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

```
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/flush s1
table successfully flushed
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/flush s2
table successfully flushed
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/flush s3
table successfully flushed
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s2
 port  VLAN  MAC                Age
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s3
 port  VLAN  MAC                Age
iris@SDN-NFV:~/mininet$
```

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

```
8e:49:2e:8f:16:3c   Broadcast           ARP      42 Who has 10.0.0.4? Tell 10.0.0.1
22:a6:21:9d:0a:2a   8e:49:2e:8f:16:3c   ARP      42 10.0.0.4 is at 22:a6:21:9d:0a:2a
10.0.0.1            10.0.0.4            ICMP     98 Echo (ping) request  id=0x225a,
10.0.0.4            10.0.0.1            ICMP     98 Echo (ping) reply    id=0x225a,
```

如第一條所示，h1 broadcast ARP request，h4 收到後會將發送端 h1 的 IP-MAC address 對應加入 ARP table.

3. How does h1 knows h4's MAC address? Take screenshot on Wireshar.

```
8e:49:2e:8f:16:3c   Broadcast           ARP      42 Who has 10.0.0.4? Tell 10.0.0.1
22:a6:21:9d:0a:2a   8e:49:2e:8f:16:3c   ARP      42 10.0.0.4 is at 22:a6:21:9d:0a:2a
10.0.0.1            10.0.0.4            ICMP     98 Echo (ping) request  id=0x225a,
10.0.0.4            10.0.0.1            ICMP     98 Echo (ping) reply    id=0x225a,
```

如第二條所示，h4 收到 h1 的 ARP broadcast 後，unicast ARP reply 告知發送端 h1 自己的 MAC address。

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

首次 ping 時，發送端的 ARP table 沒有目的端的 IP-MAC address 對應，需要先 broadcast ARP Request 查詢目的端的 MAC address，所以比較慢。

下次 ping 時，發送端的 ARP table 有目的端的 IP-MAC address 對應，可以直接將目的端的 MAC-address 填入 L2 header，所以比較快。

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

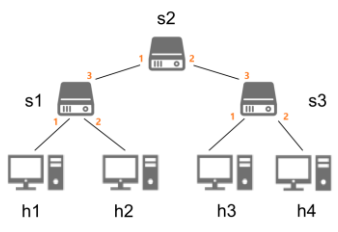
註 ： 此圖為 flush 後，h1 二度 ping h4 留下的 switch tables

左圖橘字標示 port, 右圖綠字標示 switch table 的 mac address 是誰

h1 發出封包→(S1.eth1 進 S1.eth3 出)

→(S2.eth1 進 S2.eth2 出)

→(S3.eth3 進 S3.eth2 出)→h4 收到封包(h4 icmp reply h1 為此反向)




```
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    1     0  8e:49:2e:8f:16:3c    9 h1
    3     0  22:a6:21:9d:0a:2a    9 h4
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s2
 port  VLAN  MAC                Age
    2     0  22:a6:21:9d:0a:2a   13 h4
    1     0  8e:49:2e:8f:16:3c   13 h1
iris@SDN-NFV:~/mininet$ sudo ovs-appctl fdb/show s3
 port  VLAN  MAC                Age
    2     0  22:a6:21:9d:0a:2a   14 h4
    3     0  8e:49:2e:8f:16:3c   14 h1
```

1. Can h1 ping h4 successfully before enabling STP?

No.

```
mininet> h1 ping h4 -c 5
15:21:30.854      Main Warn 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.

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4118ms
```

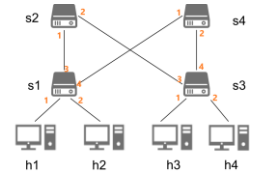
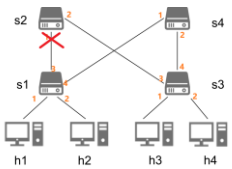
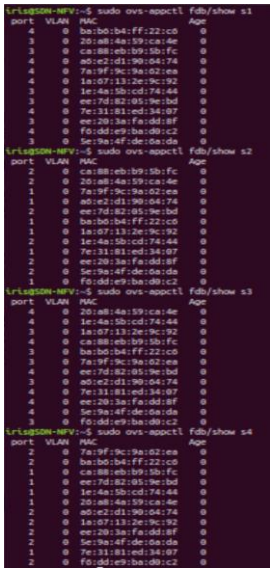
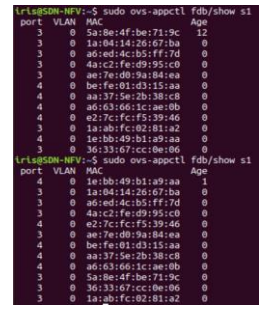
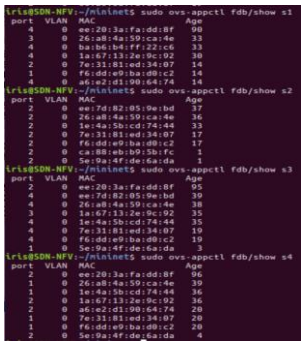
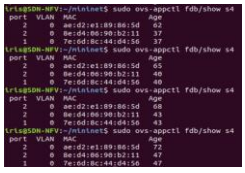
2. Can h1 ping h4 successfully after STP enabled?

Yes.

```
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=0.400 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.040 ms

--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4089ms
rtt min/avg/max/mdev = 0.037/0.111/0.400/0.144 ms
```

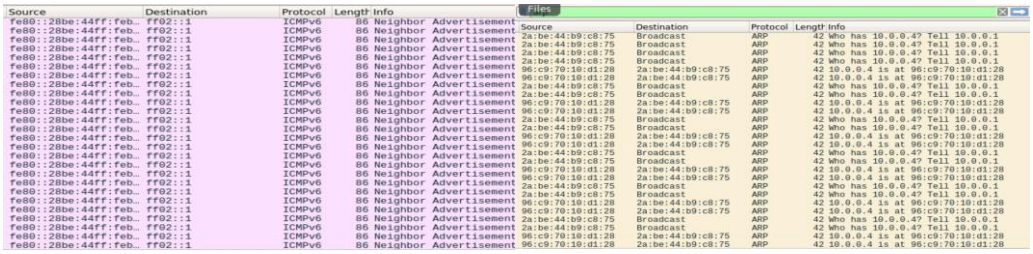
3. Show s1 MAC tables before and after enables STP and explain the differences.

<table>
<tr><td>Before STP</td><td>After STP</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td> 
</td><td> 
</td><td> 
</td><td> 
</td></tr>
<tr><td>(圖一)</td><td>(圖二)</td><td>(圖三)</td><td>(圖四)</td></tr>
<tr><td colspan="2">每個 swich 通往其他 swich 的 port 都是通的</td><td colspan="2">S2 的 eth1 被 disabled。
S2 只剩一個 port，所以收到封包後不會再傳回 S1 loop 一次</td></tr>
<tr><td colspan="2">在開啟 STP 前 MAC table 更替頻繁，如圖二所示，因為 switch receives the same frames on different ports.</td><td colspan="2">MAC table 較為穩定，如圖四所示</td></tr>
</table>

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

Observed:

(1)使用 STP 前，電腦風扇狂轉有種要爆的感覺，wireshark 裡面充滿 Neighbor Advertisements，用 filter 看 ARP protocol 也能觀察到 ARP storm.



(2)enable STP 且 ping 過後，switch table disable 的方法會浮動:

有可能 disable S2-eth1，有可能 disable S2-eth2，也有可能 disable S1-eth4……

網路上寫的原因是:

"When a BPDU propagates from the root bridge toward the leaves of the tree, the age field increments each time the BPDU goes though a bridge. Eventually, the bridge discards the BPDU when the age field goes beyond maximum age. (If the root is too far away from some bridges of the network, this issue can occur.)"

Learned:

複習很多忘記的觀念:

(1)ARP table 為 IP-MAC 對應

(2)MAC address table 為 switch port-MAC 對應

(3)broadcast storm 的原因是 network 中有 loop，所以 broadcast 複製而多出來的封包會留在網路中，再繼續廣播出去，使網路中封包越來越多。