Introduction to Wireless and Mobile Networking
Hw4 – Report

B03901026 許凱傑

I MATLAB code

1-1 Main Function

(1) Parameter setting

```
% Parameter setting
ISD     = 500;              % inter site distance
side    = ISD/sqrt(3);
num_MS = 50;
sim_time = 1000;
T       = 27 + 273.15;
B_tot   = 10e6;
H_BS    = 1.5;              % height of Base station
H_B     = 50;               % height of building
H_MS    = 1.5;              % height of mobile station
H_T     = H_BS + H_B;
P_BS    = 33 - 30;          % BS power = 33 dBm
P_MS    = 0  - 30;          % MS power = 0  dBm
G_R_dB = 14;
G_T_dB = 14;
label = 10;

BS_X = side*[-3,-3,-3,-1.5,-1.5,-1.5,-1.5,0,0,0,0,0,1.5,1.5,1.5,1.5,3,3,3];
BS_Y = ISD*[-1,0,1,-1.5,-0.5,0.5,1.5,-2,-1,0,1,2,-1.5,-0.5,0.5,1.5,-1,0,1];
```

(2) Get the MS and the BS scatter.

```
[x,y] = hexagon(side,0,0,50,1);
[edgeX,edgeY] = hexagonborder(side,0,0,1);
```

(3) Get the Shannon capacity of each MS

```
%% 4-2 Shannon capacity to distance
B = B_tot / num_MS;
Noise = myThermalNoise(T, B);

% get distance b/w each MS & BS
d = zeros(19,num_MS);
for i = 1:19
    d_x = x-BS_X(i);
    d_y = y-BS_Y(i);
    d(i,:) =  sqrt(d_x.^2 + d_y.^2);
end
```

```
G_C = G_two_ray_ground(H_T, H_MS, d);
G_C_dB = todB(G_C);
power = P_BS + G_T_dB + G_R_dB + G_C_dB;
power = fromdB(power);
P_R_MS =  power(10,:);
I_t = sum(power,1) - power(10,:);
SINR = mySINR( P_R_MS, I_t, Noise);

capacity = B * log2(1+fromdB(SINR));
distance = sqrt(x .^ 2 + y .^ 2);
```

Introduction to Wireless and Mobile Networking
Hw4 – Report

B03901026 許凱傑

(4) Simulation

a CBR(constant bits rate)

```
buffersize = 1e6;
missrate = zeros(1,3);
CBR = 1e6 * [1,0.5,0.1];
```

```
for type = 1 : 3
    buffer = zeros(1,num_MS);
    miss = zeros(1,num_MS);
    rate = CBR(type);
    for t = 1:sim_time
        data = rate + buffer;
        oversize = data - capacity;
        overflow = oversize > 0;
        buffer(~overflow) = 0;
        oversize(~overflow) = 0;
        temp = 0; % how many bits store in buffer
        for i = 1 : num_MS
            if overflow(i)
                if temp + oversize(i) <= buffersize
                    temp = temp + oversize(i);
                    buffer(i) = oversize(i);
                else
                    stop = i;
                    store = buffersize - temp;
                    loss = oversize(i) - store;
                    miss(i) = miss(i) + loss ;
                    buffer(i) = store;
                    break;
                end
            end
        end
        miss(stop+1:num_MS) = miss(stop+1:num_MS) + oversize(stop+1:num_MS);
        buffer(stop+1:num_MS) = 0;
    end
    missrate(type) = sum(miss) / (sim_time * num_MS * rate);
end
```

i There are 3 types of CBR. Each is corresponding to high, medium and low load, respectively.

ii I divide buffer into <u>num_MS</u> sections and the sum of the bits stored in each section cannot exceed the <u>buffersize</u>.

iii In every time slot, I will add CBR with the stored bits in according buffer section as data and calculate the difference between the data and Shannon capacity. If the difference larger than 0, then put the overflow in the buffer as many as it can store. All of the overflow bits cannot put in the buffer will be considered as miss bits.

iv After <u>sim_time</u>, I will divide the sum of the miss bits from each MS by the total traffic bits and get the <u>missrate</u>.

Introduction to Wireless and Mobile Networking

Hw4 – Report

B03901026 許凱傑

b Traffic arrival for each mobile device follows Poisson distribution.

```
buffersize = 1e6;
missrate = zeros(1,3);
lambda = 1e6 * [1.5,0.5,0.2];
```

```
for type = 1 : 3
    buffer = zeros(1,num_MS);
    miss = zeros(1,num_MS);
    for t = 1:sim_time
        rate = poissrnd(lambda(type));
        data = rate + buffer;
        oversize = data - capacity;
        overflow = oversize > 0;
        buffer(~overflow) = 0;
        oversize(~overflow) = 0;
        temp = 0; % how many bits store in buffer
        for i = 1 : num_MS
            if overflow(i)
                if temp + oversize(i) <= buffersize
                    temp = temp + oversize(i);
                    buffer(i) = oversize(i);
                else
                    stop = i;
                    store = buffersize - temp;
                    loss = oversize(i) - store;
                    miss(i) = miss(i) + loss ;
                    buffer(i) = store;
                    break;
                end
            end
        end
        miss(stop+1:num_MS) = miss(stop+1:num_MS) + oversize(stop+1:num_MS);
        buffer(stop+1:num_MS) = 0;
    end
    missrate(type) = sum(miss) / (sim_time * num_MS * rate);
end
```

i The only difference between this simulation and previous one is how the traffic bits generate. In this simulation, the traffic bits follow Poisson distribution with 3 types of lambda, which is the mean and the variance of the distribution. Each lambda is corresponding to high, medium and low load, respectively.

ii In every time slot, I will generate a Poisson random variable and add it with the stored bits in according buffer section as data. The other steps are the same as previous simulation.

Introduction to Wireless and Mobile Networking
Hw4 – Report

B03901026 許凱傑

1-2 Functions

(1) Number ↔ dB

```
% transfer to dB
function dB = todB(x)
dB = 10*log10(x);
end
```

```
% transfer from dB
function x = fromdB(dB)
x = 10.^(dB/10);
end
```

(2) Two-ray-ground model

```
% two-ray-ground-model
function G_d = G_two_ray_ground(H_t, H_r, d)
G_d = (H_t * H_r)^2 ./ (d .^ 4);
end
```

(3) Thermal Noise

```
% My Thermal Noise
function N_T = myThermalNoise(Temperature,Bandwidth)
k = physconst('Boltzmann');
N_T = k*Temperature*Bandwidth;
end
```

(4) SINR

```
% SINR in dB
function SINR = mySINR_dB(S, I, N)
SINR = 10*log10(S/(I+N));
end
```

(5) Hexagon

Can randomly create num_MS MSs

By dividing each cell into 3 regions, and use 2 vectors combination to get each MS's location.

```
function [vectorX,vectorY] = hexagon(side,x0,y0,num_MS,dra
  %side = side size;,(x0,y0) exagon center coordinates;
  L = linspace(0, 2*pi, 7);
  edgeX = side * cos(L)+x0;
  edgeY = side * sin(L)+y0;
  if draw
      plot(edgeX,edgeY,'r','Linewidth',1);
      scatter(x0,y0,'filled','g');
  end

  if num_MS >= 1
      ISD = side*sqrt(3);
      ai = [side,0,0];
      aj = [-side/2, ISD/2];
      ak = [-side/2, -ISD/2];
```

```
        temp  = randi(3,1,num_MS);
        tempx = rand(1,num_MS);
        tempy = rand(1,num_MS);
        for a =1:num_MS
            if temp(a) == 1
                x(a) = tempx(a)*ai(1) + tempy(a)*aj(1);
                y(a) = tempy(a)*aj(2);
            elseif temp(a) == 2
                x(a) = tempx(a)*ai(1) + tempy(a)*ak(1);
                y(a) = tempy(a)*ak(2) ;
            else temp(a) == 3
                x(a) = tempx(a)*aj(1) + tempy(a)*ak(1);
                y(a) = tempx(a)*aj(2) + tempy(a)*ak(2);
            end
        end;
        vectorX = x+x0;
        vectorY = y+y0;
        if draw
            scatter(vectorX,vectorY,10,'b','filled');
        end
    end;
```

Introduction to Wireless and Mobile Networking

Hw4 – Report

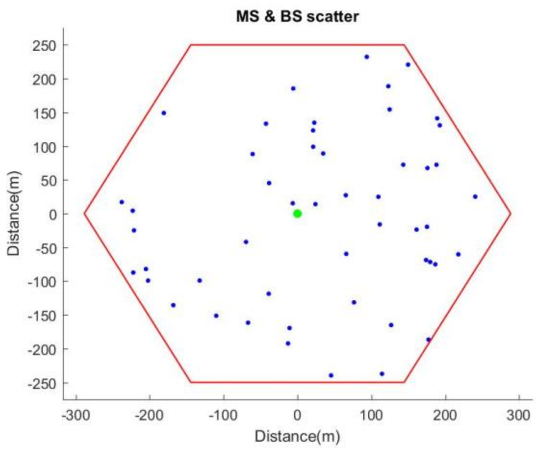
B03901026 許凱傑

(6) Hexagonborder

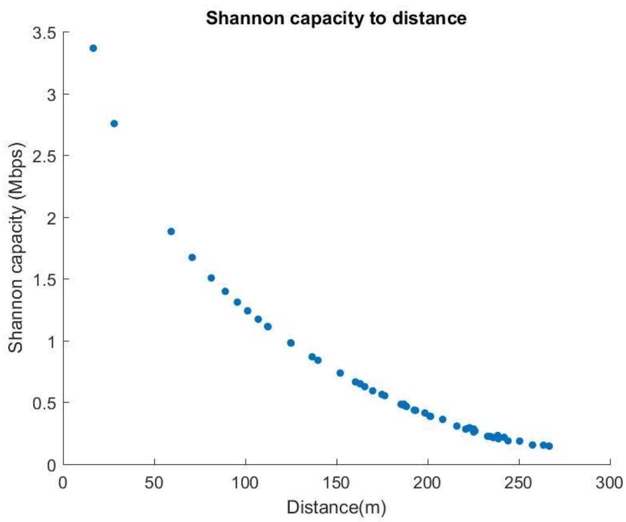
```
function [edgeX,edgeY] = hexagonborder(side,x0,y0, draw)
   %side = side size;,(x0,y0) exagon center coordinates;
   % draw(true) : draw the BS scatter and the subcell
   L = linspace(0, 2*pi,7);
   edgeX = side * cos(L)+x0;
   edgeY = side * sin(L)+y0;
   if draw == 1
       plot(edgeX,edgeY,'r','Linewidth',1);
       scatter(x0,y0,'filled','g');
   end
```

II Report

1-1 *Please plot the location of the central BS and 50 uniformly random distributed mobile devices in the central cell.*



1-2 Please plot a figure with **Shannon Capacity** (bits/s) of a mobile device in a central BS as y-axis, and with the distance between the corresponding mobile device and the central BS as x-axis. Also, **write down** how to calculate the Shannon capacity of the mobile device in the central cell in your report

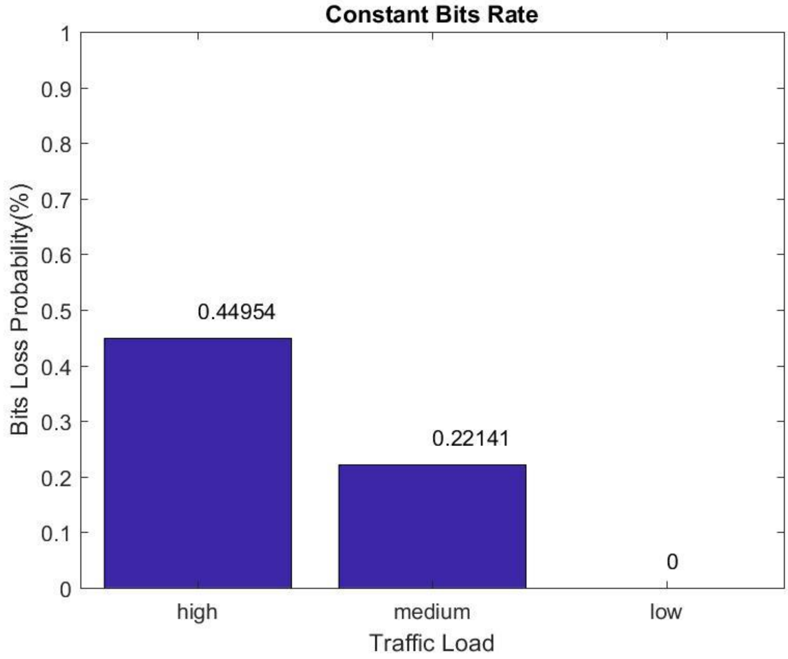


$$\text{Shannon capacity} = B \times log_2(1 + SINR)$$

Introduction to Wireless and Mobile Networking
Hw4 – Report

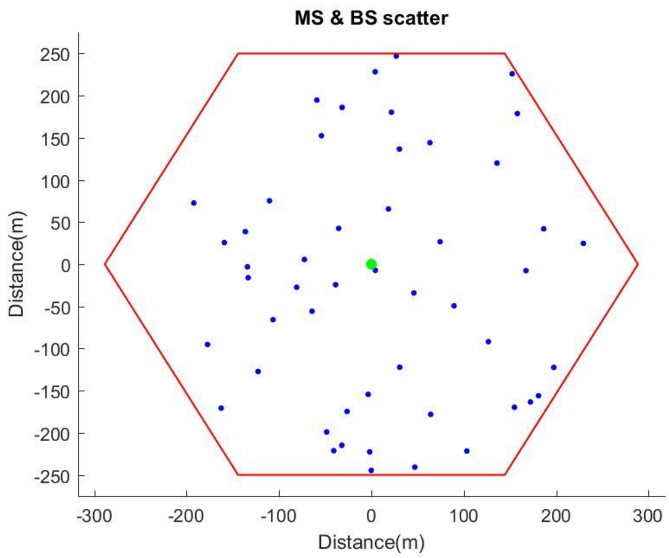
B03901026 許凱傑

1-3 *design the CBR parameters {Xl, Xm, Xh} by yourself so that you can get different values of packet loss rate in the total simulation duration for each CBR parameter. Write down the CBR parameters you used in your simulation. Plot a histogram figure with the bits loss probability in the central BS as the y-axis and the traffic load as the x-axis.*



$$CBR = \{0.1Mbps, 0.5Mbps, 1Mbps\}$$

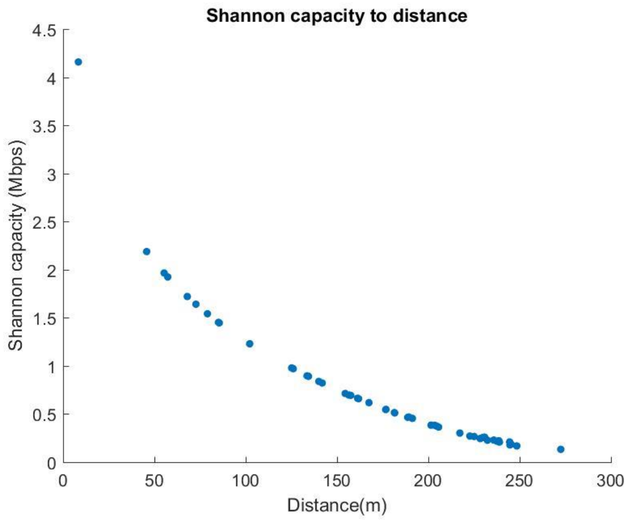
B-1 *Please plot the location of the central BS and 50 uniformly random distributed mobile devices in the central cell.*



Introduction to Wireless and Mobile Networking
Hw4 – Report

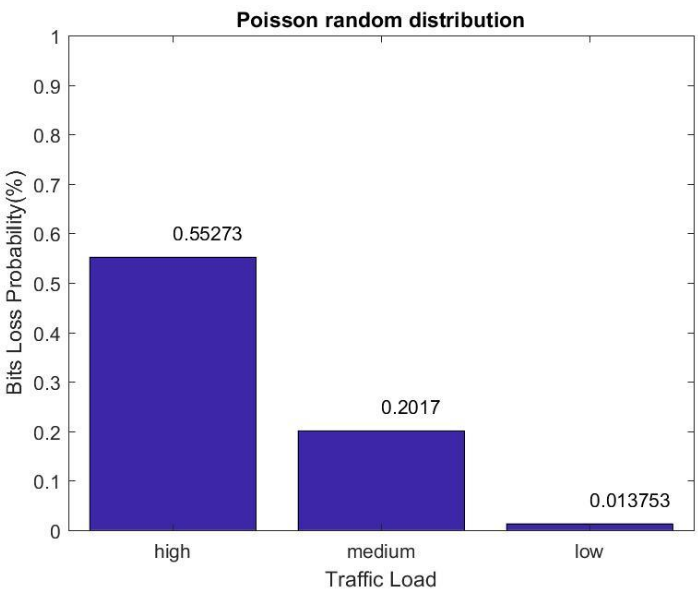
B03901026 許凱傑

B-2 Please plot a figure with **Shannon Capacity** (bits/s) of a mobile device in a central BS as y-axis, and with the distance between the corresponding mobile device and the central BS as x-axis. Also, **write down** how to calculate the Shannon capacity of the mobile device in the central cell in your report



$$\text{Shannon capacity} = \mathrm{B} \times log_2(1 + SINR)$$

B-3 *Design the Poisson traffic arrival parameters {λ_l,λ_m, λ_h} by yourself so that you can get different values of packet loss rate in the total simulation duration for each Poisson traffic arrival parameter.* ***Write down*** *the Poisson traffic arrival parameters you used in your simulation. Plot a histogram figure with the* ***bits loss probability*** *in the central BS as the y-axis and* ***the traffic load*** *as the x-axis.*



$$\lambda = \{0.2Mbps, 0.5Mbps, 1.5Mbps\}$$