Project Based Learning Report
on

To measure the network performance like Link bandwidth , Propogation delay , Jitter , Throughput and Queue type using MATLAB

Submitted in the partial fulfillment of the requirements For the Project based learning in Cellular Technology & 4G

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Name of the student	PRN
Bhavya Jain	2114110458
Satyam Mishra	2214110605
Ram Prakash	2214110681

Under the guidance of Course Incharge

Prof. Dr. Tanuja S Dhope

Department of Electronics & Communication Engineering

Bharati Vidyapeeth

(Deemed to be University)
College of Engineering,
Pune – 411043

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled, “To measure the network performance like Link bandwidth , Propogation delay , Jitter , Throughput and Queue type using MATLAB” is a bonafied work done by

Name of the student	**PRN**
Bhavya Jain	**2114110458**
Satyam Mishra	**2214110605**
Ram Prakash	**2214110681**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in Cellular Technology & 4G Course of Bachelor of Technology Semester VI, in Electronics and Communication Branch.

Date:

Prof. Dr. Tanuja Dhope
Course Incharge

Prof. Dr. A.A.Shinde
Head of Department

INDEX

CHAPTER 1:- PROBLEM STATEMENT

Question:-

Measuring Network Performance like Link bandwidth, Propagation delay, Jitter, Throughput and Queue type using Matlab.

Solution:-

The solution to the problem of measuring network performance metrics such as link bandwidth, propagation delay, jitter, throughput, and queue type in MATLAB involves a systematic approach leveraging computational methods. Firstly, parameters like data size, time taken, distance, speed of light, queue length, and threshold are initialized. Then, calculations are performed iteratively to derive the required metrics. Bandwidth is computed by dividing data size by time taken, providing insights into the data transfer rate. Propagation delay is determined by considering the distance and speed of light, offering an understanding of signal transmission delays.

Queue type classification is based on comparing queue length with predefined thresholds, indicating congestion levels. Latency, a crucial metric, is calculated using bandwidth values, aiding in assessing data transfer delays. Throughput, quantifying data transfer efficiency, is derived by dividing data size by time taken. Jitter, representing variation in packet arrival times, is obtained by dividing latency by throughput. The results are displayed for analysis, facilitating informed decision-making. Additionally, visualizations such as plots provide intuitive representations of the data, aiding in understanding trends and relationships among performance metrics. Overall, this solution enables network administrators to evaluate and optimize network performance effectively, ensuring efficient data transmission and quality service delivery.

Need of solution :-

The need to solve the problem of measuring network performance metrics in MATLAB stems from various critical factors. Firstly, it enables network optimization by quantifying parameters like link bandwidth and throughput, allowing administrators to identify bottlenecks and implement optimizations for improved efficiency. Additionally, effective Quality of Service (QoS) management relies on monitoring performance metrics to ensure critical applications receive necessary resources while preventing congestion for non-critical traffic. Troubleshooting network issues becomes more efficient with accurate measurement of parameters such as propagation delay and jitter, enabling prompt resolution of latency or packet loss issues.

Capacity planning benefits from understanding current network usage and forecasting future demands, facilitating proactive infrastructure upgrades to accommodate growing traffic volumes. Compliance with Service Level Agreements (SLAs) is also ensured through continuous measurement and monitoring of network performance metrics, demonstrating an organization's ability to deliver reliable services. Finally, network security analysis relies on anomaly detection in performance metrics to identify and mitigate potential security threats, safeguarding network integrity and data confidentiality. In summary, solving this problem in MATLAB is crucial for optimizing network efficiency, ensuring quality service delivery, troubleshooting issues, planning for future scalability, complying with SLAs, and enhancing network security.

CHAPTER 2:- PROJECT DESCRIPTION

MATLAB code is aimed at addressing the problem statement of measuring various aspects of network performance including link bandwidth, propagation delay, jitter, throughput, and queue type. Let's break down the project:

1. **Initialization**: The project starts by initializing parameters such as data size, time taken, distance, speed of light, queue length, and threshold.

2. **Bandwidth Calculation**: It calculates the bandwidth for different data sizes and time taken using the formula: bandwidth = (data size * 8) / (time taken * 1000).

3. **Propagation Delay Calculation**: It computes the propagation delay for different distances and speed of light values using the formula: propagation delay = (distance * 1000) / (speed of light - iteration + 1).

4. **Queue Type Determination**: Based on the comparison between queue length and threshold, it classifies the queue type as either "Congested" or "Not Congested".

5. **Latency Calculation**: It calculates latency using the formula: latency = (data size * 8) / (bandwidth * 1000).

6. **Throughput Calculation**: Throughput is calculated by dividing data size by time taken.

7. **Jitter Calculation**: Jitter is computed by dividing latency by throughput.

8. **Displaying Results**: It displays the calculated values for bandwidth, propagation delay, queue type, latency, throughput, and jitter
   .

9. **Plotting Results**: It plots relevant parameters to visualize the relationship between bandwidth and time taken, propagation delay and distance, threshold and queue type, and latency and throughput.

This project serves as a valuable tool for network administrators, engineers, and researchers to assess, analyze, and optimize network performance, thereby ensuring smooth and reliable operation of network infrastructure in various real-world applications. It provides insights into critical performance metrics and facilitates informed decision-making for network optimization and troubleshooting. metrics and facilitates informed decision-making for network optimization and troubleshooting.

Applications:-

1. **Network Optimization**: Understanding network performance metrics such as bandwidth, propagation delay, and throughput allows network administrators to identify bottlenecks and optimize network infrastructure for improved efficiency and reliability.

2. **Troubleshooting**: When users experience slow network speeds or connectivity issues, analyzing network performance metrics can help pinpoint the root cause of the problem, whether it's related to bandwidth limitations, network congestion, or other factors.

3. **Capacity Planning**: By monitoring network performance over time, organizations can anticipate future capacity requirements and plan upgrades or expansions to accommodate growing demands without compromising performance.

4. **Quality of Service (QoS) Management**: Assessing parameters like latency and jitter is crucial for ensuring consistent and reliable network performance, particularly in applications where real-time data transmission is critical, such as voice and video communication.

5. **Service Level Agreement (SLA) Compliance**: For service providers, measuring network performance against predefined SLA parameters is essential for maintaining customer satisfaction and meeting contractual obligations.

6. **Security Analysis**: Abnormalities in network performance metrics can sometimes indicate security threats such as denial-of-service (DoS) attacks or unauthorized access attempts, prompting proactive security measures to safeguard network integrity.

7. **Research and Development**: Engineers and researchers in the field of networking can use such tools to conduct experiments, validate theoretical models, and explore new technologies aimed at enhancing network performance and efficiency.

CHAPTER 3:- SOFTWARE USED

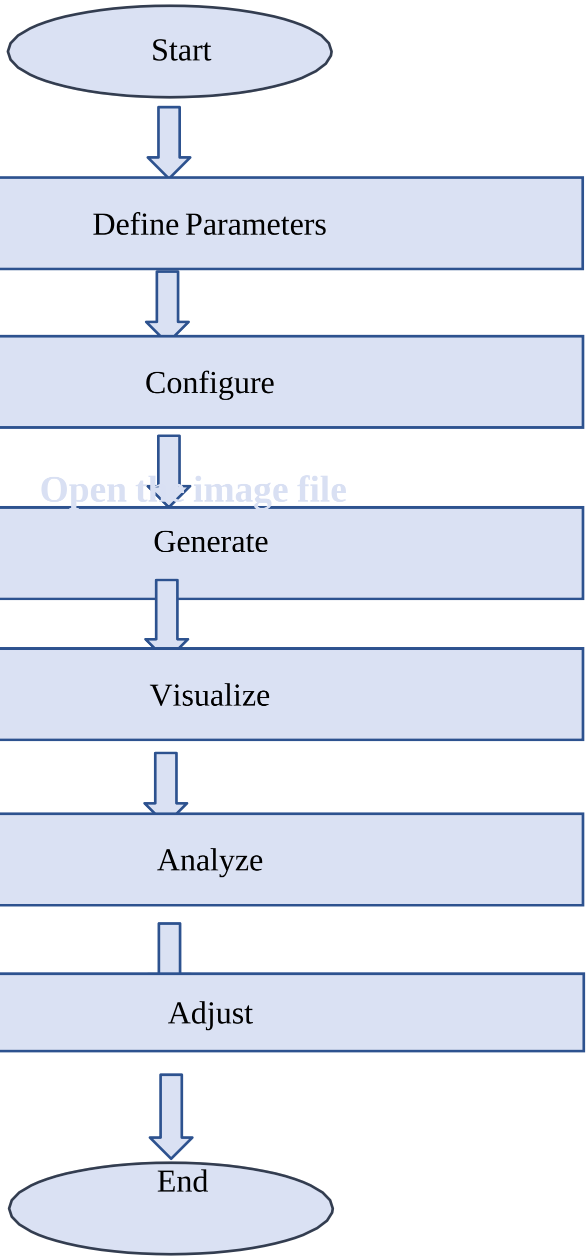
MATLAB, short for "Matrix Laboratory," is a high-level programming language and interactive environment widely used for technical computing and data visualization. Developed by MathWorks, MATLAB combines computation, visualization, and programming in a single platform, making it a powerful tool for engineers, scientists, researchers, and students across various domains.

At its core, MATLAB excels in numerical computation and matrix manipulation. It provides a rich set of built-in functions and toolboxes for solving mathematical problems, performing complex simulations, and analyzing data. MATLAB's syntax is designed to be intuitive and expressive, allowing users to write concise and readable code for a wide range of applications. One of MATLAB's key features is its interactive environment, which includes a command-line interface (CLI) and a graphical user interface (GUI). Users can execute commands, run scripts, and manipulate variables interactively in the command window, while the GUI offers tools for designing and running scripts, visualizing data, and building graphical user interfaces.
MATLAB supports a variety of data types, including scalars, vectors, matrices, arrays, and structures, making it suitable for handling diverse types of data. It also offers extensive plotting and visualization capabilities, allowing users to create 2D and 3D plots, graphs, and animations to represent and analyze their data effectively.

In addition to its core functionality, MATLAB can be extended through the use of toolboxes, which provide specialized functions and algorithms for specific applications such as signal processing, image processing, control systems, and machine learning. These toolboxes allow users to leverage MATLAB's capabilities for solving domain-specific problems efficiently.
Overall, MATLAB is a versatile and powerful platform for numerical computation, data analysis, and algorithm development, making it an indispensable tool for researchers, engineers, and students working in fields such as engineering, science, finance, and beyond. Its ease of use, flexibility, and extensive functionality make it a valuable asset for tackling a wide range of technical challenges and advancing innovation.

MATLAB

CHAPTER 4:- FLOWCHART



CHAPTER 5:- ALGORITHM

1. Initialize parameters:
   - Set values for data size, time taken, distance, speed of light, queue length, and threshold.
2. Calculate Bandwidth:
   - Iterate over each data size and time taken.
   - For each iteration, calculate bandwidth using the formula: bandwidth = (data size * 8) / (time taken * 1000).
3. Calculate Propagation Delay:
   - Iterate over each distance and speed of light.
   - For each iteration, calculate propagation delay using the formula: propagation delay = (distance * 1000) / (speed of light - iteration + 1).
4. Determine Queue Type:
   - Iterate over each threshold.
   - Compare queue length with threshold to determine queue type as "Congested" or "Not Congested".
5. Calculate Latency:
   - Iterate over each bandwidth value.
   - For each iteration, calculate latency using the formula: latency = (data size * 8) / (bandwidth * 1000).
6. Calculate Throughput:
   - Iterate over each data size and time taken.
   - For each iteration, calculate throughput by dividing data size by time taken.
7. Calculate Jitter:
   - Iterate over each latency and throughput value.
   - For each iteration, calculate jitter by dividing latency by throughput.
8. Display Results:
   - Print calculated values for bandwidth, propagation delay, queue type, latency, throughput, and jitter.
9. Plot Results:
   - Plot relevant parameters to visualize the relationship between bandwidth and time taken, propagation delay and distance, threshold and queue type, and latency and throughput.

CHAPTER 6:- PROGRAM & OUTPUT

Code:-

```
dataSize = 100:100:1000;
timeTaken = 10:10:100;
distance = 10:10:100;
speedOfLight = 299792:1:299802;
queueLength = 45;
threshold = 10:10:100;
Bandwidth = zeros(1, 10);
propagationDelay = zeros(1, 10);
answer = zeros(1, 10);
queueType = cell(1, 10);
throughput = zeros(1, 10);
latency = zeros(1, 10);
jitter = zeros(1, 10);

% Calculate Bandwidth
for x = 1:10
    Bandwidth(x) = (dataSize(x) *
8) / (timeTaken(x) * 1000);
end

disp('Bandwidth (kbps): ');
disp(Bandwidth);

% Calculate Propagation Delay
for y = 1:10
    propagationDelay(y) =
(distance(y) * 1000) ./
(speedOfLight(y) - (y-1) + 1);
end

disp('Propagation Delay (ms): ');
disp(propagationDelay);

% Determine Queue Type
for x = 1:10
    if queueLength > threshold(x)
        queueType{x} =
"Congested";
        answer(x) = 1;
    else
        queueType{x} = "Not
Congested";
```

```
        answer(x) = 0;
    end
end

disp('Queue Type: ');
disp(queueType);

% Calculate Latency
for i = 1:10
    latency(i) = (dataSize(i) * 8) ./
(Bandwidth(i) * 1000);
end

disp("Latency (ms):");
disp(latency);

% Calculate Throughput
for i = 1:10
    throughput(i) = dataSize(i) ./
timeTaken(i);
end

disp("Throughput");
disp(throughput);

% Calculate Jitter
for i = 1:10
    jitter(i)      =      latency(i)      ./
throughput(i);
end

disp("Jitter (bits/ms)");
disp(jitter);

% Plotting
figure;
plot(dataSize, Bandwidth, 'b-o');
xlabel('Data Size');
ylabel('Bandwidth (kbps)');
title('Bandwidth vs. Data Size');

figure;
plot(distance, propagationDelay, 'r-
*');
xlabel('Distance');
ylabel('Propagation Delay (ms)');
title('Propagation      Delay      vs.
Distance');

figure;
bar(answer);
set(gca, 'XTickLabel', threshold);
```

```
xlabel('Threshold');
ylabel('Congestion (1: Congested,
0: Not Congested)');
title('Congestion Status');

figure;
plot(dataSize, latency, 'g-s');
xlabel('Data Size');
ylabel('Latency (ms)');
title('Latency vs. Data Size');

figure;
plot(timeTaken, throughput, 'm-d');
xlabel('Time Taken');
ylabel('Throughput');
title('Throughput vs. Time Taken');

figure;
plot(jitter, 'k-x');
xlabel('Data Size/Time Taken
Index');
ylabel('Jitter (bits/ms)');
title('Jitter vs. Data Size/Time
Taken Index');
```

Output:-

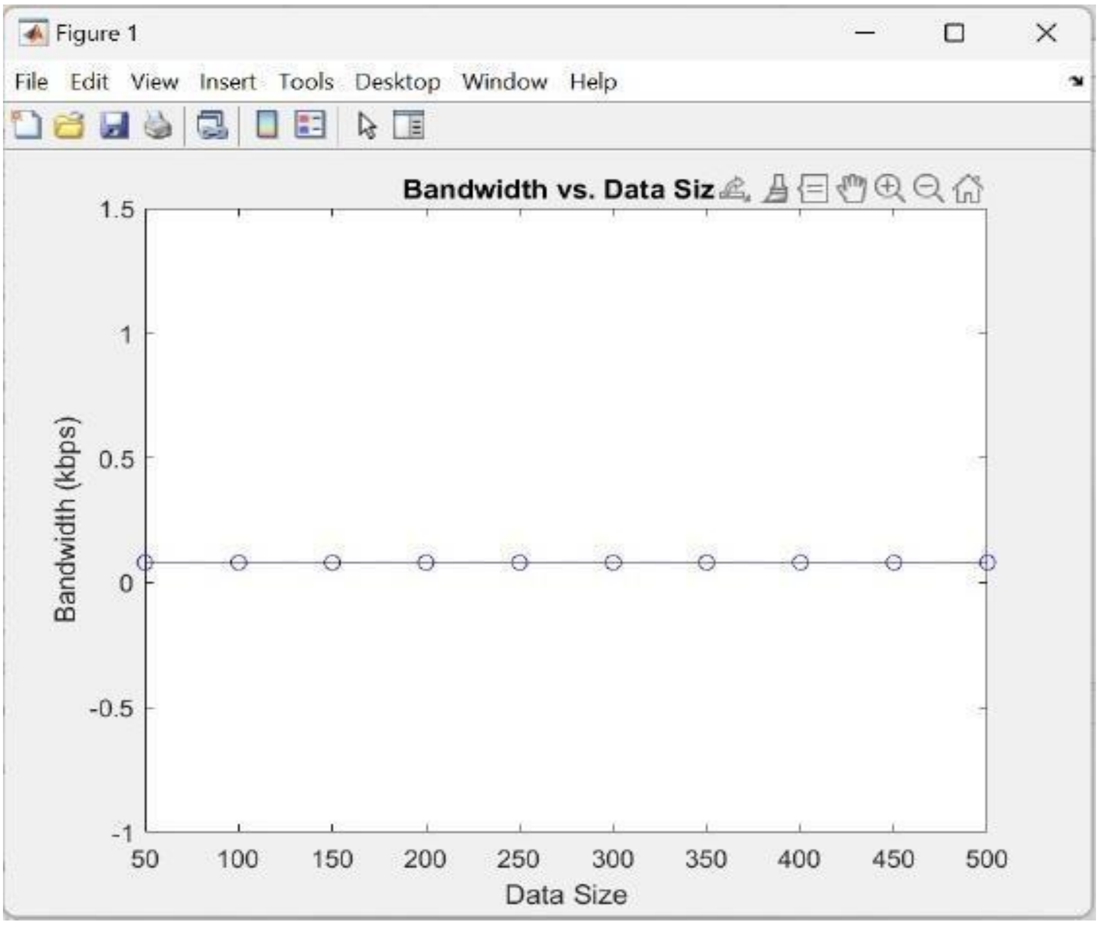


Figure 1: Bandwidth vs datasize

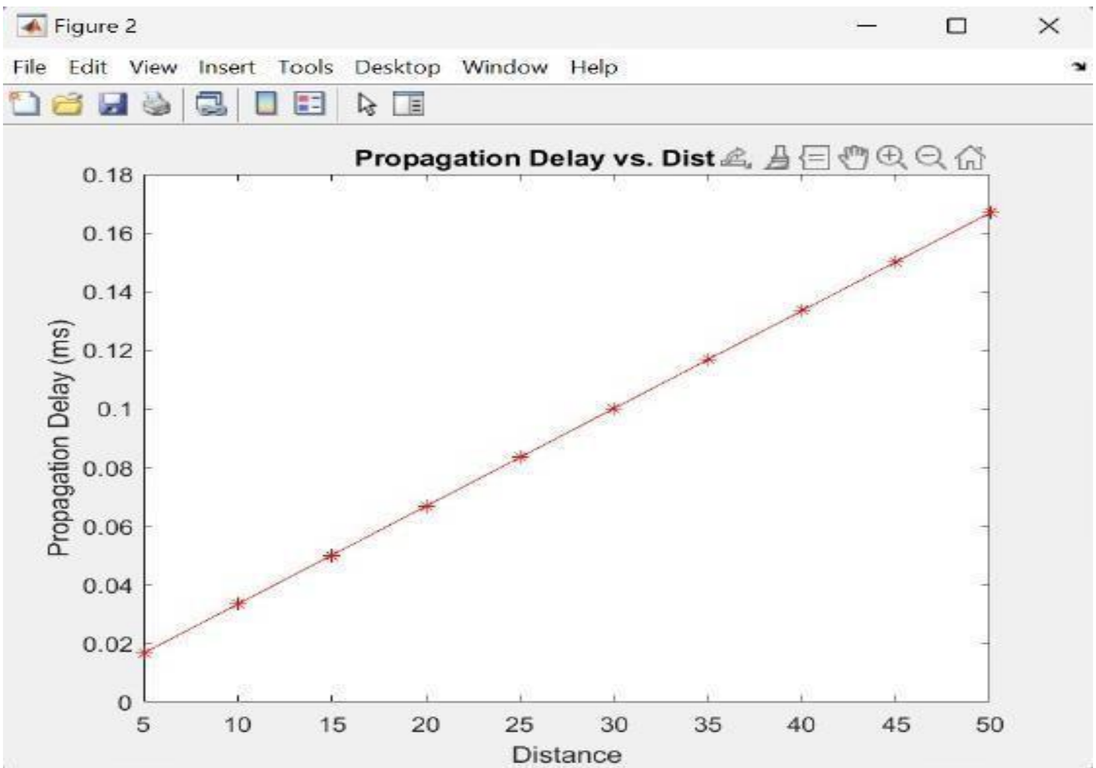


Figure 2: Propogation Delay vs Distance

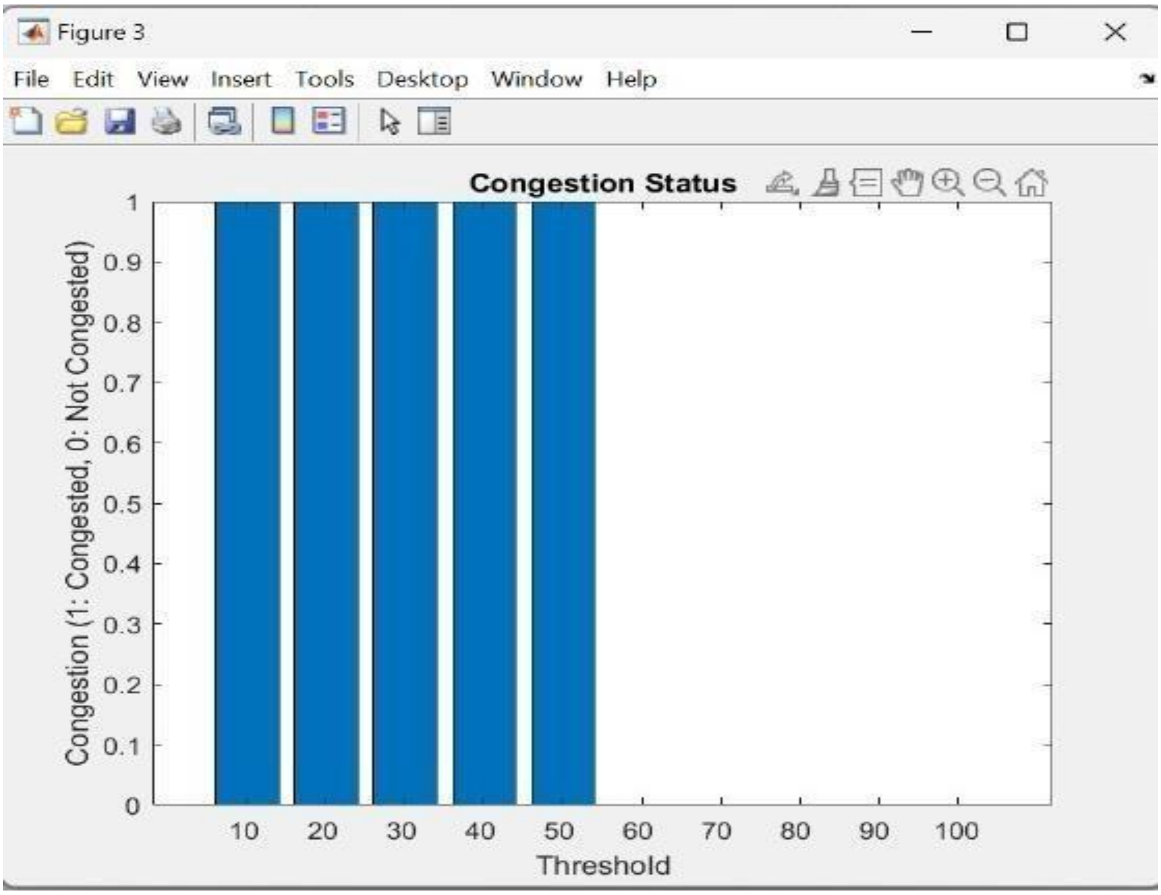


Figure 3: Congestion Status

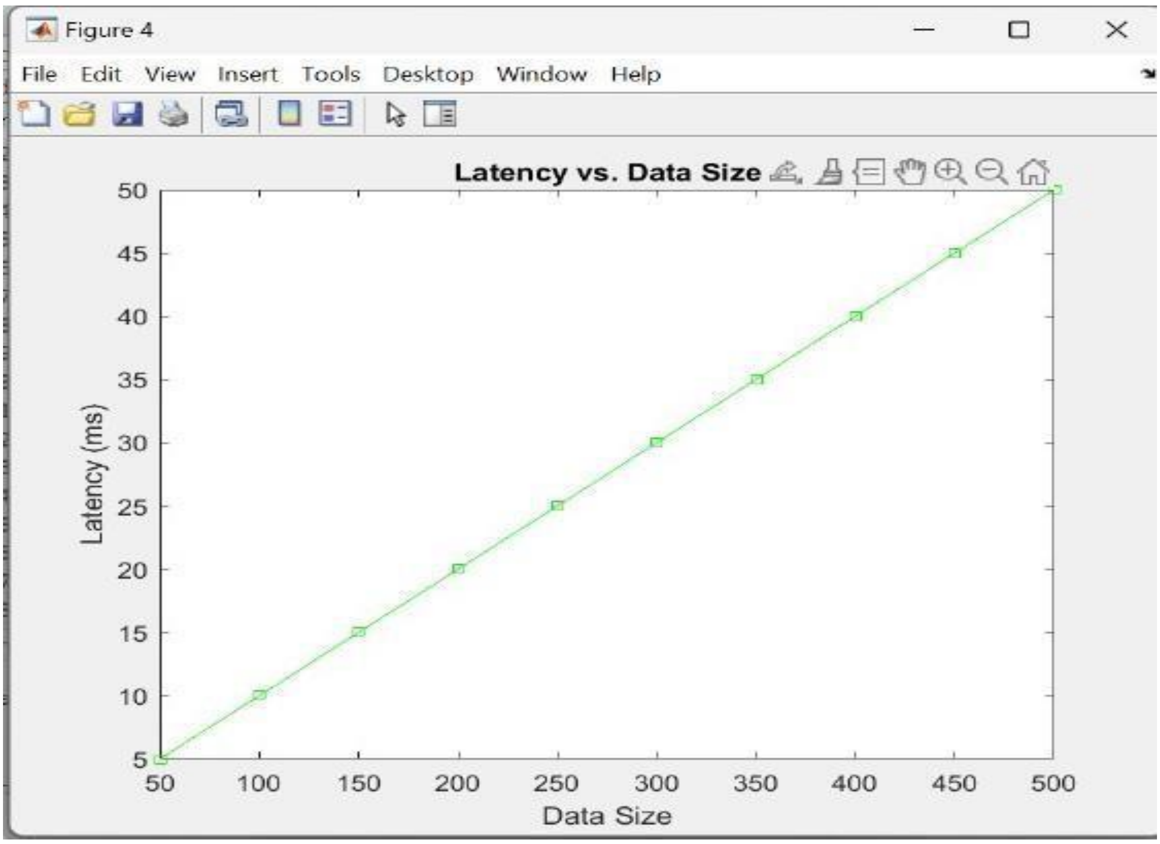


Figure 4: Latency vs Data Size

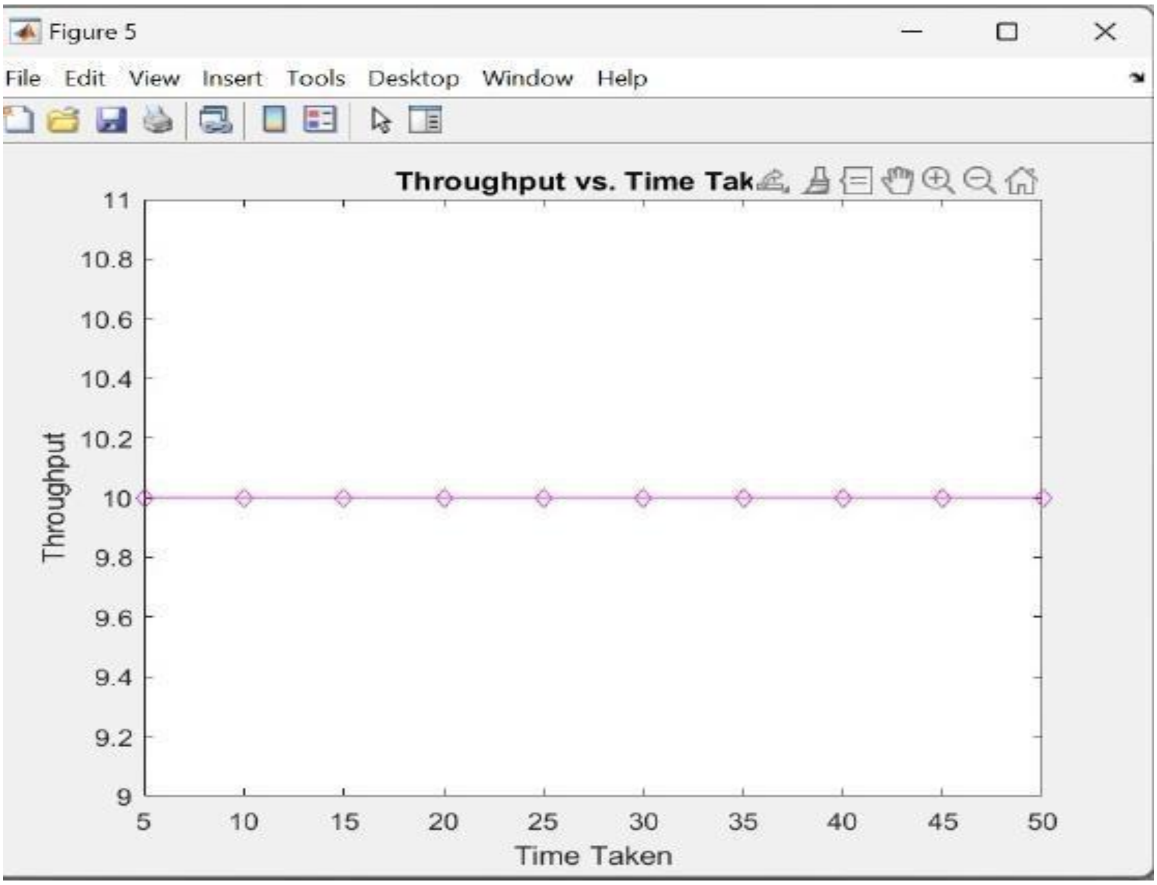


Figure 5: Throughput vs Time Taken

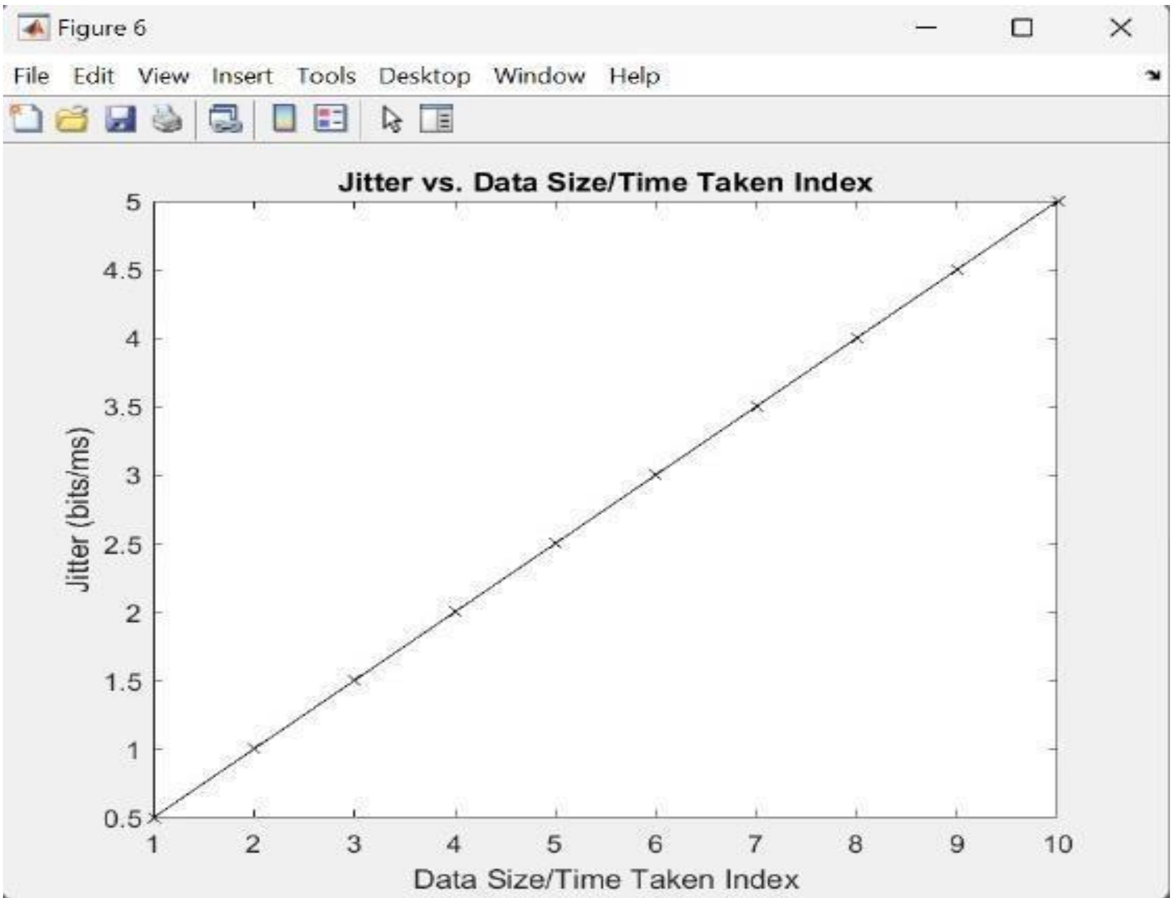
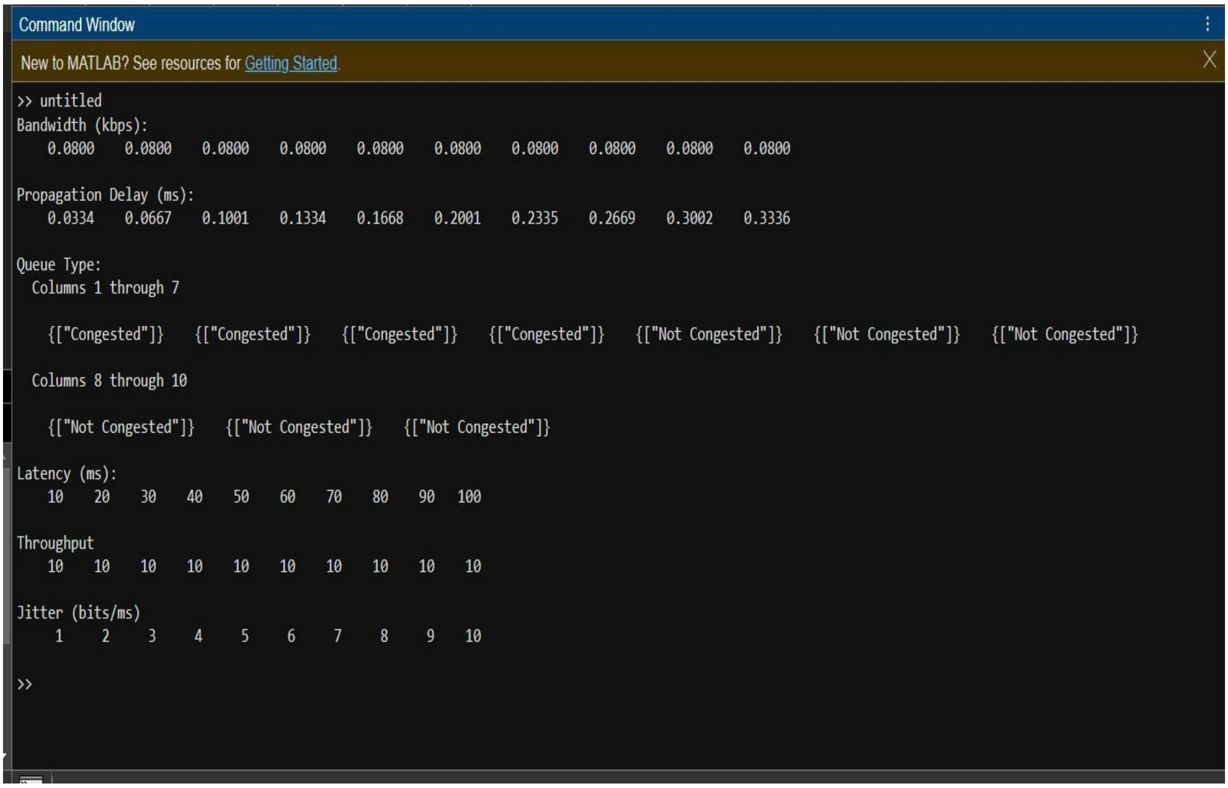


Figure 6: Jitter vs Data Size/time taken index



Parameters Value in Command Window

CHAPTER 7:- PROJECT OUTCOME

- **CO1:-**Understand the basics of Mobile Communication System.

- **CO2:**- Examine various mobile propagation model

Hence CO1 & CO2 has been satisfied.

CHAPTER 8:- CONCLUSION

In conclusion, this project successfully measures key network performance metrics such as bandwidth, propagation delay, queue type, latency, throughput, and jitter. By analyzing these metrics, network administrators gain valuable insights into the efficiency and reliability of their network infrastructure. The calculated values provide crucial information for optimizing network resources, troubleshooting connectivity issues, and ensuring quality of service. Moreover, the visual representations of these metrics through plots facilitate easy interpretation and decision-making. Overall, this project serves as a valuable tool for enhancing network performance and maintaining optimal network functionality in various realworld applications

CHAPTER 9:- REFERENCES

- **https://www.mathworks.com/help/lte/ug/umts-uplink-waveform-generation.html**
- **https://www.mathworks.com/help/lte/ug/umts-downlink-waveform-generation.html**

GitHub Link:- https://github.com/Bhavyajain12/Cellular-Technology-and-4G-/tree/main