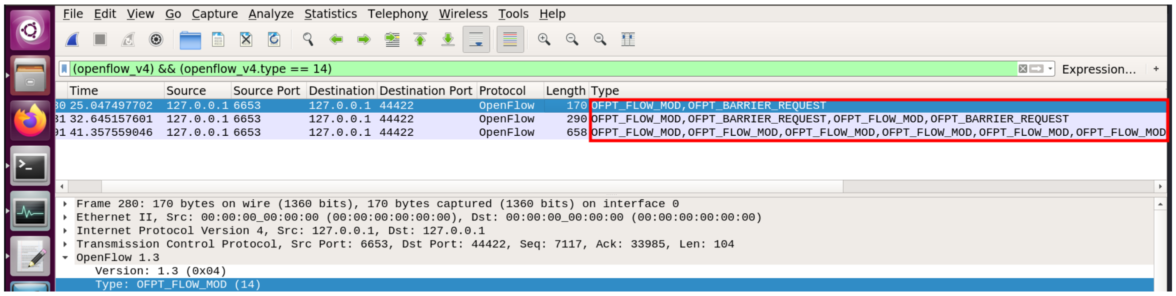
Lab 2 - OpenFlow + Flow Rule

0866007 胡孝德

Part 1: Answer Questions

1. How many OpenFlow headers of type “OFPT_FLOW_MOD” are there among all the packets?

- 9 個，如圖所示。

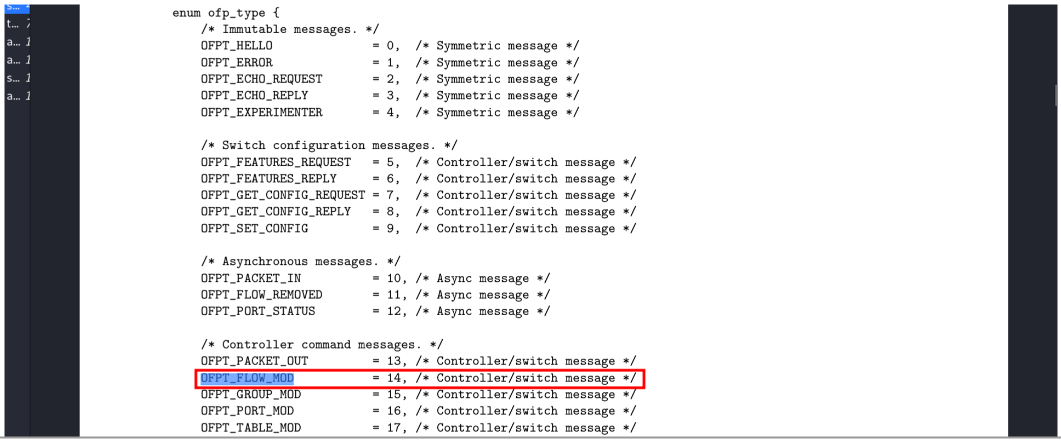

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

```
(openflow_v4) && (openflow_v4.type == 14)
```

```
 Time            Source    Source Port Destination Destination Port Protocol  Length Type
30 25.047497702  127.0.0.1 6653        127.0.0.1   44422            OpenFlow     170 OFPT_FLOW_MOD,OFPT_BARRIER_REQUEST
31 32.645157601  127.0.0.1 6653        127.0.0.1   44422            OpenFlow     290 OFPT_FLOW_MOD,OFPT_BARRIER_REQUEST,OFPT_FLOW_MOD,OFPT_BARRIER_REQUEST
91 41.357559046  127.0.0.1 6653        127.0.0.1   44422            OpenFlow     658 OFPT_FLOW_MOD,OFPT_FLOW_MOD,OFPT_FLOW_MOD,OFPT_FLOW_MOD,OFPT_FLOW_MOD,OFPT_FLOW_MOD
```

```
▸ Frame 280: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 6653, Dst Port: 44422, Seq: 7117, Ack: 33985, Len: 104
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
```

> - 由於每個封包可能有多個 OFPT_FLOW_MOD header，一個一個將封包點開來很容易算錯。於是將 openflow 的 type 作為一個欄位一次全部顯示出來，再做統計。
> - OFPT_FLOW_MOD 的 type id 可在 Openflow Spec 當中查到，如圖所示。



```
enum ofp_type {
    /* Immutable messages. */
    OFPT_HELLO              = 0,  /* Symmetric message */
    OFPT_ERROR              = 1,  /* Symmetric message */
    OFPT_ECHO_REQUEST       = 2,  /* Symmetric message */
    OFPT_ECHO_REPLY         = 3,  /* Symmetric message */
    OFPT_EXPERIMENTER       = 4,  /* Symmetric message */

    /* Switch configuration messages. */
    OFPT_FEATURES_REQUEST   = 5,  /* Controller/switch message */
    OFPT_FEATURES_REPLY     = 6,  /* Controller/switch message */
    OFPT_GET_CONFIG_REQUEST = 7,  /* Controller/switch message */
    OFPT_GET_CONFIG_REPLY   = 8,  /* Controller/switch message */
    OFPT_SET_CONFIG         = 9,  /* Controller/switch message */

    /* Asynchronous messages. */
    OFPT_PACKET_IN          = 10, /* Async message */
    OFPT_FLOW_REMOVED       = 11, /* Async message */
    OFPT_PORT_STATUS        = 12, /* Async message */

    /* Controller command messages. */
    OFPT_PACKET_OUT         = 13, /* Controller/switch message */
    OFPT_FLOW_MOD           = 14, /* Controller/switch message */
    OFPT_GROUP_MOD          = 15, /* Controller/switch message */
    OFPT_PORT_MOD           = 16, /* Controller/switch message */
    OFPT_TABLE_MOD          = 17, /* Controller/switch message */
```

2. Flow Rules

a. What are the matching fields and the correspondng actions in each of “OFPT_FLOW_MOD” messages?

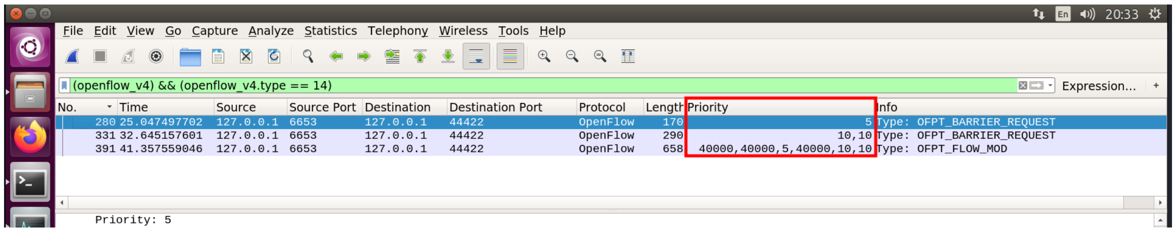
- 1st message
  - matching fields
    - OFPXMT_OFB_ETH_TYPE
  - actions
    - OFPIT_CLEAR_ACTIONS
    - OFPIT_APPLY_ACTIONS

- 2nd message
  - matching fields
    - OFPXMT_OFB_IN_PORT
    - OFPXMT_OFB_ETH_DST
    - OFPXMT_OFB_ETH_SRC
  - actions
    - OFPIT_APPLY_ACTIONS
- 3rd message
  - matching fields
    - OFPXMT_OFB_IN_PORT
    - OFPXMT_OFB_ETH_DST
    - OFPXMT_OFB_ETH_SRC
  - actions
    - OFPIT_APPLY_ACTIONS
- 4th message
  - matching fields
    - OFPXMT_OFB_ETH_TYPE
  - actions
    - OFPIT_CLEAR_ACTIONS
    - OFPIT_APPLY_ACTIONS
- 5th message
  - matching fields
    - OFPXMT_OFB_ETH_TYPE
  - actions
    - OFPIT_CLEAR_ACTIONS
    - OFPIT_APPLY_ACTIONS
- 6th message
  - matching fields
    - OFPXMT_OFB_ETH_TYPE
  - actions
    - OFPIT_CLEAR_ACTIONS
    - OFPIT_APPLY_ACTIONS
- 7th message
  - matching fields
    - OFPXMT_OFB_ETH_TYPE
  - actions
    - OFPIT_CLEAR_ACTIONS
    - OFPIT_APPLY_ACTIONS
- 8th message
  - matching fields
    - OFPXMT_OFB_IN_PORT
    - OFPXMT_OFB_ETH_DST
    - OFPXMT_OFB_ETH_SRC
  - actions

        - OFPIT_APPLY_ACTIONS
- 9th message
    - matching fields
        - OFPXMT_OFB_IN_PORT
        - OFPXMT_OFB_ETH_DST
        - OFPXMT_OFB_ETH_SRC
    - actions
        - OFPIT_APPLY_ACTIONS

b. What are the values of the priority fields of all “OFPT_FLOW_MOD” messages?

- 按照時間順序，priority 分別是 5, 10, 10, 40000, 40000, 5, 40000, 10, 10，如圖所示。


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
20:33
Expression...

```
(openflow_v4) && (openflow_v4.type == 14)
```

No.	Time	Source	Source Port	Destination	Destination Port	Protocol	Length	Priority	Info
280	25.047497702	127.0.0.1	6653	127.0.0.1	44422	OpenFlow	170	5	Type: OFPT_BARRIER_REQUEST
331	32.645157601	127.0.0.1	6653	127.0.0.1	44422	OpenFlow	290	10,10	Type: OFPT_BARRIER_REQUEST
391	41.357559046	127.0.0.1	6653	127.0.0.1	44422	OpenFlow	658	40000,40000,5,40000,10,10	Type: OFPT_FLOW_MOD

```
Priority: 5
```

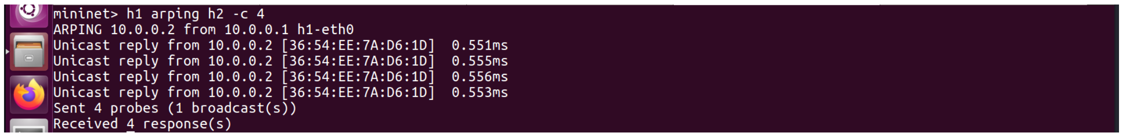
Part 2: Install Flow Rules

ARP:

- Flow Rule

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000001",
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "ALL"
      }
    ]
  },
  "selector": {
    "criteria": [
      {
        "type": "ETH_TYPE",
        "ethType": "0x0806"
      }
    ]
  }
}
```

- Verify



```
mininet> h1 arping h2 -c 4
ARPING 10.0.0.2 from 10.0.0.1 h1-eth0
Unicast reply from 10.0.0.2 [36:54:EE:7A:D6:1D]  0.551ms
Unicast reply from 10.0.0.2 [36:54:EE:7A:D6:1D]  0.555ms
Unicast reply from 10.0.0.2 [36:54:EE:7A:D6:1D]  0.556ms
Unicast reply from 10.0.0.2 [36:54:EE:7A:D6:1D]  0.553ms
Sent 4 probes (1 broadcast(s))
Received 4 response(s)
```

> - ARP 的 ETH_TYPE (0x0806) 可由以下網址查出。
>   https://zh.wikipedia.org/wiki/以太类型
>   (https://zh.wikipedia.org/wiki/%E4%BB%A5%E5%A4%AA%E7%B1%BB%E5%9E%8B)

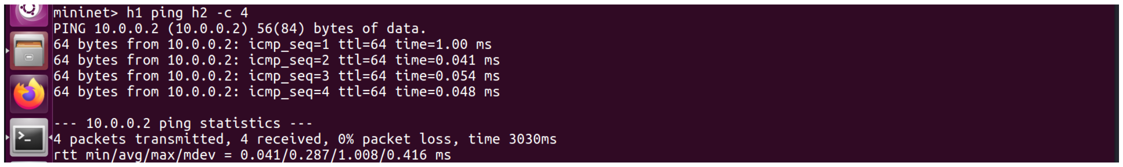
IPv4:

- Flow Rules

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000001",
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "2"
      }
    ]
  },
  "selector": {
    "criteria": [
      {
        "type": "ETH_TYPE",
        "ethType": "0x0800"
      },
      {
        "type": "IPV4_DST",
        "ip": "10.0.0.2/32"
      }
    ]
  }
}
```

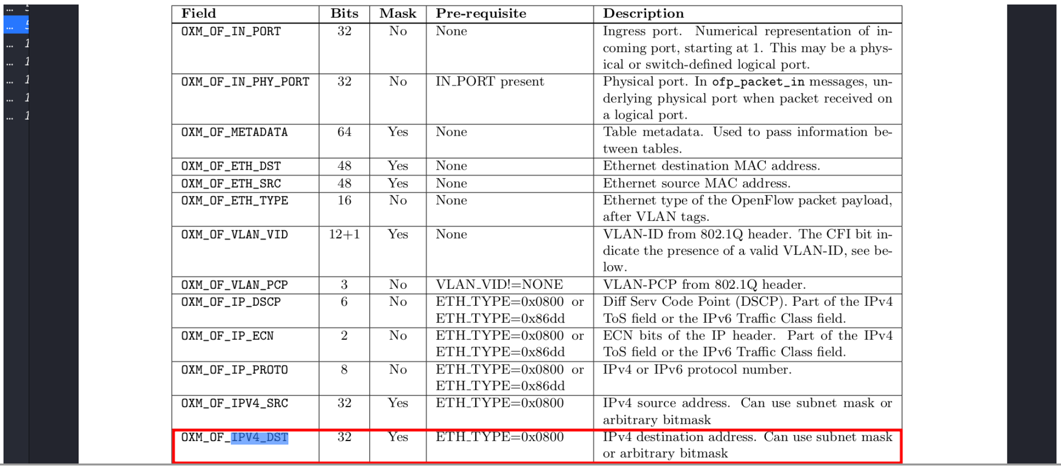
```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000001",
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "1"
      }
    ]
  },
  "selector": {
    "criteria": [
      {
        "type": "ETH_TYPE",
        "ethType": "0x0800"
      },
      {
        "type": "IPV4_DST",
        "ip": "10.0.0.1/32"
      }
    ]
  }
}
```

- Verify



```
mininet> h1 ping h2 -c 4
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=1.00 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.048 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3030ms
rtt min/avg/max/mdev = 0.041/0.287/1.008/0.416 ms
```

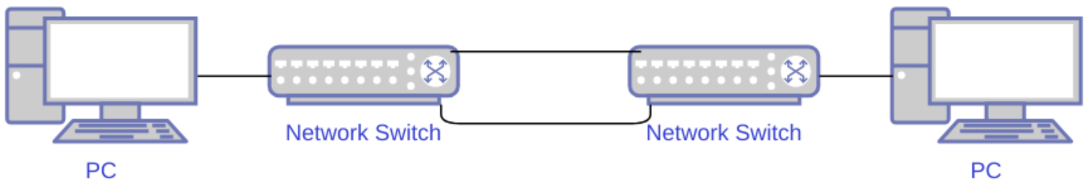
- Criteria IPV4_DST 跟 ETH_TYPE 具有 dependency，可由 Openflow Spec 得知，如圖所示。



Field	Bits	Mask	Pre-requisite	Description
OXM_OF_IN_PORT	32	No	None	Ingress port. Numerical representation of incoming port, starting at 1. This may be a physical or switch-defined logical port.
OXM_OF_IN_PHY_PORT	32	No	IN_PORT present	Physical port. In ofp_packet_in messages, underlying physical port when packet received on a logical port.
OXM_OF_METADATA	64	Yes	None	Table metadata. Used to pass information between tables.
OXM_OF_ETH_DST	48	Yes	None	Ethernet destination MAC address.
OXM_OF_ETH_SRC	48	Yes	None	Ethernet source MAC address.
OXM_OF_ETH_TYPE	16	No	None	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_VLAN_VID	12+1	Yes	None	VLAN-ID from 802.1Q header. The CFI bit indicate the presence of a valid VLAN-ID, see below.
OXM_OF_VLAN_PCP	3	No	VLAN_VID!=NONE	VLAN-PCP from 802.1Q header.
OXM_OF_IP_DSCP	6	No	ETH_TYPE=0x0800 or ETH_TYPE=0x86dd	Diff Serv Code Point (DSCP). Part of the IPv4 ToS field or the IPv6 Traffic Class field.
OXM_OF_IP_ECN	2	No	ETH_TYPE=0x0800 or ETH_TYPE=0x86dd	ECN bits of the IP header. Part of the IPv4 ToS field or the IPv6 Traffic Class field.
OXM_OF_IP_PROTO	8	No	ETH_TYPE=0x0800 or ETH_TYPE=0x86dd	IPv4 or IPv6 protocol number.
OXM_OF_IPV4_SRC	32	Yes	ETH_TYPE=0x0800	IPv4 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV4_DST	32	Yes	ETH_TYPE=0x0800	IPv4 destination address. Can use subnet mask or arbitrary bitmask

Part 3: Create Topology with Broadcast Storm

- Topology



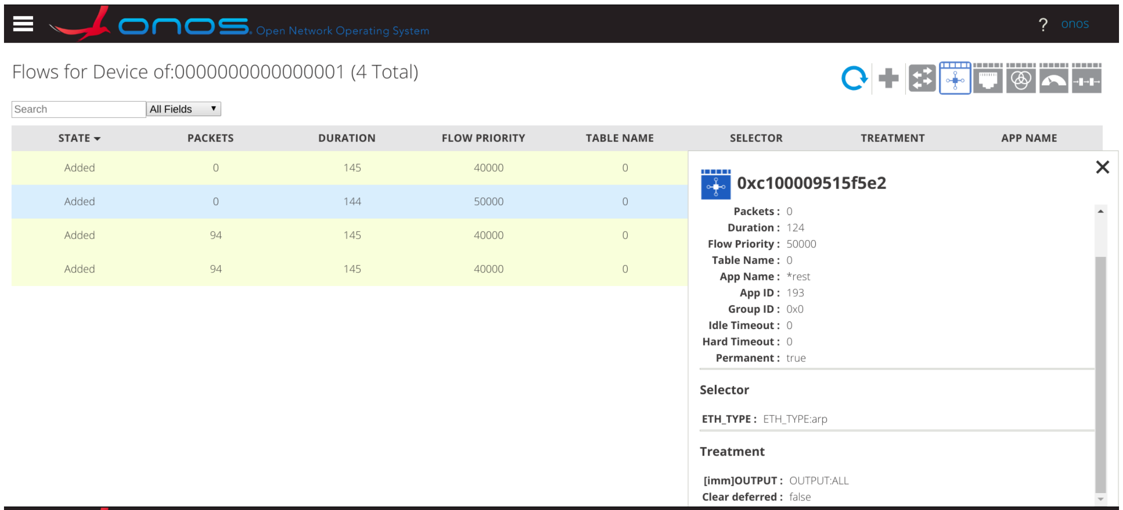
- Mininet Custom Topology Script

```
from mininet.topo import Topo


class Project2_Topo_0866007(Topo):
        def __init__(self):
                Topo.__init__(self)

                # Add hosts
                h1 = self.addHost('h1')
                h2 = self.addHost('h2')

                # Add switches
                s1 = self.addSwitch('s1')
                s2 = self.addSwitch('s2')

                # Add links
                self.addLink(h1, s1)
                self.addLink(h2, s2)
                self.addLink(s1, s2)
                self.addLink(s1, s2)

topos = {'topo_0866007': Project2_Topo_0866007}
```

- Flow Rules

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000001",
  "treatment": {
        "instructions": [
          {
                "type": "OUTPUT",
                "port": "ALL"
          }
        ]
  },
  "selector": {
        "criteria": [
          {
                "type": "ETH_TYPE",
                "ethType": "0x0806"
          }
        ]
  }
}
```

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "deviceId": "of:0000000000000002",
  "treatment": {
        "instructions": [
          {
                "type": "OUTPUT",
                "port": "ALL"
          }
        ]
  },
  "selector": {
        "criteria": [
          {
                "type": "ETH_TYPE",
                "ethType": "0x0806"
          }
        ]
  }
}
```



Flows for Device of:0000000000000001 (4 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	145	40000	0			
Added	0	144	50000	0			
Added	94	145	40000	0			
Added	94	145	40000	0			

0xc100009515f5e2

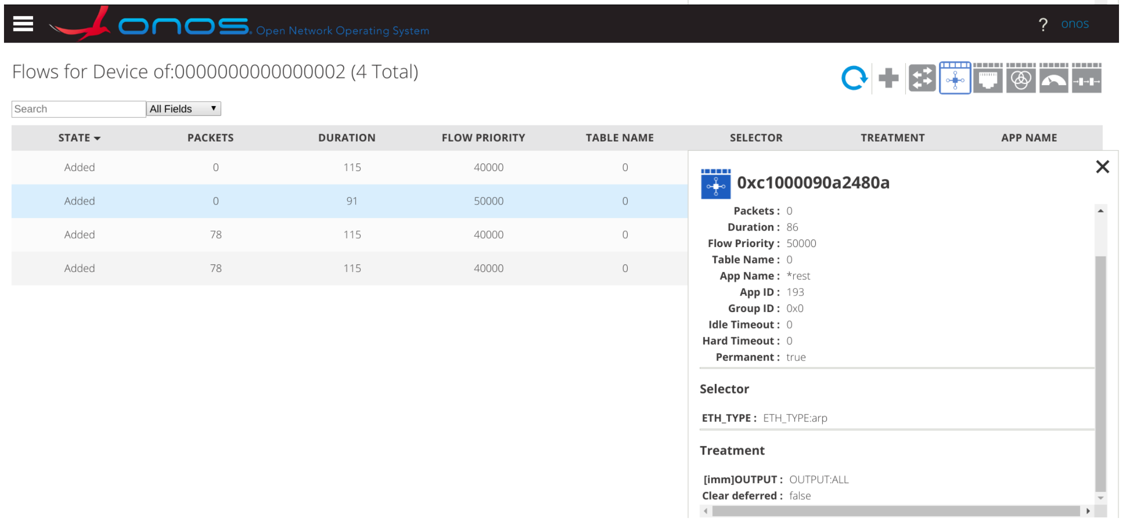
Packets :	0
Duration :	124
Flow Priority :	50000
Table Name :	0
App Name :	*rest
App ID :	193
Group ID :	0x0
Idle Timeout :	0
Hard Timeout :	0
Permanent :	true

Selector

ETH_TYPE : ETH_TYPE:arp

Treatment

[imm]OUTPUT : OUTPUT:ALL
Clear deferred : false



Flows for Device of:0000000000000002 (4 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	115	40000	0			
Added	0	91	50000	0			
Added	78	115	40000	0			
Added	78	115	40000	0			

0xc1000090a2480a

Packets :	0
Duration :	86
Flow Priority :	50000
Table Name :	0
App Name :	*rest
App ID :	193
Group ID :	0x0
Idle Timeout :	0
Hard Timeout :	0
Permanent :	true

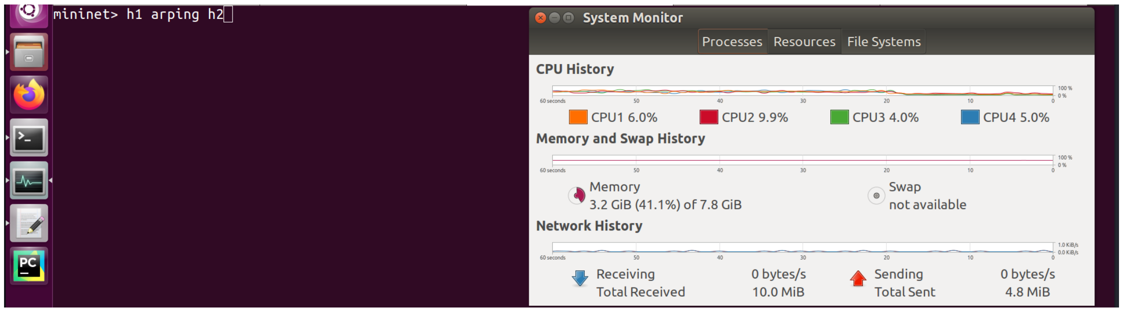
Selector

ETH_TYPE : ETH_TYPE:arp

Treatment

[imm]OUTPUT : OUTPUT:ALL
Clear deferred : false

- 由 h1 發送 arping 封包給 h2 前後之節圖可看出 CPU 的用量有明顯的上升。



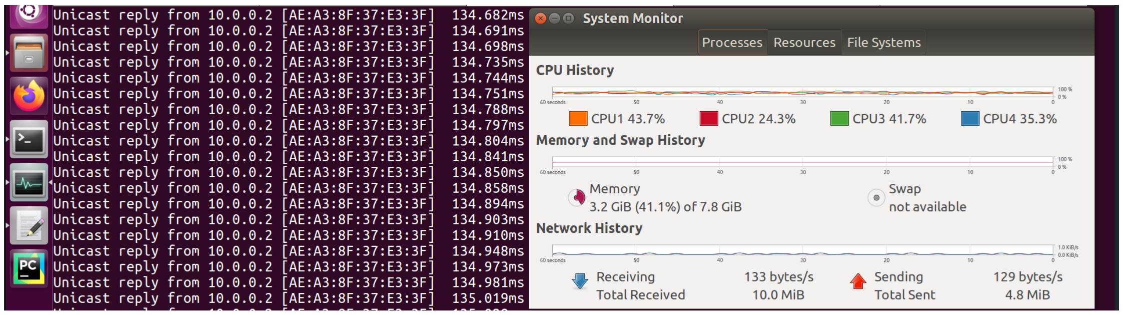
```
mininet> h1 arping h2
```

System Monitor

CPU History: CPU1 6.0% CPU2 9.9% CPU3 4.0% CPU4 5.0%

Memory and Swap History: Memory 3.2 GiB (41.1%) of 7.8 GiB; Swap not available

Network History: Receiving 0 bytes/s, Total Received 10.0 MiB; Sending 0 bytes/s, Total Sent 4.8 MiB



```
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.682ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.691ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.698ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.735ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.744ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.751ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.788ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.797ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.804ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.841ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.850ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.858ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.894ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.903ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.910ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.948ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.973ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  134.981ms
Unicast reply from 10.0.0.2 [AE:A3:8F:37:E3:3F]  135.019ms
```

System Monitor

CPU History: CPU1 43.7% CPU2 24.3% CPU3 41.7% CPU4 35.3%

Memory and Swap History: Memory 3.2 GiB (41.1%) of 7.8 GiB; Swap not available

Network History: Receiving 133 bytes/s, Total Received 10.0 MiB; Sending 129 bytes/s, Total Sent 4.8 MiB

- 當兩個交換器（a, b）之間存在不只一條路徑時，會在網路當中形成一個圓環（loop）。當廣播封包通過 a 交換機，然後送到 b 交換機時，b 會透過圓環這條路徑又將廣播封包傳送回 a。且 layer 2 封包當中沒有 TTL（Time To Live）header，因此這個循環是永久性的。當這

樣的廣播封包量很大時，便會拖垮整個網路的效能，甚至導致網路中斷。