

```

%% Summary
% 네트워크를 구성하고, 링크별로 연결여부(0/1), 딜레이([0,1]사이 실수) 등을
추가함
% 스위치/라우터는 고정간격 격자형으로 배치되어 있고,
% 격자가 겹치는 부분중 일부에 MEC가 위치하게 된다. MEC는 확률적으로 생성함

%% Init
clc;clear;close all;

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

%% System Param

% 생성되는 파일을 저장할 하위 디렉토리
subdir = 'save/';

% 시나리오 ID 구성 : X_Y_Z_AAAA
% X : N_NODE_PER_ROW/COL 를 의미
% Y : DELAY_MAX 를 의미(DELAY_MIN=0 으로 고정)
% Z : MEC 를 생성할 확률값
% AAAA ; 임의로 붙인 순서번호

% scenario_id = '4_1.0_0.3_0001'; % ILP/LR subopt 확인용 시나리오
%scenario_id = '5_1.0_0.3_0001'; % 작은 네트워크 실험용
scenario_id = '10_1.0_0.3_0001'; % Large 네트워크 실험용

strings = split(scenario_id, '_');

% 네트워크에 총 몇개의 스위치/라우터로 격자를 만들지...
% MEC/스위치/라우터 등등은 격자 형태로 연결되어 있다고 가정
% NODE = MEC/스위치/라우터 등 모두를 의미
% int32로 캐스팅 하지 말것! 아래에서 floor 연산에서 오류남
N_NODE_PER_ROW = str2double(strings(1)); % 하나의 행에 몇개의 노드가 배치될지
N_NODE_PER_COL = N_NODE_PER_ROW; % 하나의 열에... 정사각형 형태를 가정
N_NODE = N_NODE_PER_ROW * N_NODE_PER_COL; % 전체 노드 수

% 링크 딜레이를 어떻게 설정할지...Uniform distribution을 가정
DELAY_MIN = 0.0; % 이걸 바꾸지 말자...

```

```

DELAY_MAX = str2double(strings(2));

% MEC 생성 확률
mec_gen_prob = str2double(strings(3));

% 코드를 실행하면 여러개의 파일이 생성되는데, 파일의 이름 기본값 설정
out_filename = scenario_id; % 저장용 파일의 이름의 기본값을 설정

% 터미널 출력값을 텍스트 형태로 txt 파일에 저장
% "arguments must contain a char vect..." 라는 오류 나오면
% 파일명을 구성하는 모든 텍스트를 ""가 아닌 ''로 사용하기
diary_filename = strcat(subdir, out_filename, '.txt');
if( isfile(diary_filename) )
    delete(diary_filename);
end
diary(diary_filename); % 시작!

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

% 인접행렬 만들기
adj = zeros(N_NODE, N_NODE);
% 딜레이 값이 추가된 인접행렬 만들기: adj(0/1) + link delay
adjDelay = zeros(N_NODE, N_NODE);

for node=1:1:N_NODE
    %fprintf("Node : %d\n", node); % 디버깅용

    % 각 노드를 기준으로 자신의 우측과 하단만 연결하면 되잔아?

    % 1. 우측 노드 연결
    right_node = node+1; % 우측 노드의 번호
    if( right_node <= N_NODE && rem(node, N_NODE_PER_ROW) > 0 )
        % 인접행렬은 대칭이다.
        adj(node, right_node) = 1;
        adj(right_node, node) = 1;

        % random delay 추가
        d = (DELAY_MAX-DELAY_MIN)*rand()+DELAY_MIN;
    end
end

```

```

        adjDelay(node, right_node) = 1 + d;
        adjDelay(right_node, node) = 1 + d;
    end

% 2. 아래쪽 노드 연결
lower_node = node + N_NODE_PER_ROW;
if( lower_node <= N_NODE ) % 유효한 노드이면 연결
    % 인접행렬은 대칭이다.
    adj(node, lower_node) = 1;
    adj(lower_node, node) = 1;

    % random delay 추가
    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
netw = Network(adj, adjDelay);

% 그래프 객체 생성
% https://www.mathworks.com/help/matlab/ref/graph.html
%G = graph(adj);
G = netw.getGraph();
%Gdelay = graph(adjDelay);
Gdelay = netw.getGraphPlusDelay();

% matlab 의 그래프를 이용해서 인접행렬을 그림으로 그리는 방법
f1 = figure;
plot(G, 'Layout', 'auto');
set(gca, 'XTick', []);
set(gca, 'YTick', []);
% 그림파일로 저장
saveas(f1, strcat(subdir, out_filename, "-graph.png"));

% shortest path 계산 및 저장
% https://www.mathworks.com/help/matlab/ref/graph.distances.html

```

```

%fprintf("Calc shortest path with adj matrix.Wn");
%distanceMatrix = distances(G);

% edge 에 weight 을 추가
% 위에서 adj mat 만들때 추가하면 될듯? undirected 니까, 양방향 모두 추가
% 'positive' 옵션 주면 weighted graph 에서도 최단거리 계산할 수 있음
%fprintf("Calc shortest path with adj+delay matrix.Wn");
%distanceDelayMatrix = distances(Gdelay, 'Method', 'positive');

%% Part 2. 배경에 격자를 그리고, 원하는 위치에 MEC 를 배치하는 방법

% 격자가 겹치는 모든 곳에는 switch/router 가 있다고 가정
% 서비스를 제공하는 active 한 MEC 만 격자위에 표시

horizontal_spacing = 1; % plot 에서의 수평 간격
vertical_spacing = 1; % plot 에서의 수직 간격
%N_NODE_PER_COL = N_NODE_PER_COL;
%N_NODE_PER_ROW = N_NODE_PER_ROW;
margin_gap = 0.5; % 그림 그릴때 여백 추가용

mec_id = [];
total_mec_count = 0;
for i=1:1:N_NODE
    if( rand < mec_gen_prob )
        % MEC 를 확률적으로 생성
        mec_id(end+1) = i;
        total_mec_count = total_mec_count + 1;
    end
end

mec_num = size(mec_id,2); % MEC 개수 확인

% 검증
assert(mec_num == total_mec_count);
clear mec_num
assert(total_mec_count <= N_NODE);
for i=1:1:total_mec_count
    assert(mec_id(i) <= N_NODE);

```

```

end

% MEC 클래스 인스턴스 만들어서 저장하기
% 클래스의 배열 만들기 : 0 값이 들어가야 함
mecs = MEC.empty(total_mec_count,0);
for i=1:1:total_mec_count
    mecs(i) = MEC(mec_id(i));
end

% 화면에 MEC 그리기
% MEC의 정보를 매트릭스로 저장 : 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,
    % MEC의 정보(x,y)를 mec_info에 저장하기
    mec = mec_id(i);
    mec_info(i,MEC_ID_IND) = mec;
    % GRID에서 현재 mec의 행렬상 좌표값 계산
    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);
    % GRID에서 현재 mec의 지리적(plot) 좌표값 계산
    mec_x = (mec_col) * horizontal_spacing;
    mec_y = (mec_row) * vertical_spacing;
    % Plot 좌표값을 저장
    mec_info(i,MEC_X_IND) = mec_x;
    mec_info(i,MEC_Y_IND) = mec_y;
end

```

```

% 그림으로 그리기
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]);
ylim([0-0.5, N_NODE_PER_COL-0.5]);
xticks([1:1:N_NODE_PER_ROW]);
xticklabels({});
yticks([0:1:N_NODE_PER_COL-1]);
yticklabels({});
%xlabel('x coordinate index');
%ylabel('y coordinate index');
grid on;
ax.GridAlpha = 1; % Grid line 을 완전히 불투명하게
ax.GridColor = [0 0 0]; % 검은색
box off;
set(gca, 'XColor', 'none');
set(gca, 'YColor', 'none');

% plot 에서 text 출력 때, 가시성을 높이려고 text 의 좌표를 약간 이동
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
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");
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

```

```
end
```

```
%% Finalizing
```

```
fprintf("Diary... off\n");
```

```
diary off % txt 파일에 터미널 출력값 저장(diary) 종료
```

```
fprintf("Saving to %s.mat\n", out_filename); % mat 파일에 데이터 저장
```

```
% figure handle 은 저장용량을 많이 차지하므로 mat 파일 저장 전에 삭제함
```

```
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
```