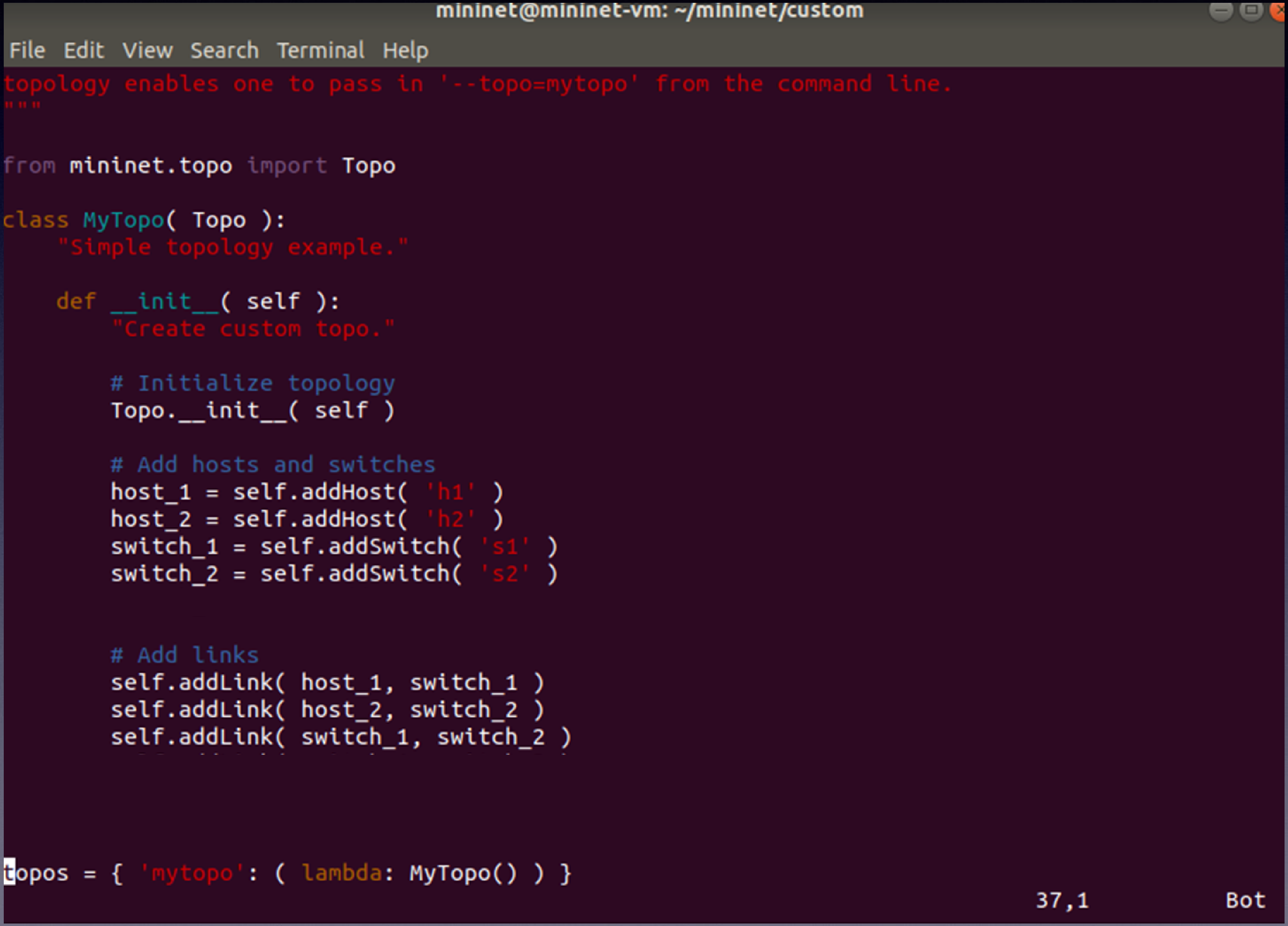
Lab 1
Mininet/ONOS Installation

Installation

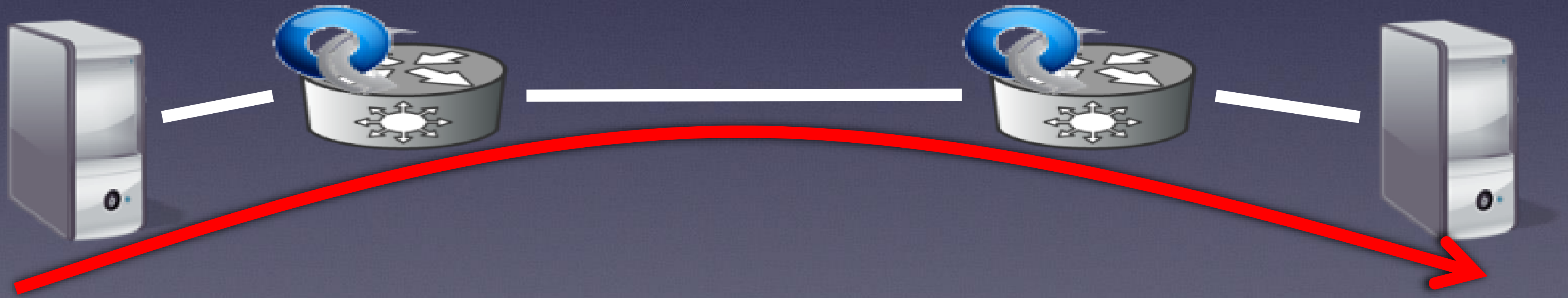
- Check the following website
- https://hackmd.io/_YRqnX0WSdmzGSQ68ObnUg?view

test.py



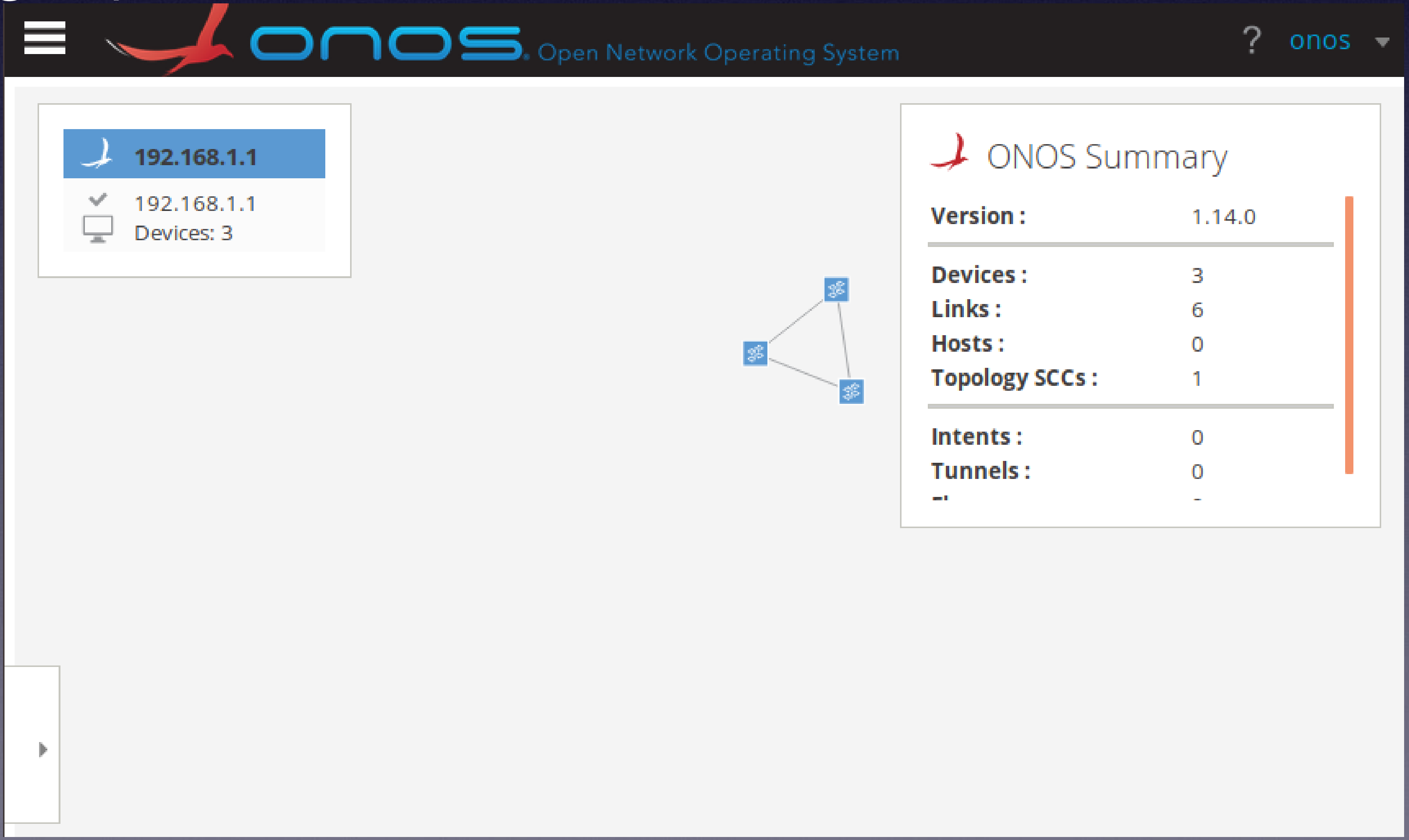
```
topology enables one to pass in '--topo=mytopo' from the command line.
"""

from mininet.topo import Topo

class MyTopo( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        host_1 = self.addHost( 'h1' )
        host_2 = self.addHost( 'h2' )
        switch_1 = self.addSwitch( 's1' )
        switch_2 = self.addSwitch( 's2' )


        # Add links
        self.addLink( host_1, switch_1 )
        self.addLink( host_2, switch_2 )
        self.addLink( switch_1, switch_2 )




topos = { 'mytopo': ( lambda: MyTopo() ) }
                                                            37,1          Bot
```

Topology



Topology via Web Page

- http://localhost:8181/onos/ui/index.html
- Login/password = onos/rocks



192.168.1.1

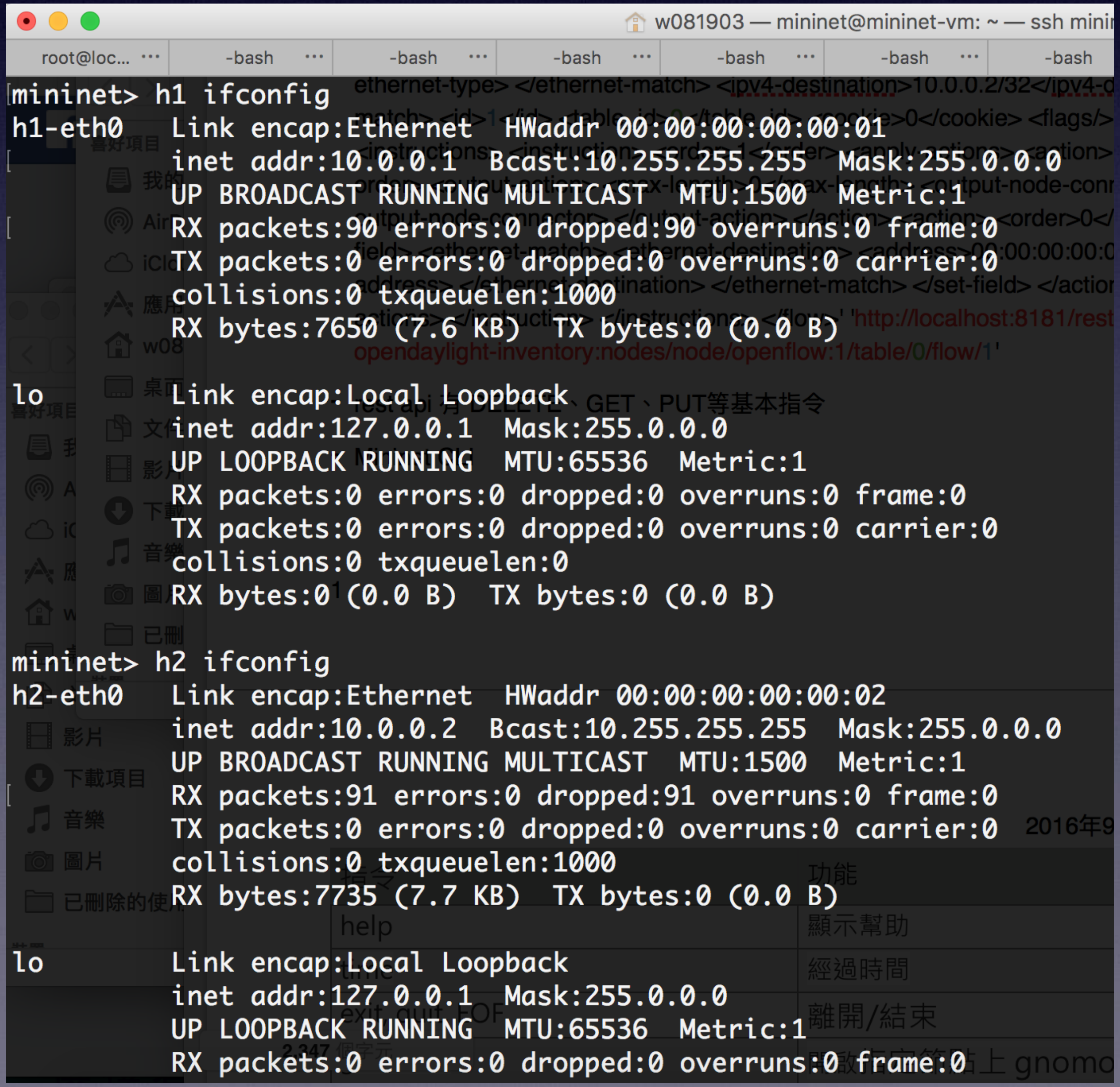
192.168.1.1
Devices: 3

ONOS Summary

Version :	1.14.0
Devices :	3
Links :	6
Hosts :	0
Topology SCCs :	1
Intents :	0
Tunnels :	0

Mininet test

- In mininet environment we can use h1 ifconfig to check what h1’s ip address it is.



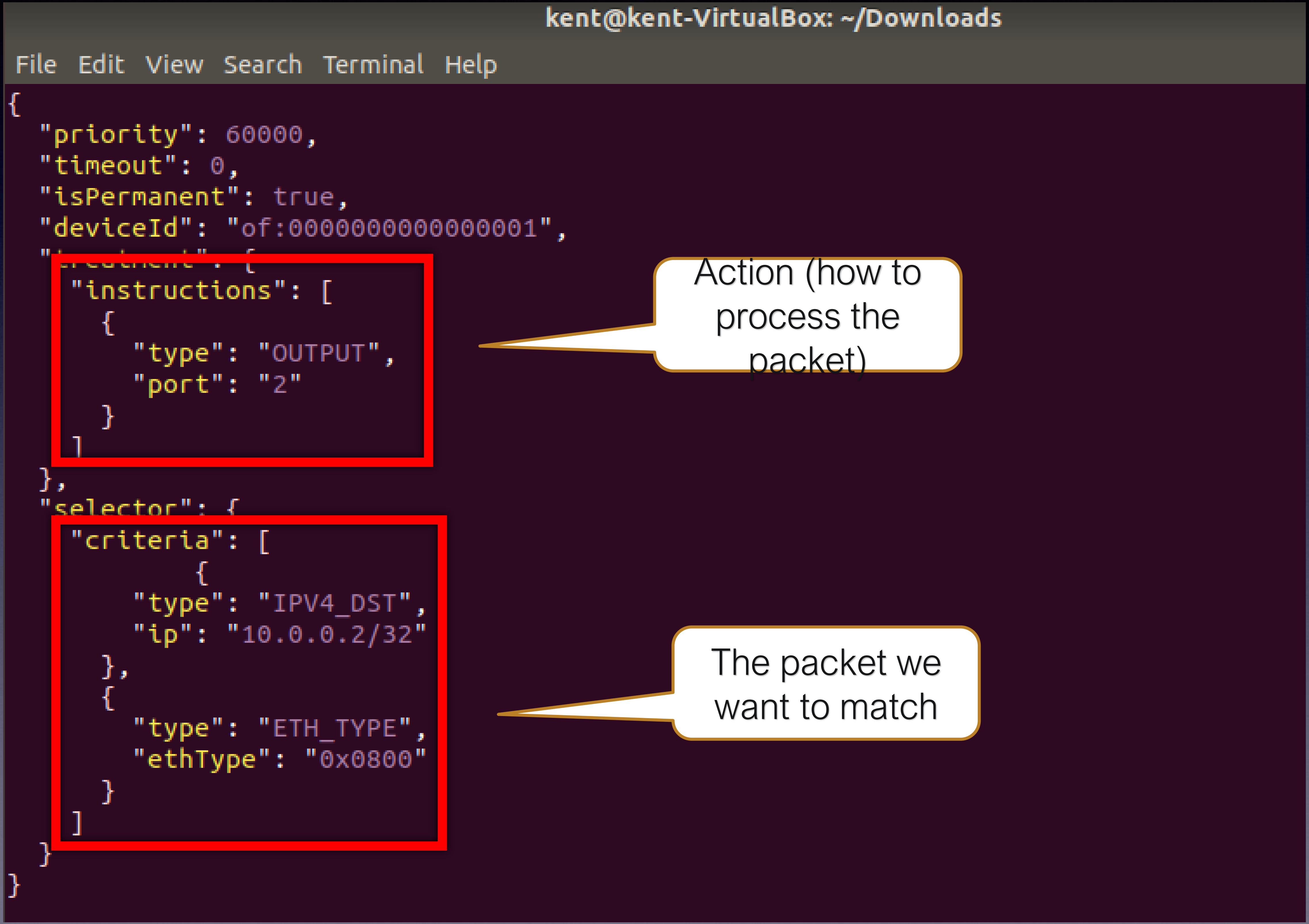
```
mininet> h1 ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:90 errors:0 dropped:90 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7650 (7.6 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mininet> h2 ifconfig
h2-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:02
          inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:91 errors:0 dropped:91 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7735 (7.7 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

Install flow to vSwitch

- We can use Rest API to install flow from controller to vswitch
- curl -X POST -H "content-type:application/json" http://localhost:8181/onos/v1/flows/of:0000000000000001 -d @./test_1.json --user onos:rocks

Switch ID
Json file for the rule

Install flow to vSwitch


kent@kent-VirtualBox: ~/Downloads
File Edit View Search Terminal Help

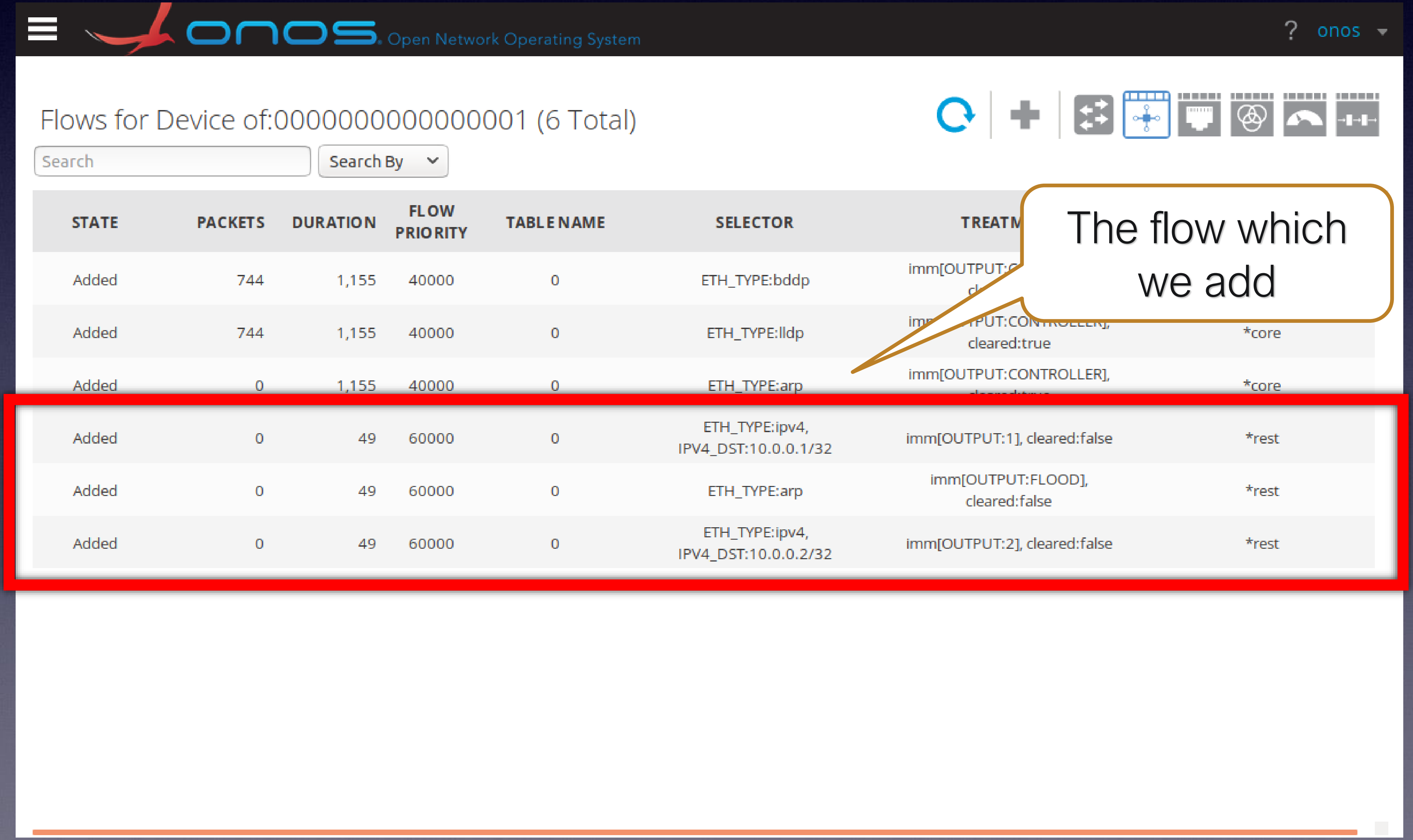
```
{
  "priority": 60000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000001",
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "2"
      }
    ]
  },
  "selector": {
    "criteria": [
            {
        "type": "IPV4_DST",
        "ip": "10.0.0.2/32"
      },
      {
        "type": "ETH_TYPE",
        "ethType": "0x0800"
      }
    ]
  }
}
```

Action (how to process the packet)

The packet we want to match

Check Flow

- ONOS web site



Flows for Device of:0000000000000001 (6 Total)

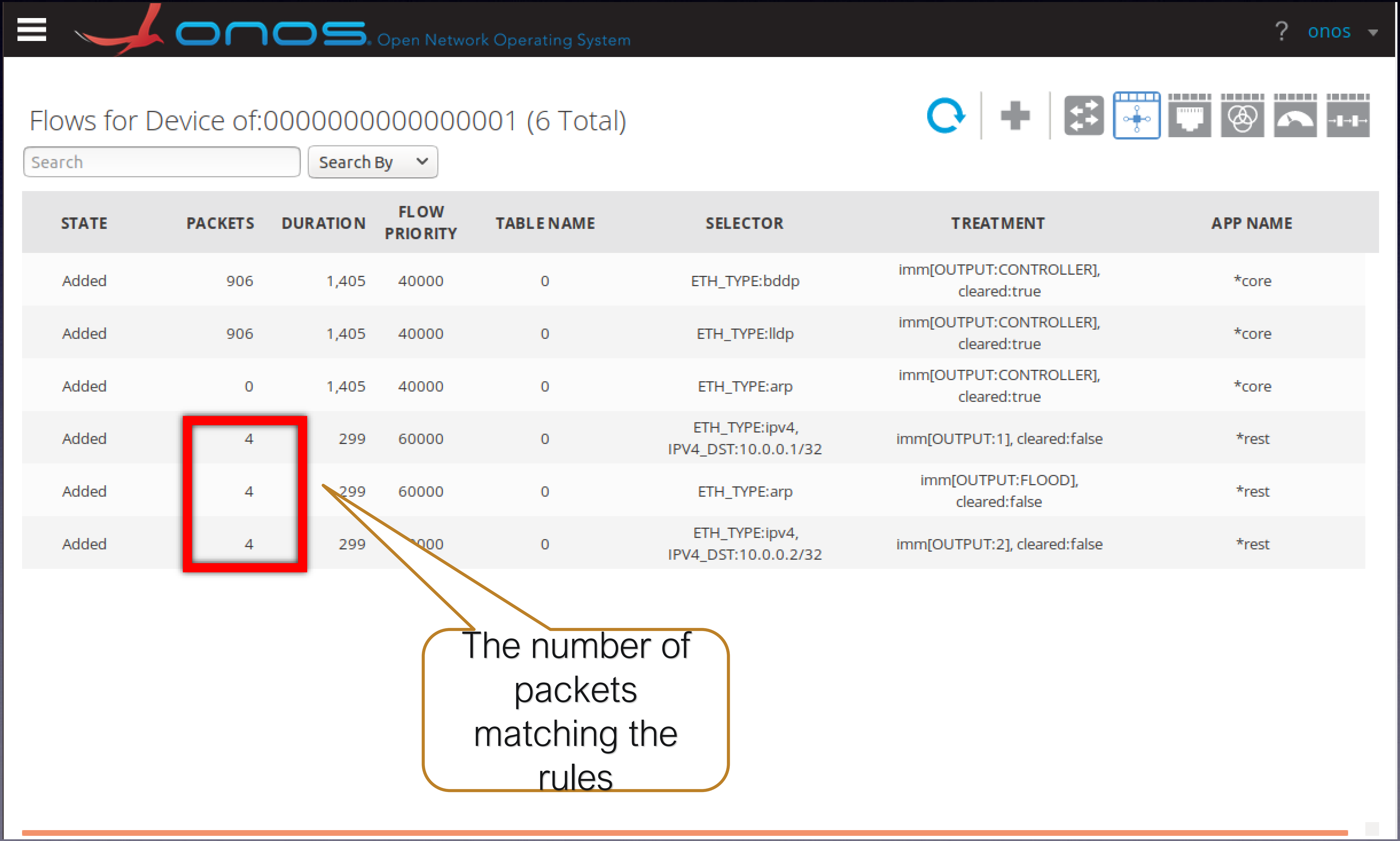
STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATM	
Added	744	1,155	40000	0	ETH_TYPE:bddp	imm[OUTPUT:C cl	
Added	744	1,155	40000	0	ETH_TYPE:lldp	im PUT:CONTROLLER], cleared:true	*core
Added	0	1,155	40000	0	ETH_TYPE:arp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	0	49	60000	0	ETH_TYPE:ipv4, IPV4_DST:10.0.0.1/32	imm[OUTPUT:1], cleared:false	*rest
Added	0	49	60000	0	ETH_TYPE:arp	imm[OUTPUT:FLOOD], cleared:false	*rest
Added	0	49	60000	0	ETH_TYPE:ipv4, IPV4_DST:10.0.0.2/32	imm[OUTPUT:2], cleared:false	*rest

The flow which we add

Test Traffic

- Use iperf or ping
- h1 ping h2
- iperf

Check Flow



Flows for Device of:0000000000000001 (6 Total)

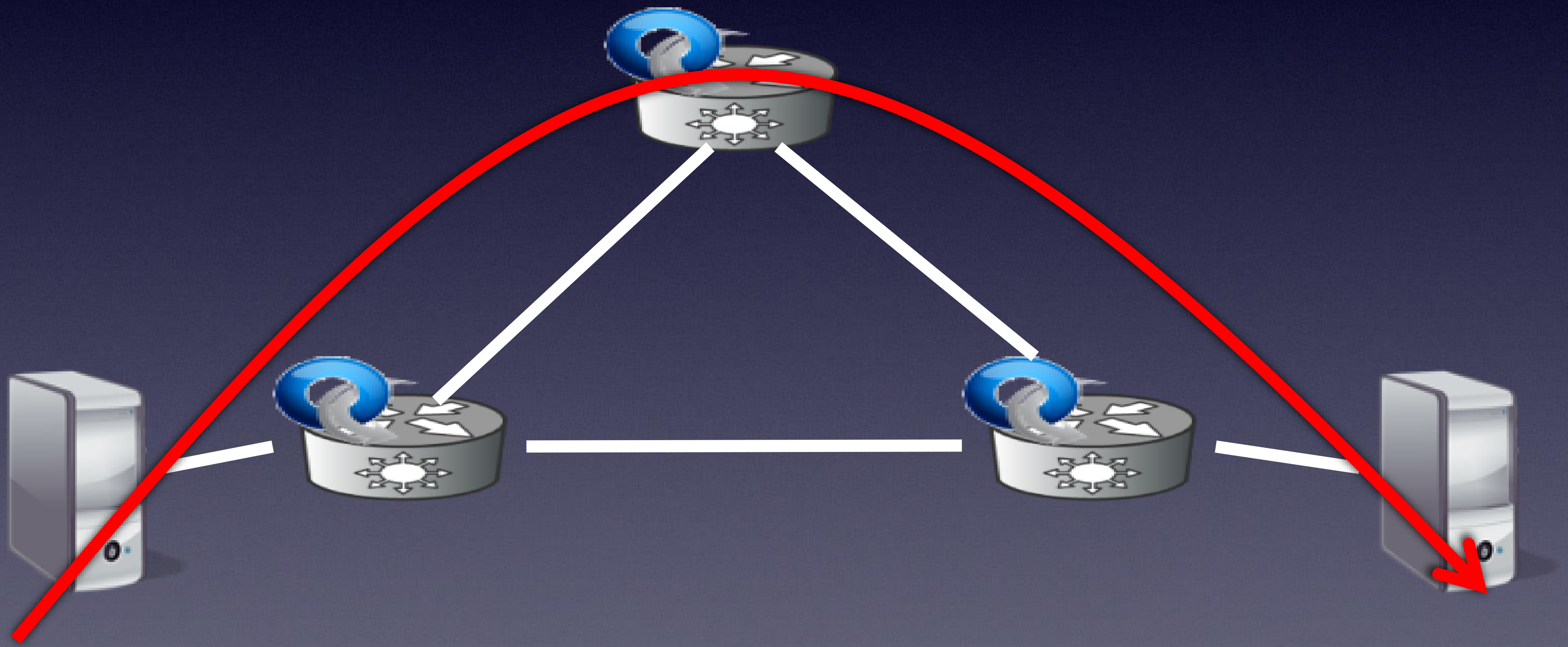
STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	906	1,405	40000	0	ETH_TYPE:bddp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	906	1,405	40000	0	ETH_TYPE:lldp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	0	1,405	40000	0	ETH_TYPE:arp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	4	299	60000	0	ETH_TYPE:ipv4, IPV4_DST:10.0.0.1/32	imm[OUTPUT:1], cleared:false	*rest
Added	4	299	60000	0	ETH_TYPE:arp	imm[OUTPUT:FLOOD], cleared:false	*rest
Added	4	299	60000	0	ETH_TYPE:ipv4, IPV4_DST:10.0.0.2/32	imm[OUTPUT:2], cleared:false	*rest

The number of packets matching the rules

Homework

- 產生有兩條路徑的topology
- h1 ping h2
- 讓ping packet走較長的路徑
- 在moodle上傳三張螢幕截圖:
    - Switch s3的forwarding table
    - Ping 成功的畫面
    - ONOS GUI中的topology
- Due: 11/5, 2020 (THU.)

Topology



reference

- ONOS and mininet
    - http://www.stackguy.com/archives/248
- json for rule
    - https://wiki.onosproject.org/display/ONOS/Flow+Rules
- curl to install rules
    - https://wiki.onosproject.org/display/ONOS/REST