% 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