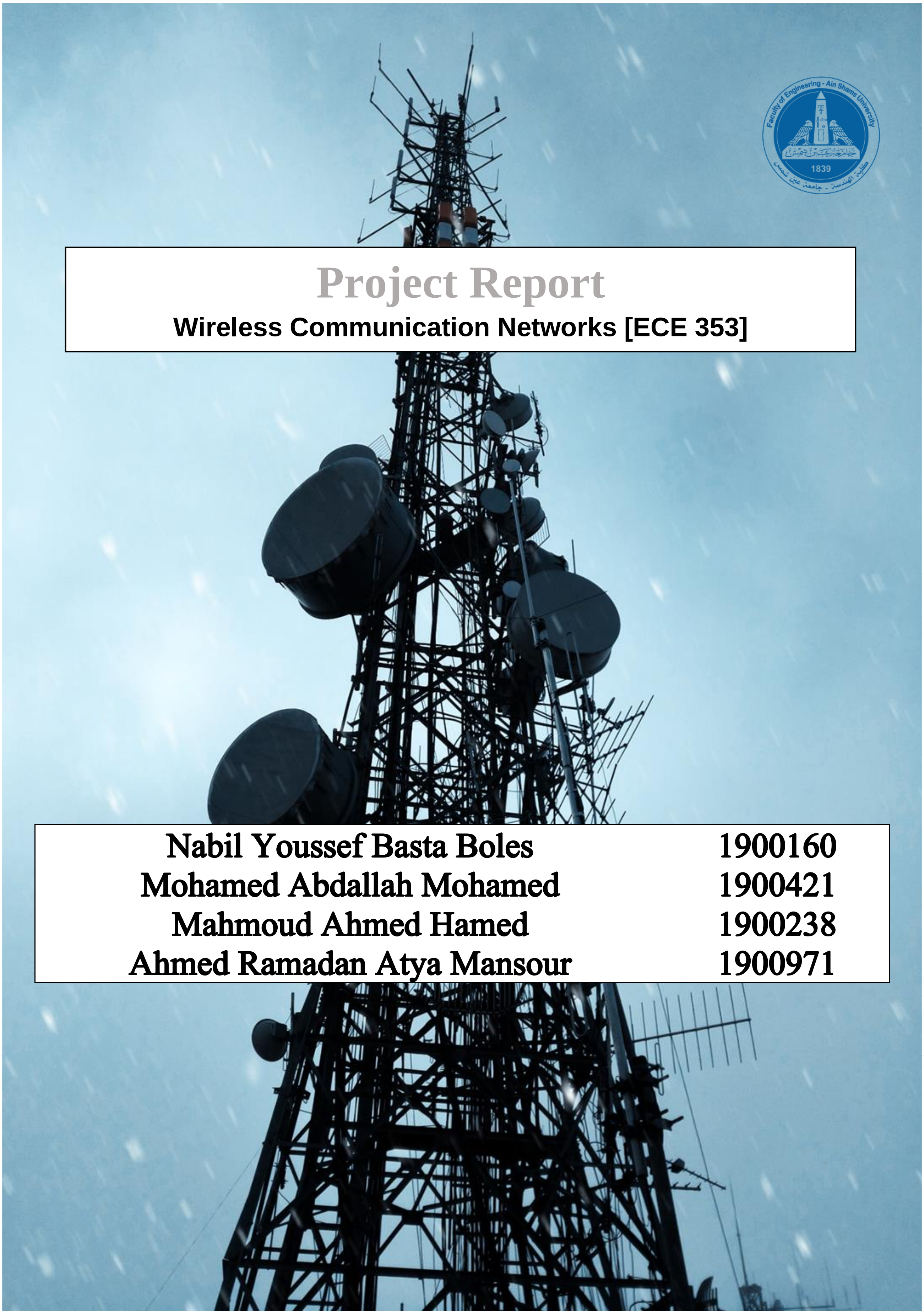


Project Report

Wireless Communication Networks [ECE 353]

Nabil Youssef Basta Boles	1900160
Mohamed Abdallah Mohamed	1900421
Mahmoud Ahmed Hamed	1900238
Ahmed Ramadan Atya Mansour	1900971

Contents

Hand Analysis

PAGE
DATE

[1] Cluster size (N)

$$SiR = \frac{(\sqrt{3N}-1)^n}{i}$$

$n=4$

i = no. of interfeers

$$N = \frac{1}{3}\left(\left(i\,10^{\frac{SiR}{10}}\right)^{\frac{1}{n}}+1\right)^2 \rightarrow N = i^2+ik+k^2$$

[2] number of Cells [3] traffic intensity (Cell & sector)

$$k = \frac{S}{N}, \quad C = \frac{k}{sectors}$$

k = Channell / cell

C = channl / sector

$$GoS = \frac{\frac{A_{sector}^{c}}{C!}}{\sum_{k=0}^{c}\frac{A_{sector}^{k}}{k!}}$$

traffic intensity Per Cell = A sector * sectors → Acell

$$\text{number of users/cell} = \frac{A_{cell}}{A_u}$$

$$\text{number of cells} = \frac{\text{user density} * \text{Area}}{\text{number of user/cell}}$$

$A_u = 0.025$ traffic intensity Per user

[4] Cell Radius

$$\text{Cell Area} = \frac{\text{Total Area}}{\text{number of Cells}}$$

$$\text{Cell Radius} = \left(\frac{\text{Cell Area}}{1.5\sqrt{3}}\right)^{\frac{1}{2}}$$

$$\text{Area} = \frac{3\sqrt{3}}{2}R^2$$

$$R = \left(\frac{\text{Area}}{1.5\sqrt{3}}\right)^{\frac{1}{2}}$$

[5] Power transmitted

Hata model

$CH = 0.8 + [(1.1 \log(f) - 0.7) h_a - 1.56 \log_{10}(f)]$

$$\text{Path loss} = 69.55 + 26.16 \log_{10} f - 13.82 \log h_a - CH + [44.9 - 6.55 \log_{10}(h_a)] \log R$$

$f = 900$ MHz

$h_a = 20$

Worst case * R = cell Radius

$P_{tx} = MS\text{-sensitivity} + \text{Path loss}$

MS = -95 dBm

[6] Received power

$$\text{Path loss} = 69.55 + 26.16 \log f - 13.82 \log h_a - CH + [44.9 - 6.55 \log_{10}(h_a)] \log(\text{distance})$$

distance $\xrightarrow{\text{Range}}$ 0 $\rightarrow$ cell Radius

$P_{rx} = P_{tx} - \text{Path loss}$

Part A

Main Part (A) MATLAB CODE

```
clear;
clc;
```

```
%% Get the input parameters
GOS = input('Enter the GOS(in decimal): ');
cityArea = input('Enter the city area (in km^2): ');
userDensity = input('Enter the user density (in users/km^2): ');
SIRmin = input('Enter the SIRmin (in dB): ');
sectorizationMethod = input('Enter the sectorization method (1 for omnidirectional, 2
for 3-sector, 3 for 6-sector): ');
if sectorizationMethod == 1
    i = 6; % numInterferers
    numSectors=1;
elseif sectorizationMethod == 2
    i = 2;
    numSectors=3;
elseif sectorizationMethod == 3
    i = 1;
    numSectors=6;
else disp('Wrong')
end
```

```
%% Givens
S = 340; %numChannels
f = 900; % band 900 MHz
BS = 20; % heightBaseStation
MS = 1.5; % heightMobileStation
Au = 0.025; %  the traffic intensity per user equals 0.025 Erlang
n = 4; %  the path loss exponent equals 4
MS_Sensitivity = -95; % MS sensitivity equals -95 dBm
```

```
%% Calculate the cluster size (N)
clusterSize =calClusterSize(i,SIRmin,n);
%% Calculate Traffic  intensity of Cell & Sector
[trafficIntensityPerSector,trafficIntensityPerCell] =
calTrafficIntensity(S,clusterSize,GOS,numSectors);
%% Calculate the number of cells
[numCells] = calNumCells(Au,trafficIntensityPerCell,userDensity,cityArea);
%% Calculate the cell radius (R)
[cellRadius] = calCellRadius(cityArea,numCells);
%% Calculate the base station transmitted power
HataCh = 0.8+(((1.1*log10(f))-0.7)*MS - 1.56*log10(f)) ;
pathLoss = 69.55+(26.16*log10(f))-(13.82*log10(BS))-HataCh+((44.9-
6.55*log10(BS))*log10(cellRadius)) ; % power loss
Ptx = MS_Sensitivity + pathLoss ;             %in dBm
```

```
%% Plot the MS received power in dBm versus the receiver distance from the BS
figure;
distance = (cellRadius/100):0.01:cellRadius;
```

```
pathLoss2 = 69.55+(26.16*log10(f))-(13.82*log10(BS))-HataCh+((44.9-
6.55*log10(BS))*log10(distance)) ;
receivedPower = Ptx - pathLoss2; % Prx dBm
plot(distance, receivedPower, 'b')
xlabel('Receiver distance (Km)');
ylabel('Received power (dBm)');
title('MS received power versus receiver distance');
```

```
%% Display the results
fprintf('The cluster size is %d\n',clusterSize);
fprintf('The number of cells is %4.2f\n',numCells);
fprintf('The cell radius is %4.2f km\n',cellRadius);
fprintf('The traffic intensity per cell is %4.2f Erlang\n',trafficIntensityPerCell);
fprintf('The traffic intensity per sector is %4.2f Erlang\n',trafficIntensityPerSector);
fprintf('The Base station transmitted power is %4.2f dBm\n',Ptx);
```

Cluster Size Function

```
function [clusterSize]=calClusterSize(i,SIRmin,n)
    N=(1/3)*((i*10^(SIRmin/10))^(1/n)+1)^2;
    testValues =[1,3,4,7,9,12,13,16,19,21,25,27,28,31,36,37,39];
    % check if N is acceptable value
    for i= 1:length(testValues)
        if testValues(i)> N
            clusterSize = testValues(i);
            break
        end
    end
    end
```

Traffic Intensity per Sector & Cell Function

```
function [trafficIntensityPerSector,trafficIntensityPerCell] =
calTrafficIntensity(S,clusterSize,GOS,numSectors)
    K =floor ( S / clusterSize); % numChannelsPerCell
    C =floor( K / numSectors) ; % channels per sector
    % to get Pr close value to GOS
    for A=1:1000
        Pr= (A^C/factorial(C))/sum(A.^([0:C])./cumprod([0,0:C-1]+1));
    %The probability of blocking is calculated using the Erlang B formula.
        if GOS<=Pr
         break
        end
    end
    Erlang = @(A1) (A1^C/factorial(C))/sum(A1.^([0:C])./cumprod([0,0:C-1]+1));
    trafficIntensityPerSector = fzero(@(A1) Erlang(A1)-GOS, A);
    %{ to find the value of A that minimizes the difference between the Erlang
probability of blocking and the GOS.
    %}
    trafficIntensityPerCell=trafficIntensityPerSector*numSectors;
    end
```

Number of Cells Function

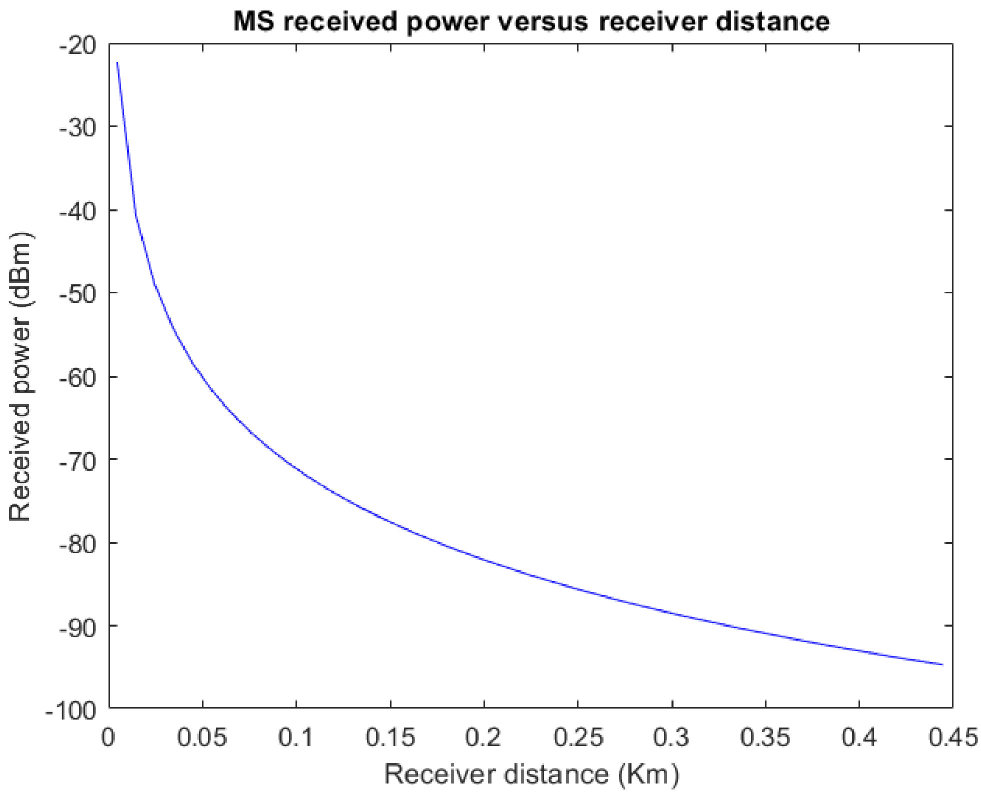
```
function [numCells] =
calNumCells(Au,trafficIntensityPerCell,userDensity,cityArea)
    UsersPerCell = trafficIntensityPerCell / Au; % total number users per cell
    numCells = ceil((userDensity * cityArea) / UsersPerCell);
    end
```

Cell Radius Function

```
function [cellRadius] = calCellRadius(cityArea,numCells)
    cellArea = cityArea / numCells;
    cellRadius = sqrt(cellArea / (1.5*sqrt(3)));
    end
```

Example

```
Command Window
Enter the GOS(in decimal): 0.01
Enter the city area (in km^2): 100
Enter the user density (in users/km^2): 1400
Enter the SIRmin (in dB): 19
Enter the sectorization method (1 for omnidirectional, 2 for 3-sector, 3 for 6-sector): 1
The cluster size is 12
The number of cells is 188.00
The cell radius is 0.45 km
The traffic intensity per cell is 18.64 Erlang
The traffic intensity per sector is 18.64 Erlang
The Base station transmitted power is 21.31 dBm
fx >>
```



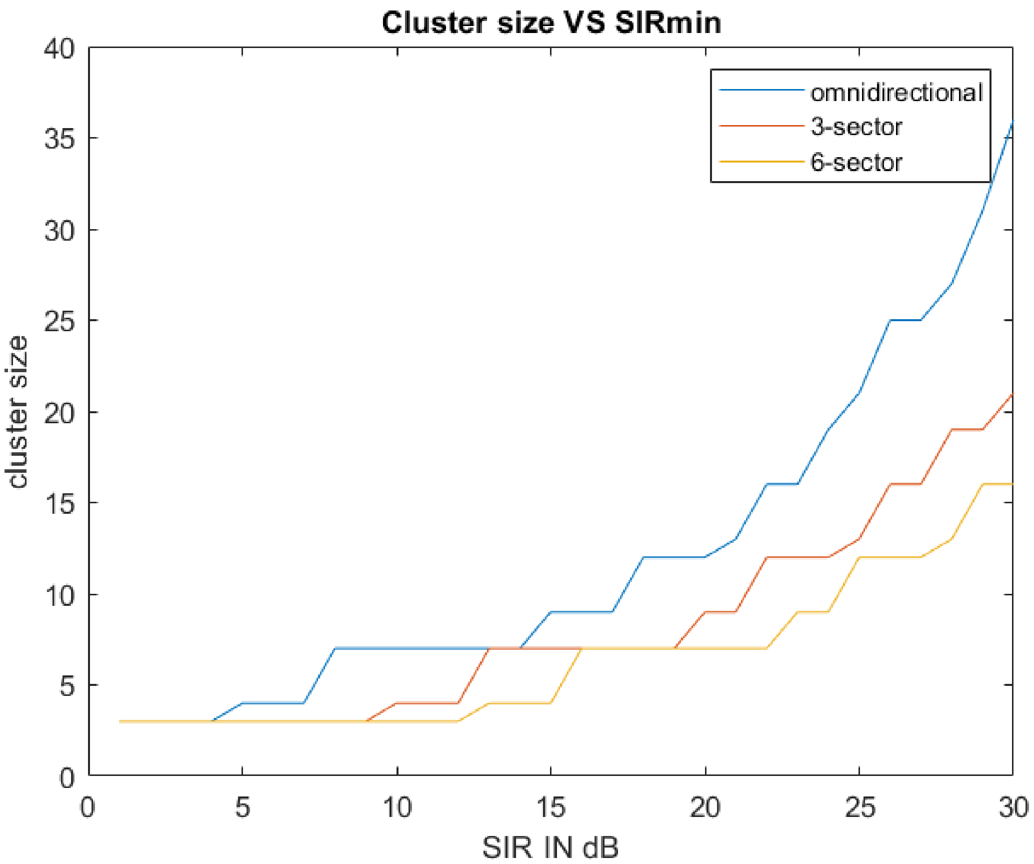
Comment: The further the distance between the receiver and the base station, the lower the received power. This is because the signal travels a longer path and experiences more loss.

Part B

1) Plot the cluster size versus $SIRmin$ with range from $1dB$ to $30\ dB$.

MATLAB CODE

```
%% PART B
%% 1)Plot the cluster size versus SIRmin with range from 1dB to 30 dB.
clear;
clc;
SIRmin = 1:1:30;
n=4;
for i=1:1:30
    [N1(i)]=calClusterSize(6,i,n);
    [N2(i)]=calClusterSize(2,i,n);
    [N3(i)]=calClusterSize(1,i,n);
end
plot(SIRmin,N1,SIRmin,N2,SIRmin,N3)
title('Cluster size VS SIRmin');
xlabel("SIR IN dB");
legend('omnidirectional','3-sector','6-sector')
ylabel("cluster size");
```



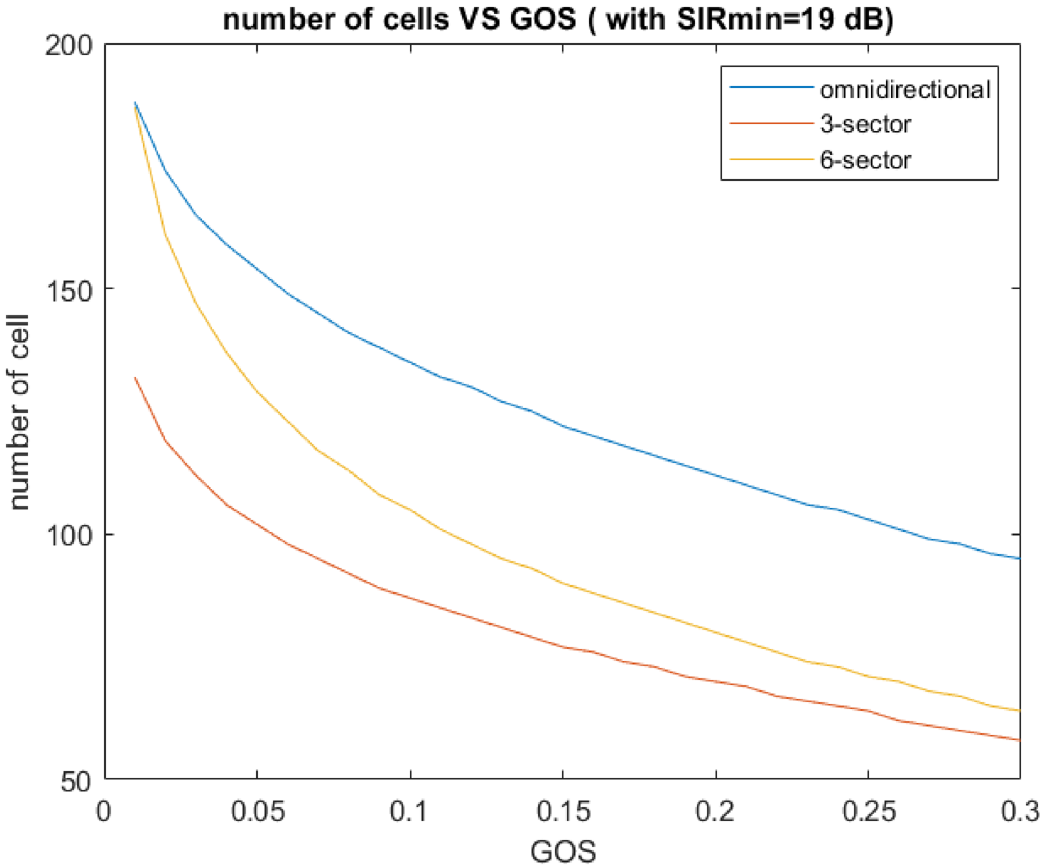
Comment: As the cluster size increases, the distance between co-channels increases, which decreases interference and increases SIR. For the same SIR, the cluster size decreases as the number of sectors per cell increases. This is because each sector will cover a different channel, so less cluster size is needed to cover the same area.

2) For $SIRmin$ = 19dB & user density= 1400 $users/km2$

MATLAB CODE

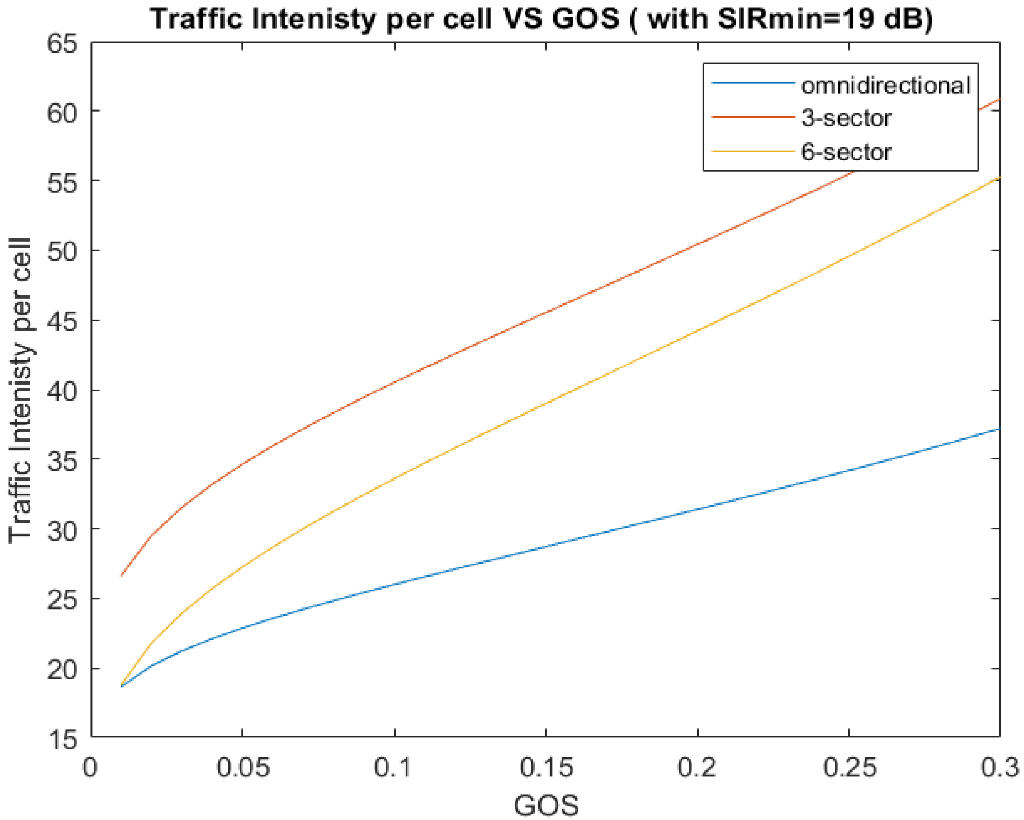
```
%% 2)  For SIRmin = 19dB & user density= 1400 users/km2
%% (i)  Plot the number of cells versus GOS (1% to 30%).
%% (ii) Plot the traffic intensity per cell versus GOS (1% to 30%).
clear;
clc;
userDensity=1400;
cityArea=100;
SIRmin=19;
Au=0.025;
n=4;
S=340;
GOS = 0.01:0.01:0.3;
[N1] = calClusterSize(6,SIRmin,n);
[N2] = calClusterSize(2,SIRmin,n);
[N3] = calClusterSize(1,SIRmin,n);
k=1;
for i=0.01:0.01:0.3
    [trafficIntensityPerSector1(k),trafficIntensityPerCell1(k)] =
calTrafficIntensity(S,N1,i,1);
    [trafficIntensityPerSector2(k),trafficIntensityPerCell2(k)] =
calTrafficIntensity(S,N2,i,3);
    [trafficIntensityPerSector3(k),trafficIntensityPerCell3(k)] =
calTrafficIntensity(S,N3,i,6);
    k = k + 1;
end
for i=1:length(GOS)
    [numCells1(i)] =
calNumCells(Au,trafficIntensityPerCell1(i),userDensity,cityArea);
    [numCells2(i)] =
calNumCells(Au,trafficIntensityPerCell2(i),userDensity,cityArea);
    [numCells3(i)] =
calNumCells(Au,trafficIntensityPerCell3(i),userDensity,cityArea);
end
figure
plot(GOS,numCells1,GOS,numCells2,GOS,numCells3)
title('number of cells VS GOS ( with SIRmin=19 dB) ');
legend('omnidirectional','3-sector','6-sector')
xlabel("GOS");
ylabel("number of cell");
figure
plot(GOS,trafficIntensityPerCell1,GOS,trafficIntensityPerCell2,GOS,trafficIntens
ityPerCell3)
title('Traffic Intenisty per cell VS GOS ( with SIRmin=19 dB) ');
legend('omnidirectional','3-sector','6-sector')
xlabel("GOS");
ylabel("Traffic Intenisty per cell");
```

(i) Plot the number of cells versus GOS (1% to 30%).



Comment: The graph shows that as the probability of blocking increases, the number of cells decreases. This is reasonable because as the number of cells decreases, the area covered by each cell increases, which means that there are more users competing for each cell's resources.

(ii) Plot the traffic intensity per cell versus GOS (1%to 30%).



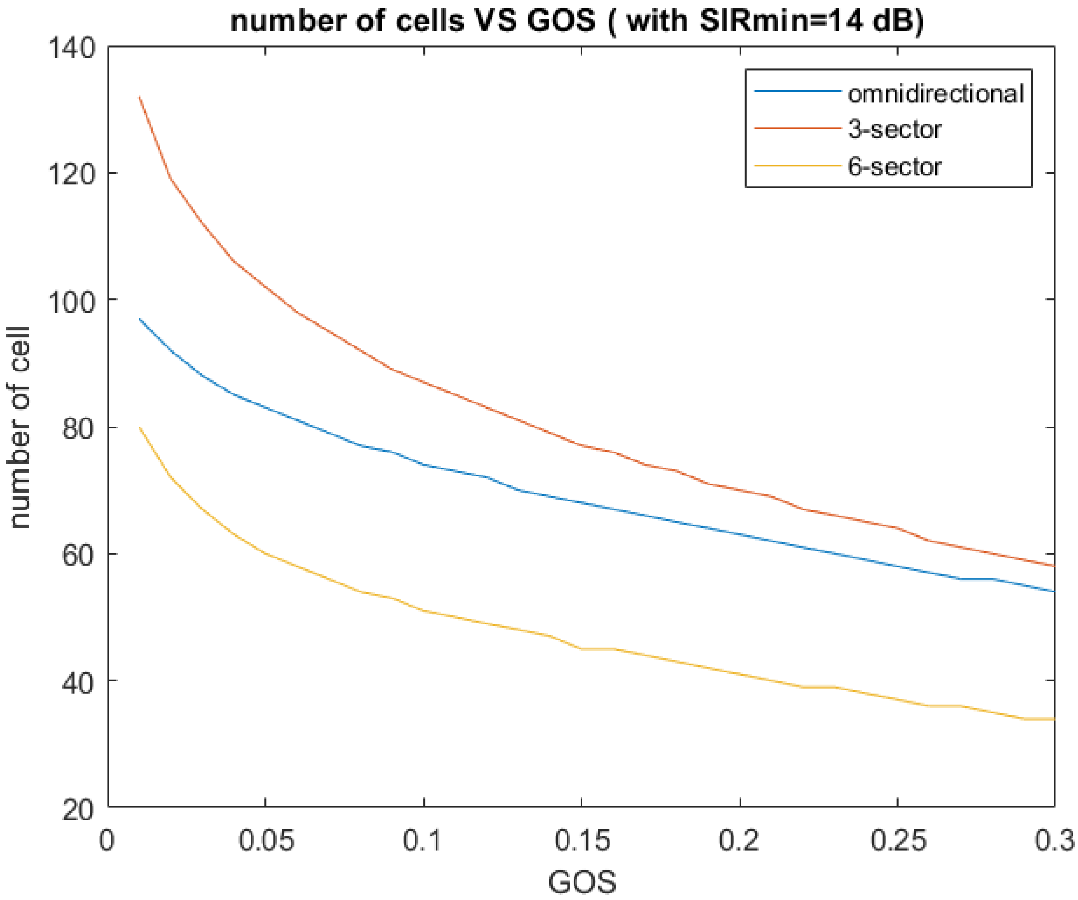
Comment: The graph shows that as traffic intensity increases, so does the blocking probability. This means that when there is more traffic, there is a greater chance that a call will be blocked due to a lack of available resources.

3) At $SIRmin$ = 14dB & user density= 1400 $users/km2$,

MATLAB CODE

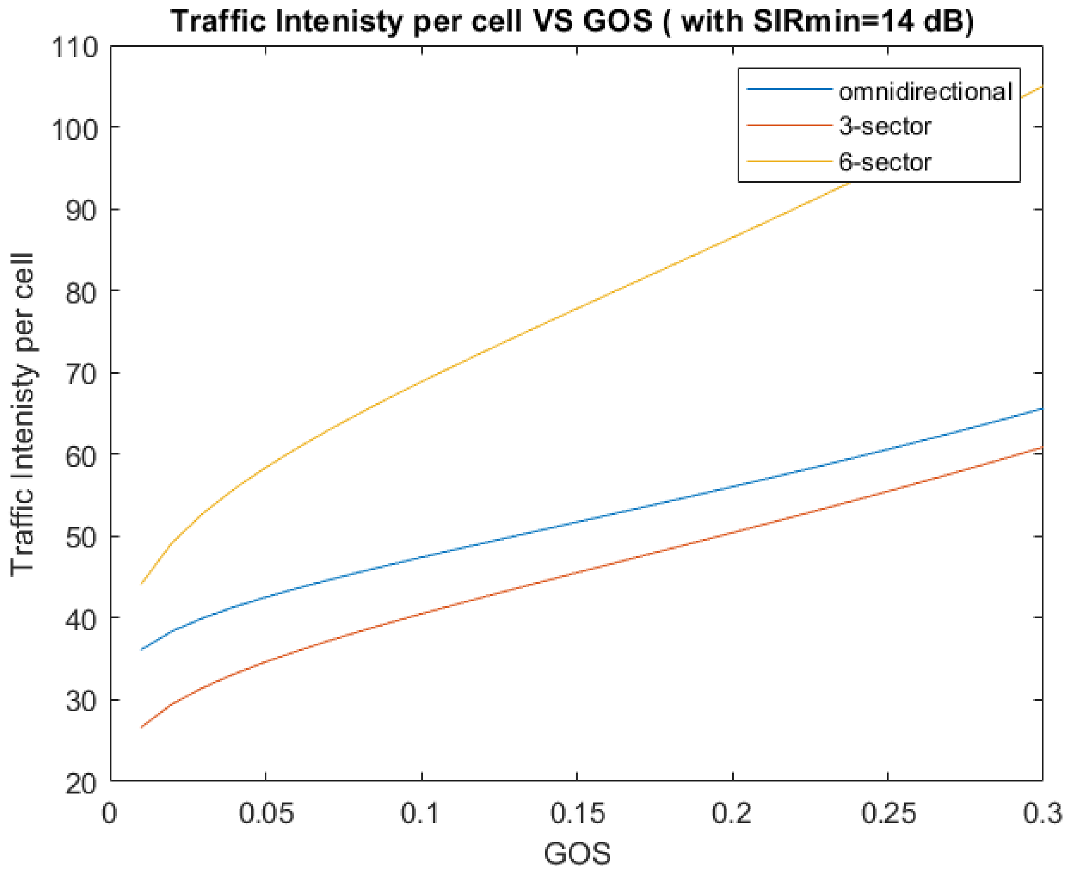
```
%%  3)  At SIRmin = 14dB & user density= 1400 users/km2
%% (i)  Plot the number of cells versus GOS (1% to 30%).
%% (ii) Plot the traffic intensity per cell versus GOS (1% to 30%).
clear;
clc;
userDensity=1400;
cityArea=100;
SIRmin=14;
Au=0.025;
n=4;
S=340;
GOS = 0.01:0.01:0.3;
[N1] = calClusterSize(6,SIRmin,n);
[N2] = calClusterSize(2,SIRmin,n);
[N3] = calClusterSize(1,SIRmin,n);
k=1;
for i=0.01:0.01:0.3
    [trafficIntensityPerSector1(k),trafficIntensityPerCell1(k)] =
calTrafficIntensity(S,N1,i,1);
    [trafficIntensityPerSector2(k),trafficIntensityPerCell2(k)] =
calTrafficIntensity(S,N2,i,3);
    [trafficIntensityPerSector3(k),trafficIntensityPerCell3(k)] =
calTrafficIntensity(S,N3,i,6);
    k = k + 1;
end
for i=1:length(GOS)
    [numCells1(i)] =
calNumCells(Au,trafficIntensityPerCell1(i),userDensity,cityArea);
    [numCells2(i)] =
calNumCells(Au,trafficIntensityPerCell2(i),userDensity,cityArea);
    [numCells3(i)] =
calNumCells(Au,trafficIntensityPerCell3(i),userDensity,cityArea);
end
figure
plot(GOS,numCells1,GOS,numCells2,GOS,numCells3)
title('number of cells VS GOS ( with SIRmin=14 dB) ');
legend('omnidirectional','3-sector','6-sector')
xlabel("GOS");
ylabel("number of cell");
figure
plot(GOS,trafficIntensityPerCell1,GOS,trafficIntensityPerCell2,GOS,trafficIntens
ityPerCell3)
title('Traffic Intenisty per cell VS GOS ( with SIRmin=14 dB) ');
legend('omnidirectional','3-sector','6-sector')
xlabel("GOS");
ylabel("Traffic Intenisty per cell");
```

(i) Plot the number of cells versus GOS (1% to 30%).



<u>Comment:</u> The graph shows that as the probability of blocking increases, the number of cells decreases. This is reasonable because as the number of cells decreases, the area covered by each cell increases, which means that there are more users competing for each cell's resources.

(ii) Plot the traffic intensity per cell versus GOS (1% to 30%).



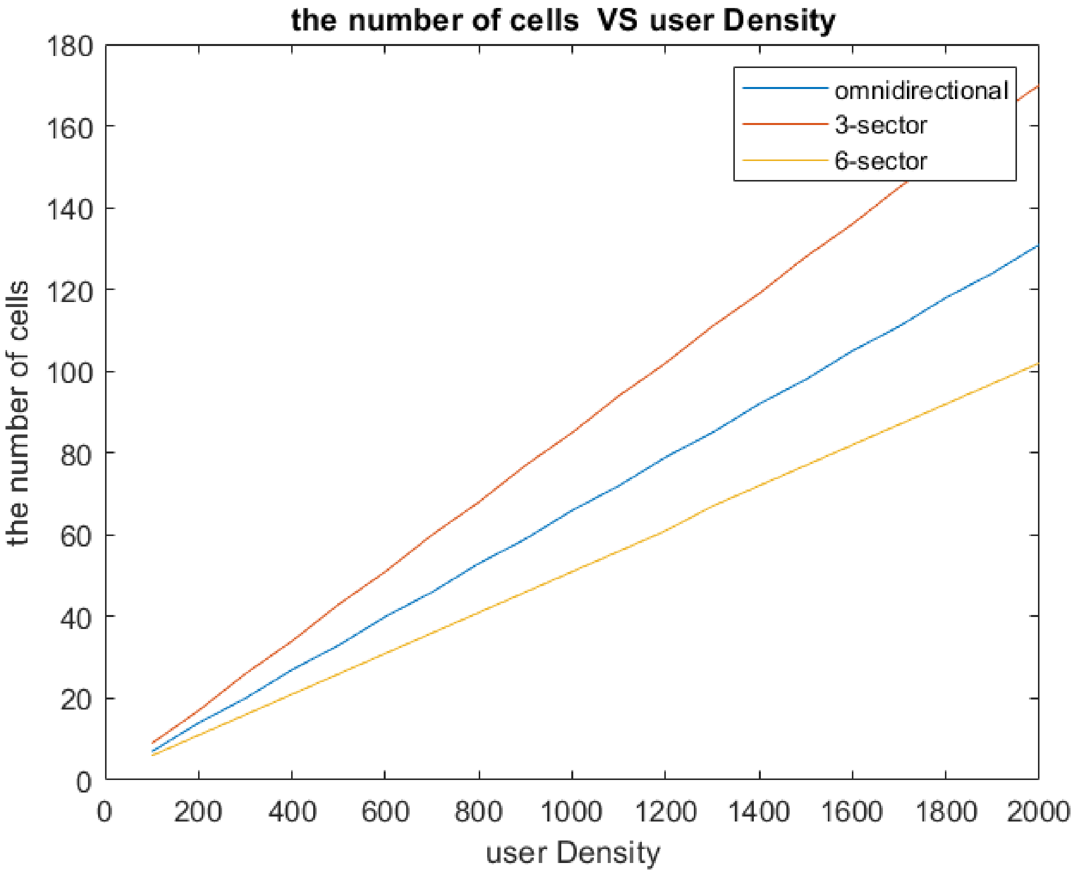
<u>Comment:</u> The graph shows that as traffic intensity increases, so does the blocking probability. This means that when there is more traffic, there is a greater chance that a call will be blocked due to a lack of available resources.

4) At $SIRmin$ = 14dB & GOS= 2%,

MATLAB CODE

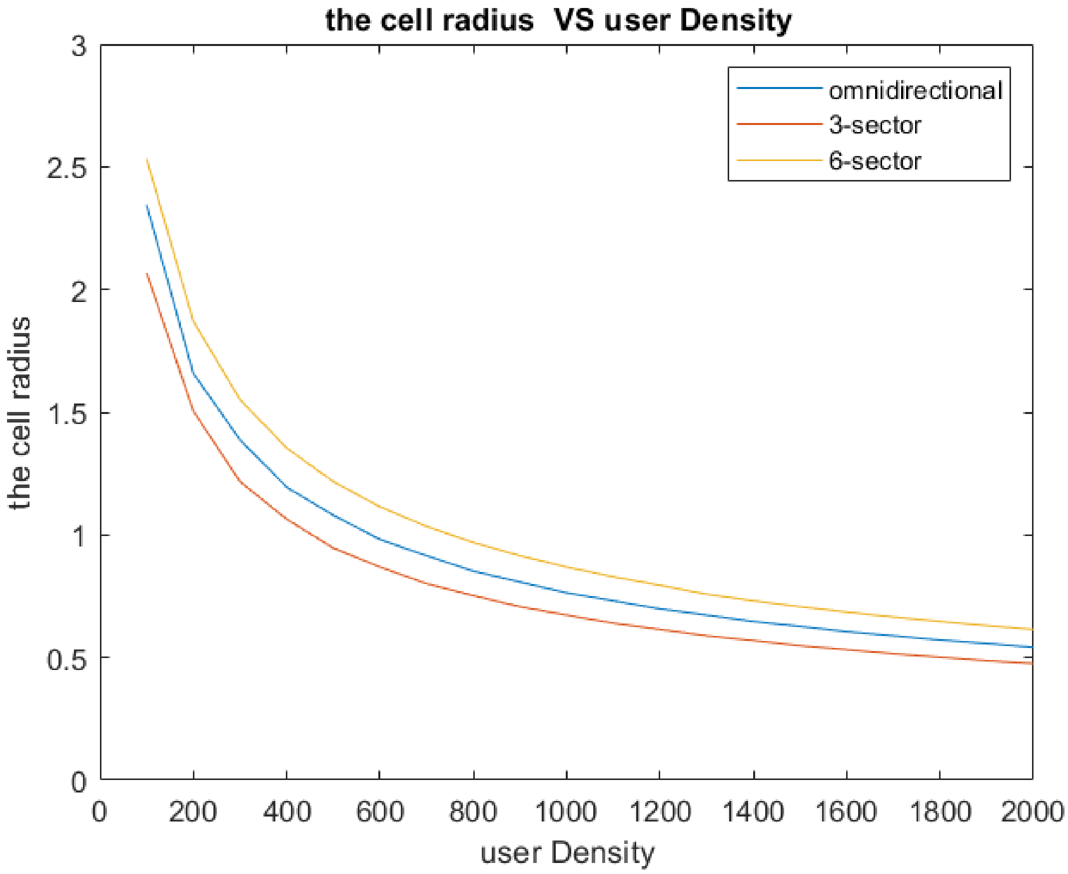
```
%% 4)  At SIRmin = 14dB & GOS= 2%,
%% (i) Plot the number of cells versus user density(100 to 2000 users/km2 ).
%% (ii) Plot the cell radius versus user density (100 to 2000 users/km2 ).
clear;
clc;
userDensity=100:100:2000;
cityArea=100;
SIRmin=14;
Au=0.025;
n=4;
S=340;
GOS = 0.02;
[N1] = calClusterSize(6,SIRmin,n);
[N2] = calClusterSize(2,SIRmin,n);
[N3] = calClusterSize(1,SIRmin,n);
[trafficIntensityPerSector1,trafficIntensityPerCell1] =
calTrafficIntensity(S,N1,GOS,1);
[trafficIntensityPerSector2,trafficIntensityPerCell2] =
calTrafficIntensity(S,N2,GOS,3);
[trafficIntensityPerSector3,trafficIntensityPerCell3] =
calTrafficIntensity(S,N3,GOS,6);
k=1;
for i=100:100:2000
    [numCells1(k)] = calNumCells(Au,trafficIntensityPerCell1,i,cityArea);
    [numCells2(k)] = calNumCells(Au,trafficIntensityPerCell2,i,cityArea);
    [numCells3(k)] = calNumCells(Au,trafficIntensityPerCell3,i,cityArea);
    k = k + 1;
end
for i= 1:length(userDensity)
[cellRadius1(i)] = calCellRadius(cityArea,numCells1(i));
[cellRadius2(i)] = calCellRadius(cityArea,numCells2(i));
[cellRadius3(i)] = calCellRadius(cityArea,numCells3(i));
end
figure
plot(userDensity,numCells1,userDensity,numCells2,userDensity,numCells3)
title('the number of cells  VS user Density  ');
legend('omnidirectional','3-sector','6-sector')
xlabel("user Density");
ylabel("the number of cells");
figure
plot(userDensity,cellRadius1,userDensity,cellRadius2,userDensity,cellRadius3
)
title('the cell radius  VS user Density  ');
legend('omnidirectional','3-sector','6-sector')
xlabel("user Density");
ylabel("the cell radius");
```

(i) Plot the number of cells versus user density (100 to 2000 $users/km2$).



<u>Comment:</u> The number of cells in a cellular network is directly related to the user density. In areas with a high user density, more cells are needed to provide adequate coverage and capacity.

(ii) Plot the cell radius versus user density (100 to 2000 $users/km2$).



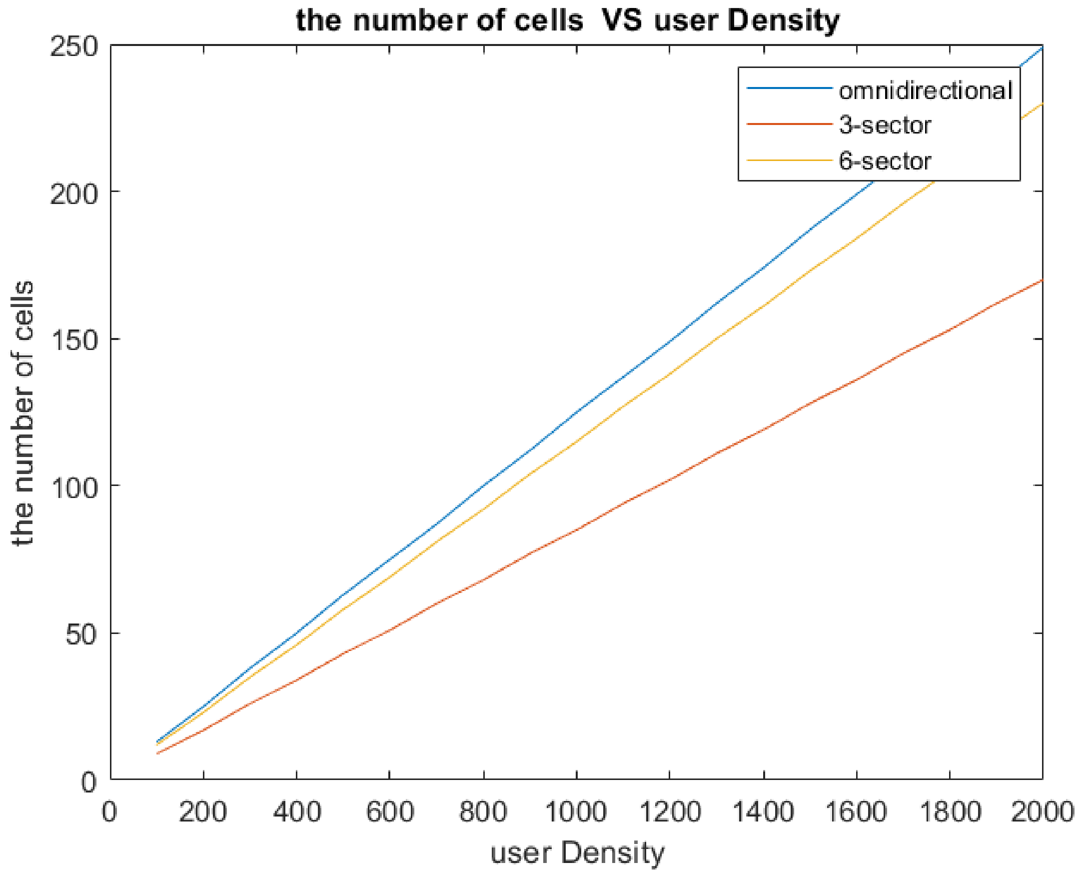
<u>Comment:</u> There is an optimal user density for each cell radius. This is the user density at which the cell radius is the smallest while still providing adequate coverage and capacity for the users.

5) At $SIRmin$ = 19dB & GOS= 2%,

MATLAB CODE

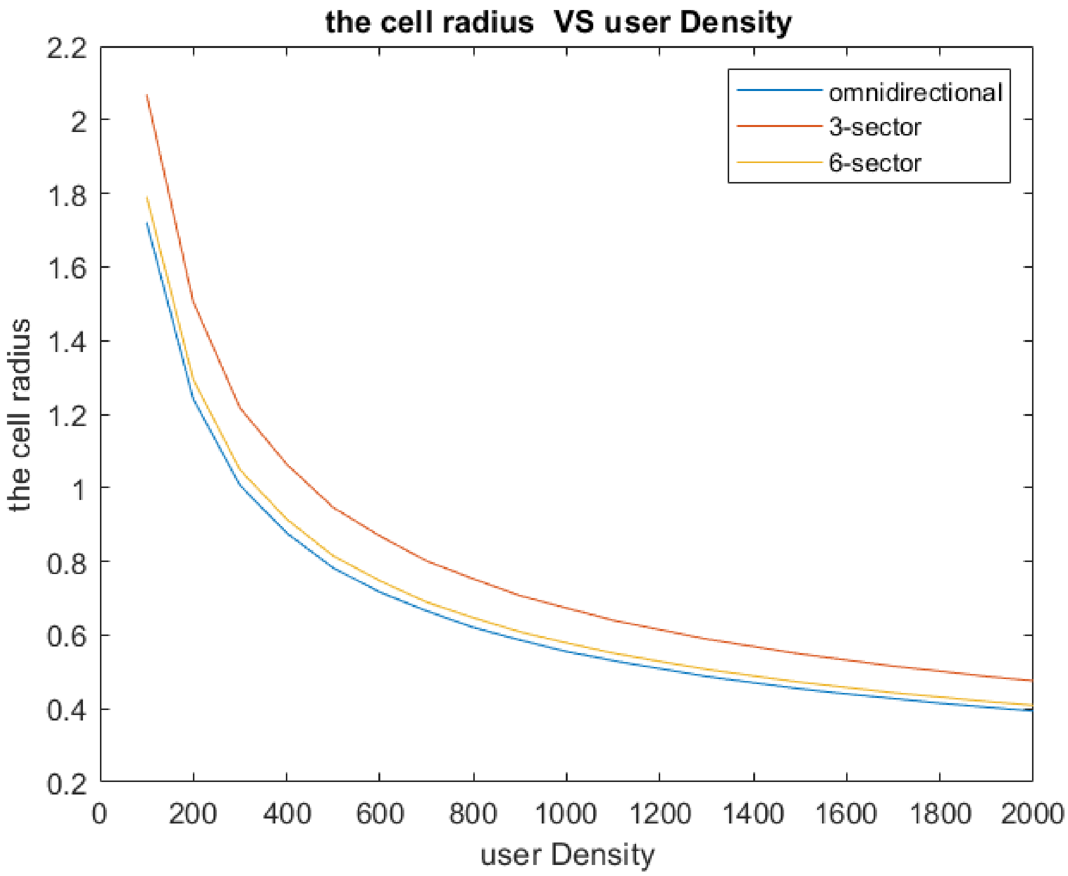
```
%% 4)   At SIRmin = 19dB & GOS= 2%,
%% (i) Plot the number of cells versus user density(100 to 2000 users/km2 ).
%% (ii) Plot the cell radius versus user density (100 to 2000 users/km2 ).
clear;
clc;
userDensity=100:100:2000;
cityArea=100;
SIRmin=19;
Au=0.025;
n=4;
S=340;
GOS = 0.02;
[N1] = calClusterSize(6,SIRmin,n);
[N2] = calClusterSize(2,SIRmin,n);
[N3] = calClusterSize(1,SIRmin,n);
[trafficIntensityPerSector1,trafficIntensityPerCell1] =
calTrafficIntensity(S,N1,GOS,1);
[trafficIntensityPerSector2,trafficIntensityPerCell2] =
calTrafficIntensity(S,N2,GOS,3);
[trafficIntensityPerSector3,trafficIntensityPerCell3] =
calTrafficIntensity(S,N3,GOS,6);
k=1;
for i=100:100:2000
    [numCells1(k)] = calNumCells(Au,trafficIntensityPerCell1,i,cityArea);
    [numCells2(k)] = calNumCells(Au,trafficIntensityPerCell2,i,cityArea);
    [numCells3(k)] = calNumCells(Au,trafficIntensityPerCell3,i,cityArea);
    k = k + 1;
end
for i= 1:length(userDensity)
[cellRadius1(i)] = calCellRadius(cityArea,numCells1(i));
[cellRadius2(i)] = calCellRadius(cityArea,numCells2(i));
[cellRadius3(i)] = calCellRadius(cityArea,numCells3(i));
end
figure
plot(userDensity,numCells1,userDensity,numCells2,userDensity,numCells3)
title('the number of cells  VS user Density  ');
legend('omnidirectional','3-sector','6-sector')
xlabel("user Density");
ylabel("the number of cells");
figure
plot(userDensity,cellRadius1,userDensity,cellRadius2,userDensity,cellRadius3
)
title('the cell radius  VS user Density  ');
legend('omnidirectional','3-sector','6-sector')
xlabel("user Density");
ylabel("the cell radius");
```

(i) Plot the number of cells versus user density (100 to 2000 $users/km2$).



<u>Comment:</u> The number of cells in a cellular network is directly related to the user density. In areas with a high user density, more cells are needed to provide adequate coverage and capacity.

(ii) Plot the cell radius versus user density (100 to 2000 $users/km2$).



<u>Comment:</u> There is an optimal user density for each cell radius. This is the user density at which the cell radius is the smallest while still providing adequate coverage and capacity for the users.