

SDN Experiment 3 Report

SDN Experiment 3 Report

How to redirect all UDP packets to leftmost UDP server

Successful redirection result screenshot

How to redirect all UDP packets to leftmost UDP server

- 在原本的 `mac_to_port` 外，另外使用兩個 dictionary

- `ip_to_port` 紀錄 host 與 switch 的相連的 port
- `ip_to_mac` 紀錄 host 的 MAC address

```
21     def __init__(self, *args, **kwargs):
22         super(SimpleSwitch13, self).__init__(*args, **kwargs)
23         self.mac_to_port = {}
24         self.ip_to_port = {}
25         self.ip_to_mac = {}
```

- 延續實驗一的實作 pingfull 部份，在 `packet_in_handler` 中
 - 利用 ping 時候產生的 ARP 封包完成前面兩個 dictionary 的紀錄：利用 ARP reply packet 可以知道 host (IP address) 會對應的 switch port 以及 switch 的 MAC address)

```
91         if ARP:
92             if ARP.opcode == arp.ARP_REPLY:
93                 ip_src = ARP.src_ip
94                 ip_dst = ARP.dst_ip
95                 # learn where IP address should use which port
96                 self.ip_to_port[dpid][ip_src] = in_port
97                 self.ip_to_mac[ip_dst] = ARP.dst_mac
```

- 下 Flow entries 到對應的 switch
 - 使用 `udp_match` 偵測 `ip_proto` 是否顯示為 UDP 封包，來抓到所有 UDP 封包可以，針對這些 UDP 封包，要透過 `redirect_actions` 將 Ethernet 和 IP 的目的地位址 (`eth_dst` & `ipv4_dst`) 改成 leftmost UDP server 的 MAC address 和 IP 位址 "10.0.0.1"，並把 priority 設大於 1（我的程式碼設置為 50）才能將封包都導到 leftmost server

```
116     if out_port != ofproto.OFPP_FLOOD:
117         match = parser.OFPMatch(in_port=in_port, eth_dst=dst)
118         self.add_flow(datapath, 1, match, actions)
119         udp_match = parser.OFPMatch(eth_type=ether.ETH_TYPE_IP, ip_proto=inet.IPPROTO_UDP)
120
121         if '10.0.0.1' in self.ip_to_port[dpid]:
122             redirect_out_port = self.ip_to_port[dpid]['10.0.0.1']
123             redirect_mac = self.ip_to_mac['10.0.0.1']
124             redirect_actions = [parser.OFPActionSetField(eth_dst=redirect_mac),
125                                parser.OFPActionSetField(ipv4_dst='10.0.0.1'),
126                                parser.OFPActionOutput(redirect_out_port)]
127             self.add_flow(datapath, 50, udp_match, redirect_actions)
```

- 當 UDP server 收到封包後要回覆給 UDP client 時，也會 match 到上面所下的 match rule，於是透過在 match 中增加另外一個欄位 `ipv4_src='10.0.0.1'`，表示若 UDP 封包是由 leftmost UDP server (位址 10.0.0.1) 送出，則不像上面改 Ethernet 和 IP 的目的地位址，而是指定 switch 讓封包往對應的 port 送；priority 設定則需要比上面的 match entry 還大（我設置了 100），從 10.0.0.1 送出的 UDP 封包才能在 match 到兩個 entries 時，採用此 entry 中的 action

```
111 send_back_match = parser.OFPMatch(eth_type=ether.ETH_TYPE_IP, ip_proto=inet.IPPROTO_UDP, ipv4_src='10.0.0.1')
112 send_back_actions = [parser.OFPACTIONOutput(out_port)]
113 self.add_flow(datapath, 100, send_back_match, send_back_actions)
```

如此一來就能成功 redirect UDP packets to leftmost server !

Successful redirection result screenshot

Leftmost UDP server h01 成功收到 h02 原本要送往 h05 (IP=10.0.0.5)的 UDP packet

```
"Node: h02" (在 ubuntu)
root@ubuntu:~# ./client 10.0.0.5 test.txt
test_file
|00

"Node: h01" (在 ubuntu)
root@ubuntu:~# ./server
Server is ready to receive !!
Can strike Cntrl-c to stop Server >>
Received data from 10.0.0.2 : 54289
test_file

Can Strike Crttl-c to stop Server >>
█
```