% Parameters

num\_nodes = 10;

num\_channels = 1; % Single channel

channel\_capacity = 1000; % bits per second

packet\_arrival\_rate = 0.1;

transmission\_time = 50; % milliseconds

energy\_consumption\_per\_packet = 5; % Joules

simulation\_time = 1000; % milliseconds

% Initialization

num\_packets = round(simulation\_time \* packet\_arrival\_rate);

received\_packets = zeros(num\_nodes, 1);

dropped\_packets = zeros(num\_nodes, 1);

average\_delay = zeros(num\_nodes, 1);

throughput = zeros(num\_nodes, 1);

energy\_consumption = zeros(num\_nodes, 1);

efficiency = zeros(num\_nodes, 1);

for node = 1:num\_nodes

successful\_transmissions = 0;

total\_delay = 0;

total\_energy = 0;

for packet = 1:num\_packets

if rand() < packet\_arrival\_rate

% Simulate transmission

transmission\_delay = transmission\_time;

total\_delay = total\_delay + transmission\_delay;

% Update energy consumption

total\_energy = total\_energy + energy\_consumption\_per\_packet;

successful\_transmissions = successful\_transmissions + 1;

received\_packets(node) = received\_packets(node) + 1;

else

dropped\_packets(node) = dropped\_packets(node) + 1;

end

end

average\_delay(node) = total\_delay / successful\_transmissions;

throughput(node) = successful\_transmissions \* transmission\_time / simulation\_time;

energy\_consumption(node) = total\_energy;

efficiency(node) = throughput(node) / energy\_consumption(node);

end

% Calculate average energy, average throughput, received packet rate, dropped packet rate

avg\_energy\_single\_channel = mean(energy\_consumption);

avg\_throughput\_single\_channel = mean(throughput);

received\_packet\_rate\_single\_channel = mean(received\_packets) / simulation\_time;

dropped\_packet\_rate\_single\_channel = mean(dropped\_packets) / simulation\_time;

% Display metrics for single-channel scenario...

disp('Metrics for Single-Channel Scenario:');

disp(['Average Energy Consumption: ', num2str(avg\_energy\_single\_channel), ' Joules']);

disp(['Average Throughput: ', num2str(avg\_throughput\_single\_channel), ' bits per second']);

disp(['Received Packet Rate: ', num2str(received\_packet\_rate\_single\_channel), ' packets per millisecond']);

disp(['Dropped Packet Rate: ', num2str(dropped\_packet\_rate\_single\_channel), ' packets per millisecond']);

disp(newline);

% Display received and dropped packets for Single-Channel Scenario

disp('Received and Dropped Packets for Single-Channel Scenario:');

for node = 1:num\_nodes

disp(['Node ', num2str(node), ': Received = ', num2str(received\_packets(node)), ', Dropped = ', num2str(dropped\_packets(node))]);

end

disp(newline);

% Parameters

num\_nodes = 10;

num\_channels = 5; % Multi channel

channel\_capacity = 1000; % bits per second

packet\_arrival\_rate = 0.1;

transmission\_time = 50; % milliseconds

energy\_consumption\_per\_packet = 5; % Joules

simulation\_time = 1000; % milliseconds

% Initialization

num\_packets = round(simulation\_time \* packet\_arrival\_rate);

received\_packets = zeros(num\_nodes, 1);

dropped\_packets = zeros(num\_nodes, 1);

average\_delay = zeros(num\_nodes, 1);

throughput = zeros(num\_nodes, 1);

energy\_consumption = zeros(num\_nodes, 1);

efficiency = zeros(num\_nodes, 1);

for node = 1:num\_nodes

successful\_transmissions = 0;

total\_delay = 0;

total\_energy = 0;

for packet = 1:num\_packets

if rand() < packet\_arrival\_rate

selected\_channel = randi(num\_channels);

% Simulate channel contention

contention\_delay = rand() \* transmission\_time;

% Simulate transmission

transmission\_delay = transmission\_time;

total\_delay = total\_delay + transmission\_delay + contention\_delay;

% Update energy consumption

total\_energy = total\_energy + energy\_consumption\_per\_packet;

successful\_transmissions = successful\_transmissions + 1;

received\_packets(node) = received\_packets(node) + 1;

else

dropped\_packets(node) = dropped\_packets(node) + 1;

end

end

average\_delay(node) = total\_delay / successful\_transmissions;

throughput(node) = successful\_transmissions \* transmission\_time / simulation\_time;

energy\_consumption(node) = total\_energy;

efficiency(node) = throughput(node) / energy\_consumption(node);

end

% Calculate average energy, average throughput, received packet rate, dropped packet rate

avg\_energy\_multichannel = mean(energy\_consumption);

avg\_throughput\_multichannel = mean(throughput);

received\_packet\_rate\_multichannel = mean(received\_packets) / simulation\_time;

dropped\_packet\_rate\_multichannel = mean(dropped\_packets) / simulation\_time;

% Display metrics for multichannel scenario...

disp('Metrics for Multichannel Scenario:');

disp(['Average Energy Consumption: ', num2str(avg\_energy\_multichannel), ' Joules']);

disp(['Average Throughput: ', num2str(avg\_throughput\_multichannel), ' bits per second']);

disp(['Received Packet Rate: ', num2str(received\_packet\_rate\_multichannel), ' packets per millisecond']);

disp(['Dropped Packet Rate: ', num2str(dropped\_packet\_rate\_multichannel), ' packets per millisecond']);

disp(newline);

% Display received and dropped packets for Multichannel Scenario

disp('Received and Dropped Packets for Multichannel Scenario:');

for node = 1:num\_nodes

disp(['Node ', num2str(node), ': Received = ', num2str(received\_packets(node)), ', Dropped = ', num2str(dropped\_packets(node))]);

end

disp(newline);

% Plotting

% Metrics comparison plot

figure;

subplot(2, 2, 1);

plot(1:num\_nodes, energy\_consumption, 'o-', 1:num\_nodes, repmat(avg\_energy\_single\_channel, 1, num\_nodes), '--');

xlabel('Node');

ylabel('Energy Consumption (Joules)');

legend('Multichannel', 'Single-Channel Avg', 'Location', 'best');

title('Energy Consumption Comparison');

subplot(2, 2, 2);

plot(1:num\_nodes, throughput, 'o-', 1:num\_nodes, repmat(avg\_throughput\_single\_channel, 1, num\_nodes), '--');

xlabel('Node');

ylabel('Throughput (bits/s)');

legend('Multichannel', 'Single-Channel Avg', 'Location', 'best');

title('Throughput Comparison');

subplot(2, 2, 3);

plot(1:num\_nodes, received\_packets / simulation\_time, 'o-', 1:num\_nodes, repmat(received\_packet\_rate\_single\_channel, 1, num\_nodes), '--');

xlabel('Node');

ylabel('Received Packet Rate (packets/ms)');

legend('Multichannel', 'Single-Channel Avg', 'Location', 'best');

title('Received Packet Rate Comparison');

subplot(2, 2, 4);

plot(1:num\_nodes, dropped\_packets / simulation\_time, 'o-', 1:num\_nodes, repmat(dropped\_packet\_rate\_single\_channel, 1, num\_nodes), '--');

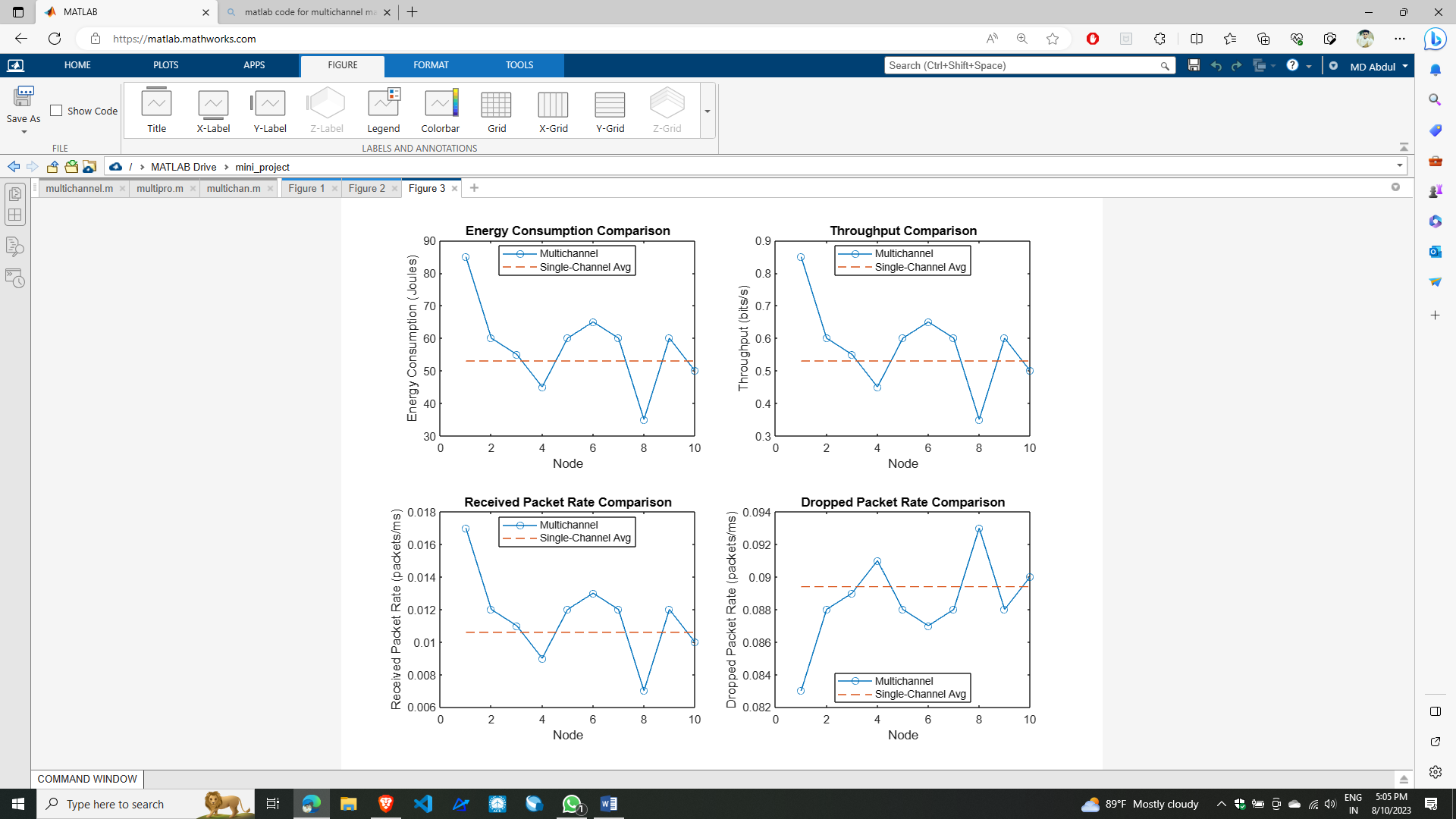
xlabel('Node');

ylabel('Dropped Packet Rate (packets/ms)');

legend('Multichannel', 'Single-Channel Avg', 'Location', 'best');

title('Dropped Packet Rate Comparison');

**OUTPUT:**



**COMMAND WINDOW:**

Metrics for Single-Channel Scenario:  
Average Energy Consumption: 53 Joules  
Average Throughput: 0.53 bits per second  
Received Packet Rate: 0.0106 packets per millisecond  
Dropped Packet Rate: 0.0894 packets per millisecond  
  
  
Received and Dropped Packets for Single-Channel Scenario:  
Node 1: Received = 8, Dropped = 92  
Node 2: Received = 7, Dropped = 93  
Node 3: Received = 10, Dropped = 90  
Node 4: Received = 11, Dropped = 89  
Node 5: Received = 17, Dropped = 83  
Node 6: Received = 12, Dropped = 88  
Node 7: Received = 12, Dropped = 88  
Node 8: Received = 11, Dropped = 89  
Node 9: Received = 9, Dropped = 91  
Node 10: Received = 9, Dropped = 91  
  
  
Metrics for Multichannel Scenario:  
Average Energy Consumption: 57.5 Joules  
Average Throughput: 0.575 bits per second  
Received Packet Rate: 0.0115 packets per millisecond  
Dropped Packet Rate: 0.0885 packets per millisecond  
  
  
Received and Dropped Packets for Multichannel Scenario:  
Node 1: Received = 17, Dropped = 83  
Node 2: Received = 12, Dropped = 88  
Node 3: Received = 11, Dropped = 89  
Node 4: Received = 9, Dropped = 91  
Node 5: Received = 12, Dropped = 88  
Node 6: Received = 13, Dropped = 87  
Node 7: Received = 12, Dropped = 88  
Node 8: Received = 7, Dropped = 93  
Node 9: Received = 12, Dropped = 88  
Node 10: Received = 10, Dropped = 90