# PROJECT BASED LEARNING PROJECT

on

Wi-Fi simulator Using MATLAB

Submitted in the partial fulfillment of the requirements

For the Project based learning in Cellular Technology & 4G

In branch

Electronics & Communication Engineering

By

**PRN-2114110466 MOHAMMAD KAZIM**

**PRN-2114110452 ARYAN GUPTA**

**PRN-2114110470 RISHI KUMAR RANJAN**

Under the guidance of Course In charge

**Prof. Dr. TANUJA S. DHOPE**

Department of Electronic & Communication Engineering

Bharati Vidyapeeth, (Deemed to be University)

College of Engineering, Pune-411043

Academic Year: - 2023-24

## Bharati Vidyapeeth (Deemed to be University) College of Engineering,

**Pune – 411043**

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**CERTIFICATE**

Certified that the Project Based Learning report entitled, “Wi-Fi simulator Using MATLAB” is a bonafide work done by

## PRN-2114110470 RISHI RANJAN

## PRN-2114110452 ARYAN GUPTA

## PRN-2114110466 MOHAMMAD KAZIM

in partial fulfillment of the requirement for the award of credits for Project Based Learning (PBL) in **“**Cellular Technology & 4G**”,** of Bachelor of Technology semester VI, in Electronics and Communication Engineering.

**Date:**

**Prof. Dr. Tanuja S. Dhope Prof. Dr. Arundhati A.Shinde**

**Course In-charge Professor & Head**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
| 1. | Problem Statement & Solution | 4 |
| 2. | Description about Project | 5 |
| 3. | Applications | 6 |
| 4. | Introduction to MATLAB & Tools | 7 |
| 5. | Flowchart | 8 |
| 6. | Algorithm | 9 |
| 7. | Code and Output | 10-17 |
| 8. | Advantages and Disadvantages | 18 |
| 9. | Course Outcome & Conclusion | 19 |
| 10. | References & GitHub-Link | 20 |

**Chapter-1**

**Problem Statement: -**

Wi-Fi simulation is crucial for testing new technologies, educating students, optimizing networks, analyzing security, testing IoT environments, and planning emergency responses, ensuring the reliability and security of wireless communication systems across various applications and scenarios.

**Solution: -**

Set the parameters according to your simulation requirements:

numNodes: Number of Wi-Fi nodes in the network.

channelCapacity: Maximum channel capacity (not used in the provided code).

simTime: Simulation time in seconds.

timeStep: Time step for simulation.

dataRate: Data transmission rate in Mbps.

thresholdSNR: Signal-to-Noise Ratio (SNR) threshold for data transmission.

interferenceLevel: Interference level for random noise.

transmitPower: Transmit power in dBm.

frequency: Operating frequency in Hertz.

pathLossExponent: Path loss exponent for signal attenuation.

shadowingStdDev: Standard deviation of log-normal shadowing.

Initialize the positions of Wi-Fi nodes randomly within a given area.

Simulate Wi-Fi communication:

Iterate over each node.

For each node, generate random data and calculate the signal strength between that node and every other node.

If the signal strength exceeds the SNR threshold, simulate data transmission from the current node to the other node.

Record the transmission speed and print relevant information.

Visualize the transmission rate for each node.

# Chapter-2

# DESCRIPTION ABOUT PROJECT: -

A Wi-Fi simulator in MATLAB is a tool that can simulate the behavior of a Wi-Fi network in various scenarios. It can model the physical and medium access control (MAC) layer parameters of the Wi-Fi network, as well as the scenario parameters of the environment where the network is deployed. The simulator can generate transmitter site objects and receiver site objects with isotropic antenna elements, and calculate the statistics of the Wi-Fi network, including the throughput, packet loss ratio, and average packet latency. The simulator can also generate PCAP files for further analysis and visualize the node performance using helper objects in MATLAB. The WLAN Toolbox in MATLAB provides networking and multi-node system-level simulation capabilities for Wi-Fi systems, which can help characterize the performance of Wi-Fi networks by including effects of MAC layer modeling, shared communication channels, data traffic patterns, channel contentions.

Network Topology Configuration: The simulator allows users to define the layout of the Wi-Fi network, including the placement of access points (APs), clients, and potential sources of interference.

Signal Propagation Modeling: Implementing realistic signal propagation models such as Free Space Path Loss (FSPL), log-distance path loss, and fading models to simulate the attenuation and distortion of signals as they travel through the environment.

Channel Modeling: Incorporating channel characteristics such as noise, multipath fading, and interference from other wireless devices to accurately emulate real-world Wi-Fi channel conditions.

Protocol Simulation: Simulating Wi-Fi protocols such as IEEE 802.11a/b/g/n/ac/ax to analyze their performance in terms of throughput, latency, and packet loss under different scenarios.

Mobility Simulation: Introducing mobility models for clients and access points to study the impact of movement on network performance and handover mechanisms.

Traffic Generation: Generating various types of traffic patterns (e.g., voice, video, data) to evaluate the QoS (Quality of Service) provided by the Wi-Fi network.

Performance Metrics: Calculating performance metrics such as throughput, packet loss rate, delay, and jitter to assess the effectiveness of different network configurations and protocols.

Visualization and Analysis: Providing visualization tools to plot network topology, signal strength maps, throughput graphs, and other relevant metrics for better understanding and analysis of simulation results.

## Chapter-3

## Applications: -

**Network Planning and Optimization:** Wi-Fi simulation tools are used to plan and optimize the deployment of Wi-Fi networks in different environments such as homes, offices, campuses, and public spaces. By simulating different configurations, antenna placements, and channel allocations, network engineers can optimize network coverage, capacity, and performance.

**Performance Evaluation:** Wi-Fi simulation allows for the evaluation of network performance under different scenarios, including varying numbers of users, traffic loads, mobility patterns, and interference conditions. This helps in identifying potential bottlenecks, optimizing network parameters, and ensuring satisfactory user experience.

**Protocol Development and Testing:** Wi-Fi simulation is valuable for developing and testing new protocols, algorithms, and technologies for Wi-Fi networks. Researchers and developers can simulate different aspects of Wi-Fi protocols, such as medium access control (MAC) protocols, routing protocols, quality of service (QoS) mechanisms, and security protocols, in a controlled environment.

**IoT and Smart Environments**: With the proliferation of Internet of Things (IoT) devices and smart environments, Wi-Fi simulation plays a crucial role in designing and optimizing Wi-Fi networks to support a large number of connected devices with diverse communication requirements. Simulations help in understanding the impact of IoT traffic on network performance and in designing efficient communication strategies.

**Education and Training:** Wi-Fi simulation tools are used in educational settings to teach networking concepts, protocols, and technologies. Students can simulate real-world Wi-Fi scenarios, analyze network behavior, and experiment with different configurations in a virtual environment, facilitating hands-on learning and skill development.

**Chapter-4**

**Introduction to MATLAB:-**

MATLAB stands as a cornerstone in engineering, scientific, and academic realms, offering a versatile programming language and a robust numerical computing environment. Its intuitive syntax simplifies complex computations, while its extensive library of functions empowers users to tackle diverse tasks with ease. With an array of built-in toolboxes covering various disciplines, from signal processing to machine learning, MATLAB serves as a one-stop solution for data analysis, algorithm development, and simulation. Its flexibility and efficiency make it indispensable for professionals and researchers alike, enabling groundbreaking discoveries and innovative solutions across countless domains**.**

Some MATLAB functions used in the project are :

**Random Number Generators**.

rand() generates uniformly distributed random numbers between 0 and 1. It is used to initialize node positions and generate random data.

randi([min, max], rows, cols) generates random integers within the specified range [min, max]. It is used to simulate data transmission rates.

**norm():**

norm(v) computes the Euclidean norm (magnitude) of vector v. It calculates the distance between nodes by finding the Euclidean distance between their positions.

**for Loop:**

The for loop iterates over nodes and time steps to simulate Wi-Fi communication for each node over time.

**if Statement:**

The if statement conditionally executes code based on signal strength and transmission status. It determines whether a node should transmit data based on the calculated signal strength.

**Chapter-5**

**FLOWCHART: -**

A diagram of a process

Description automatically generated

Define Parameters

Start

Configure

Generate

Validate

Visualize

Analyze

|

Adjust

Re-validate

**Chapter-6**

**Algorithm: -**

1. Start

2. Initialize parameters:

- numNodes: Number of Wi-Fi nodes

- channelCapacity: Maximum channel capacity

- simTime: Simulation time in seconds

- timeStep: Time step for simulation

- dataRate: Data transmission rate (Mbps)

- thresholdSNR: Signal-to-Noise Ratio (SNR) threshold for data transmission

- interferenceLevel: Interference level for random noise

- transmitPower: Transmit power (dBm)

- frequency: Operating frequency in Hertz

- pathLossExponent: Path loss exponent for signal attenuation

- shadowingStdDev: Standard deviation of log-normal shadowing

3. Initialize nodes with random positions (nodePositions)

4. Initialize signal strengths matrix (signalStrengths) with zeros

5. For each node from 1 to numNodes:

a. Create a figure for the current node

b. Initialize empty arrays for plotting (x, y)

c. For each time step from 0 to simTime with a step of timeStep:

i. Generate random data for each node (nodeData)

ii. For each other node j from 1 to numNodes:

- Skip self-communication (if node == j)

- Calculate the distance between the current node and node j

- Calculate signal strength based on distance, path loss, and shadowing:

\* Avoid division by zero

\* Calculate path loss (assuming free space path loss)

\* Calculate log-normal shadowing

- Update signal strength in the signal strengths matrix

- Simulate data transmission if SNR is sufficient:

\* Record transmitting nodes for plotting

\* Calculate transmission speed

\* Print transmission information

d. Calculate average data rate for each receiver node:

- Identify unique receiver nodes

- Calculate mean transmission speed for each receiver node

e. Plot the graph for the current node:

- Plot unique receiver nodes against average data rates

- Set title, xlabel, ylabel

- Turn on grid

6. End

**Chapter-7**

**CODE: -**

% Wi-Fi Simulator in MATLAB

% Parameters

numNodes = 10;

% Number of Wi-Fi nodes

channelCapacity = 100;

% Maximum channel capacity

simTime = 10;

% Simulation time in seconds

timeStep = 0.1;

% Time step for simulation

% New parameters

dataRate = 10;

% Data transmission rate (Mbps)

thresholdSNR = 10;

% Signal-to-Noise Ratio (SNR) threshold for data transmission

interferenceLevel = 5;

% Interference level for random noise

transmitPower = 15;

% Transmit power (dBm)

frequency = 2.4e9;

% Operating frequency in Hertz (e.g., 2.4 GHz)

pathLossExponent = 2;

% Path loss exponent for signal attenuation

shadowingStdDev = 3;

% Standard deviation of log-normal shadowing

% Initialize nodes with random positions

nodePositions = rand(numNodes, 2);

% Initialize signal strengths

signalStrengths = zeros(numNodes, numNodes);

% Simulate Wi-Fi communication

for node = 1:numNodes

% Create a figure for each node

figure;

% Initialize variables for plotting

x = [];

y = [];

for time = 0:timeStep:simTime

% Generate random data for each node

nodeData = randi([1, 10], 1, numNodes);

for j = 1:numNodes

if node ~= j % Skip self-communication

% Calculate signal strength based on distance, path loss, and shadowing

distance = norm(nodePositions(node, :) - nodePositions(j, :));

% Avoid division by zero

if distance == 0

signalStrength = transmitPower + randn() \* interferenceLevel;

else

pathLoss = (distance^pathLossExponent) / (frequency / 2e9); % Assuming free space path loss

shadowing = 10^((randn() \* shadowingStdDev) / 10); % Log-normal shadowing

signalStrength = transmitPower - 10 \* log10(pathLoss) + shadowing + randn() \* interferenceLevel;

end

% Update signal strength

signalStrengths(node, j) = signalStrength;

% Simulate data transmission if SNR is sufficient

if signalStrengths(node, j) > thresholdSNR

% Record the transmitting nodes for plotting

x = [x, j];

transmissionSpeed = nodeData(node) \* dataRate \* timeStep \* rand(); % Adjust as needed

y = [y, transmissionSpeed];

fprintf('Node %d transmitting data to Node %d at time %.2f\n', node, j, time);

fprintf('Transmission Speed: %.2f Mbps\n', transmissionSpeed);

end

end

end

end

% Calculate average data rate for each receiver node

uniqueReceivers = unique(x);

averageDataRates = zeros(size(uniqueReceivers));

for k = 1:length(uniqueReceivers)

receiverIdx = uniqueReceivers(k);

averageDataRates(k) = mean(y(x == receiverIdx));

end

% Plot the graph for the current node

plot(uniqueReceivers, averageDataRates, 'o-');

title(['Average Transmission Rate for Node ', num2str(node)]);

xlabel('Receiver Node');

ylabel('Average Transmission Rate (Mbps)');

    grid on;

end

**Output and Graph:-**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated



A graph showing the average transmission rate for node 2

Description automatically generated

A graph showing the average transmission rate for node 3

Description automatically generated

A graph showing the average transmission rate

Description automatically generated

A graph showing the average transmission rate

Description automatically generated

A graph showing a transmission rate

Description automatically generated

A graph showing the average transmission rate

Description automatically generated

A graph showing the average transmission rate for node 8

Description automatically generated

A graph showing the average transmission rate for node 9

Description automatically generated

A graph showing the average transmission rate for node 10

Description automatically generated

## Chapter-8

## Advantages: -

* Cost-Effective Testing: Allows testing without physical hardware, saving costs.
* Controlled Environment: Enables manipulation of network parameters for testing.
* Scalability: Simulates large-scale Wi-Fi networks for testing scalability.
* Repeatable Experiments: Facilitates repeating experiments under identical conditions for reliable results.
* Rapid Prototyping: Accelerates the development of new Wi-Fi technologies and applications through rapid experimentation.

**Disadvantage: -**

* **Accuracy and realism issues**: Simulated environments may not accurately replicate real-world WiFi conditions, leading to unreliable results.
* **Resource-intensive:** Running WiFi simulations can require significant processing power and time.
* **Costly setup:** Setting up WiFi simulation environments often requires specialized software, hardware, and expertise.
* **Lack of real-time data:** Simulations cannot provide real-time insights into network performance.
* **Dependency on assumptions:** Simulations rely on assumptions about network behavior and environmental factors.

**Chapter-9**

**COURSE OUTCOME: -**

**CO1:-** Understand the basics of Wi-Fi Simulator in MATLAB.

**CO5:**- Examine the Wi-Fi Simulator in MATLAB.

Hence CO1 & CO5 has been satisfied.

**CONCLUSION: -**

In conclusion, the development of the WiFi simulator in MATLAB has culminated in a robust and versatile platform for analyzing and optimizing wireless communication systems. Through this endeavor, we've delved into various aspects of WiFi behavior, including signal strength, interference, and channel capacity. The simulator's capabilities have enabled researchers and engineers to explore different modulation schemes, coding techniques, and channel models, offering valuable insights for improving system performance.

**Chapter-10**

**References: -**

* T. S. Rappaport, “Wireless Communications: Principles and practice”, Pearson, 2nd Edition, 2010.
* Raj Pandya, “Mobile & Personnel communication Systems and Services”, Prentice Hall India, 2001.
* T. L. Singal, “Wireless Communications”, Tata McGraw Hill, 2nd Edition, 2011.
* <https://www.mathworks.com/help/lte/ug/umts-wi-fi> simulator.

**GitHub - Link: -**

https://github.com/Aryangpt/wifi-simulator