

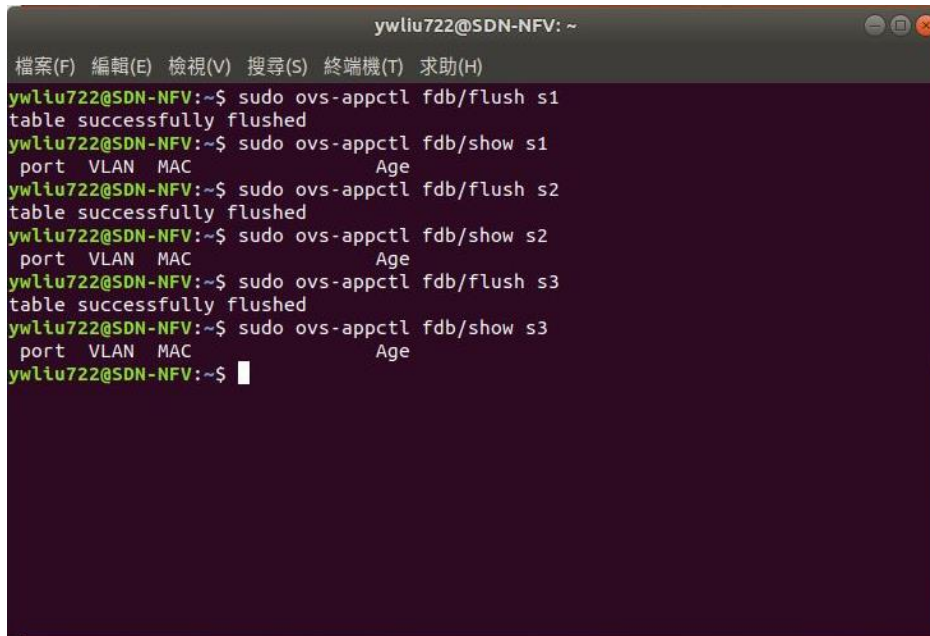
網路系統總整與實作 Lab #1

Layer 2 Forwarding and MAC Learning

0716236 劉耀文

Part 1 : A Tree Topology

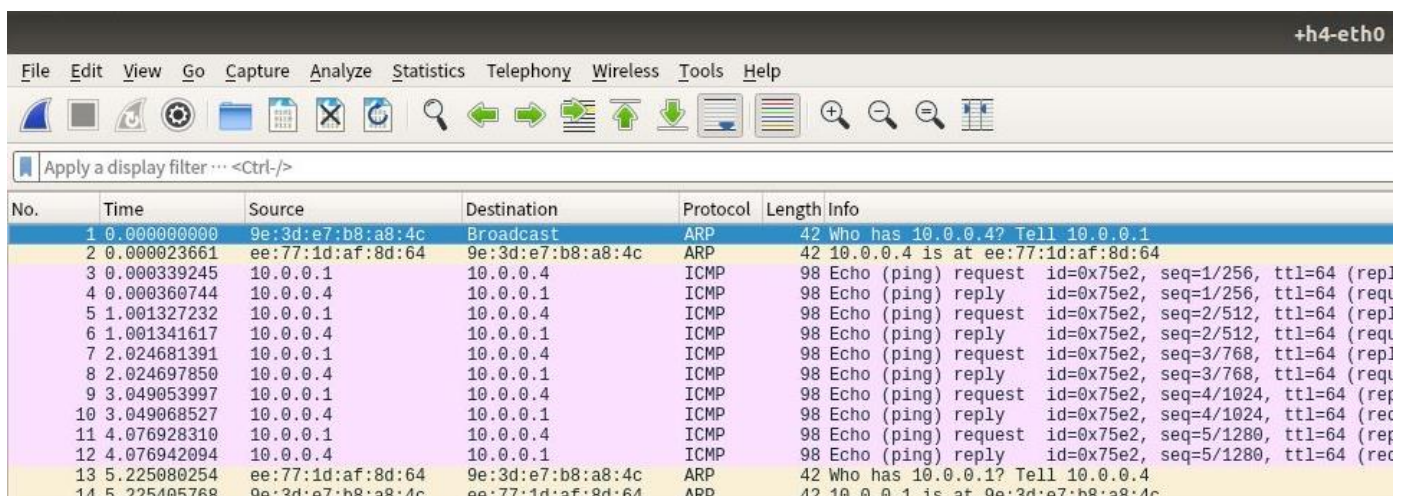
1. Flush all switch tables and take screenshots to show the switch tables of all switches.



```
ywliu722@SDN-NFV: ~  
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/flush s1  
table successfully flushed  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s1  
port VLAN MAC Age  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/flush s2  
table successfully flushed  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s2  
port VLAN MAC Age  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/flush s3  
table successfully flushed  
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s3  
port VLAN MAC Age  
ywliu722@SDN-NFV:~$
```

➔ 在執行完 flush 後，可以看到每一個 switch 的 switch table 都是空的，已進行接下來的步驟。

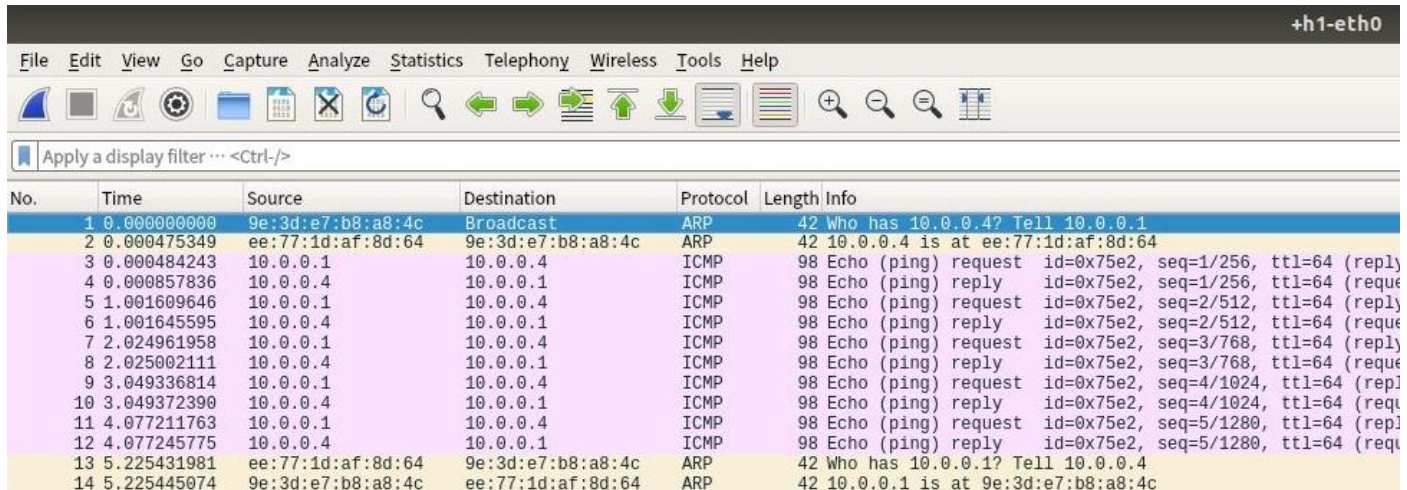
2. How does h4 knows h1's MAC address? Take screenshot on Wireshark to verify your answers.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	9e:3d:e7:b8:a8:4c	Broadcast	ARP	42	Who has 10.0.0.4? Tell 10.0.0.1
2	0.000023661	ee:77:1d:af:8d:64	9e:3d:e7:b8:a8:4c	ARP	42	10.0.0.4 is at ee:77:1d:af:8d:64
3	0.000339245	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=1/256, ttl=64 (repl
4	0.000360744	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=1/256, ttl=64 (requ
5	1.001327232	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=2/512, ttl=64 (repl
6	1.001341617	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=2/512, ttl=64 (requ
7	2.024681391	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=3/768, ttl=64 (repl
8	2.024697850	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=3/768, ttl=64 (requ
9	3.049053997	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=4/1024, ttl=64 (repl
10	3.049068527	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=4/1024, ttl=64 (rec
11	4.076928310	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=5/1280, ttl=64 (repl
12	4.076942094	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=5/1280, ttl=64 (rec
13	5.225080254	ee:77:1d:af:8d:64	9e:3d:e7:b8:a8:4c	ARP	42	Who has 10.0.0.1? Tell 10.0.0.4
14	5.225405768	9e:3d:e7:b8:a8:4c	ee:77:1d:af:8d:64	ARP	42	10.0.0.1 is at 9e:3d:e7:b8:a8:4c

➔ 此圖為 h4 上 wireshark 所監聽得到之封包。h1 發出 ping 指令的 ICMP 封包之前，會先廣播一個 ARP 封包來得知 h4 的 MAC address 來進行 ping 指令的 ICMP 封包交換，h4 便是以接收到的 ARP 封包(封包 no.1)得知 h1 的 MAC address，並透過 unicast 的方式回傳一個 ARP 封包給 h1。

3. How does h1 know h4's MAC address? Take screenshot on Wireshark to verify your answers.

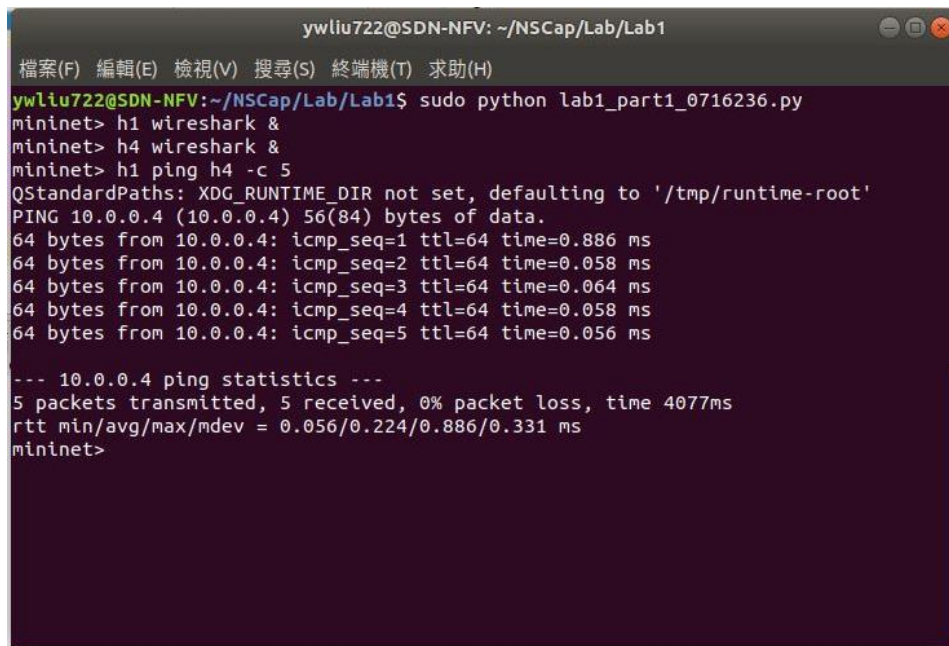


The screenshot shows a Wireshark capture on interface h1-eth0. The packet list contains 14 packets. Packets 1 and 2 are ARP requests and replies. Packets 3 through 13 are ICMP Echo (ping) requests and replies. Packet 14 is an ARP request. The packet details pane shows the structure of the selected packet (No. 14).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	9e:3d:e7:b8:a8:4c	Broadcast	ARP	42	Who has 10.0.0.4? Tell 10.0.0.1
2	0.000475349	ee:77:1d:af:8d:64	9e:3d:e7:b8:a8:4c	ARP	42	10.0.0.4 is at ee:77:1d:af:8d:64
3	0.000484243	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=1/256, ttl=64 (reply)
4	0.000857836	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=1/256, ttl=64 (request)
5	1.001609646	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=2/512, ttl=64 (reply)
6	1.001645595	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=2/512, ttl=64 (request)
7	2.024961958	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=3/768, ttl=64 (reply)
8	2.025002111	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=3/768, ttl=64 (request)
9	3.049336814	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=4/1024, ttl=64 (reply)
10	3.049372390	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=4/1024, ttl=64 (request)
11	4.077211763	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x75e2, seq=5/1280, ttl=64 (reply)
12	4.077245775	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x75e2, seq=5/1280, ttl=64 (request)
13	5.225431981	ee:77:1d:af:8d:64	9e:3d:e7:b8:a8:4c	ARP	42	Who has 10.0.0.1? Tell 10.0.0.4
14	5.225445074	9e:3d:e7:b8:a8:4c	ee:77:1d:af:8d:64	ARP	42	10.0.0.1 is at 9e:3d:e7:b8:a8:4c

→ 此圖為 h1 上 wireshark 所監聽得到之封包。在廣播 ARP 封包被 h4 接受後，h4 會用 unicast 的方式回傳一個 ARP reply 封包(封包 no.2)給 h1，此時 h1 便可以此更新 ARP table。

4. Why does the first ping have a longer delay?



```
ywliu722@SDN-NFV: ~/NSCap/Lab/Lab1
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo python lab1_part1_0716236.py
mininet> h1 wireshark &
mininet> h4 wireshark &
mininet> h1 ping h4 -c 5
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.886 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.056 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4077ms
rtt min/avg/max/mdev = 0.056/0.224/0.886/0.331 ms
mininet>
```

→ 因為事先把 flush 每一個 switch 的 switch table，所以第一個 ping 所發出的 ICMP 封包無法得知目標 host 的 MAC address，故需要先廣播一個 ARP 封包，每一個 switch 接到此封包便會繼續廣播下去，直到找到目標並在得到回傳之 ARP 封包後更新 switch table，因此第一次的 ping 指令會比接下來 4 次要花上更多時間。

5. Show the switch tables and identify the entries that constitute the path of Ping.

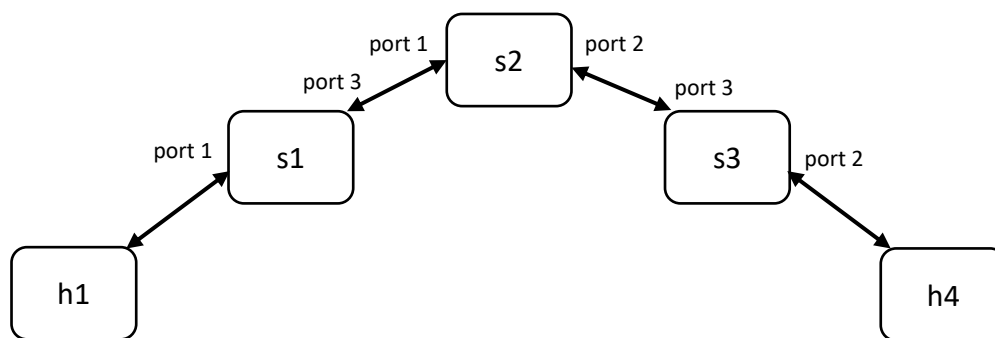
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	9e:3d:e7:b8:a8:4c	Broadcast	ARP	42	Who has 10.0.0.4? Tell 10.0.0.1
2	0.000023661	ee:77:1d:af:8d:64	9e:3d:e7:b8:a8:4c	ARP	42	10.0.0.4 is at ee:77:1d:af:8d:64

```

ywliu722@SDN-NFV: ~
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s1
port VLAN MAC Age
3 0 a2:93:95:57:1f:6a 23
3 0 ee:77:1d:af:8d:64 8
1 0 9e:3d:e7:b8:a8:4c 8
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s2
port VLAN MAC Age
1 0 9e:3d:e7:b8:a8:4c 10
2 0 ee:77:1d:af:8d:64 10
ywliu722@SDN-NFV:~$ sudo ovs-appctl fdb/show s3
port VLAN MAC Age
3 0 9e:3d:e7:b8:a8:4c 12
2 0 ee:77:1d:af:8d:64 11
ywliu722@SDN-NFV:~$

```

→ 由上圖之封包 1 以及封包 2 可以得知 h1 以及 h4 之 MAC address 分別爲 “9e:3d:e7:b8:a8:4c”以及“ee:77:1d:af:8d:64”，再結合 3 個 switch 的 switch table 可以得到下圖之關係。



- s1 分別透過 port 1 以及 port 3 和 h1 以及 s2 連接
- s2 分別透過 port 1 以及 port 2 和 s1 以及 s3 連接
- s3 分別透過 port 3 以及 port 2 和 s2 以及 h4 連接

Part 2 : A Leaf-Spine Topology

1. Can h1 ping h4 successfully before enabling STP?

```
ywliu722@SDN-NFV: ~/NSCap/Lab/Lab1
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo python lab1_part2_0716236.py
[sudo] password for ywliu722:
mininet> h1 wireshark &
[1] 2705
mininet> h1 ping h4 -c 5
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4066ms
pipe 4
mininet>
```

→ 因為 4 個 switch 會形成一個 cycle 且 ARP 封包會一直廣播給所有 port 所連接的裝置，故會無法正確送達 ARP 封包進而導致無法順利更新 switch table，故無法建立 h1 到 h4 之間的路徑。

2. Can h1 ping h4 successfully after STP enabled?

```
ywliu722@SDN-NFV: ~/NSCap/Lab/Lab1
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
mininet> h1 ping h4 -c 5
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4066ms
pipe 4
mininet> h1 ping h4 -c 5
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.529 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.047 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.044 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.041/0.141/0.529/0.194 ms
mininet>
```

→ 透過開啓 4 個 switch 的 STP 協議，使得 4 個互相連接的 switch 可以事先溝通那些 port 該發送或接收哪些資料，哪些 port 不需要做，來避免 cycle 的形成，進而避免無止境的 ARP 封包廣播，最後便可以順利從 h1 找到一條到 h4 的路徑。

3. Show s1 MAC tables before and after enables STP and explain the differences.

```
ywliu722@SDN-NFV: ~/NSCap/Lab/Lab1
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s1
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s2
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s3
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s4
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/show s1
port  VLAN  MAC                               Age
3      0      26:ca:04:22:3b:aa                1
4      0      62:3a:16:0c:81:59                1
3      0      ee:a8:da:b2:c0:92                1
3      0      96:70:2e:f0:01:79                1
3      0      be:fc:7f:83:f5:65                1
4      0      72:4e:fa:2a:fa:0b                1
4      0      f6:b2:d8:e4:20:60                1
4      0      1e:99:3b:1e:b0:cd                1
3      0      6e:47:3f:59:ff:2b                1
3      0      86:47:2f:26:e4:2c                1
4      0      4e:d6:bc:aa:97:1f                1
3      0      92:b1:5f:18:22:d1                1
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$
```

→ 此圖為開啓 STP 協議前 s1 之 switch table，因為路徑形成 cycle 導致產生無止境的 ARP 封包廣播，所以不同的 port 會有重複的 MAC address 出現，使得到達一個裝置會有兩條路徑。

```
ywliu722@SDN-NFV: ~/NSCap/Lab/Lab1
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s3
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/flush s4
table successfully flushed
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/show s1
port  VLAN  MAC                               Age
3      0      26:ca:04:22:3b:aa                1
4      0      62:3a:16:0c:81:59                1
3      0      ee:a8:da:b2:c0:92                1
3      0      96:70:2e:f0:01:79                1
3      0      be:fc:7f:83:f5:65                1
4      0      72:4e:fa:2a:fa:0b                1
4      0      f6:b2:d8:e4:20:60                1
4      0      1e:99:3b:1e:b0:cd                1
3      0      6e:47:3f:59:ff:2b                1
3      0      86:47:2f:26:e4:2c                1
4      0      4e:d6:bc:aa:97:1f                1
3      0      92:b1:5f:18:22:d1                1
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$ sudo ovs-appctl fdb/show s1
port  VLAN  MAC                               Age
1      0      4e:d6:bc:aa:97:1f                2
2      0      92:b1:5f:18:22:d1                2
3      0      86:47:2f:26:e4:2c                2
ywliu722@SDN-NFV:~/NSCap/Lab/Lab1$
```

→ 此圖為開啓 STP 協議後 s1 之 switch table，正如上個小題所述，開啓 4 個 switch 的 STP 協議，使得 4 個互相連接的 switch 可以事先溝通，避免路徑上產生 cycle 讓兩裝置間能夠有唯一路徑。

4. What have you observed and learned from this lab?

→ Observed: 不是把線接起來網路就能夠順利傳輸資料

→ Learned: 簡易模擬網路架設、STP 協議、基本 mininet 操作、L2 封包傳輸