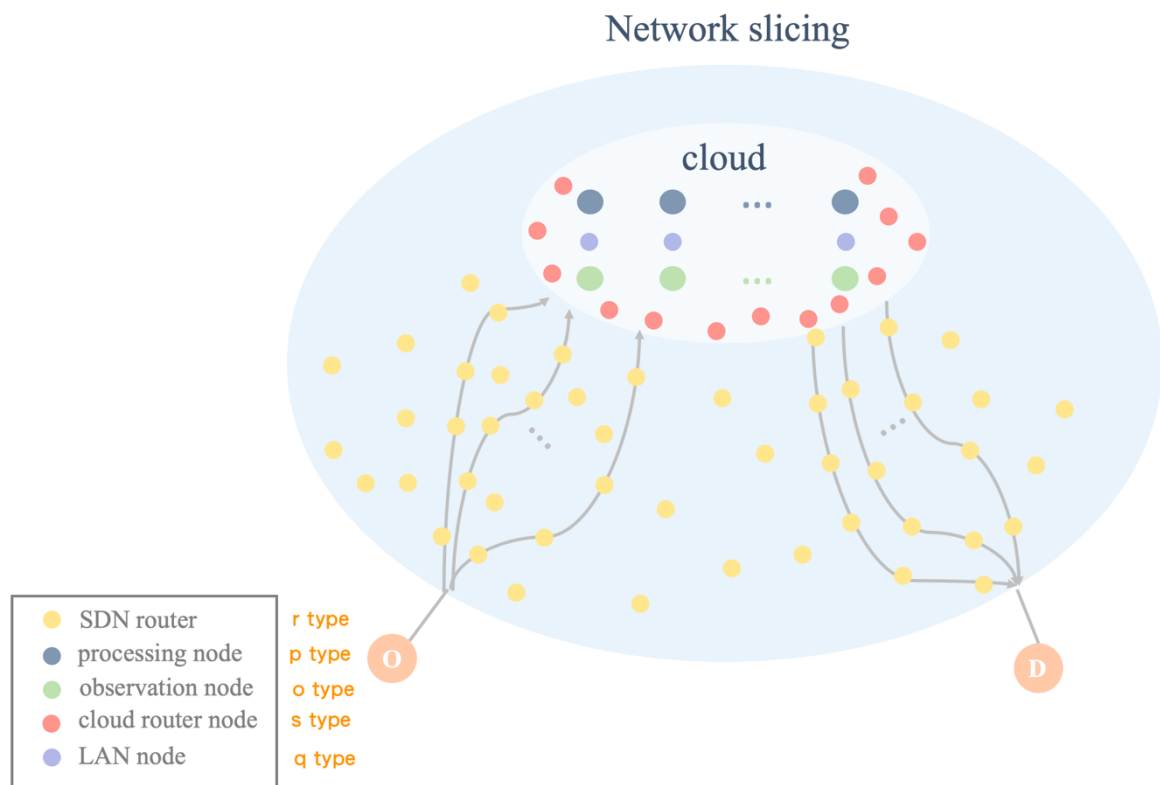


Deployment Model 規格

<https://hackmd.io/@8PJJVUNVTY2iSYumK3Rtpg/ByEJD9byv/edit>



每條路徑型態/順序 : r_type -> s_type -> o_type -> q_type -> p_type -> s_type -> r_type

1. User pair

- 數量
- 每個用戶對所要求的 traffic (unit)
- 每個用戶對所要求之路徑數量

每個用戶對新增一 Flow entry (origin 端傳送到 destination 端)

POST/api/openflow/flowentry/

(Flow entry 中之 in_port 為來源端，actions 中的 OUTPUT port 為目的端)

2. Node

- 數量
- capacity level
- cost

共五種節點類型，包括 SDN router, cloud router, observer, LAN, processor，

每個節點新增一 switch

POST/api/openflow/switch/

節點類型可調整 `node_type` 參數，包含 legacy, host, network, switch :

legacy : 傳統網路設備

host : 終端設備

network : 網路服務

switch : OpenFlow Switch

3. Link

- 數量
- capacity level
- cost

欲建置則新增一 link

POST/api/openflow/link/

設定 `bandwidth` 參數來指定相對應的建置頻寬

Config txt File

config.txt

config.txt 檔為使用者設定 user pair 數量、各 user pair 要求的 traffic、各 user pair 要求的傳輸路徑、五種類型之節點數量，以及節點與鏈結的 capacity level 與相對應之 cost level。以下順序為 config.txt 中依序填入值的相對應變數：

- User pair 數量
- User pair 要求的 traffic (值在 10~60 間，以逗號分格)
- User pair 要求的傳輸路徑數，(以 2, 2, 8, 6, ... 表示，以逗號分隔)

值需在 2 到 $\text{cloud router node (s_type node) 數量} // 2$ 之間

e.g. 若 s_type node 有 17 個，user pair 所指定的傳輸路徑數量就需在 (2, 8) 之間

- 節點數量

| 節點類型 | 設定限制 |
|-----------------------------------|---|
| SDN router node (type r) | 給定數量不能 $< \text{user_pair_num} - 3$ (通常設定 $\text{user_pair_num} - 3$) |
| Observer node (type o) | 給定數量不得少於 user_pair_num 的一半 (通常設定 $\text{user_pair_num} - 5$) |
| Processor node (type p) | 給定數量不得少於 user_pair_num 的一半 (通常設定 $\text{user_pair_num} - 5$) |
| LAN node (type q) | 給定數量不得少於 user_pair_num 的一半 (通常設定 $\text{user_pair_num} - 5$) |
| Cloud SDN router node (type s) | 給定數量不能 $< \text{user_pair_num} - 3$ (可設定與 SDN router node 一樣) |

節點與鏈結的 `capacity` 與 `cost level` 更動需在 `Class.py` 這份 `python` 檔中，`Node()` 這個 `class` 的 `assign_attribute` function 中去更動，裡面的 `type` 包括 `r, o, p, q, s`，分別對應五種類型的節點。鏈結的 `capacity` 與 `cost` 更動需在 `Link()` 這個 `class` 中的 `generate_capacity_cost()` 中去做更動。

需要注意的是，當 `user pair` 數量在 40(含)以上時，需將 `LR.py` 中第 73 行的 `init_value` 改為 1。

在我的論文中，節點會做 `node splitting`，因此每種類型的節點程式都以 `link` 型態表示，所以假設有一個 ID 為 1 的 `SDN router (type r)` 節點，在程式中會以 `link (r1->r1)` 的形式表示，這條鏈結的起點與終點都是同一個節點 (`r1`)。

另外，`user pair` 的 `origin` 跟 `destination` 為哪兩個節點，程式目前是以 `random` 形式，若要指定 `origin` 與 `destination` 端的話，需在 `main.py` 檔中加上第 175 與 176 行去指定。

```
170 user_pairs = UserpairSet()
171 for i in range(USERPAIR_NUM):
172     pair = UserPair(i, r_loc, adjacency, comm_links, arti_nodes, s_loc.length())
173     pair.traffic = traffic_list[i]
174     pair.path_number = pathnum_list[i]
175     pair.origin = r_loc.get_link(0)
176     pair.destination = r_loc.get_link(1)
177     user_pairs.add_pair(pair)
```

未來若要變成可供使用者設定的話，可在 `config.txt` 中再加入給使用者輸入每對 `user pair` 的起始端與終端個要是哪兩個節點，而要注意所指定的節點是從 `r type links` 中挑選，也就是程式中的 `r_loc` (可以建置 `r type u` 節點的所有位置)。

output.txt

`output.txt` 這份 `txt` 檔內會顯示出模型的最終建置結果，哪些節點與鏈結被建置，以及各個元件相對應的建置 `capacity` 為哪個 `level`。