Project 2. OpenFlow

312551015 Sean 韋詠祥

Part 1: Answer Questions

Q1. How many OFPT_FLOW_MOD packets are there

When we enable `org.onosproject.openflow` and start the Mininet topology, the controller will send 3 `OFPFC_ADD` command within one packet, respectively for `lldp`, `arp`, `bddp` (broadcast domain discovery protocol).
After we enable `org.onosproject.fwd`, there will be 1 packet with `OFPFC_ADD` command to match IPv4 with priority 5

Then we use the command `h1 ping h2 -c 5` in Mininet, the controller will send two `OFPFC_ADD` command. The first one is the rule for h1 send packet to h2, the latter is for the opposite direction.

After 10 seconds timeout, two `OFPFC_DELETE_STRICT` command will be issued.

When we closed the Mininet, after four `OFPT_FLOW_REMOVED` command, it will be four `OFPFC_ADD` command for `lldp`, `arp`, `bddp`, `ipv4` respectively.

2. What are the fields in each OFPT_FLOW_MOD messages?

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

Match fields	Action	Timeout value
ETH_TYPE = LLDP	Output = Controller	0
ETH_TYPE = ARP	Output = Controller	0
ETH_TYPE = BDDP	Output = Controller	0
ETH_TYPE = IPv4	Output = Controller	0
IN_PORT = 1, ETH_SRC = h1, ETH_DST = h2	Output = Port 2	10 seconds
IN_PORT = 2, ETH_SRC = h2, ETH_DST = h1	Output = Port 1	10 seconds

3. What are the Idle Timeout values for all flow rules?

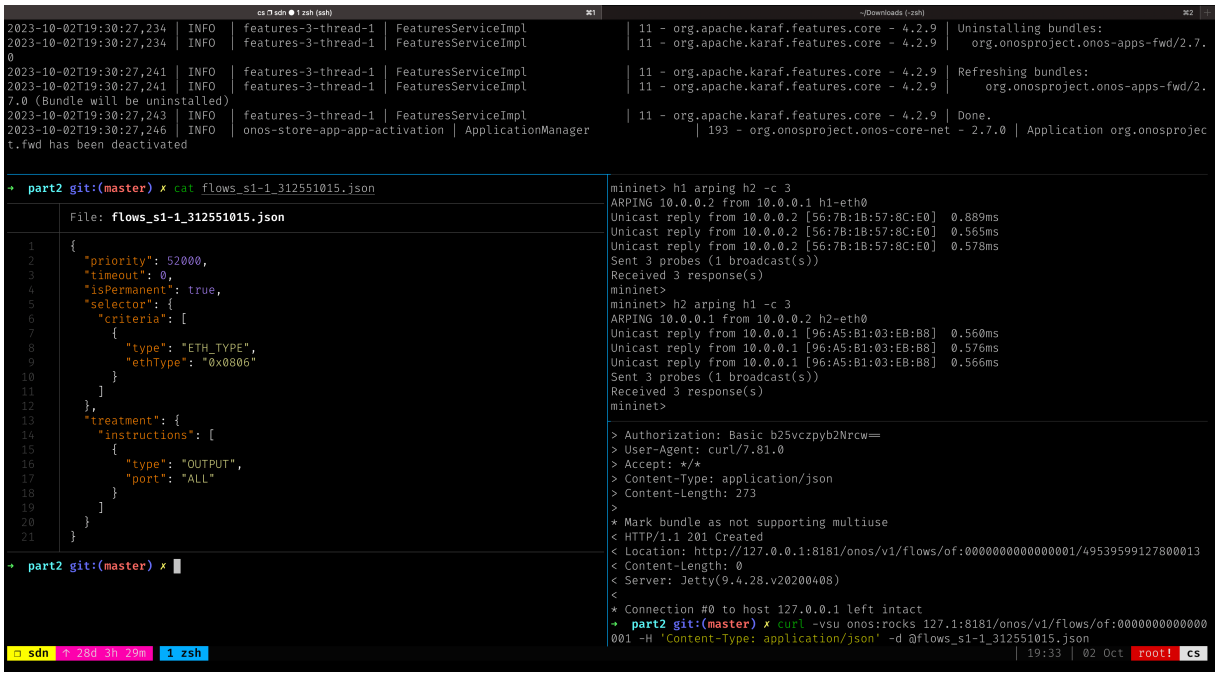
In the Web GUI, we can see the default rules (`lldp`, `arp`, `bddp`, `ipv4`) have Idle Timeout value of 0, which means there have no timeout limit.
For the two rules added when we ping from h1 to h2, it have timeout value of 10 seconds.

Part 2: Install Flow Rules

1. ARP

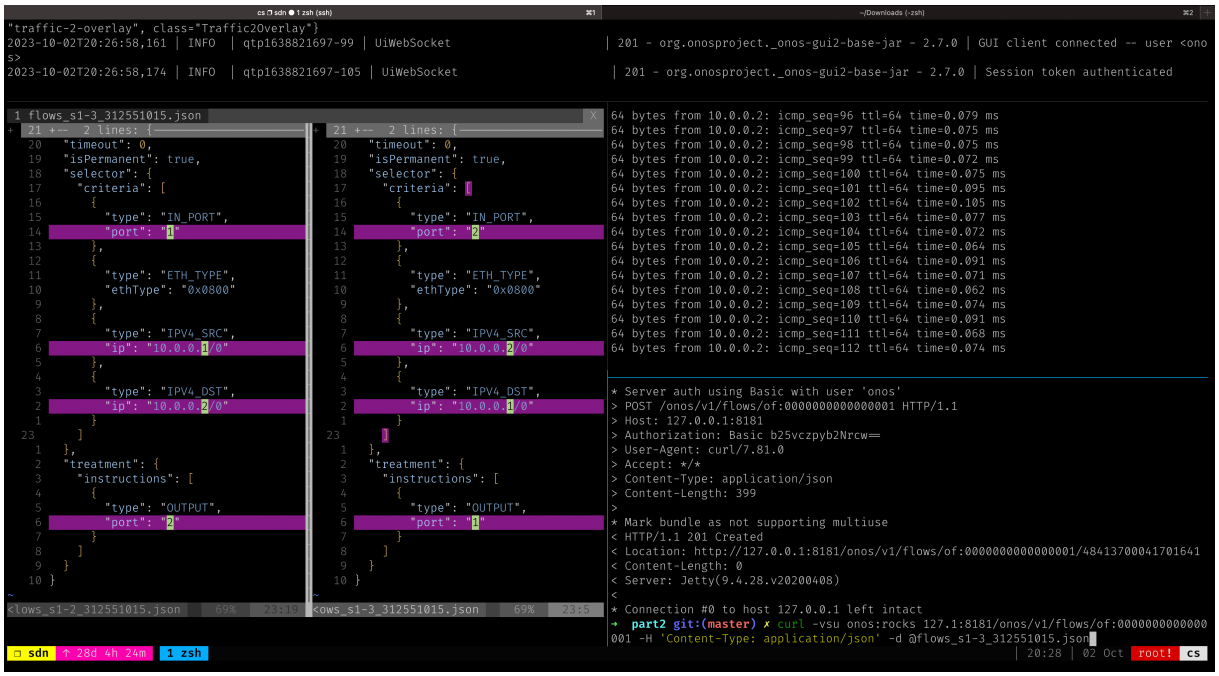
Use criteria `ETH_TYPE = 0x0806` (ARP) to match packets.

And the instructions is to `OUTPUT` to `ALL` ports.



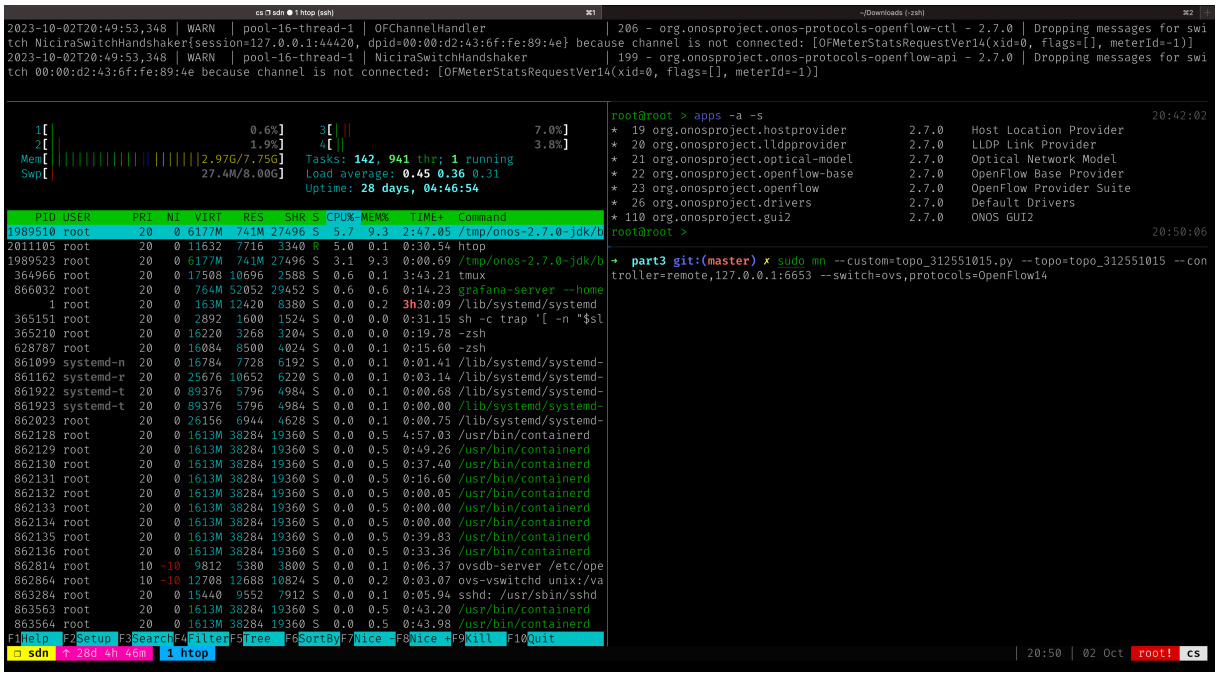
2. IPv4 Ping

For second part we use `IPV4_SRC` and `IPV4_DST` to decide where the packet should go.



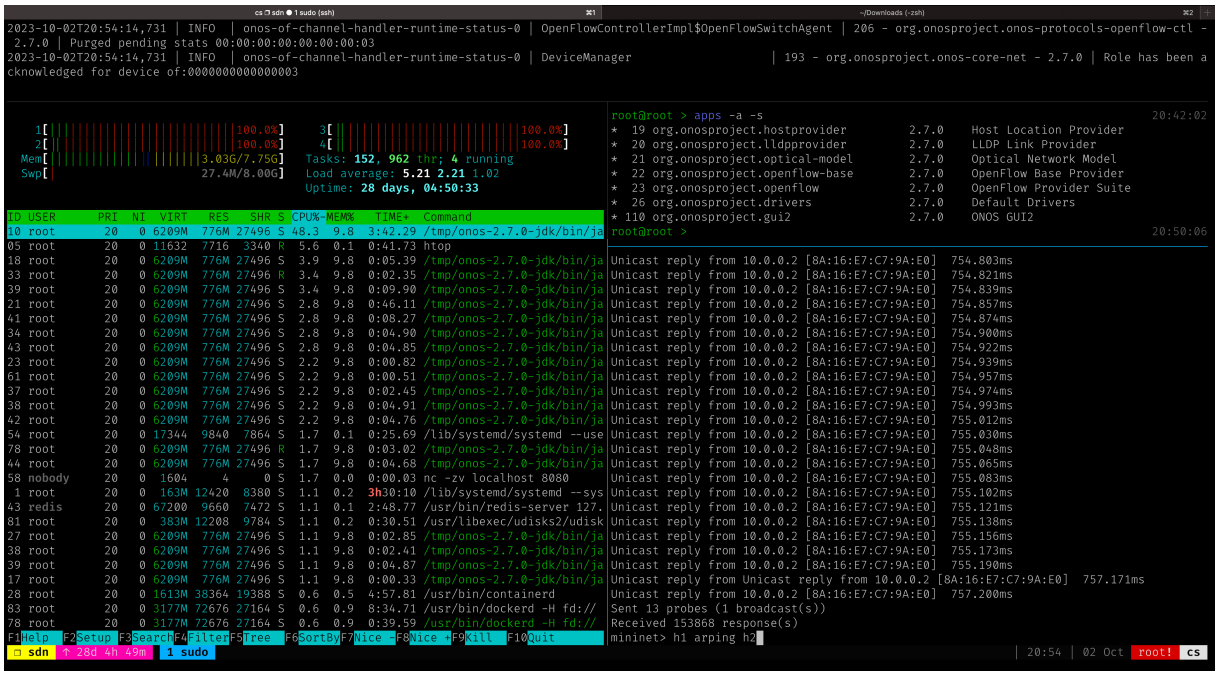
Part 3: Broadcast Storm

Before the experiment, my average CPU load is about 40%.



In my experiment environment design, I use four switch fully mesh with others. And install the flow rule from part 2-1, which forward all the ARP packets to all ports.

When we use `h1 ping h2` command in Mininet, the average CPU load is over 500%.



Part 4: Trace ReactiveForwarding

When we start the Mininet default topology and let h1 send a ICMP ping to h2, h1 will first lookup the MAC address of h2.
We can see a `OFPT_PACKET_IN` and `OFPT_PACKET_OUT` pair for ARP request, which is `10.0.0.1` asking who has `10.0.0.2`.
Then h2 respond to the ARP request, along with its MAC address, generate another pair of `OFPT_PACKET_IN` and `OFPT_PACKET_OUT` packets.
Once the MAC address of h2 is known, h1 will send out an ICMP packet (with `OFPT_PACKET_IN` in Wireshark), and the packet will be forwarded to h2.

Inside the `ReactiveForwarding` program, when it received the ARP request from h1, since the destination is `00:00:00:00:00:00`, it flood the packet to all ports.
It will add a rule `(IN_PORT = 1, ETH_SRC = h1, ETH_DST = h2) → OUT_PORT = 2` to flow table. And add the rule for opposite direction for packets send from h2 to h1.

Part 5: What we learned from this project

When I am doing Part 2-2 writing flow rule for IPv4, I was stuck because not fulfilled the “dependency”. I thought OpenFlow engine will warn me or reject the rule if I didn’t meet the requirement.
It take me a long time to realized it could be wrong, then follow the hint in the slide p.36 to add `ETH_TYPE` and other fields.