Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Department of Computer Engineering

Academic Year: 2024-25 **Name of Student: Krushnali Biradar**
Semester: VI **Student ID:22102039**
Class / Branch: T.E Comp **Date of Performance:4/3/25**
Subject: MC **Date of Submission:4/3/25**
Name of Instructor: Prof. Tanvi

Experiment No.6

Aim:

To setup & configuration of Wireless Access Point (AP). Analyze the Wi-Fi communication range in the presence of the access point (AP) and the base station (BS). Consider BS and AP are static. Find out the maximum distance to which two way communications is possible. Try multiple iterations by adjusting its distance in the code and test it.

Program:

```
import numpy as np


# Constants
TX_POWER_BS = 20  # Transmit power of base station in dBm
TX_POWER_AP = 15  # Transmit power of access point in dBm
SIGNAL_THRESHOLD = -90  # Signal threshold for reception in dBm
FREQUENCY = 2.4e9  # Frequency in Hz (2.4 GHz for Wi-Fi)


# Function to calculate received signal strength using Friis transmission equation
def calculate_received_power(distance):
    # Friis transmission equation for free space
    path_loss = (20 * np.log10(distance) + 20 * np.log10(FREQUENCY) - 147.55)
    rx_power_bs = TX_POWER_BS - path_loss
    rx_power_ap = TX_POWER_AP - path_loss
```

```
    return rx_power_bs, rx_power_ap


# Function to check if communication is possible at a given distance
def is_communication_possible(distance):
    rx_power_bs, rx_power_ap = calculate_received_power(distance)
    # Check if both base station and access point can maintain a signal above the threshold
    return rx_power_bs > SIGNAL_THRESHOLD and rx_power_ap > SIGNAL_THRESHOLD


# Find maximum distance for communication using binary search for more accuracy
def find_max_communication_distance():
    min_distance = 1  # Start distance (minimum is 1 meter)
    max_distance = 10000  # Arbitrary maximum distance (10 km)
    epsilon = 1  # Desired accuracy (in meters)

    # Perform binary search for maximum distance where communication is still possible
    while (max_distance - min_distance) > epsilon:
        mid_distance = (min_distance + max_distance) / 2
        if is_communication_possible(mid_distance):
            min_distance = mid_distance  # We can communicate at this distance, so try further
        else:
            max_distance = mid_distance  # Communication fails, try shorter distances

    return min_distance
```

```
# Test the communication range

max_distance = find_max_communication_distance()

print("Maximum distance for two-way communication:", round(max_distance, 2), "m")
```

Output: