

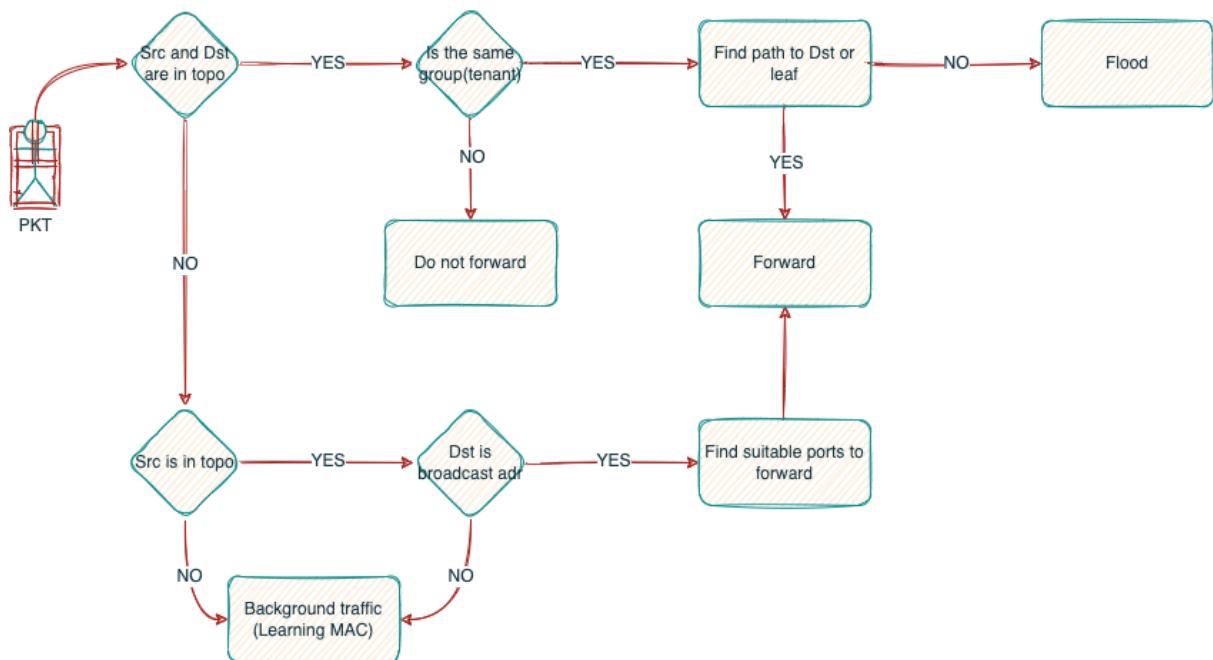
Data Center Network Project 4 Report

310551017

王志嘉

A. Explanation of design implementation

a. Flowchart

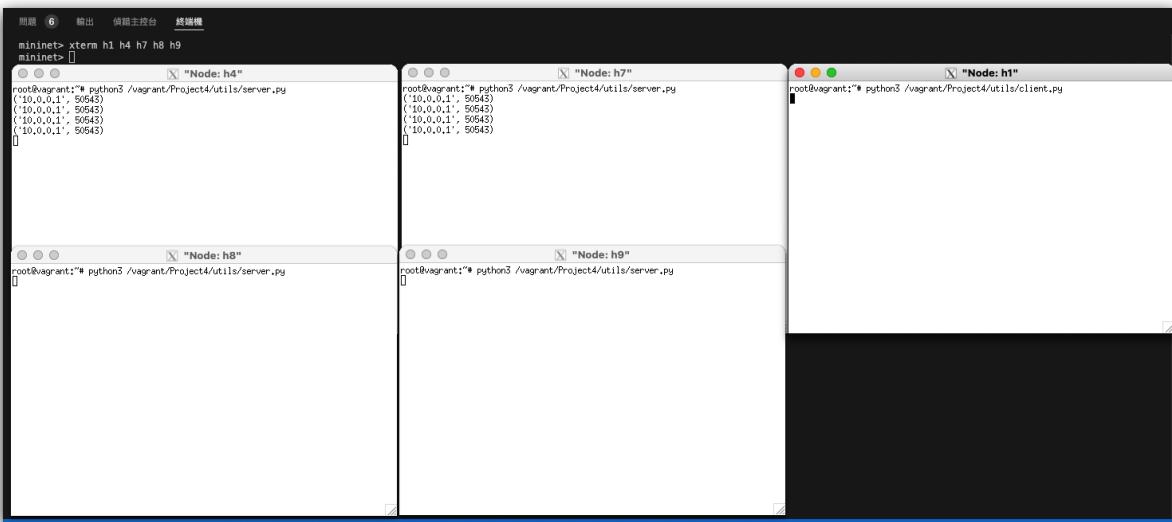


b. Screenshot

i. Pingall

```
vagrant@vagrant:/vagrant/Project4$ sudo mn --controller remote,ip=127.0.0.1 --topo tree,depth=4 --switch default,protocols=OpenFlow13 --mac --arp
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16
*** Adding links:
(s1, s2) (s1, s3) (s2, s3) (s3, s4) (s4, s5) (s5, s6) (s6, s7) (s7, s8) (s8, s9) (s9, s10) (s10, s11) (s11, s12) (s12, s13) (s13, s14) (s14, s15) (s15, s16) (s16, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Starting controller
c0
*** Starting 15 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 --> X X X X X h7 X X h10 X X h13 X X h16
h2 --> X X X X h5 X X h8 X X h11 X X h14 X X
h3 --> X X X X h6 X X h9 X X h12 X X h15 X
h4 --> h1 X X X X h7 X X h10 X X h13 X X h16
h5 --> X X h2 X X X X h8 X X h11 X X h14 X X
h6 --> X X h3 X X X X h9 X X h12 X X h15 X
h7 --> h1 X X h4 X X X X h10 X X h13 X X h16
h8 --> X h2 X X h5 X X X X h11 X X h14 X X
h9 --> X X h3 X X h6 X X X X h12 X X h15 X
h10 --> h1 X X h4 X X h7 X X X X h13 X X h16
h11 --> X h2 X X h5 X X h8 X X X X h14 X X
h12 --> X X h3 X X h6 X X h9 X X X X h15 X
h13 --> h1 X X h4 X X h7 X X X h10 X X X X h16
h14 --> X h2 X X h5 X X h8 X X h11 X X X X
h15 --> X X h3 X X h6 X X h9 X X h12 X X X X
h16 --> h1 X X h4 X X h7 X X X h10 X X h13 X X
*** Results: 70% dropped (70/240 received)
```

ii. UDP server and client



B. Difficulties Or Bottleneck

我在這個專案中遇到了以下的困難，一個問題是尋找有效的拆分租戶的方法，另一個則是廣播流量的問題。在我開發第一個版本時，我是直接在程式裡寫死所有要拆分的 MAC 地址以拆分各個租戶，儘管它是能動的，但還是很醜而且很難改動。所以我又重構了第二個版本，本來想使用 etcd 或是 hashicorp consul 來全自動地做出服務(裝置)發現的效果，但由於我還沒有想到一個很好的 reload 方法，所以我改成用簡單的 json 設定檔將裝置與租戶給靜態的關聯起來，以簡單地達成本專案的目的。而廣播流量的問題則是我的 controller 無法學習正確的 MAC 地址，且無法通過 UDP 測試，所以我添加了新的邏輯來處理 controller 所不知道的租戶廣播流量。

C. Advantage and disadvantage

a. Advantage

由於我們的網路拓樸是相當簡單的，所以我們可以使用簡單和少量的 json 設定檔去組織它(只需要租戶組和 leaf-to-host 即可)，即可完成本次專案要求。

b. Disadvantage

由於目前的設計是靜態性的一次導入所有的租戶設定，所以每當有新的租戶或是要修改時，得要將 controller 重啟並重新學習 MAC 地址，因此會影響到現有網路，且有些不切實際。未來展望是改成動態的去 reload 才是最實際的作法。