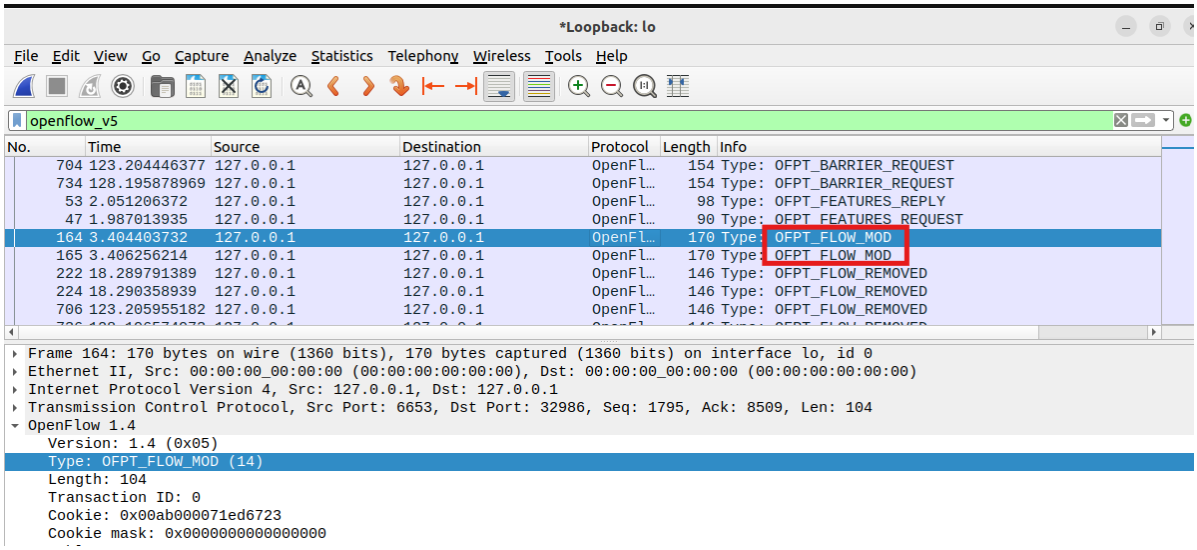
SDN-NFV Lab2

313552018 林哲緯

Part 1: Answer Questions

1. **How many OpenFlow headers with type “OFPT_FLOW_MOD” and command “OFPFC_ADD” are there among all the packets?**

   There are 2 OpenFlow headers with type “OFPT_FLOW_MOD” and command “OFPFC_ADD” among all the packets.



2. **What are the match fields and the corresponding actions in each “OFPT_FLOW_MOD” message?**

3. **What are the Idle Timeout values for all flow rules on s1 in GUI?**

Match fields	actions	Timeout values
IN_PORT:1	OUTPUT:2	0
IN_PORT:2	OUTPUT:1	0

Part 2: Install Flow Rules

Install **two** flow rules to forward IPv4 packets

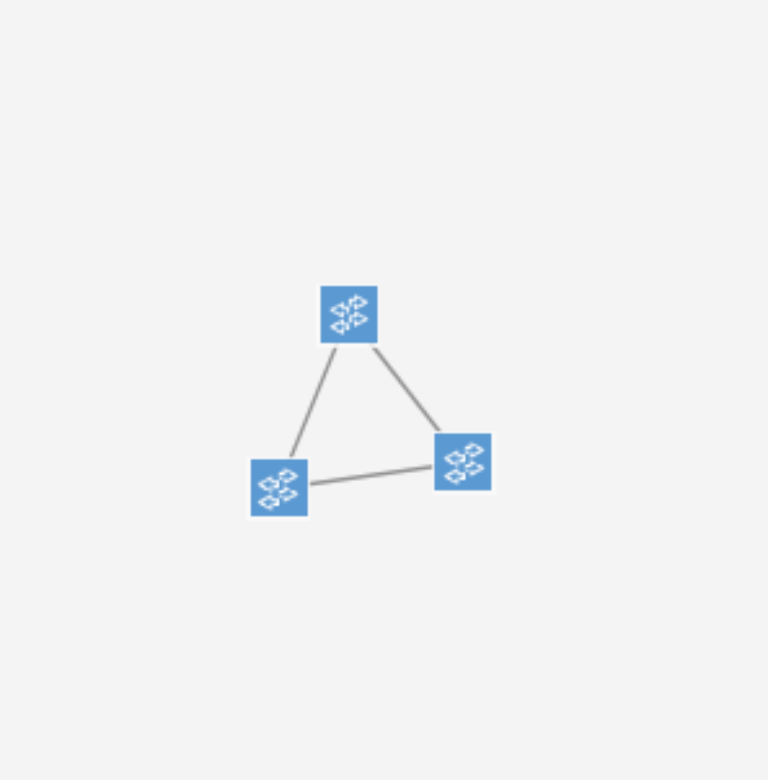
```
mininet> h1 arping h2
ARPING 10.0.0.2
42 bytes from 2e:7d:a9:9f:eb:46 (10.0.0.2): index=0 time=1.882 msec
42 bytes from 2e:7d:a9:9f:eb:46 (10.0.0.2): index=1 time=5.120 usec
42 bytes from 2e:7d:a9:9f:eb:46 (10.0.0.2): index=2 time=6.892 usec
42 bytes from 2e:7d:a9:9f:eb:46 (10.0.0.2): index=3 time=3.790 usec
42 bytes from 2e:7d:a9:9f:eb:46 (10.0.0.2): index=4 time=5.501 usec
```

Install **one** flow rule to forward ARP packets

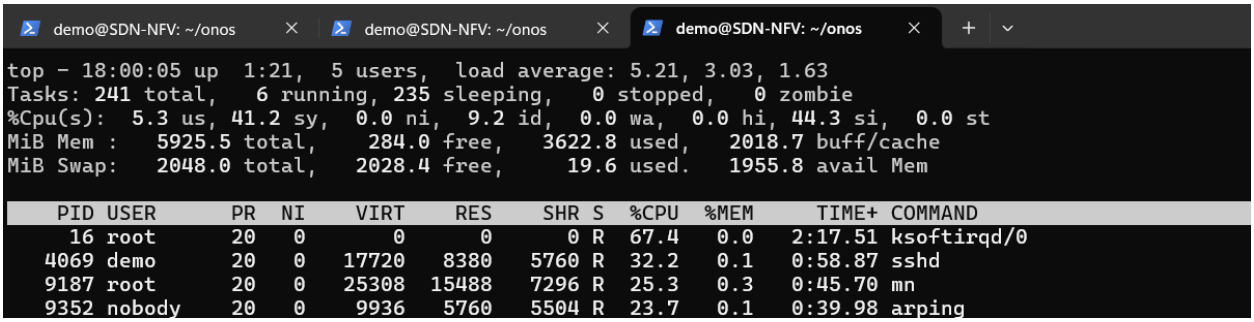
```
mininet> h1 ping h2 -c 3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.07 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.302 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.068 ms
```

Part 3: Create Topology with Broadcast Storm

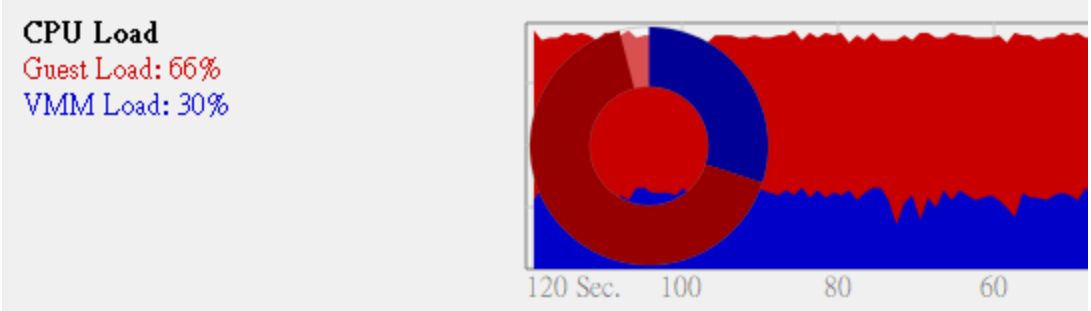
I installed three flow rules on each of the three switches to forward ARP packets to "ALL" ports.



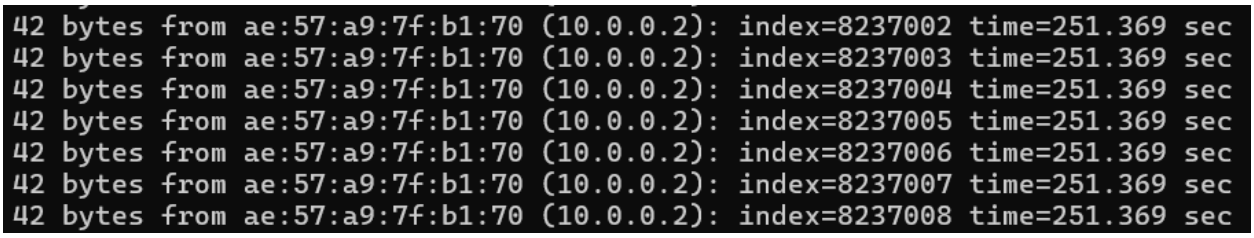
The topology, two hosts are connected to two of the switches.



Using the command `top` to observe cpu usage.



Observe the cpu usage through VirtualBox UI.



Broadcast storm, it makes 8 million packets in about four minutes.

Part 4: Trace ReactiveForwarding

1. Controller adds two flow rules to the switch through the packets with type “OFPT_FLOW_MOD”. (control plain)
2. When we command h1 ping h2, h1 broadcasts an ARP request to get h2's MAC address (data plane). The switch then generates a "PACKET_IN" event to the controller, and the controller responds with a "PACKET_OUT" event to the switch. I guess it’s because switch doesn’t have the flow rule based on IP, the controller needs to inform the switch of h2's MAC address to deliver the ARP request (control plain).
3. h2 receives the ARP request, it replies with its MAC address to h1 (data plain). It generates a pair of “PACKET_IN” and “PACKET_OUT” again. (control plain).
4. h1 sends an ICMP Echo request to h2 using h2's MAC address (data plane). The "PACKET_IN" and "PACKET_OUT" events, as explained in steps 2 and 3, occur again here (control plane).
5. The two flow rules added by the controller will first be deleted by "FLOW_REMOVE" packets after the default timeout of 10 seconds. When exiting Mininet, the switch sends "FLOW_REMOVE" packets to the controller to delete the default flow rules (control plain).

What I’ve learned or solved

1. If the dependency of match fields is overlooked (e.g., the prerequisite for `IPV4_DST` is `ETH_TYPE`), the rule will remain in the "PENDING_ADD" state rather than transitioning to "ADDED".
2. Get familiar with match fields and actions by reading openflow spec.

3. Observe the communication between switch and controller by using wireshark.