SDN Lab2

- 110550137 Yu-Sheng Lin

Part1

Q1

1. How many OpenFlow headers with type “OFPT_FLOW_MOD” and command “OFPFC_ADD” are there?
2. What are the match fields and the corresponding actions in each “OFPT_FLOW_MOD” message?
3. What are the timeout values for all flow rules on s1?

There are 6 distinct "OFPT_FLOW_MOD" headers during the experiment.

Match Field	Actions	Timeout value
ETH_TYPE=802.1	CLEAR, OFPAT_OUTPUT	0
ETH_TYPE=IPv4	CLEAR, OFPAT_OUTPUT	0
ETH_TYPE=Unknown(0x8942)	CLEAR, OFPAT_OUTPUT	0
ETH_TYPE=ARP	CLEAR, OFPAT_OUTPUT	0
IN_PORT=2	OFPAT_OUTPUT_PORT=1	0
IN_PORT=1	OFPAT_OUTPUT_PORT=2	0

Part2

Flow Rule - ARP

```
288 packets transmitted, 8 packets received,  97% unanswered (0 extra)
rtt min/avg/max/std-dev = 0.012/0.014/0.020/0.003 ms
mininet> h1 arping h2
ARPING 10.0.0.2
42 bytes from 9a:e9:7a:6f:a1:51 (10.0.0.2): index=0 time=24.591 usec
42 bytes from 9a:e9:7a:6f:a1:51 (10.0.0.2): index=1 time=11.717 usec
42 bytes from 9a:e9:7a:6f:a1:51 (10.0.0.2): index=2 time=13.125 usec
42 bytes from 9a:e9:7a:6f:a1:51 (10.0.0.2): index=3 time=12.177 usec
42 bytes from 9a:e9:7a:6f:a1:51 (10.0.0.2): index=4 time=11.700 usec
^C
--- 10.0.0.2 statistics ---
5 packets transmitted, 5 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 0.012/0.015/0.025/0.005 ms
mininet>
```

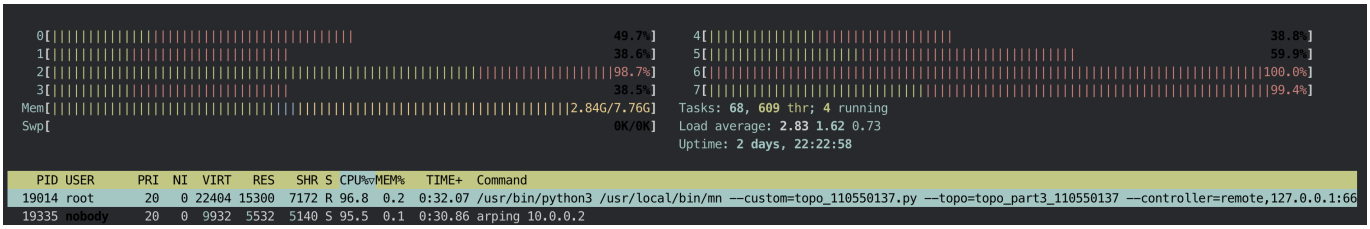
Flow Rule - Ping

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.080 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5092ms
rtt min/avg/max/mdev = 0.080/0.291/1.266/0.436 ms
mininet>
```

Part3

Broadcast Storm

- The arp packets flooding among all nodes. Because all node just re-broadcast the receiced arp packet again and again which cause the broadcast storm.
- CPU Ultilization



Part4

The `ReactivePacketProcessor` is the main part of a application dealing with packet forwarding. After passing a context object into the process function, the function will check if the context has been handled. If not record the packet into metric. After that, the function will determine the packet type, if the packet is control packet, drop it. If the packet is IPv4 multicast then let other application handle it. If the packet destination is unknown, then flood and bail the packet.

After the determination, it attempt to calculate a path which can be use to forward the packet to the right host. First, using `topologyService.getPaths` to get paths, if we are on a edge switch, we just simply forward out the packet to destination. If not the case, we get a set of paths that can lead us to the destination edge switch, and use `pickForwardPathIfPossible` to get the path.

After we obtain the path, we can just simply invoke `installRule` to let go the packets. `installRule` is the part that forwarding the packet to specified port.