Orchestrator使用指南

1. 我們的orchestrator主要的目的與功能為：
2. 建置一個虛擬網路，展示我們的演算法
3. 利用NAPA API將網路圖像化，變得更易讀
4. 利用NAPA API控制flow table
5. 將以上三點利用網頁前後端來操控，並讓使用者能夠用GUI調整演算法中的參數
6. 操作方法：
   1. GUI的操作已經在ppt以及demo中顯示，這裡就不再詳細介紹
   2. 首先，在linux系統中執行命令 sudo python3 flask\_server.py，會在port 5000開啟orchestrator的後端
   3. 接著訪問前端所在的網址，即可看到GUI介面，我們之前是使用ngrok來產生網址
7. 各個檔案的功能
   1. csv file & txt file
      * capacity.csv是各種不同的node的capacity
      * comm.csv是communication link的config
      * cost.csv 是各種不同node的cost
      * config.txt是學姊演算法的Input
   2. python file
      * Class.py、LR.py是惟之學姊的code，為了方便我們使用稍微進行了一些更改（針對input進行修改）
      * show\_path.py是讓學姊的output變得易讀且供其他function使用
      * catKnight.py是利用mininet以及NAPA api建置網路節點、產生flow entry的function
      * run\_topo.py是使用catKnight.py的function來建置網路拓樸，可單獨使用也可以從前後端呼叫執行
      * flask\_main.py是學姊的main.py的更動版本，為後端的一部分
      * flask\_server.py是網頁的後端，獲取前端的request並執行上述檔案中的function
   3. output file
      * 前端按下calculate path之後，flask\_main.py會產生link.csv、adjacency.csv兩個檔案，而show\_path.py會在./path中產生result.json檔案
      * 也會同時觸發run\_topo.py，建置網路拓樸與flow entry
   4. web directory & frontend