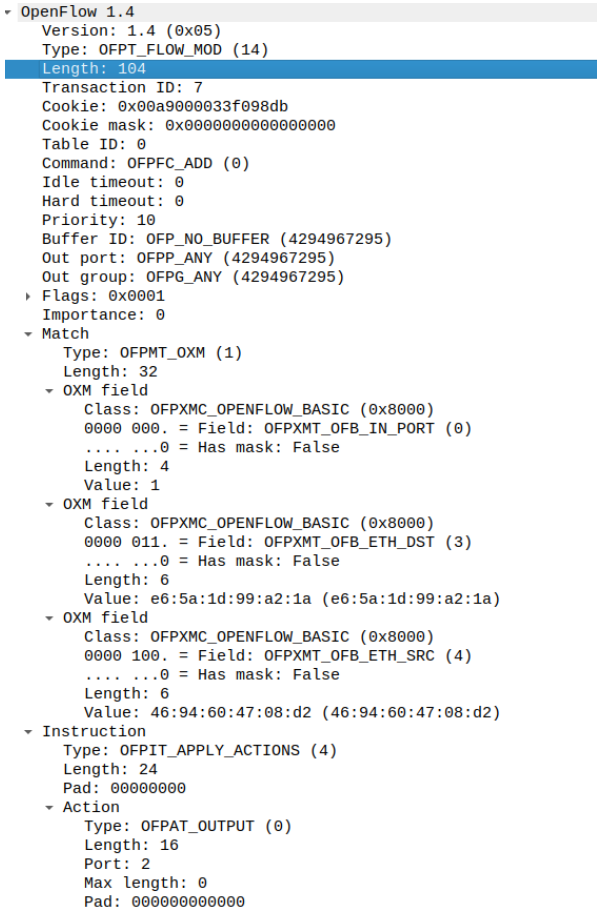
Project2

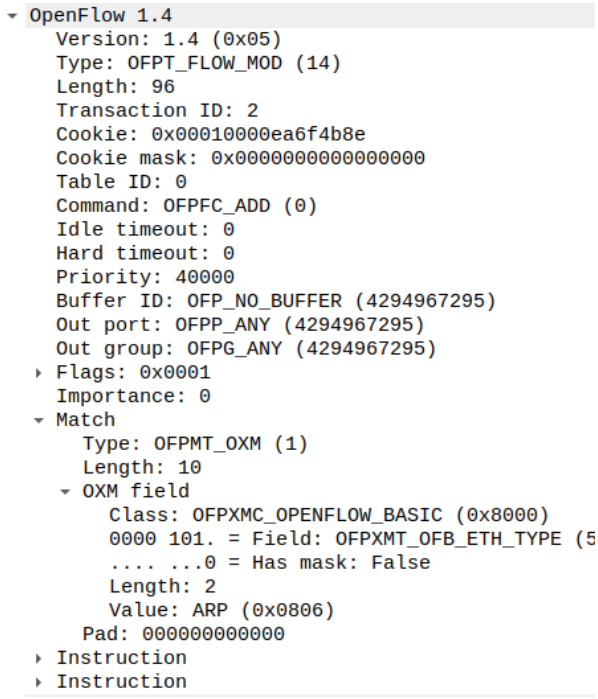
Part 1: Answer Questions

1. How many OpenFlow headers with type “OFPT_FLOW_ and command “OFPFC_ are there among all the packets?

2. What are the match fields and the corresponding actions in each “OFPT_FLOW_ message?

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





Ans: There are **2** distinct “OFPT_FLOW_ headers during the experiment

Match Fields	Actions	Timeout Values
ETH_TYPE(5) = IPv4(0x0800)	OFPAT_OUTPUT(0) Port = OFPP_CONTROLLER(4294967293)	0
IN_PORT(0) = 1 ETH_DST(3) = e6:5a:1d:99:a2:1a ETH_SRC(4) = 46:94:60:47:08:d2	OFPAT_APPLY_ACTIONS(4) Port = 2	0

Part 2: Install Flow Rules

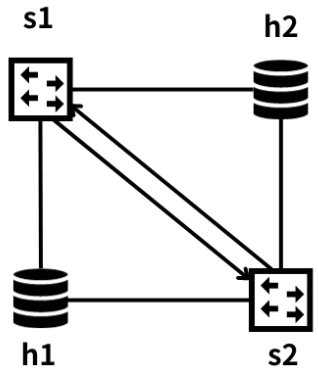
- **Arping**

```
mininet> h1 arping h2
ARPING 10.0.0.2
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=0 time=118.929 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=1 time=3.620 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=2 time=2.635 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=3 time=2.637 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=4 time=3.116 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=5 time=2.735 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=6 time=3.710 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=7 time=2.819 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=8 time=2.919 usec
42 bytes from 9a:b1:a5:da:b0:23 (10.0.0.2): index=9 time=3.996 usec
^C
--- 10.0.0.2 statistics ---
10 packets transmitted, 10 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 0.003/0.015/0.119/0.035 ms
```

- **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=0.255 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.037 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.037 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7164ms
rtt min/avg/max/mdev = 0.031/0.063/0.255/0.072 ms
```

Part 3: Create Topology with Broadcast Storm



My topology can shown as an illustration on the left side, and we can see there is a loop between two switches. So once any of these two switches receive a package, it will

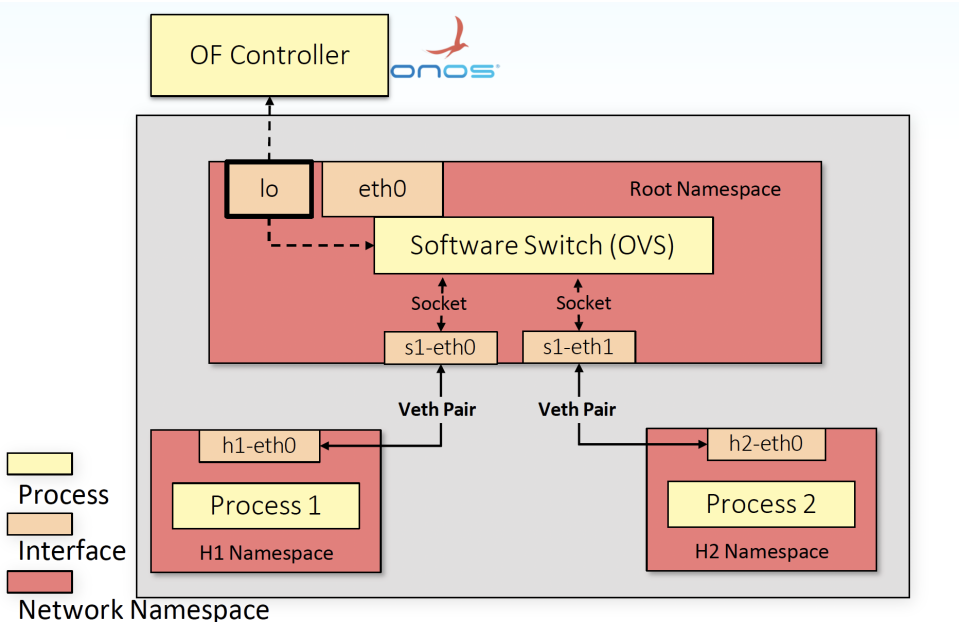
transfer to the other one, which will cause a terrible event called “broadcast storm” and consume a lot of resource.

Then we can see the picture on the right hand side, it show that the CPU usage of VM was almost 100%.

Part 4: Trace ReactiveForwarding

Control Plane Operations

During the time between when h1 initiates the ping and h2 receives the first ICMP request, I observed the following control plane operations:



1. ONOS, particularly the "ReactiveForwarding" application, will receive the ICMP packet from h1.
2. The "ReactiveForwarding" application will examine the packet and determine that there is no pre-existing flow rule for the ICMP traffic between h1 and h2.
3. The application will then generate a flow rule for the switch to handle ICMP traffic between these two hosts.
4. The flow rule will be installed in the switch(OVS) by the controller, specifying the actions required to forward ICMP traffic from h1 to h2.

Data Plane Operations

Observe the data plane operations during the time when h1 pings h2:

- When h1 sends the ICMP echo requests, the switch receives these packets.
- The switch matches the packets against the flow rule installed by the controller.
- The switch forwards the packets to the appropriate port to reach h2.

What you’ve learned or solved.

- understand the concept how controller interacts with switches on SDN
- understand the specifics of flowrule and what actions it can make
- the concept and reason of broadcast storm
- gotten much more familiar with OpenFlow protocol and Wireshark