

# CNLAB Lab3 Report

B07902002 連崇安 B07902034 王昱凱 B07902126 謝宗儒

## Environment

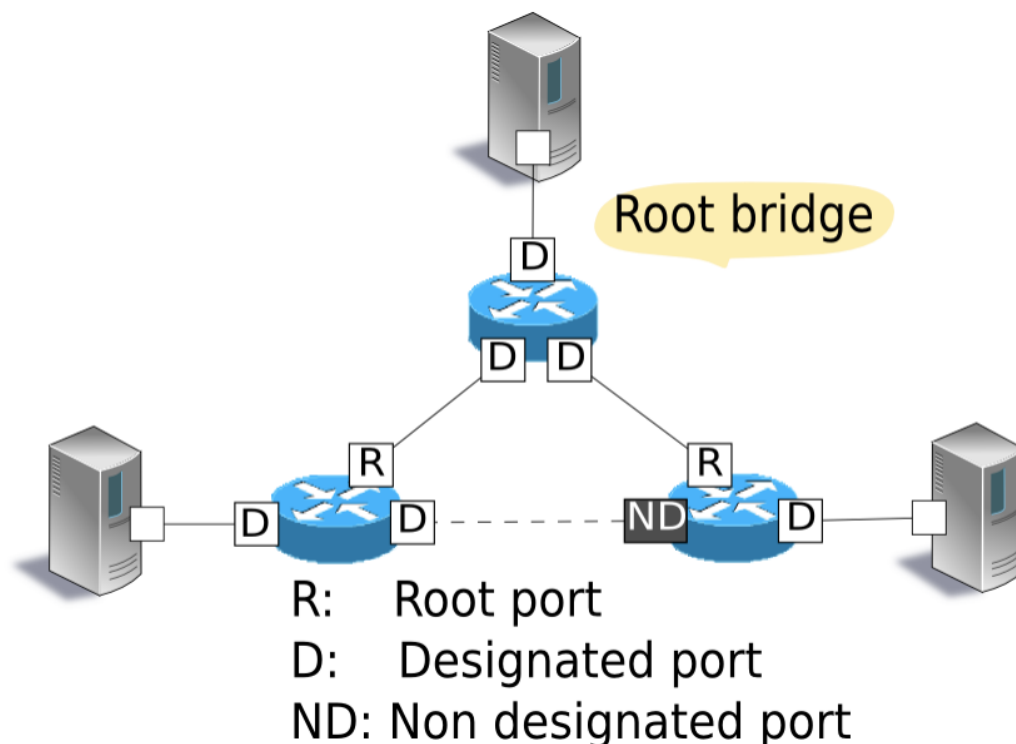
我們測試的環境為 Ubuntu 18.04，使用的 virtual network topology 是用 mininet 創建的，並且使用 Ryu 作為 SDN controller，兩者都是以 python 實作的，而使用的 protocol 為 OpenFlow13

## Explain the pros and cons that there are loops in a network topology

當 network topology 有 loops 的好處在於，loops 中的任兩個 hosts 都必定有大於一條以上的 path，如果當其中一條 path 因為 congestion 或者當中的 link, switch, router 有發生問題時，這兩個 hosts 仍然可以透過另外一條 path 來傳送資料，所以有 loops 存在可以避免當其中一條 path 出問題時導致兩個 hosts 無法互相傳送資料，而壞處在於可能有 broadcast storm 的問題產生，當 topology 有 loops 並且 switch 使用 flood 時就會發生 broadcast storm，switch 會使用 flood 通常是因為不知道封包要送到哪個 port，因此透過 flood 的方式來學習 ARP table，但是在有 loops 的 topology 使用這種學習方法會導致封包在 loop 中不斷被 broadcast 並造成網路癱瘓

## Explain the broadcast storm and how you handle it in this lab. Is there any better solution to handle broadcast storms under SDN? If yes, explain how; if no, explain why

在上一小題中有提到，broadcast storm 產生的原因是 topology 中有 loops 存在，如果 switch 使用 flood 來學習 ARP table 的話會導致封包在 loop 中不斷被 broadcast 並造成網路癱瘓，而我們這次在 data center topology 中解決 broadcast storm 的方式為使用 STP (spanning tree protocol)，而 STP 的處理方式如下圖：



會將 port 分類為 root, designated, non designated，並且將 non designated port 設為 disable 來避免 broadcast storm 發生，但這種解決方法也會導致 topology 沒有 loops 存在，如果 congestion 等問題發生時就無法以 re-route 的方式解決，所以並不是最好的處理方法，由於我們可以使用 ryu 來作為 controller，而 controller 能夠得到整個 topology 的 global view，在有這個資訊的情況下我們就不需要藉由 flood 來學習 ARP table，可以藉由一些 shortest-path algorithm 來決定 path，就能夠不透過 flood 直接將封包送到指定的 port，既能避免 broadcast storm 又能夠 re-route path，是比較好的處理方法