to the destination port without having to flood the network with unnecessary traffic. This significantly improves the throughput.

**Q.)** Run pingall in both cases and report the installed rules on switches.

**Output:**

**For Learning_Switch:**

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=1692.759s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=1692.756s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=1384.981s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=1384.977s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth1"
 cookie=0x0, duration=1104.167s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=1104.165s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth1"
 cookie=0x0, duration=692.911s, table=0, n_packets=296100, n_bytes=19542620, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=692.908s, table=0, n_packets=296102, n_bytes=12955431660, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth1"
 cookie=0x0, duration=8.642s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=8.639s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=8.626s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=8.623s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=8.600s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=8.597s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=8.582s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=8.580s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=8.546s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=8.543s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=1701.012s, table=0, n_packets=28, n_bytes=1656, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=1692.772s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth1"
 cookie=0x0, duration=1692.763s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=1384.997s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth1"
 cookie=0x0, duration=1384.983s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=1104.179s, table=0, n_packets=8, n_bytes=616, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth1"
 cookie=0x0, duration=1104.172s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=692.932s, table=0, n_packets=296100, n_bytes=19542620, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth1"
 cookie=0x0, duration=692.914s, table=0, n_packets=296102, n_bytes=12955431660, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=8.594s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth1"
 cookie=0x0, duration=8.586s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=8.558s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth1"
 cookie=0x0, duration=8.548s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=8.519s, table=0, n_packets=3, n_bytes=238, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=8.515s, table=0, n_packets=2, n_bytes=140, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=1701.019s, table=0, n_packets=24, n_bytes=1376, priority=0 actions=CONTROLLER:65535
```

**For Controller_Hub:**

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=729.223s, table=0, n_packets=4923, n_bytes=7142630, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=729.215s, table=0, n_packets=7289, n_bytes=7298786, priority=0 actions=CONTROLLER:65535
```

## Part 2:

**Q.)** Run pingall and report the results.

**Output:**

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X h5
h2 -> h1 h3 h4 X
h3 -> h1 h2 h4 X
h4 -> X h2 h3 h5
h5 -> h1 X X h4
*** Results: 30% dropped (14/20 received)
```

**Q.) a.)** Report the installed rules on both switches.

**Output:**

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=90.926s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=90.924s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=90.912s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=90.909s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=87.815s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=87.813s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth1"
 cookie=0x0, duration=87.796s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=87.793s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=87.778s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=87.775s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=84.665s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=84.659s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=94.649s, table=0, n_packets=44, n_bytes=2240, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=87.827s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth1"
 cookie=0x0, duration=87.819s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=87.790s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth1"
 cookie=0x0, duration=87.780s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=84.680s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth1"
 cookie=0x0, duration=84.666s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=71.570s, table=0, n_packets=7, n_bytes=518, priority=1,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=71.568s, table=0, n_packets=6, n_bytes=420, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=94.658s, table=0, n_packets=68, n_bytes=3360, priority=0 actions=CONTROLLER:65535
```

**b.)** Can you think of ways to minimize the number of firewall rules on the switch?

**Solution:** The following are ways to minimize the number of firewall rules:

- Instead of writing separate rules for individual IP addresses or services we can aggregate the rules whenever possible. For example, if you want to allow access to a range of IP addresses, we can use the CIDR notation to specify the range in a single rule.
- We can use object groups to group related IP addresses or ports together. This helps to reduce the number of firewall rules we need to create and manage.

**Q.)** Suppose the network operator intends to implement firewall policies in real time. How can she ensure that the pre-existing rules do not interfere with the firewall policy?

**Solution:** To ensure that pre-existing firewall rules do not interfere with a real-time firewall policy, the network operator can take the following steps:

- Identify the pre-existing rules that are relevant to the new policy: This can be done by reviewing the firewall logs and identifying the rules that are being triggered by the traffic that is covered by the new policy.
- Modify or remove the pre-existing rules that are against the new policy
- Test the new policy.
- Monitor the new policy.

## Part 3:

**Q.)** Have the three hosts (H1, H2, and H3) ping the virtual IP and report the installed rule on the switches.

```
mininet> h1 ping -c 5 10.0.0.42
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
64 bytes from 10.0.0.42: icmp_seq=2 ttl=64 time=20.5 ms
64 bytes from 10.0.0.42: icmp_seq=3 ttl=64 time=0.489 ms
64 bytes from 10.0.0.42: icmp_seq=4 ttl=64 time=0.086 ms
64 bytes from 10.0.0.42: icmp_seq=5 ttl=64 time=0.059 ms

--- 10.0.0.42 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4043ms
rtt min/avg/max/mdev = 0.059/5.280/20.489/8.782 ms
```

```
mininet> h2 ping -c 5 10.0.0.42
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
64 bytes from 10.0.0.42: icmp_seq=2 ttl=64 time=19.9 ms
64 bytes from 10.0.0.42: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 10.0.0.42: icmp_seq=4 ttl=64 time=0.114 ms
64 bytes from 10.0.0.42: icmp_seq=5 ttl=64 time=0.060 ms

--- 10.0.0.42 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4044ms
rtt min/avg/max/mdev = 0.060/5.101/19.929/8.561 ms
```

```
mininet> h3 ping -c 5 10.0.0.42
PING 10.0.0.42 (10.0.0.42) 56(84) bytes of data.
64 bytes from 10.0.0.42: icmp_seq=2 ttl=64 time=11.6 ms
64 bytes from 10.0.0.42: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.0.42: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 10.0.0.42: icmp_seq=5 ttl=64 time=0.072 ms

--- 10.0.0.42 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4039ms
rtt min/avg/max/mdev = 0.058/2.937/11.553/4.974 ms
```

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=322.509s, table=0, n_packets=4, n_bytes=392, priority=20,icmp,in_port="s1-eth2",nw_dst=10.0.0.42 actions=mod_nw_dst:10.0.0.4,output:"s1-eth1"
 cookie=0x0, duration=236.527s, table=0, n_packets=2, n_bytes=196, priority=20,icmp,in_port="s1-eth3",nw_dst=10.0.0.42 actions=mod_nw_dst:10.0.0.5,output:"s1-eth1"
 cookie=0x0, duration=113.415s, table=0, n_packets=2, n_bytes=196, priority=20,icmp,in_port="s1-eth4",nw_dst=10.0.0.42 actions=mod_nw_dst:10.0.0.4,output:"s1-eth1"
 cookie=0x0, duration=322.506s, table=0, n_packets=4, n_bytes=392, priority=20,icmp,in_port="s1-eth1",dl_dst=00:00:00:00:00:01,nw_src=10.0.0.4 actions=mod_nw_src:10.0.0.42,output:"s1
-eth2"
 cookie=0x0, duration=236.525s, table=0, n_packets=2, n_bytes=196, priority=20,icmp,in_port="s1-eth1",dl_dst=00:00:00:00:00:02,nw_src=10.0.0.5 actions=mod_nw_src:10.0.0.42,output:"s1
-eth3"
 cookie=0x0, duration=113.415s, table=0, n_packets=2, n_bytes=196, priority=20,icmp,in_port="s1-eth1",dl_dst=00:00:00:00:00:03,nw_src=10.0.0.4 actions=mod_nw_src:10.0.0.42,output:"s1
-eth4"
 cookie=0x0, duration=321.493s, table=0, n_packets=4, n_bytes=168, priority=10,in_port="s1-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=154.778s, table=0, n_packets=4, n_bytes=168, priority=10,in_port="s1-eth3",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth1"
 cookie=0x0, duration=113.416s, table=0, n_packets=5, n_bytes=210, priority=10,in_port="s1-eth4",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s1-eth1"
 cookie=0x0, duration=27.969s, table=0, n_packets=0, n_bytes=0, priority=10,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s1-eth4"
 cookie=0x0, duration=24.897s, table=0, n_packets=0, n_bytes=0, priority=10,in_port="s1-eth1",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth3"
 cookie=0x0, duration=20.545s, table=0, n_packets=0, n_bytes=0, priority=10,in_port="s1-eth1",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth2"
 cookie=0x0, duration=324.891s, table=0, n_packets=16, n_bytes=840, priority=0 actions=CONTROLLER:65535
*** s2 ------------------------------------------------------------------------
 cookie=0x0, duration=321.492s, table=0, n_packets=7, n_bytes=462, priority=10,in_port="s2-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=321.486s, table=0, n_packets=4, n_bytes=336, priority=10,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth1"
 cookie=0x0, duration=154.781s, table=0, n_packets=5, n_bytes=266, priority=10,in_port="s2-eth1",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=154.776s, table=0, n_packets=2, n_bytes=140, priority=10,in_port="s2-eth3",dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:02 actions=output:"s2-eth1"
 cookie=0x0, duration=101.337s, table=0, n_packets=6, n_bytes=308, priority=10,in_port="s2-eth1",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04 actions=output:"s2-eth2"
 cookie=0x0, duration=101.325s, table=0, n_packets=2, n_bytes=140, priority=10,in_port="s2-eth2",dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:03 actions=output:"s2-eth1"
 cookie=0x0, duration=324.895s, table=0, n_packets=11, n_bytes=798, priority=0 actions=CONTROLLER:65535
```

Q.) If you were to implement a load balancing policy that considers the load on these servers, what additional steps would you take?

**Solution:** Steps that can be taken to implement a load balancing policy are:

- Configure the load balancer to use a weighted round robin algorithm. This would ensure that traffic is distributed more evenly to the servers, based on their respective processing capacity.
- Use a health check mechanism to monitor the availability of the servers. This would allow the load balancer to quickly identify and remove any unavailable servers from the pool.