

## %% Summary

```
% 네트워크를 구성하고, 링크별로 연결여부(0/1), 딜레이([0,1]사이 실수) 등을
추가함
% 配置網絡，為每條鏈路添加連接狀態（0/1）、延遲（[0,1]之間的實數）等
% 스위치/라우터는 고정간격 격자형으로 배치되어 있고,
% 交換機/路由器以固定間距排列成網格，MEC 位於重疊
% 격자가 겹치는 부분중 일부에 MEC 가 위치하게 된다. MEC 는 확률적으로
생성함
% 網格的一部分中。 MEC 是概率生成的
```

## %% Init

```
clc; clear; close all;
% clc: 清空指令視窗
% clear: 清空所有東西以釋放記憶體
% close all: 清空所有圖表
```

```
%addpath("../Z-common");
```

```
% 把資料夾加入 search path
% 不然他會存到預設路徑
% e.g. `%USERPROFILE%/Documents/MATLAB` in Windows
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
```

## %% System Param

```
% 생성되는 파일을 저장할 하위 디렉토리
% 存放生成文件的子目錄
```

```
subdir = 'save/';
```

```
% Path for generated files
```

```
% 시나리오 ID 구성 : X_Y_Z_AAAA
```

```
% 場景 ID 配置 : X_Y_Z_AAAA
```

```
% X : N_NODE_PER_ROW/COL 를 의미
```

```
% X : 表示 N_NODE_PER_ROW/COL
```

```

% Y : DELAY_MAX 를 의미(DELAY_MIN=0 으로 고정)
%   Y : 表示 DELAY_MAX (固定為 DELAY_MIN=0)
% Z : MEC 를 생성할 확률값
%   Z : 產生 MEC 的概率
% AAAA ; 임의로 붙인 순서번호
%   AAAA : 隨機分配的序列號

%scenario_id = '4_1.0_0.3_0001';
% ILP/LR subopt 확인용 시나리오
%   檢查 ILP/LR subopts 的場景
%
%scenario_id = '5_1.0_0.3_0001';
% 작은 네트워크 실험용
%   用於小型網絡實驗
%
scenario_id = '10_1.0_0.3_0001';
% Large 네트워크 실험용
%   大型網絡實驗

strings = split(scenario_id, '_');
% Split scenario_id variable into an array of strings

% 네트워크에 총 몇개의 스위치/라우터로 격자를 만들지...
%   在網絡上創建網格總共需要多少個交換機/路由器...
% MEC/스위치/라우터 등등은 격자 형태로 연결되어 있다고 가정
%   假設 MEC/交換機/路由器等以網格模式連接。
% NODE = MEC/스위치/라우터 등 모두를 의미
%   NODE = 表示所有 MEC/交換機/路由器等。
% int32 로 캐스팅 하지 말것! 아래에서 floor 연산에서 오류남
%   不要強制轉換為 int32 ! floor 的計算會出現錯誤

N_NODE_PER_ROW = str2double(strings(1));
% 하나의 행에 몇개의 노드가 배치될지
%   一行放置多少個節點
N_NODE_PER_COL = N_NODE_PER_ROW;
% 하나의 열에... 정사각형 형태를 가정
%   在一列中.....假設一個正方形

```

```

N_NODE = N_NODE_PER_ROW * N_NODE_PER_COL;
% 전체 노드 수
% 節點總數

% 링크 딜레이를 어떻게 설정할지...Uniform distribution 을 가정
% 如何設置鏈接延遲...假設均勻分佈
DELAY_MIN = 0.0;
% 이걸 바꾸지 말자...
% 我們不要改變這個...
% (前面 scenario_id 有提過 MIN 預設為 0)
DELAY_MAX = str2double(strings(2));
% 由 scenario_id 定義最大延遲

% MEC 생성 확률
% MEC 生成概率
% 一樣由 scenario_id 定義
mec_gen_prob = str2double(strings(3));

% 코드를 실행하면 여러개의 파일이 생성되는데, 파일의 이름 기본값 설정
% 運行代碼時，會創建幾個文件。設置默認文件名
out_filename = scenario_id;
% 저장용 파일의 이름의 기본값을 설정
% 設置保存文件名的默認值 (預設為 scenario_id)

% 터미널 출력값을 텍스트 형태로 txt 파일에 저장
% 將文本格式的終端輸出保存到 txt 文件
% "arguments must contain a char vect..." 라는 오류 나오면
% 如果你得到一個錯誤說 "arguments must contain a char vect ..."
% 파일명을 구성하는 모든 텍스트를 ""가 아닌 "로 사용하기
% 使用構成文件名的所有文本作為 " 而不是 ""
% (TL;DR: 用單引號)
diary_filename = strcat(subdir, out_filename, '.txt');

% 輸出檔名: diary_filename
% 如果檔案存在就刪掉重建
if( isfile(diary_filename) )
    delete(diary_filename);
end

```

```

diary(diary_filename);
% 시작!
% 開始!

%%%%%%%%%%
%%%%%%%%%%
%%

%% Part 1. Adjacency Matrix 만들고, matlab 의 graph 로 그림 그리기
%% 創建 Adjacency Matrix 並用 matlab 繪製

% 인접행렬 만들기
% 創建 Adjacency Matrix (初始化為都沒連接)
adj = zeros(N_NODE, N_NODE);
% 딜레이 값이 추가된 인접행렬 만들기: adj(0/1) + link delay
% 創建具有附加延遲值的鄰接矩陣 (初始化為無延遲)
adjDelay = zeros(N_NODE, N_NODE);

% node = 1; node <= N_NODE; node++
for node=1:1:N_NODE
    %fprintf("Node : %d\n", node);
    % 디버깅용
    % 用於除蟲

    % 각 노드를 기준으로 자신의 우측과 하단만 연결하면 되잔아?

    % 1. 우측 노드 연결
    % 右節點連接
    right_node = node + 1;
    % 우측 노드의 번호
    % 右節點數量
    if ( right_node <= N_NODE && rem(node, N_NODE_PER_ROW) > 0 )
        % 인접행렬은 대칭이다.
        % Adjacency Matrix 是對稱的
        adj(node, right_node) = 1;
        adj(right_node, node) = 1;

        % random delay 추가

```

```

    % 添加隨機延遲
    % rand(): 回傳 (0, 1) 間的隨機數
    d = (DELAY_MAX - DELAY_MIN) * rand() + DELAY_MIN;
    adjDelay(node, right_node) = 1 + d;
    adjDelay(right_node, node) = 1 + d;
end

% 2. 아래쪽 노드 연결
% 連接下層節點
lower_node = node + N_NODE_PER_ROW;

% 유효한 노드이면 연결
% 如果有效節點連接
if( lower_node <= N_NODE )
    % 인접 행렬은 대칭이다.
    % Adjacency Matrix 是對稱的
    adj(node, lower_node) = 1;
    adj(lower_node, node) = 1;

    % random delay 추가
    % 添加隨機延遲
    d = (DELAY_MAX-DELAY_MIN) * rand() + DELAY_MIN;
    adjDelay(node, lower_node) = 1 + d;
    adjDelay(lower_node, node) = 1 + d;
end
end

% Network 클래스 객체 생성
% 創建網絡類對象
% [Debug] 沒有 Network: 下面註解掉不然會炸
%netw = Network(adj, adjDelay);

% 그래프 객체 생성
% 創建圖形 Object
% https://www.mathworks.com/help/matlab/ref/graph.html
% [Debug] 沒有 Network: 註解下面兩行取消註解下面三四行不然會炸
%G = netw.getGraph();
%Gdelay = netw.getGraphPlusDelay();

```

```

G = graph(adj);
Gdelay = graph(adjDelay);

% matlab 의 그래프를 이용해서 인접행렬을 그림으로 그리는 방법
% 如何在 matlab 中使用圖形繪製鄰接矩陣
f1 = figure;
% 建立圖形視窗
plot(G, 'Layout', 'auto');
% 在視窗 (f1) 中畫圖
set(gca, 'XTick', []);
% 把 X 軸刻度的字關掉
set(gca, 'YTick', []);
% 把 Y 軸刻度的字關掉
% 그림파일로 저장
% 將圖形另存為圖片
saveas(f1, strcat(subdir, out_filename, "-graph.png"));

% shortest path 계산 및 저장
% 計算並儲存最短路徑
% https://www.mathworks.com/help/matlab/ref/graph.distances.html
%fprintf("Calc shortest path with adj matrix.\n");
%distanceMatrix = distances(G);

% edge 에 weight 을 추가
% 將權重添加到 edge
% 위에서 adj mat 만들때 추가하면 될듯? undirected 니까, 양방향 모두 추가
% 我可以在上面製作 adj mat 添加它嗎? 因為它是無向的, 所以添加兩個方向
% 'positive' 옵션 주면 weighted graph 에서도 최단거리 계산할 수 있음
% 如果給出 'positive' 選項, 即使在加權圖中, 您也可以計算最短距離。
%fprintf("Calc shortest path with adj+delay matrix.\n");
%distanceDelayMatrix = distances(Gdelay, 'Method', 'positive');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%% Part 2. 배경에 격자를 그리고, 원하는 위치에 MEC 를 배치하는 방법
%% 如何在背景上繪製網格並將 MEC 放置在所需位置

```

```
% 격자가 겹치는 모든 곳에는 switch/router 가 있다고 가정
% 假設在網格重疊的任何地方都有一個交換機/路由器
% 서비스를 제공하는 active 한 MEC 만 격자위에 표시
% 網格上只顯示提供服務的活躍 MEC
```

```
horizontal_spacing = 1;
% plot 에서의 수평 간격
% 繪圖中的水平間距
vertical_spacing = 1;
% plot 에서의 수직 간격
% 圖中的垂直間距
%N_NODE_PER_COL = N_NODE_PER_COL;
%N_NODE_PER_ROW = N_NODE_PER_ROW;
margin_gap = 0.5;
% 그림 그릴때 여백 추가용
% 用於繪製時添加空白
```

```
mec_id = [];
total_mec_count = 0;
% i = 1; i <= N_NODE; i++
for i = 1:1:N_NODE
    if ( rand < mec_gen_prob )
        % MEC 를 확률적으로 생성
        % 概率性地產生 MEC
        mec_id(end + 1) = i;
        % 在 mec_id 後面加一個 i
        % Py: mec_id.append(i)
        total_mec_count = total_mec_count + 1;
        % 紀錄 MEC 數量
    end
end
```

```
mec_num = size(mec_id, 2);
% MEC 개수 확인
% 檢查 MEC 的數量
```

```
% 검증
```

```

% 確認

% assert: 如果條件為 false 就 Throw error
assert(mec_num == total_mec_count);
% 檢查 mec_num 是否等於 total_mec_count
clear mec_num
% 把變數 mec_num 清掉 (釋放記憶體)
assert(total_mec_count <= N_NODE);
% 檢查 total_mec_count 是否小於等於 N_NODE
% 檢查 mec_id 裡面的元素是否都小於等於 N_NODE
for i = 1:1:total_mec_count
    assert(mec_id(i) <= N_NODE);
end

% MEC 클래스 인스턴스 만들어서 저장하기
% 創建和保存 MEC 類實例
% 클래스의 배열 만들기 : 0 값이 들어가야 함
% 創建一個類數組 : 必須包含 0 個值
% [Debug] 沒有 MEC class: 下面四行註解掉不然會炸
%mecls = MEC.empty(total_mec_count,0);
%for i=1:1:total_mec_count
%    mecls(i) = MEC(mec_id(i));
%end

% 화면에 MEC 그리기
% 在螢幕上繪製 MEC
% MEC 의 정보를 매트릭스로 저장 : mec_info(mec ID, mec x, mec y)
% 將 MEC 信息保存為矩陣 : mec_info(mec ID, mec x, mec y)
N_DIM = 3;
% mec_ind, mec_x, mec_y
    MEC_ID_IND = 1;
    MEC_X_IND = 2;
    MEC_Y_IND = 3;
mec_info = zeros(total_mec_count, N_DIM);

for i = 1:1:total_mec_count
    % 랜덤하게 생성한 mec_id 값을 이용해서,
    % 使用隨機生成的 mec_id 值

```



```

% MEC 의 정보(x,y)를 mec_info 에 저장하기
% 在 mec_info 中保存 MEC 資訊 (x,y)
mec = mec_id(i);
mec_info(i,MEC_ID_IND) = mec;

% GRID 에서 현재 mec 의 행렬상 좌표값 계산
% 計算 GRID 中當前 mec 的矩陣坐標
mec_row = floor(mec/N_NODE_PER_ROW);
mec_col = mod(mec, N_NODE_PER_ROW);
if ( mec_col == 0 )
    mec_col = N_NODE_PER_ROW;
    mec_row = mec_row-1;
end

fprintf("%d %d %d\n", mec, mec_row, mec_col);
% GRID 에서 현재 mec 의 지리적 (plot) 좌표값 계산
% 計算 GRID 中當前 mec 的地理 (繪圖) 坐標
mec_x = (mec_col) * horizontal_spacing;
mec_y = (mec_row) * vertical_spacing;
% Plot 좌표값을 저장
% 保存繪圖坐標
mec_info(i,MEC_X_IND) = mec_x;
mec_info(i,MEC_Y_IND) = mec_y;
end

% 그림으로 그리기
% 畫成圖片
f2 = figure;
ax = gca;
scatter(mec_info(:,MEC_X_IND), mec_info(:,MEC_Y_IND));
xlim([0 + 0.5, N_NODE_PER_ROW + 0.5]);
% 設定 X 軸上下界
ylim([0 - 0.5, N_NODE_PER_ROW - 0.5]);
% 設定 X 軸刻度
xticks([1:1:N_NODE_PER_ROW]);
% 設定 X 軸刻度標籤
xticklabels({});
% 設定 Y 軸上下界

```

```

yticks([0:1:N_NODE_PER_COL-1]);
% 設定 Y 軸刻度
yticklabels({});
% 設定 Y 軸刻度標籤
%xlabel('x coordinate index');
%ylabel('y coordinate index');
grid on;
% 開啟圖表網格
ax.GridAlpha = 1;
% Grid line 을 완전히 불투명하게
% 讓網格完全不透明 (網格 Alpha = 100%)
ax.GridColor = [0 0 0];
% 검은색
% 黑色 (設定網格格線)
box off;
% 把圖表外框關掉只留軸線
set(gca,'XColor','none');
set(gca,'YColor','none');
% 軸線顏色全設為無色 (不顯示軸線)

% plot 에서 text 출력 때, 가시성을 높이려고 text 의 좌표를 약간 이동
% 在繪圖中輸出文字時, 稍微移動文字的坐標以增加可見性
% 顯示節點資料
text_print_gapX = 0.07;
text_print_gapY = 0.17;
for i = 1:1:total_mec_count
    text(mec_info(i,MEC_X_IND) + text_print_gapX, mec_info(i, MEC_Y_IND) +
text_print_gapY, string(mec_info(i,MEC_ID_IND)));
end

% 그림파일로 저장
% 另存為圖片文件
set(gcf,'renderer','painters');
%set(gcf,'renderer','opengl');
saveas(f2, strcat(subdir, out_filename, "-grid.png"));
saveas(f2, strcat(subdir, out_filename, "-grid.pdf"));
saveas(f2, strcat(subdir, out_filename, "-grid.eps"));
fprintf("MEC ID | MEC_X | MEC_Y : \n");

```

```

% 把 info 全 dump 出來
for i=1:1:total_mec_count
    fprintf("%d | %d | %d\n", mec_info(i,MEC_ID_IND), mec_info(i,MEC_X_IND),
mec_info(i,MEC_Y_IND));
end

%% Finalizing
fprintf("Diary... off\n");
diary off
% txt 파일에 터미널 출력값 저장(diary) 종료
% 結束保存終端輸出框 (diary) 到 txt 文件

fprintf("Saving to %s.mat\n", out_filename);
% mat 파일에 데이터 저장
% 將數據保存到 mat 文件
% figure handle 은 저장용량을 많이 차지하므로 mat 파일 저장 전에 삭제함
% figure handle 佔用大量存儲空間，所以在保存 mat 文件之前先刪除
clear f1
clear f2
clear adj
clear adjDelay
clear d
clear i
clear lower_node
clear mec
clear mec_col
clear mec_id
clear mec_row
clear mec_x
clear mec_y
clear node
clear right_node
clear G
clear Gdelay

save(strcat(subdir, out_filename, ".mat"));
% workspace 의 변수를 mat 파일로 저장

```

% 將工作區中的變數保存到 mat 文件