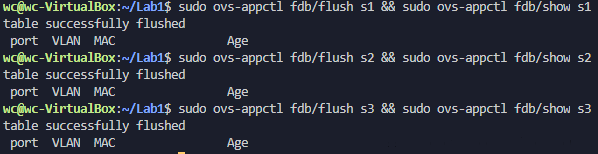
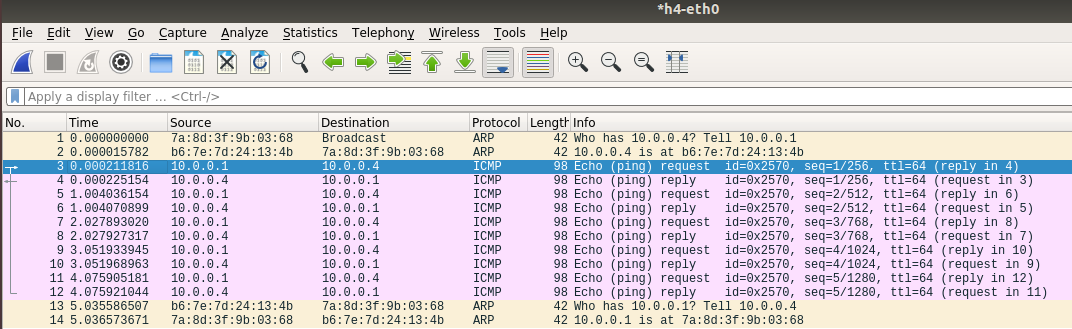
**Part1 A Tree Topology**

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

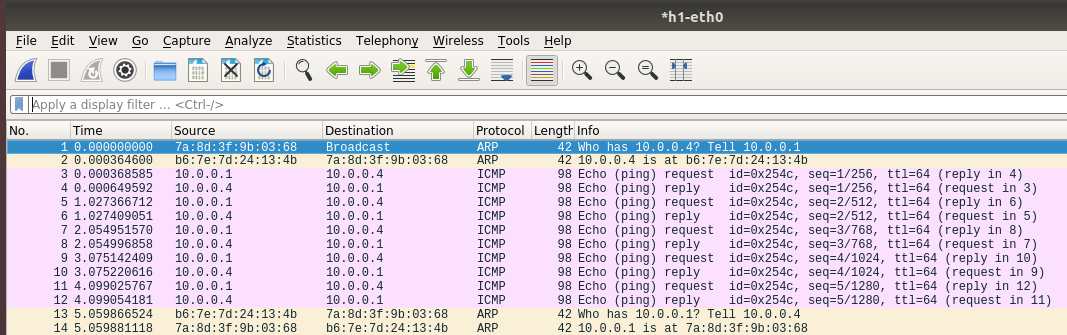


1. **How does h4 knows h1’s MAC address? Take screenshot on Wireshark to verify your answers.**

****

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。

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

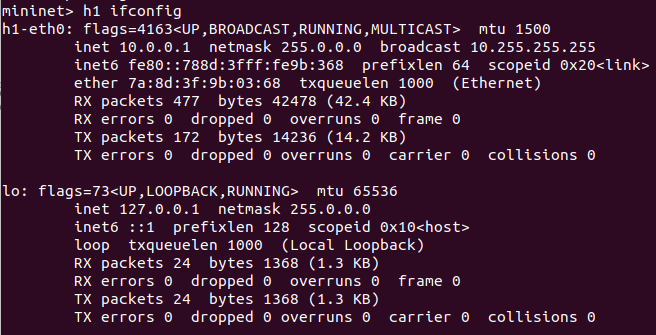
****

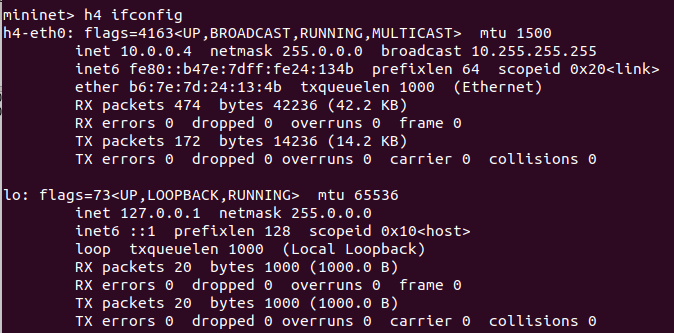
h1在發送icmp封包前會因為不知道h4的mac address是多少，而先廣播ARP給附近的所有device去獲得h4的mac address。No.2收到的封包就是h4收到廣播的ARP封包回傳給h1的ARP封包，也告訴h1自己h4的mac address是多少，所以h1就知道h4的mac address了。

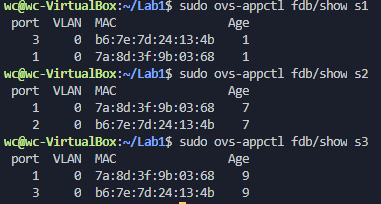
1. **Why does the first ping have a longer delay?**

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。

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

****

****

****

**Part2 A Leaf-Spine Topology**

1. **Can h1 ping h4 successfully before enabling STP?**

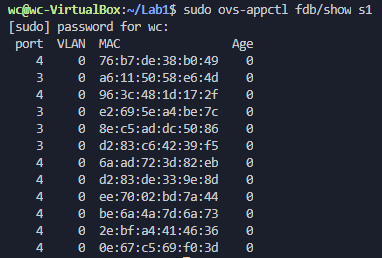
不行

1. **Can h1 ping h4 successfully after STP enabled?**

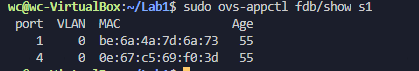
可以

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

before

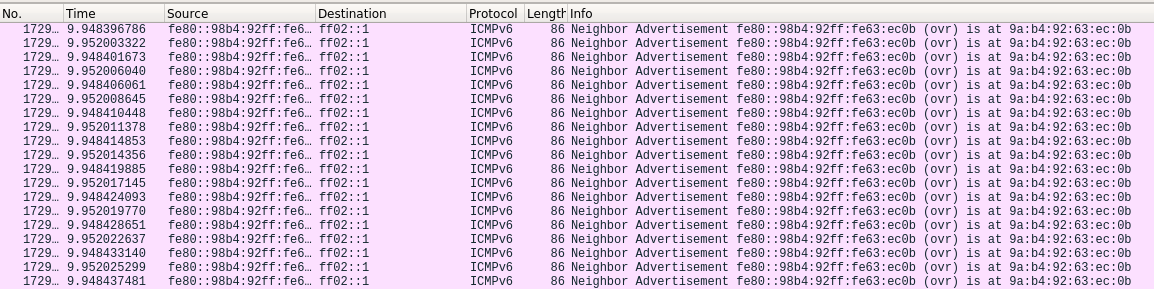


after



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。



1. **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。