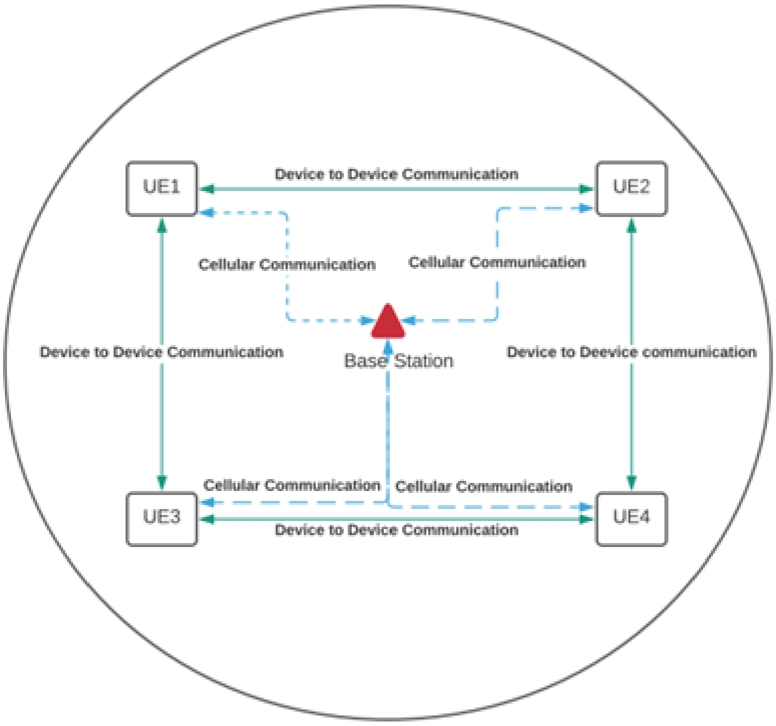
Date:2021-12-25

Working on the interim report.

Draw a D2D communication laid Cellular Network:



Finish the system model section of the Implementation section:

Energy harvesting is a technology which can be used to capture and convert those energy that we don't want to use such as thermal energy, radiant energy and sound energy into electricity and put the converted energy into the work based on users' requirements. The energy source from the ambient environment is high-quality compared to the normal batteries and capacitors which is usually used in the normal EH model. However, it is not quite effective as expected and the reason for that because when the EH models use natural sources since there is a lot of ambient sources that cannot be predicted and most of them are quite unstable [2].

So WPT is one of the technologies which can be used in the EH model to solve the issues above where the nodes can be charged from the radiation of electromagnetic field [8]. There are two types of WPT which are usually used in EH: near-field and far-field, they can be both used to harvest energy either from ambient environment and or base station. However, both have some drawbacks. For example, it is quite hard to maintain the field strengths with an appropriate level when using near-filed WPT. And when using far-field WPT where far-field means long distance, the distance from users to base stations is an important issue which needs to be considered and improved. So, considering the drawbacks of both methods mentioned in WPT, we want to introduce SWIPT in this project.

With the development of technology, the size of most of the wireless devices are becoming smaller and more energy efficient, and in the future, the radio waves will not only provide energy to operate these devices, but they will also be unified in terms of transferring energy and information [8]. There are a lot of wireless technology which can carry information and power simultaneously such as power line communication (PLC). However, it also has some technical drawbacks when it comes to energy efficiency and interference management. And that is the reason why SWIPT was introduced whose theoretical concept was first introduced in [9]. As its name indicates, SWIPT is a technology which can transfer information and energy simultaneously, it is very useful especially for notable gain of spectral efficiency and Interference Exploitation [3], which makes this technology become the basis of energy supply and exchange of information in the era of IOT.

4.2 Implementation

4.2.1 System Model

To start with the experiment mentioned in [4], first the D2D communication underlaid cellular network model need to be generated. In the system model as shown in Figure 2, there is a D2D link set and CUE set which will be denoted as $D = \{1,2,3 \dots i, \dots N\}$ and $C = \{1,2,3, \dots k, \dots M\}$ where in this project, $M \leq N$.It is noted that each D2D transmitter will be paired with one corresponding D2D Receiver. In this D2D cellular network model, for the sake of not losing generality, each D2D transmitter and Receiver should be distributed randomly while it should not be distributed out of the system following the radius of the cellular network which is set as 250 m in this project, to achieve that, the coordinates of each D2D transmitter and receiver will be randomly generated and cantered around the base station. Also, to measure the pre-matching failure rate versus communication distance, the distance of each D2D link will be changed for different occasions.

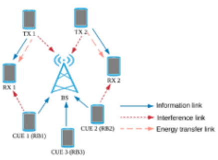


Figure 2: standard D2D communication cellular network [4]

Figure 3 plots the generated model for the scenario of 10m of D2D communication distance when the number of D2D and CUE are set as 20.

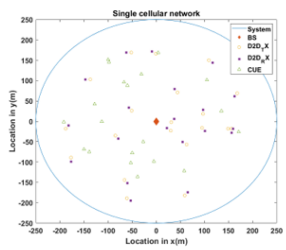


Figure 3: generated system model where communication distance is 10m

Figure 4 plots the generated model for the scenario of 20m of D2D communication distance when the number of D2D and CUE are set as 20.

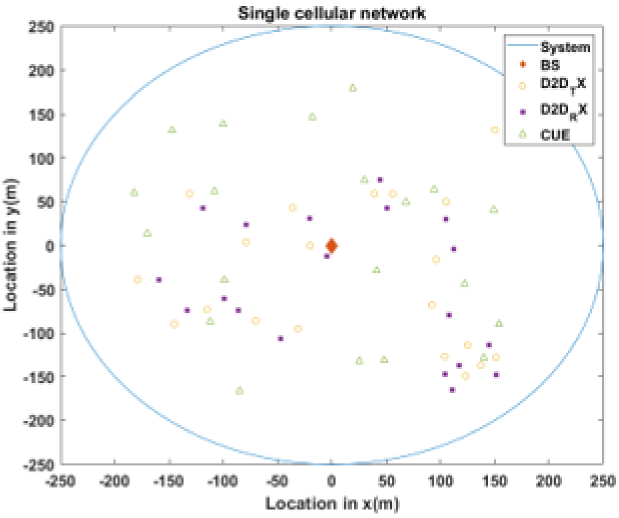


Figure 4: generated system model where communication distance is 20m

The function for generating the system has been modified since the previous one sometimes can generate some points whose coordinates are [0 0] which is the base station point. The modified code in MATLAB is:

```
function [D2D_group,CUE_group]=system_model(number_D2D,num_CUE,distance)
D2D_TX_coord=[];
CUE_coord=[];
RX_coord=[];
for i=1:number_D2D
    D2D_TX_coord
    x=randperm(400,1)-200;
    y=randperm(400,1)-200;
    check1=x==0;
    check2=y==0;
    check=check1*check2;
    isempty(D2D_TX_coord)
    if isempty(D2D_TX_coord)
        if x^(2)+y^(2)<=200^2 && check==0
           D2D_TX_coord(i,1)=x;
           D2D_TX_coord(i,2)=y;
        else
            while x^(2)+y^(2)>200^2 ||check==1
              x=randperm(400,1)-200;
              y=randperm(400,1)-200;
              check1=x==0;
              check2=y==0;
              check=check1*check2;
            end
            D2D_TX_coord(i,1)=x;
            D2D_TX_coord(i,2)=y;
        end
    elseif isempty(D2D_TX_coord)==0
        find=false;
        while find==false
            if x^(2)+y^(2)<=200^2 && ~any(D2D_TX_coord(:,1)==x)&& check==0
            D2D_TX_coord(i,1)=x;
```

```
             D2D_TX_coord(i,2)=y;
             find=true;
             else
              x=randperm(400,1)-200;
              y=randperm(400,1)-200;
              check1=x==0;
              check2=y==0;
              check=check1*check2;
             end
         end
     end
end
for j=1:num_CUE
    x_C=randperm(400,1)-200;
    y_C=randperm(400,1)-200;
    check_x_C=x_C==0;
    check_y_C=y_C==0;
    check_C=check_x_C*check_y_C;
    if isempty(CUE_coord)
        if x_C^(2)+y_C^(2)<=200^2 && check_C==0
           CUE_coord(j,1)=x_C;
           CUE_coord(j,2)=y_C;
        else
            while x_C^(2)+y_C^(2)>200^2 ||check_C==1
              x_C=randperm(400,1)-200;
              y_C=randperm(400,1)-200;
              check_x_C=x_C==0;
              check_y_C=y_C==0;
              check_C=check_x_C*check_y_C;
            end
            CUE_coord(j,1)=x_C;
            CUE_coord(j,2)=y_C;
        end
    else
        find_C=false;
        while find_C==false
            if x_C^(2)+y_C^(2)<=200^2 &&
~any(CUE_coord(:,1)==x_C)&&~any(D2D_TX_coord(:,1)==x_C)&& check_C==0
                CUE_coord(j,1)=x_C;
                CUE_coord(j,2)=y_C;
                find_C=true;
                break;
            else
                x_C=randperm(400,1)-200;
                y_C=randperm(400,1)-200;
                check_x_C=x_C==0;
                check_y_C=y_C==0;
                check_C=check_x_C*check_y_C;
            end
        end
    end
end

for i=1:size(D2D_TX_coord,1)
    temp=D2D_TX_coord(i,:);
    x=temp(1);
    y=temp(2);
    [a,b]=RX(x,y,distance);
```

```
    RX_coord(i,:)=[a b];
end
temp_D2D={};
temp_CUE={};
for i=1:number_D2D
    temp_D2D{i,1}=D2D_TX_coord(i,:);
    temp_D2D{i,2}=RX_coord(i,:);
end
D2D_group=temp_D2D;
for k=1:num_CUE
    temp_CUE{k,1}=CUE_coord(k,:);
end
for p=1:size(temp_CUE,1)
    CUE_group(p,:)=temp_CUE{p,1};
end
end
```

In this function, for each loop, it will generate random points for CUE and D2D transmitters. And for each loop, it will check if the generated D2D or CUE points are in the established cellular network which is a circle   whose radius is 250m. It will also check if the current generated points are both 0 by using the ***and*** logic. For example

```
check1=x==0;
check2=y==0;
check=check1*check2;
```