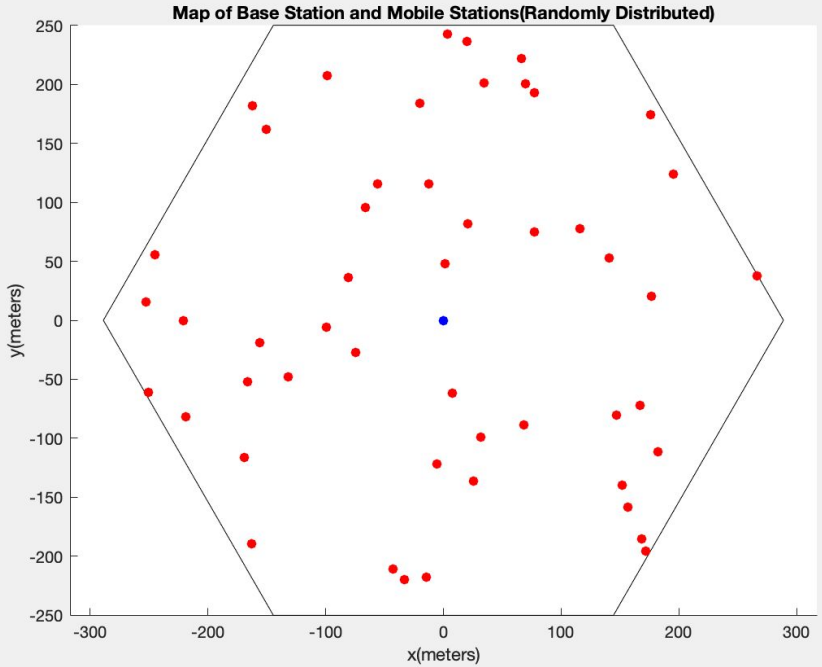
Report HW4

陳守仁 B09901116

Question 1-1



Hexagonal Cell

```
% Plot hexagonal pattern
plot(xCoords(1)+cellRadius*cosd(0:60:360),yCoords(1)+cellRadius*sind(0:60:360),'k');
scatter(xCoords(1), yCoords(1), 'filled', 'MarkerFaceColor', 'b');
```

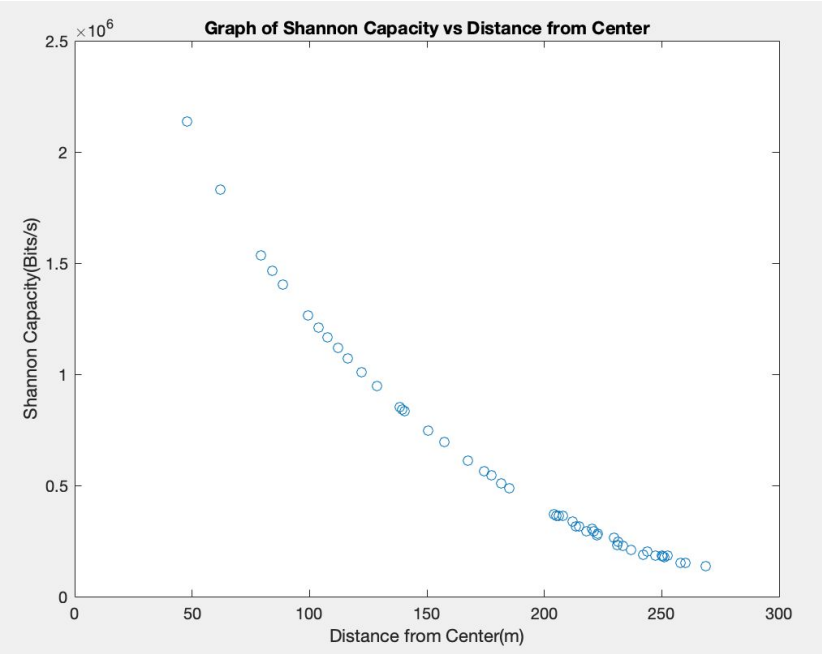
50 Randomly Distributed Mobile Cell

```
%_____________________50 mobile in each cell case _____________%

numPointsPerHex = 50;
allPoints = [];
while size(allPoints,1)<numPointsPerHex
    % Generate a random point in the hexagon
    x = (2*rand()-1)*500;
    y = (2*rand()-1)*500;
    if inpolygon(x,y,xCoords(1)+cellRadius*cosd(0:60:360),yCoords(1)+cellRadius*sind(0:60:360))
        allPoints = [allPoints; x y];
        % Add point to the list if it's inside the hexagon
    end
end
title('Map of Base Station and Mobile Stations(Randomly Distributed)')
xlabel('x(meters)')
ylabel('y(meters)')
scatter(allPoints(:,1),allPoints(:,2),'filled','r');
distance = [];
```

SINR Calculation Process

```
for i = 1:50        %creates a 50 rows and 19 columns matrix
    for j = 1:19    % each 50 MS distance to each 19 BS
        dx = allPoints(i, 1) - xCoords(j);
        dy = allPoints(i, 2) - yCoords(j);
        %xCoords and yCoords are BS centers
        distance(i, j) = sqrt(dx^2 + dy^2);
    end
end

gd = ((ht_b*ht_m)^2)./distance.^4; % 50x19 double
Pr_W = gd.*P_m_W*Gt_W*Gr_W; %50x19 double


%_____calculate interference --> add up all others except targeted_____%
Interference = zeros(size(Pr_W)); % Initialize Interference to zeros
for i = 1:size(Pr_W,1) % Iterate over rows
    for j = 1:size(Pr_W,2) % Iterate over columns

        % Add of all elements in the ith row of Pr except for the jth column
        % Add up all distance from MS#1 to all 19 BS except the 1 BS that
        % we are targeting
        Interference(i,j) = sum(Pr_W(i,[1:j-1,j+1:end]));
    end
end

%_______calculate_SINR_____%
SINR = Pr_W./(Interference+N);
SINR_dB = 10*log10(SINR);
```

Question 1-2

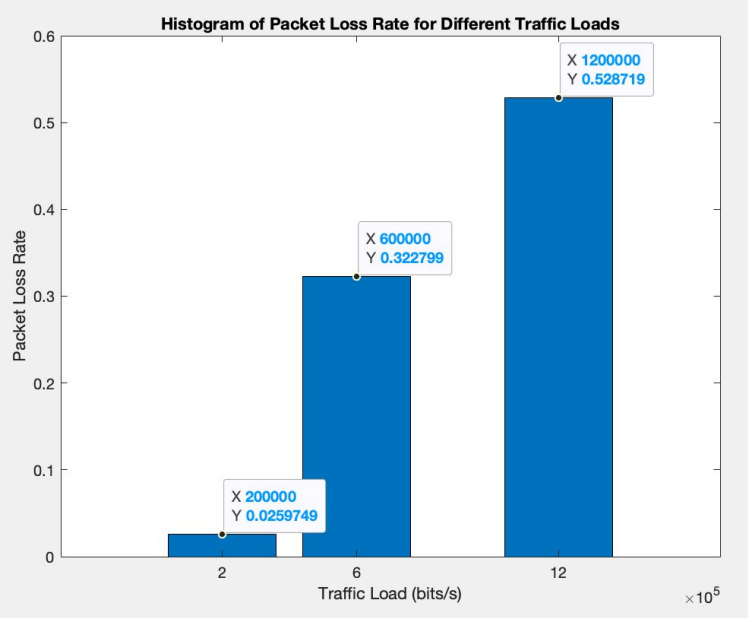


Shannon Capacitance

```
% Shannon Capacity = Bandwidth * log2(1+SINR)
eachBW = BW/50;
%shannon cap -> y-axis
%distance -> x-axis
shannonCap = zeros(50,1);
for i = 1:50
    shannonCap(i,1) = eachBW*log2(1+SINR(i,1));
end

distancefromcenter = distance(:, 1);
figure('Name','Question 1-2');

plot(distancefromcenter, shannonCap, 'o')
xlabel('Distance from Center(m)')
ylabel('Shannon Capacity(Bits/s)')
title('Graph of Shannon Capacity vs Distance from Center')
```

I use formula C = BW * log2(1+SINR)
where BW is bandwidth, C is shannon capacitance, SINR is S/(I+N)

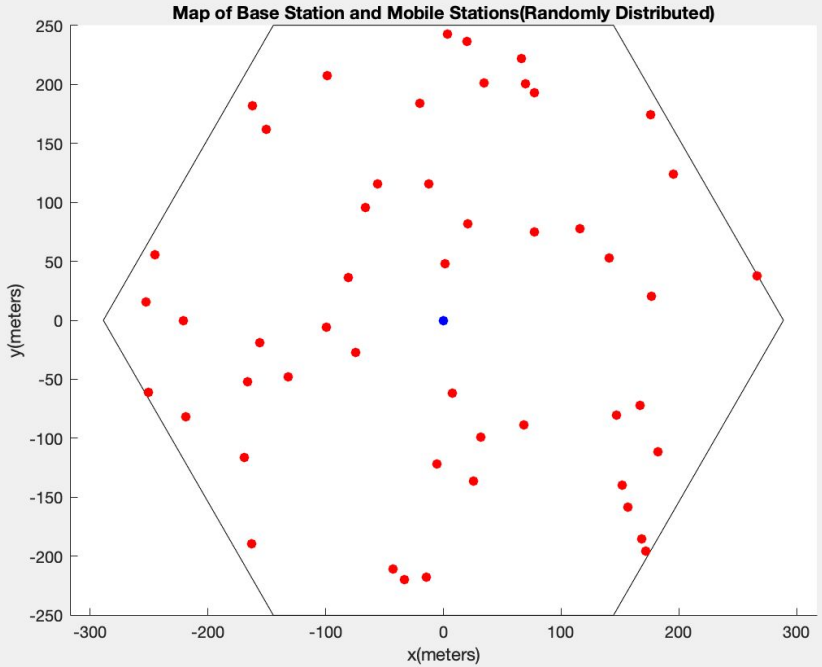
Question 1-3



CBR = [2e5, 6e5, 12e5] Bps
T = 1000s
Buffer 6M bits

```
bufferSize = 6e6;%bits
Xl=2e5; %bits per second
Xm=6e5; %bits per second
Xh=12e5; %bits per second
totalTime = 1000;%seconds

CBR = [Xl,Xm,Xh];
numBits = CBR*totalTime;

numLossBits = zeros(1,3);
for i = 1:3
    bitsArrived = zeros(1,3);
    buffer = 0;
    for t = 1 : totalTime
        for j = 1:50
            if shannonCap(j,1)>CBR(1,i)
                bitsArrived(1,i) = bitsArrived(1,i)+CBR(1,i);
            else
                bitsArrived(1,i) = bitsArrived(1,i)+shannonCap(j,1);
                buffer = buffer+CBR(1,i)-shannonCap(j,1);
            end
        end
    end
    if buffer>bufferSize
        numLossBits(1,i) = buffer-bufferSize;
    else
        numLossBits(1,i) = 0;
    end
end

lossRate = numLossBits ./ (numBits*50);


  % Plot histogram
  figure('Name','Question 1-3');
  bar([Xl, Xm, Xh], lossRate);
  xlabel('Traffic Load (bits/s)');
  ylabel('Packet Loss Rate');
  title('Histogram of Packet Loss Rate for Different Traffic Loads');
  figure('Name','Question B-1');
  hold on;
```

Bonus 1-1



Hexagonal Cell

```
% Plot hexagonal pattern
plot(xCoords(1)+cellRadius*cosd(0:60:360),yCoords(1)+cellRadius*sind(0:60:360),'k');
scatter(xCoords(1), yCoords(1), 'filled', 'MarkerFaceColor', 'b');
```

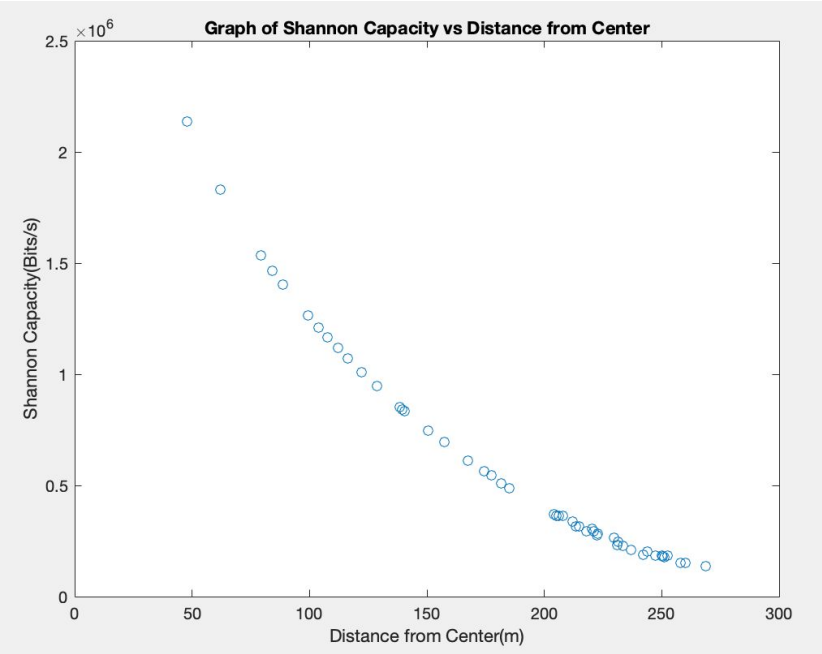
50 Randomly Distributed Mobile Cell

```
%_____________________50 mobile in each cell case _____________%

numPointsPerHex = 50;
allPoints = [];
while size(allPoints,1)<numPointsPerHex
    % Generate a random point in the hexagon
    x = (2*rand()-1)*500;
    y = (2*rand()-1)*500;
    if inpolygon(x,y,xCoords(1)+cellRadius*cosd(0:60:360),yCoords(1)+cellRadius*sind(0:60:360))
        allPoints = [allPoints; x y];
        % Add point to the list if it's inside the hexagon
    end
end
title('Map of Base Station and Mobile Stations(Randomly Distributed)')
xlabel('x(meters)')
ylabel('y(meters)')
scatter(allPoints(:,1),allPoints(:,2),'filled','r');
distance = [];
```

SINR Calculation Process →

```
for i = 1:50        %creates a 50 rows and 19 columns matrix
    for j = 1:19    % each 50 MS distance to each 19 BS
        dx = allPoints(i, 1) - xCoords(j);
        dy = allPoints(i, 2) - yCoords(j);
        %xCoords and yCoords are BS centers
        distance(i, j) = sqrt(dx^2 + dy^2);
    end
end

gd = ((ht_b*ht_m)^2)./distance.^4; % 50x19 double
Pr_W = gd.*P_m_W*Gt_W*Gr_W; %50x19 double


%_____calculate interference --> add up all others except targeted_____%
Interference = zeros(size(Pr_W)); % Initialize Interference to zeros
for i = 1:size(Pr_W,1) % Iterate over rows
    for j = 1:size(Pr_W,2) % Iterate over columns

        % Add of all elements in the ith row of Pr except for the jth column
        % Add up all distance from MS#1 to all 19 BS except the 1 BS that
        % we are targeting
        Interference(i,j) = sum(Pr_W(i,[1:j-1,j+1:end]));
    end
end

%_______calculate_SINR_____%
SINR = Pr_W./(Interference+N);
SINR_dB = 10*log10(SINR);
```

Bonus 2

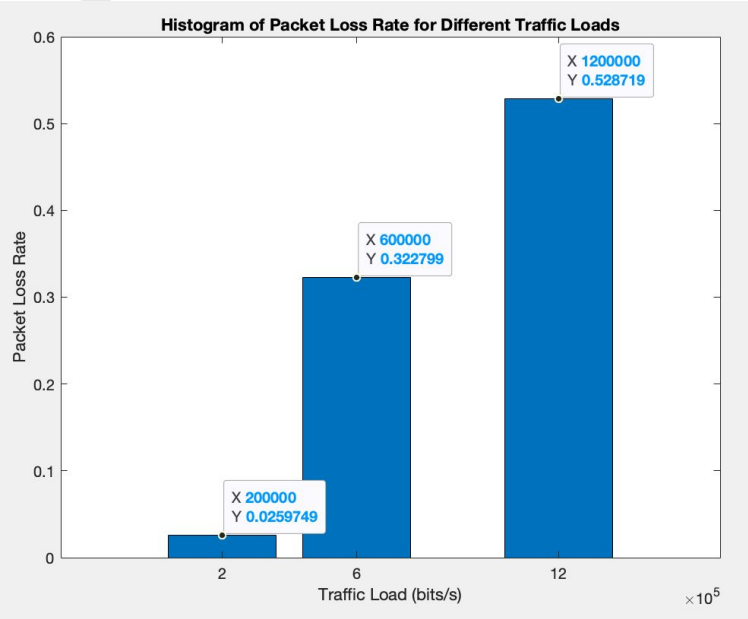


Shannon Capacitance

```
% Shannon Capacity = Bandwidth * log2(1+SINR)
eachBW = BW/50;
%shannon cap -> y-axis
%distance -> x-axis
shannonCap = zeros(50,1);
for i = 1:50
    shannonCap(i,1) = eachBW*log2(1+SINR(i,1));
end

distancefromcenter = distance(:, 1);
figure('Name','Question 1-2');

plot(distancefromcenter, shannonCap, 'o')
xlabel('Distance from Center(m)')
ylabel('Shannon Capacity(Bits/s)')
title('Graph of Shannon Capacity vs Distance from Center')
```

I use formula C = BW * log2(1+SINR)
where BW is bandwidth, C is shannon capacitance, SINR is S/(I+N)

Bonus 3



CBR = [2e5, 6e5, 12e5] Bps
T = 1000s
Buffer 6M bits
Used poissrnd function

```
bufferSize = 6e6;%bits
Xl=2e5; %bits per second
Xm=6e5; %bits per second
Xh=12e5; %bits per second
totalTime = 1000;%seconds

CBR = [Xl,Xm,Xh];
numBits = CBR*totalTime;

numLossBits = zeros(1,3);
for i = 1:3
    bitsArrived = zeros(1,3);
    buffer = 0;
    for t = 1 : totalTime
        for j = 1:50
            if shannonCap(j,1)>CBR(1,i)
                bitsArrived(1,i) = bitsArrived(1,i)+CBR(1,i);
            else
                bitsArrived(1,i) = bitsArrived(1,i)+shannonCap(j,1);
                buffer = buffer+CBR(1,i)-shannonCap(j,1);
            end
        end
    end
    if buffer>bufferSize
        numLossBits(1,i) = buffer-bufferSize;
    else
        numLossBits(1,i) = 0;
    end
end

lossRate = numLossBits ./ (numBits*50);
```

```
% Plot histogram
figure('Name','Question 1-3');
bar([Xl, Xm, Xh], lossRate);
xlabel('Traffic Load (bits/s)');
ylabel('Packet Loss Rate');
title('Histogram of Packet Loss Rate for Different Traffic Loads');
figure('Name','Question B-1');
hold on;
```