Experiment No-1:-Frequency reuse

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim

To understand the cellular network frequency reuse concept fulfilling the following objectives:

1. **Finding the co-channel cells for a particular cell.**
2. **Finding the cell clusters within certain geographic area.**

A.2. Objectives: To understand basics of cellular system

A.3. Outcomes: Students will able to carry out simulation of frequency reuse, hidden/exposed terminal.(LO-3)

A.4 Theory:

- In mobile communication systems a slot of a carrier frequency / code in a carrier frequency is a radio resource unit.
- This radio resource unit is assigned to a user in order to support a call/ session. The number of available such radio resources at a base station thus determines the number of users who can be supported in the call.
- Since in wireless channels a signal is "broadcast" i.e. received by all entities therefore one a resource is allocated to a user's it cannot be re-assigned until the user finished the call/ session. Thus the number of users who can be supported in a wireless system is highly limited.
- In order to support a large no. of users within a limited spectrum in a region the concept of frequency re-use is used.
- The signal radiated from the transmitter antenna gets attenuated with increasing distance. At a certain distance the signal strength falls below noise threshold and is no longer identifiable.

- In this region when the signal attenuates below noise floor the same radio resource may be used by another transmission to send different information.
- In term of cellular systems, the same radio resource (frequency) ***can use by two base stations which a sufficient spaced apart.*** In this way **the same frequency gets reused** in a layer- geographic area ***by two or more different base station*** different users simultaneously.

- Now what is important *is to select the set of **base stations** which will use the same set of radio resources/ channel of frequencies or **technically the co- channel cells.***
- In this context the minimum adjacent set cells which use different frequencies each is calls a cluster.
- The cellular concept is the major solution of the problem of spectral congestion and user capacity. Cellular radiorely on an intelligent allocation and channel reuse throughout a large geographical coverage region.

Cellular Frequency Reuse:

- Each cellular base station is allocated a **group of radio channels** to be used within a small geographic area called a cell.
- Base stations in adjacent cells are assigned channel groups which contain completely different channels than neighbouring cells.
- Base station antennas are designed to achieve the desired coverage within a particular cell. **By limiting the coverage area within the boundaries of a cell, the same group of channels may be used to cover different cells that are separated from one another**

by geographic distances large enough to <u>keep interference levels within tolerable limits.</u>

- **The design process of selecting and allocating channel groups for all cellular base stations within a system is called frequency reuse or frequency planning.**

Hexagonal Cell Structure:

In figure 1, cells labelled with the same letter use the same group of channels. The hexagonal cell shape is conceptual and is the simplistic model of the radio coverage for each base station. It has been universally adopted since the hexagon permits easy and manageable analysis of a cellular system. The actual radio coverage of a system is known as the footprint and is determined from old measurements and propagation prediction models. Although the real footprint is amorphous in nature, a regular cell shape is needed for systematic system design and adaptation for future growth.

If a circle is chosen to represent the coverage area of a base station, adjacent circles overlaid upon a map leave gaps or overlapping regions. A square, an equilateral triangle and a hexagon can cover the entire area without overlap and with equal area. A cell must serve the weakest mobiles typically located at the edge of the cell within the foot print. For a given distance between the center of a polygon and its farthest perimeter points, the hexagon has the largest area of the three. Thus, with hexagon, the fewest number of cell scan cover a geographic region and close approximation of a circular radiation pattern that occurs for an omni directional base antenna and free space propagation is possible.

Base station transmitters are situated either at the center of the cell (center-excited cells) or at three of the six cell vertices (edge-excited cells). Normally, omni directional antennas are used in center-exited cells and sectored directional antennas are used in edge-exited cells. Practical system design considerations permit a base station to be positioned up to onefourth the cell radius away from the ideal location.

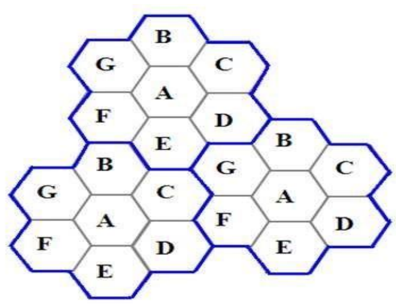
Cell Cluster:

Considering a cellular system that has a total of S duplex radio channels. If each cell is allocated a group of k channels ($k < S$)and if the S channels are divided among N cellsinto unique and disjoint channel groups of same number of channels, then,

$S = kN.$ 6.1

The N cells that collectively use the complete set of available frequencies is called a cluster. If a cluster is replicated M times within the system, the total number of duplex channels or capacity,

$C = MkN= MS.$ 6.2



Frequency reuse concept, Cells with the same letter use the same set of frequencies. A cell cluster is outline in blue color and replicated over the coverage area.

In this example,

The cluster size N = 7 and the frequency reuse factor is 1/7 since each cell contains oneseventh of the total number of available channels.

The capacity is directly proportional to M. The *factor N is called the cluster size and is typically 4, 7 or 12*. If the cluster size N is reduced while the cell size is kept constant, more clusters are required to cover a given area and hence more capacity is achieved from the

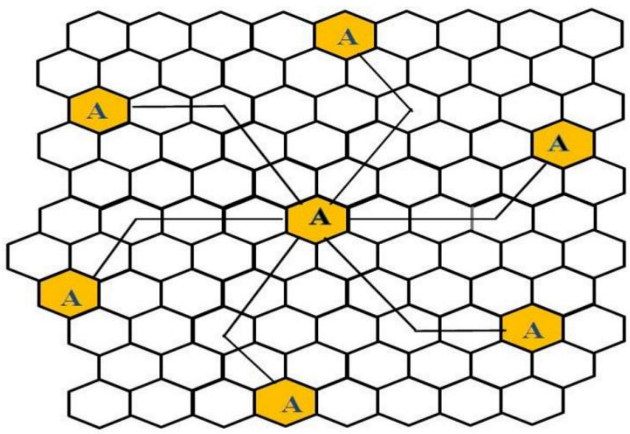
design viewpoint, the smallest possible value of N is desirable to maximize capacity over a

given coverage area. The frequency reuse factor of a cellular system is 1/N,since each cell within a cluster is assigned 1/N of the total available channels in the system.

- **Co-channel Cells:**

A larger cluster size causes the ratio between the cell radius and the distance between cochannel cells to decrease reducing co-channel interference. The value of N is a function of how much interference a mobile or base station can tolerate while maintaining a sufficient quality of communications. Since each hexagonal cell has six equidistant neighbors and the line joining the centers of any cell and each of its neighbors are separated by multiples of 60 degrees, only certain cluster sizes and cell layouts are possible. To connect without gaps between adjacent cells, the geometry of hexagons is such that the number of cells per cluster, N, can only have values that satisfy,

$N = i^2 + ij + j^2$, 6.3



Method of locating co-channel cells in a cellular system. In this figure, N=19(i.e, i =3, j=2).

In this example, N = 19 (i.e., i = 3, j = 2).

Where, i and j are non-negative integers.

To find the nearest co-channel neighbours of a particular cell,

a. move i cells along any chain of hexagons then,

b. turn 60 degrees counter-clockwise and move j cells.

A.6 Steps

Follow the instructions given below to perform the experiments.

➢ **Steps to perform Virtual lab Experiment:**

Virtual Lab: Fading Channel and Mobile Communication

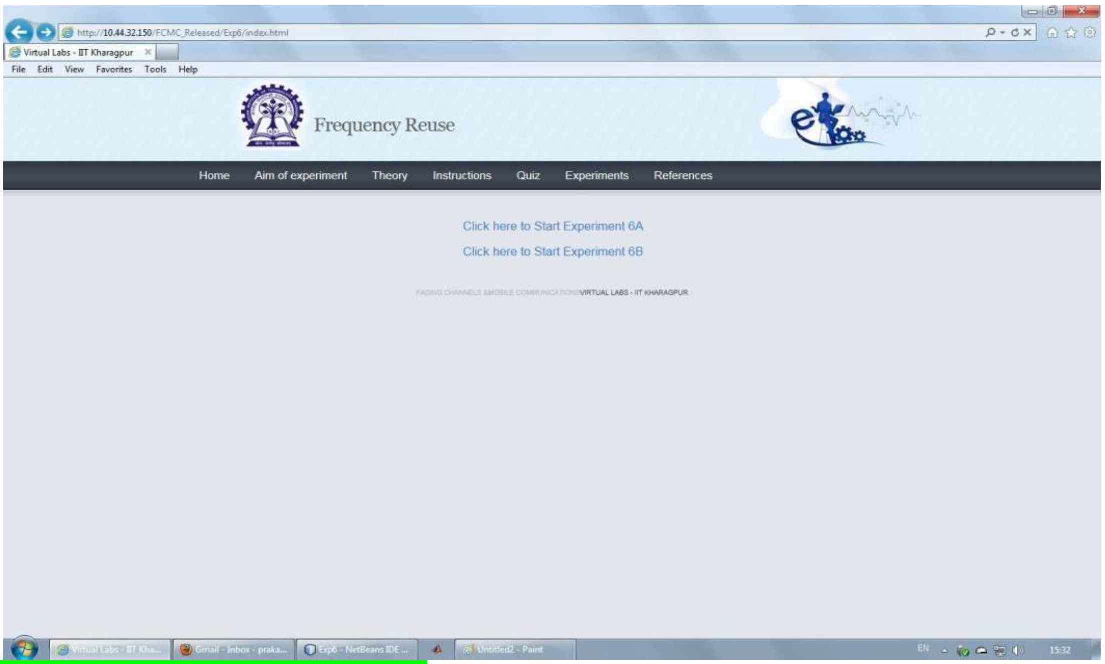
Virtual Lab Link: http://vlabs.iitkgp.ac.in/fcmc/index.html#

Experiment: Frequency Reuse (Co-channel cells and Cell Cluster)

Starting the Experiments:-

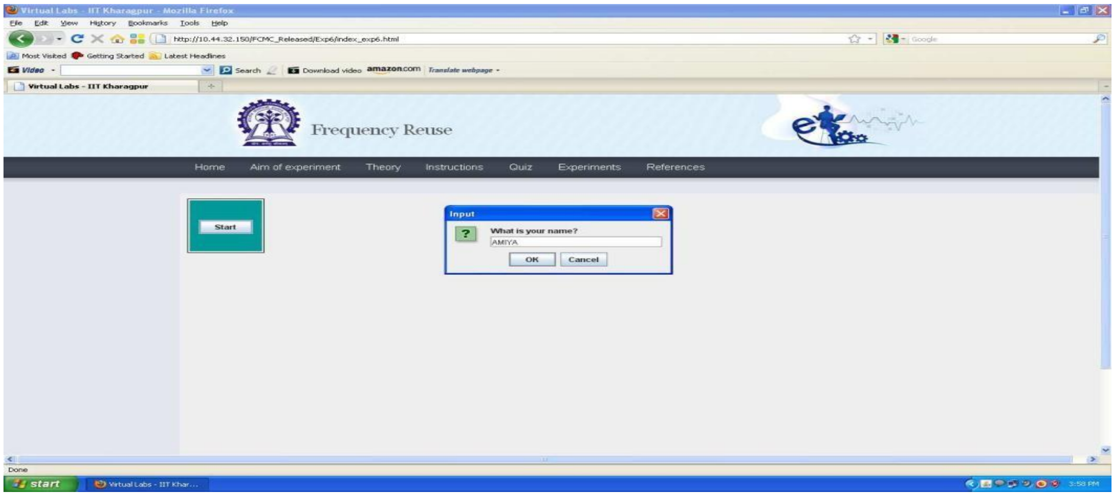
LINK: **http://vlabs.iitkgp.ernet.in/fcmc/exp6/index.html#**

- Step 1: Click on the experiment you want to do by clicking on either `Click here to start Experiment 6A (Co-channel cell)' or `Click here to start Experiment 6B (Cell cluster)'



Performing Experiment 1A:- •

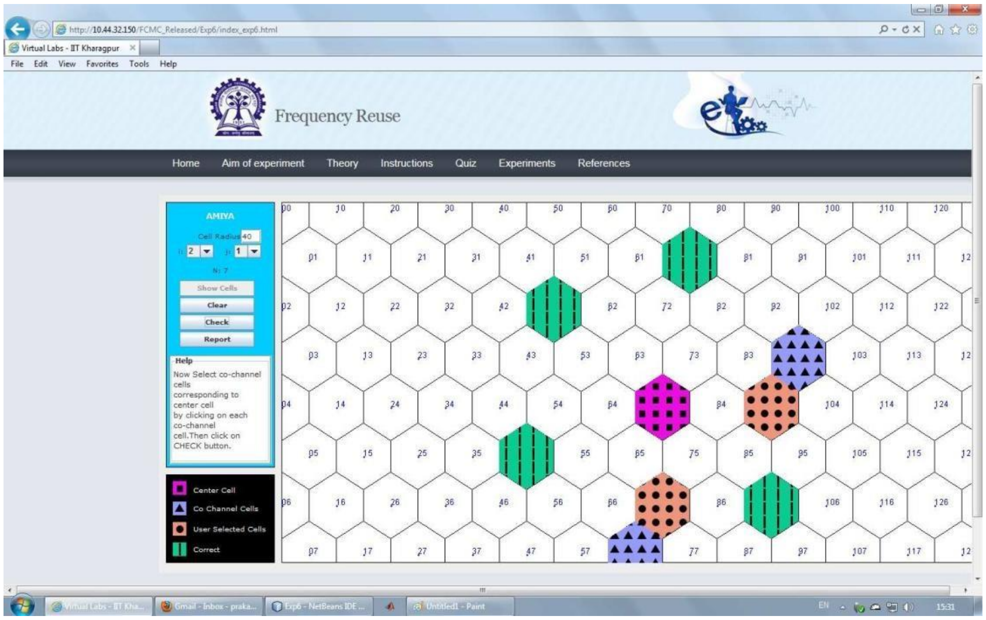
- Step 2: Let Experiment 6A (Co-channel cell) is chosen. Click on the button START. A page appears with a dialogue box asking for your name. Enter your name and click OK.



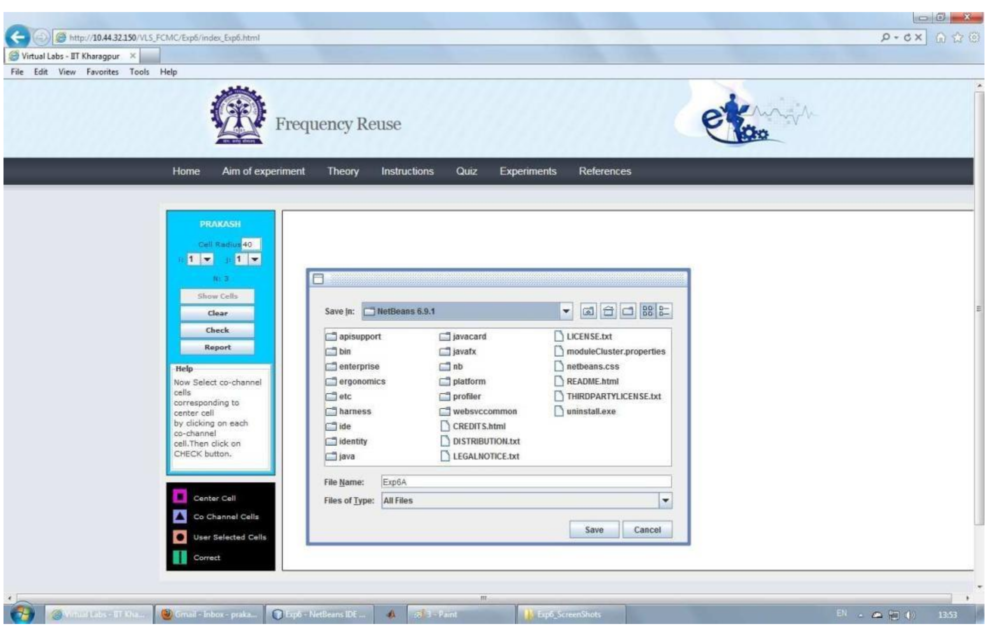
- Step 3: Choose the value of Cell Radius, i andj.
- Step 4: Click on the button Show Cells. For the given parameters, the value of Clustersize N is shown in the LHS of the page and the generated cells are shown on the RHS of the page.

- Step 5: Within the generated cells the center cell is shown in pink colour. Select the Co-channel cells in orange colour for the center cell by finding the Co-channel cells from the formula given in the theory section.

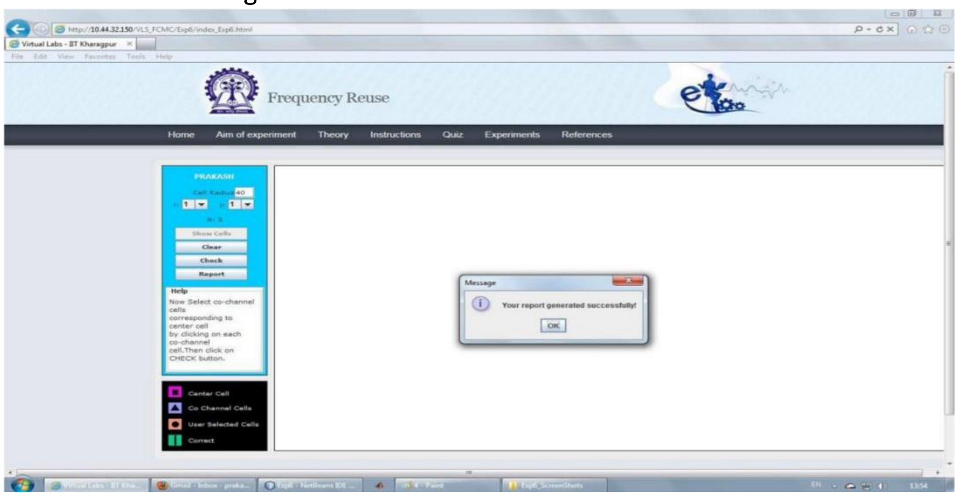
Step 6: Click on the button CHECK to see whether your manually selected Co-channel cells matchwiththecorrectCo-channelcells.Ifyou'remanuallyselectedcellsdonotmatchwiththe correct Co-channel cells then the correct Co-channel cells are displayed in sky blue colour. If your manually selected Co-channel cells match with the correct Co-channel cells then the correct Co-channel cells are over-marked in green colour.



• Step 7: Click on the button REPORT to generate the report of the experiment you have performed.



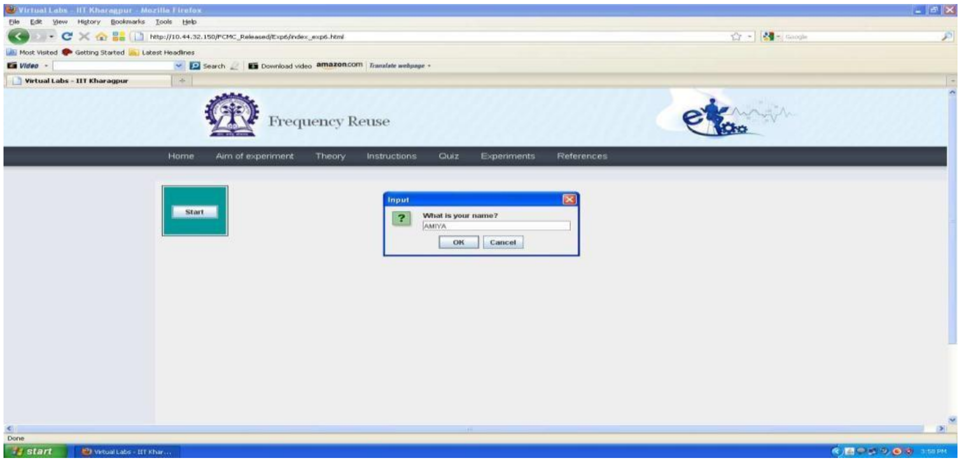
- Step 8: A dialogue box appears. Click on the button Save to save yourreport.
- Step 9: A dialogue box appears with the message that `Your report has generated successfully'. Click on button OK in the dialogue box



- Step 10: Now you can view the pdf report.
- Step 11: You can repeat the experiment by clicking the CLEAR button at the upper corner in the LHS of the page.

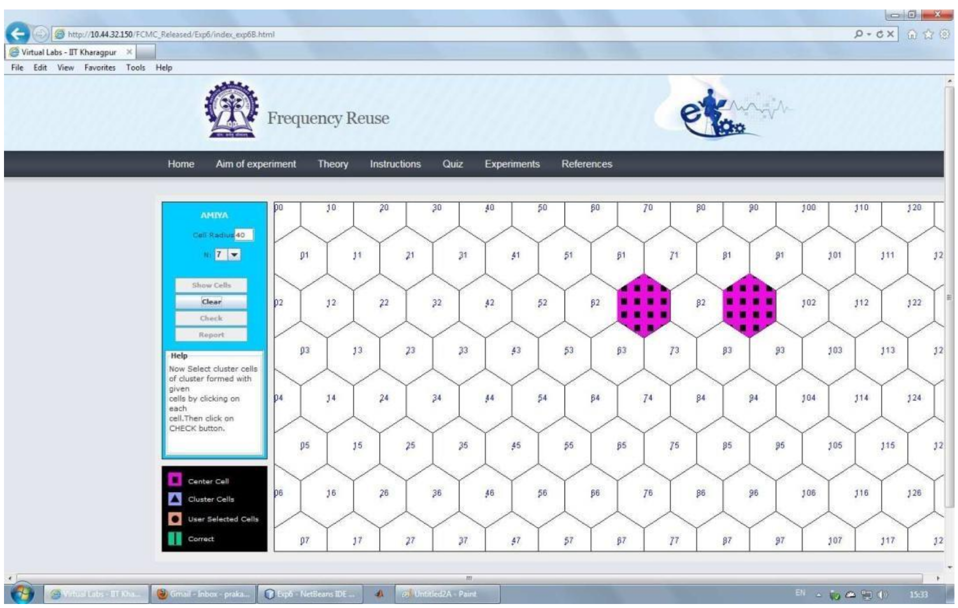
Performing Experiment 1B:-

- Step 12: Let Experiment 6B (Cell cluster) is chosen. Click on the button START. A page appears with a dialogue box asking for your name. Enter your name and click OK.

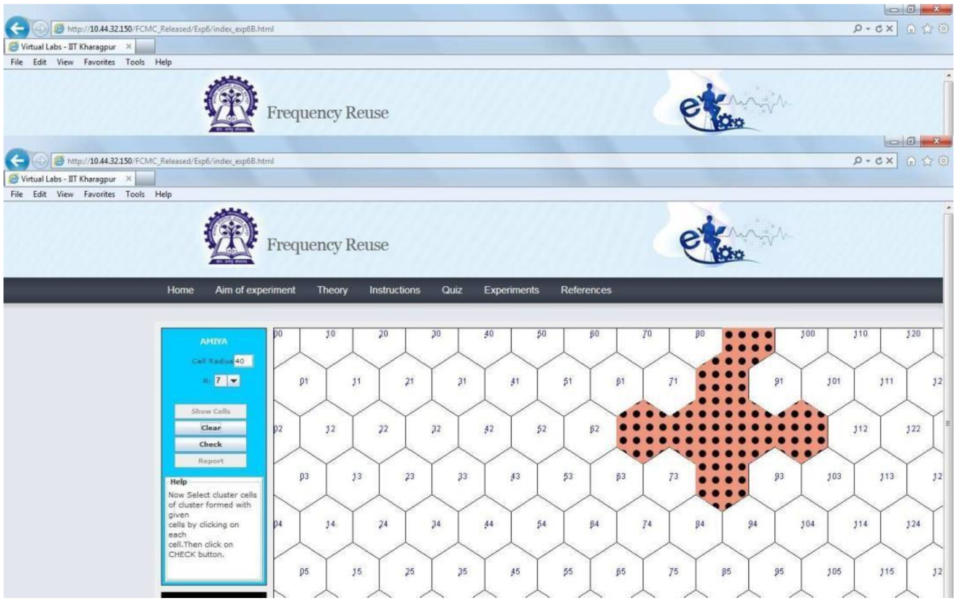


- Step 13: Choose the value of Cell Radius and Cell Cluster.

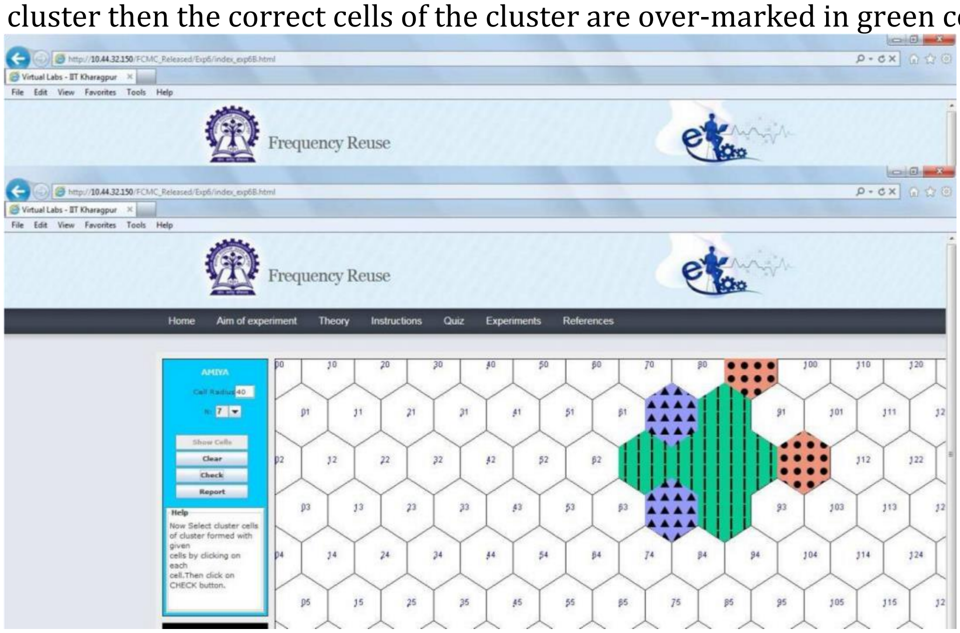
Step 14: Click on the button Show Cells. The generated cells are shown on the RHS of the page.



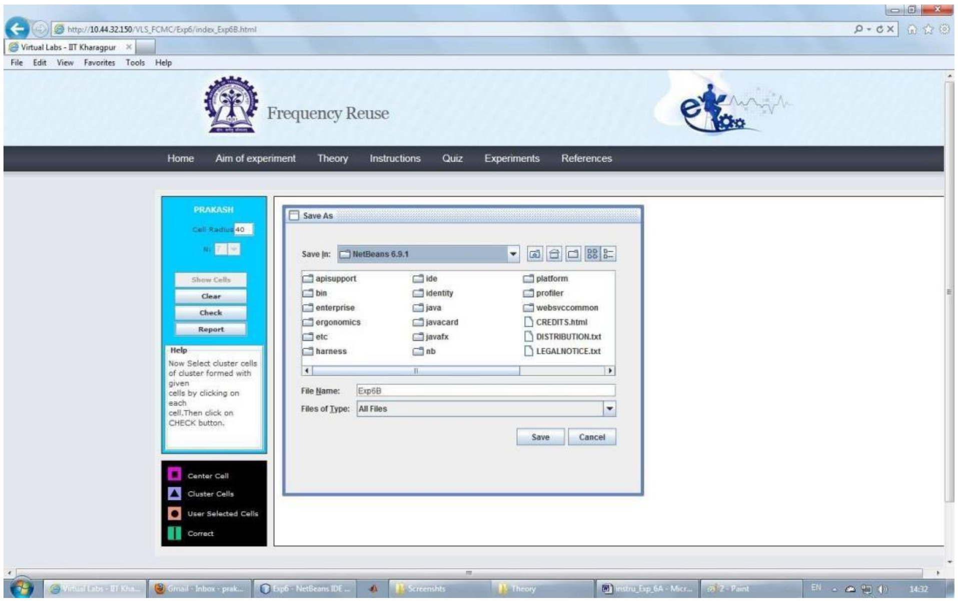
- Step 15: Within the generated cells the two extreme cells within the cell cluster is shown in pink colour. Select other cells within the cell cluster in orange colour.



- Step 16: Click on the button CHECK to see whether your manually selected cluster cells match with the correct cells of the cluster. If your manually selected cells do not match with the correct cells of the cluster then the correct cells of the cluster are displayed in sky blue colour. If the manually selected cells of the cluster match with the correct cells of the cluster then the correct cells of the cluster are over-marked in green colour.

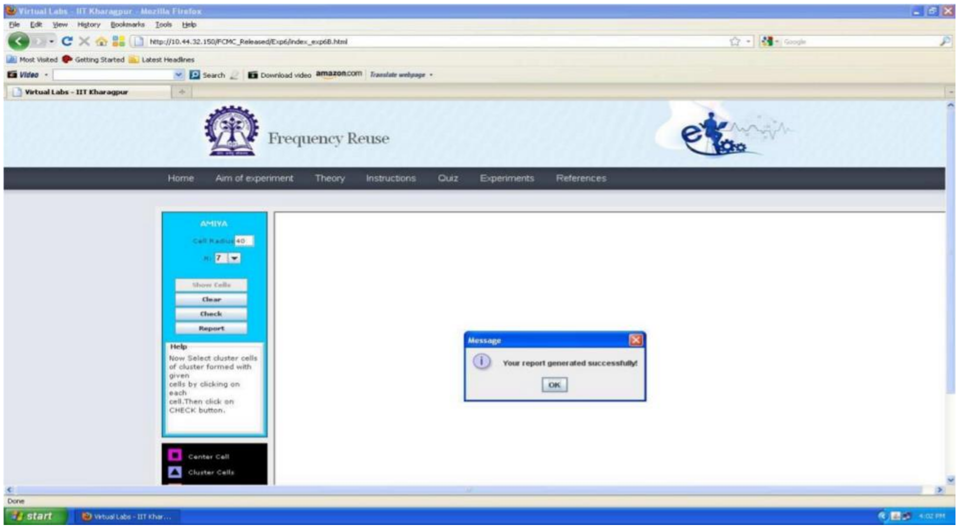


- Step 17: Click on the button REPORT to generate the report of the experiment you have performed.



- Step 18: A dialogue box appears. Click on the button Save to save your report.

- Step 19: A dialogue box appears with the message that `Your report has generated successfully'. Click on button OK in the dialogue box.



- Step 20: Now you can view the pdf report.
- Step 21: You can repeat the experiment by clicking the CLEAR button at the upper corner in LHS of the page.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student:

Co-channel.py:

```
from math import cos, sin, radians, sqrt
from tkinter import *

class Hexagon:
    def __init__(self, parent, x, y, length, color, tags):
        self.parent = parent
        self.x = x
        self.y = y
        self.length = length
        self.color = color
        self.tags = tags
        self.draw_hex()

    def draw_hex(self):
        """Draw a hexagon based on center (x, y) and edge length"""
        coords = []
        for i in range(6):
            angle_rad = radians(60 * i)
            end_x = self.x + self.length * cos(angle_rad)
            end_y = self.y + self.length * sin(angle_rad)
            coords.append((end_x, end_y))
        self.parent.create_polygon(*sum(coords, ()), fill=self.color, outline="black", tags=self.tags)

    def get_center(self):
```

```
        """Return the center coordinates of the hexagon"""
        return self.x, self.y

class FrequencyReuse(Tk):
    def __init__(self, i, j, columns=16, rows=10, edge_len=30):
        super().__init__()
        self.i = i
        self.j = j
        self.N = i**2 + i*j + j**2  # Correct cluster size calculation
        self.columns = columns
        self.rows = rows
        self.edge_len = edge_len
        self.hexagons = {}
        self.first_click = True
        self.selected_hex = None
        self.co_channel_cells = []
        self.canvas = Canvas(self, width=800, height=650, bg="#Add0e1")
        self.canvas.pack()
        self.title(f"Frequency Reuse: Cluster Size {self.N}")
        self.create_grid()
        self.create_textbox()
        self.canvas.bind("<Button-1>", self.call_back)
        self.canvas.bind("<Shift-R>", self.reset_grid)
        self.mainloop()

    def create_grid(self):
        """Create a grid of hexagons"""
        size = self.edge_len
        for c in range(self.columns):
            offset = 0 if c % 2 == 0 else size * sqrt(3) / 2
            for r in range(self.rows):
                x = c * (size * 1.5) + 50
                y = (r * (size * sqrt(3))) + offset + 15
                tag = f"{r},{c}"
                self.hexagons[tag] = Hexagon(self.canvas, x, y, self.edge_len, "#fafafa", tag)

    def create_textbox(self):
        """Create the instruction text box"""
        self.textbox = Label(self.canvas, text="Select a Hexagon", font=("Helvetica", 12), bg="white", width=80)
        self.canvas.create_window(400, 600, window=self.textbox)

    def write_text(self, text):
        """Update text in instruction box"""
        self.textbox.config(text=text)

    def reset_grid(self, event=None):
        """Reset the grid and selection"""
        self.first_click = True
        self.selected_hex = None
        self.co_channel_cells = []
        self.canvas.delete("lines")  # Clear previous lines
        for hexagon in self.hexagons.values():
```

```
            self.canvas.itemconfigure(hexagon.tags, fill="#fafafa")
        self.write_text("Select a Hexagon")

    def find_co_channel_cells(self, start_tag):
        """Find co-channel cells based on (i, j) movement"""
        r_start, c_start = map(int, start_tag.split(","))
        co_cells = [(r_start, c_start)]  # Include original cell

        # Directions to move in the hexagonal grid
        directions = [
            (self.i, self.j),
            (-self.i, -self.j),
            (self.j, -self.i-self.j),
            (-self.j, self.i+self.j),
            (self.i+self.j, -self.j),
            (-self.i-self.j, self.j)
        ]

        for dr, dc in directions:
            r_new, c_new = r_start + dr, c_start + dc
            tag_new = f"{r_new},{c_new}"
            if tag_new in self.hexagons:
                co_cells.append((r_new, c_new))

        return co_cells

    def draw_line(self, tag):
        """Draw a line from the selected hexagon to the correctly guessed co-channel cell"""
        if not self.selected_hex:
            Return

        start_x, start_y = self.hexagons[self.selected_hex].get_center()
        end_x, end_y = self.hexagons[tag].get_center()
        self.canvas.create_line(start_x, start_y, end_x, end_y, fill="blue", width=2, tags="lines")

    def call_back(self, evt):
        """Handle user clicks on hexagons"""
        selected_hex_id = self.canvas.find_closest(evt.x, evt.y)[0]
        selected_tag = self.canvas.gettags(selected_hex_id)[0]

        if self.first_click:
            # First selection
            self.first_click = False
            self.selected_hex = selected_tag
            self.co_channel_cells = self.find_co_channel_cells(selected_tag)

            # Highlight first selection
            self.canvas.itemconfigure(selected_tag, fill="green")
            self.write_text(f"Selected cell {selected_tag}. Now, select co-channel cells.")

        else:
            # Subsequent selections
```

```
            r, c = map(int, selected_tag.split(","))
            if (r, c) in self.co_channel_cells and selected_tag != self.selected_hex:
                self.canvas.itemconfigure(selected_tag, fill="green")
                self.draw_line(selected_tag)  # Draw line for correct selections
                self.write_text(f"Correct! Cell {selected_tag} is a co-channel cell.")
            else:
                self.canvas.itemconfigure(selected_tag, fill="red")
                self.write_text(f"Incorrect! Cell {selected_tag} is not a co-channel cell.")

            if len(self.co_channel_cells) >= 7:  # 1 Original + 6 Co-Cells
                self.write_text("Great! Press Shift-R to restart.")

if __name__ == '__main__':
    print("Enter i & j values. Common (i, j) values are: (1,0), (1,1), (2,0), (2,1), (3,0), (2,2)")
    i = int(input("Enter i: "))
    j = int(input("Enter j: "))
    if i == 0 and j == 0:
        raise ValueError("i & j both cannot be zero")
    elif j > i:
        raise ValueError("Value of j cannot be greater than i")
    else:
        print(f"Cluster size (N) = {i**2 + i*j + j**2}")
        FrequencyReuse(i, j)
```

Cluster.py:

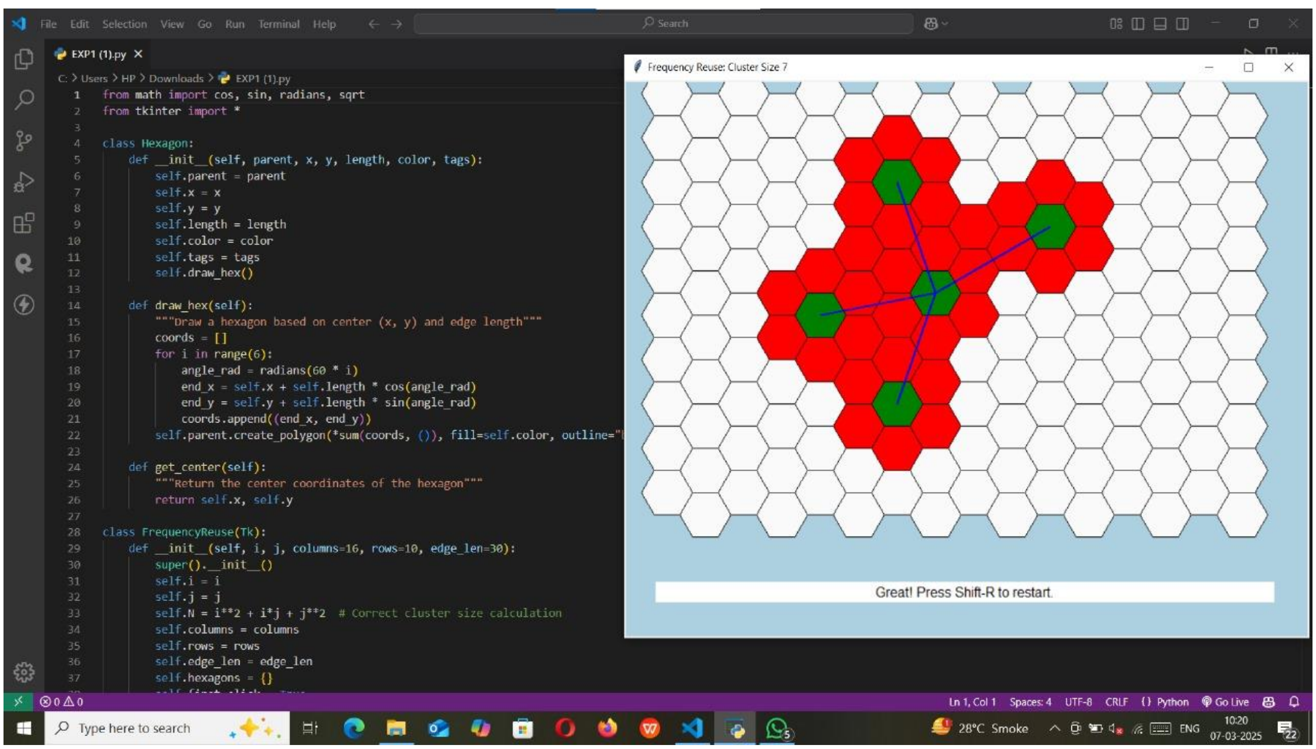
```
import tkinter as tk
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

def hexagon(x, y, size):
    """Generate hexagon coordinates."""
    angles = np.linspace(0, 2 * np.pi, 7)
    return x + size * np.cos(angles), y + size * np.sin(angles)

def plot_clusters(cluster_size):
    """Plot the frequency reuse cell clusters with green and yellow colors."""
    fig, ax = plt.subplots(figsize=(6, 6))
    size = 10
    colors = ["green", "yellow"]  # Only 2 colors for clusters

    cell_count = 1  # Start cell numbering

    for i in range(-cluster_size, cluster_size + 1):
        for j in range(-cluster_size, cluster_size + 1):
            x = i * 1.5 * size
            y = j * np.sqrt(3) * size + (i % 2) * (np.sqrt(3) / 2 * size)
            cluster_id = (i + j) % 2  # Alternating clusters
            color = colors[cluster_id]
```

```
            # Draw hexagon
            ax.fill(*hexagon(x, y, size), color=color, edgecolor='black')

            # Display cell number at center of hexagon
            text_color = "white" if color == "green" else "black"
            ax.text(x, y, str(cell_count), ha='center', va='center', fontsize=10, color=text_color, fontweight='bold')

            cell_count += 1  # Increment cell count

    ax.set_xlim(-cluster_size * 15, cluster_size * 15)
    ax.set_ylim(-cluster_size * 15, cluster_size * 15)
    ax.set_aspect('equal')
    ax.axis("off")
    return fig

def update_plot():
    """Update the plot based on user input."""
    try:
        cluster_size = int(cluster_size_var.get())
        if cluster_size < 1:
            error_label.config(text="Please enter a positive integer")
            Return

        for widget in plot_frame.winfo_children():
            widget.destroy()

        fig = plot_clusters(cluster_size)
        new_canvas = FigureCanvasTkAgg(fig, master=plot_frame)
        new_canvas.get_tk_widget().pack()
        new_canvas.draw()

        error_label.config(text="")
    except ValueError:
        error_label.config(text="Please enter a valid number")

# GUI Setup
root = tk.Tk()
root.title("Cell Clustering - Frequency Reuse")

frame = ttk.Frame(root, padding=10)
frame.pack()

# Cluster Size Input
cluster_size_var = tk.StringVar(value="3")
ttk.Label(frame, text="Cluster Size:").grid(row=0, column=0)
ttk.Entry(frame, textvariable=cluster_size_var, width=5).grid(row=0, column=1)
ttk.Button(frame, text="Update", command=update_plot).grid(row=0, column=2)

error_label = ttk.Label(frame, text="", foreground="red")
error_label.grid(row=1, column=0, columnspan=3)
```

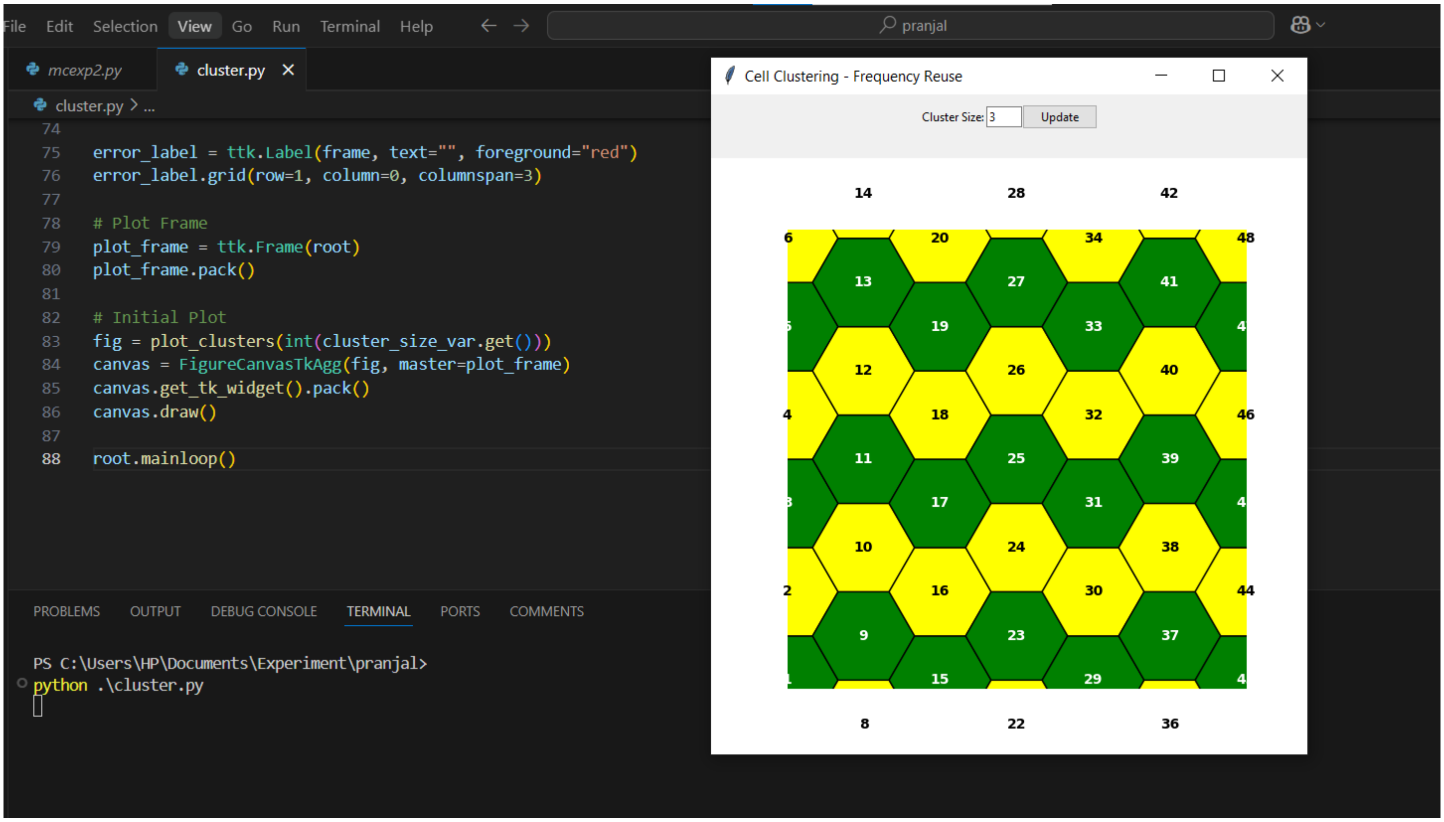
```
# Plot Frame
plot_frame = ttk.Frame(root)
plot_frame.pack()

# Initial Plot
fig = plot_clusters(int(cluster_size_var.get()))
canvas = FigureCanvasTkAgg(fig, master=plot_frame)
canvas.get_tk_widget().pack()
canvas.draw()

root.mainloop()
```

B.2 Input and Output:

Co-channel:



Cluster:



B.3 Observations and learning:

Co-channel Cells Identification:

1. Cells using the same frequency were identified based on the frequency reuse factor (1/N1/N1/N), where NNN is the cluster size.
2. The co-channel cells were separated by a sufficient distance to minimize interference.
3. Example: For a cluster size N=7N = 7N=7, co-channel cells formed a repeating pattern, maintaining adequate spacing.

Cluster Formation:

1. The entire geographic area was divided into clusters of cells.
2. Each cluster used all available frequencies (S=kNS = kNS=kN) without overlap between adjacent clusters.

Hexagonal Cell Structure:

1. The hexagonal layout was observed to provide efficient coverage without gaps or overlaps.
2. Co-channel cells were located using the formula N=i2+ij+j2N = i^2 + ij + j^2N=i2+ij+j2, ensuring equal spacing between them.

Frequency Reuse Efficiency:

1. Reuse of frequencies increased the system's capacity to accommodate more users while limiting interference.
2. Smaller cluster sizes (N=4N = 4N=4 or N=7N = 7N=7) allowed higher frequency reuse but required careful planning to control interference.

Impact of Cluster Size:

1. Larger cluster sizes reduced co-channel interference but also decreased capacity.

Smaller cluster sizes provided higher capacity but required tighter interference management.

B.4 Conclusion:

The experiment demonstrates that frequency reuse is an essential concept for efficient spectrum utilization in cellular networks. By allocating the same frequency to co-channel cells separated by sufficient distance, interference is minimized while maximizing user capacity. The hexagonal cell structure ensures optimal coverage without overlaps or gaps, making it ideal for network design. Smaller cluster sizes increase capacity but require careful interference management. Overall, frequency reuse enables scalable, high-capacity cellular systems, forming the foundation of modern communication technologies like GSM, LTE, and 5G.

B.5 Question of Curiosity

1. With Example explain Co-channel cells

Definition:

Co-channel cells are cells in a cellular network that use the same frequency for communication but are separated by a sufficient distance to prevent interference. These cells are part of the frequency reuse concept, which allows the reuse of limited frequencies in non-adjacent cells to increase system capacity.

Example:

Suppose a cellular system has a total of 21 frequencies, and each cell in a cluster of N=7N = 7 uses 3 frequencies.

These 7 cells form a cluster, and frequencies are assigned as:

Cell A: Frequencies (f1, f2, f3)

Cell B: Frequencies (f4, f5, f6)

Cell C: Frequencies (f7, f8, f9), and so on.

Now, the same frequencies (f1, f2, f3) can be reused in another cell, labeled A', which is geographically distant from Cell A. A' is a co-channel cell to A.

To locate co-channel cells geometrically:

Use the formula: N=i2+ij+j2N = i^2 + ij + j^2 where ii and jj are integers.

For N=7N = 7: i=2,j=1i = 2, j = 1.

Start at Cell A.

Move 2 cells in a straight line.

Turn 60° counterclockwise and move 1 cell to find the nearest co-channel cell.

2. Define following Term: a) cell b) Frequency Reuse c)Cell Splitting

a) Cell

A cell is a small geographical area in a cellular network covered by a base station using a specific set of frequencies.

Each cell serves users within its boundary and ensures efficient use of spectrum by allowing frequency reuse in other cells.

Cells are typically represented as hexagons for coverage analysis.

b) Frequency Reuse

Frequency reuse is the technique of using the same frequency bands in geographically separated cells within a cellular network. This allows efficient spectrum utilization and supports more users in a limited frequency spectrum.

Example: If Cell A uses frequency f1, another cell (A') separated by sufficient distance can also use f1 without causing interference.

c) Cell Splitting

Cell splitting is a technique used to increase the capacity of a cellular network by dividing a large cell into smaller cells. Each smaller cell has its own base station and operates at a lower power level.

Purpose: Accommodate more users in high-density areas by increasing the number of cells.

Example: A cell with a radius of 10 km can be split into 4 cells with a radius of 5 km each, doubling the capacity in that area.