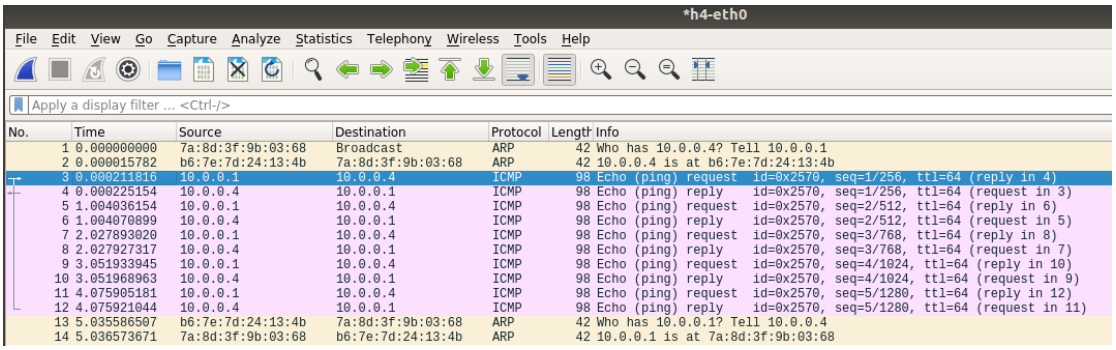
Part1 A Tree Topology

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

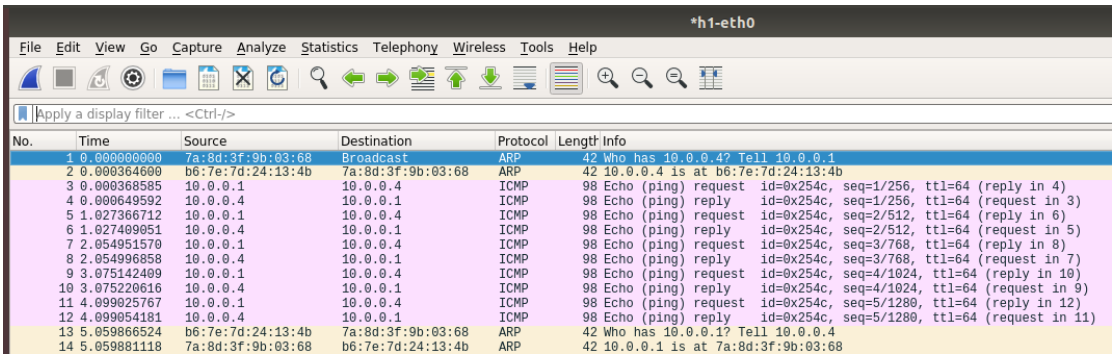
```
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/flush s1 && sudo ovs-appctl fdb/show s1
table successfully flushed
 port  VLAN  MAC                Age
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/flush s2 && sudo ovs-appctl fdb/show s2
table successfully flushed
 port  VLAN  MAC                Age
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/flush s3 && sudo ovs-appctl fdb/show s3
table successfully flushed
 port  VLAN  MAC                Age
```

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



h4 在收到 h1 的 icmp 封包之前會先收到 h1 廣播的 ARP，然後 h4 會檢查 ARP 封包的 IP 欄位是否與自己的一致，如果一致根據 ARP 協定，h4 會將 h1 的 ip 位址和 mac address 更新到自己的 ARP Cache 內，此時 h4 就會知道 h1 的 mac address 了。

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



h1 在發送 icmp 封包前會因為不知道 h4 的 mac address 是多少，而先廣播 ARP 給附近的所有 device 去獲得 h4 的 mac address。No.2 收到的封包就是

h4 收到廣播的 ARP 封包回傳給 h1 的 ARP 封包，也告訴 h1 自己 h4 的 mac address 是多少，所以 h1 就知道 h4 的 mac address 了。

4. **Why does the first ping have a longer delay?**
   我想應該是因為第一次傳送 icmp 封包時，因為不知道對方的 mac address，所以要先傳送 ARP，因此才會花比較多的時間。
5. **Show the switch tables and identify the entries that constitute the path of Ping**

```
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::788d:3fff:fe9b:368  prefixlen 64  scopeid 0x20<link>
        ether 7a:8d:3f:9b:03:68  txqueuelen 1000  (Ethernet)
        RX packets 477  bytes 42478 (42.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 172  bytes 14236 (14.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24  bytes 1368 (1.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1368 (1.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
mininet> h4 ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.4  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::b47e:7dff:fe24:134b  prefixlen 64  scopeid 0x20<link>
        ether b6:7e:7d:24:13:4b  txqueuelen 1000  (Ethernet)
        RX packets 474  bytes 42236 (42.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 172  bytes 14236 (14.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1000 (1000.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1000 (1000.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    3     0  b6:7e:7d:24:13:4b    1
    1     0  7a:8d:3f:9b:03:68    1
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/show s2
 port  VLAN  MAC                Age
    1     0  7a:8d:3f:9b:03:68    7
    2     0  b6:7e:7d:24:13:4b    7
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/show s3
 port  VLAN  MAC                Age
    1     0  7a:8d:3f:9b:03:68    9
    3     0  b6:7e:7d:24:13:4b    9
```

Part2 A Leaf-Spine Topology

1. **Can h1 ping h4 successfully before enabling STP?**
   不行
2. **Can h1 ping h4 successfully after STP enabled?**
   可以
3. **Show s1 MAC tables before and after enables STP and explain the differences.**
   before

```
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/show s1
[sudo] password for wc:
 port  VLAN  MAC                Age
    4     0  76:b7:de:38:b0:49    0
    3     0  a6:11:50:58:e6:4d    0
    4     0  96:3c:48:1d:17:2f    0
    3     0  e2:69:5e:a4:be:7c    0
    3     0  8e:c5:ad:dc:50:86    0
    3     0  d2:83:c6:42:39:f5    0
    4     0  6a:ad:72:3d:82:eb    0
    4     0  d2:83:de:33:9e:8d    0
    4     0  ee:70:02:bd:7a:44    0
    4     0  be:6a:4a:7d:6a:73    0
    4     0  2e:bf:a4:41:46:36    0
    4     0  0e:67:c5:69:f0:3d    0
```

   after

```
wc@wc-VirtualBox:~/Lab1$ sudo ovs-appctl fdb/show s1
 port  VLAN  MAC                Age
    1     0  be:6a:4a:7d:6a:73   55
    4     0  0e:67:c5:69:f0:3d   55
```

   Differences:

   在沒有設置 STP 之前，由於 switch 的接法會產 loop 所以 broadcast 的封包會一直在 loop 裡面繞且不會停止，它不但占滿了整個網路，也因此讓 switch 從四面八方收到 broadcast 的封包，所以 mac table 上的 address 才會對應到不對的 port。所以之後由 h1 送出的 icmp 封包不但

傳不到，也不知道要傳去哪裡。

No.	Time	Source	Destination	Protocol	Length	Info
1729…	9.948396786	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952003322	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948401673	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952006040	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948406061	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952008645	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948410448	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952011378	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948414853	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952014356	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948419885	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952017145	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948424093	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952019770	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948428651	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952022637	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948433140	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.952025299	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b
1729…	9.948437481	fe80::98b4:92ff:fe6…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::98b4:92ff:fe63:ec0b (ovr) is at 9a:b4:92:63:ec:0b

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

學到了 ARP 的運作方式，和電腦獲得不知道的 host 的 mac address 的方法，也複習了之前學過封包在網絡中傳遞的方式。