Lab Report

Course Title: Computer Networks Lab

Autumn 2024
Section:7BF

Lab No: 4

Name of Labwork: You'll notice that this simple "routing" is not very efficient: often the packet keeps bouncing between two nodes for a while before it is sent to a different direction. This can be improved somewhat if nodes don't send the packet back to the sender. Implement this.
Again, In this model, there is only one message underway at any given moment: nodes only generate a message when another message arrives at them. We did it this way to make it easier to follow the simulation. Change the module class so that instead, it generates messages periodically. The interval between messages should be a module parameter, returning exponentially distributed random numbers.

Student's ID : C213246
Date of Performance : 2 3 .10.2024
Date of Submission : 2.11.2024
Team Name : ProtocolPros

Marks

:

Name of Labwork: You'll notice that this simple "routing" is not very efficient: often the packet keeps bouncing between two nodes for a while before it is sent to a different direction. This can be improved somewhat if nodes don't send the packet back to the sender. Implement this.

Again, In this model, there is only one message underway at any given moment: nodes only generate a message when another message arrives at them. We did it this way to make it easier to follow the simulation. Change the module class so that instead, it generates messages periodically. The interval between messages should be a module parameter, returning exponentially distributed random numbers.

1. Introduction:

In this lab, we explore basic network routing principles and examine ways to enhance message handling in a simulated network using OMNeT++. The primary objectives are to construct a simple network of interconnected nodes, observe packet transmission behavior, and implement improvements to optimize message routing.

2. Constructing Network(NED):

In this section, we define the network structure using OMNeT++'s NED (Network Description) language. The NED file specifies the layout of nodes and their connections, setting up a topology that mimics a small network.

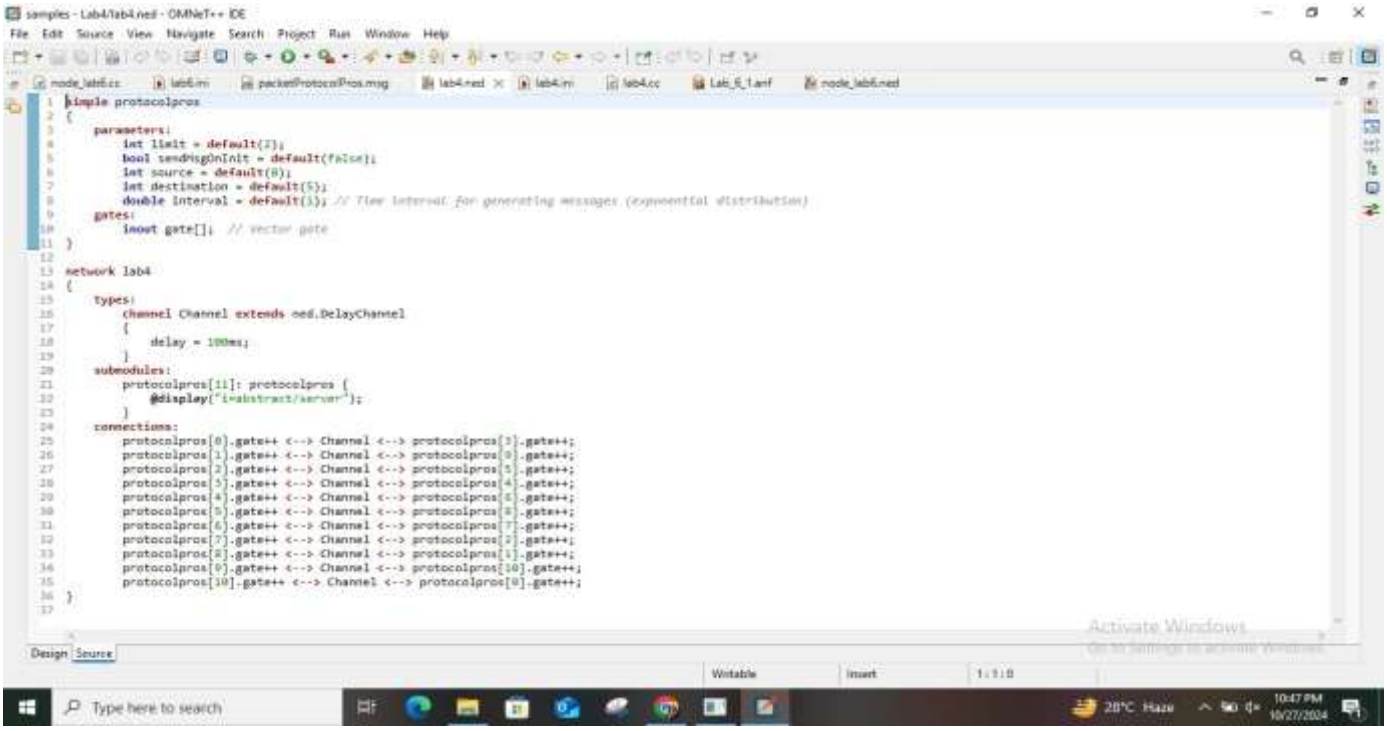

Figure 1: Laptop view of NED file

3.Building Module(C++ file):

In this section, we describe the C++ implementation that defines the behavior of each node in the network. The main functionalities include message handling, message forwarding, and periodic message generation.

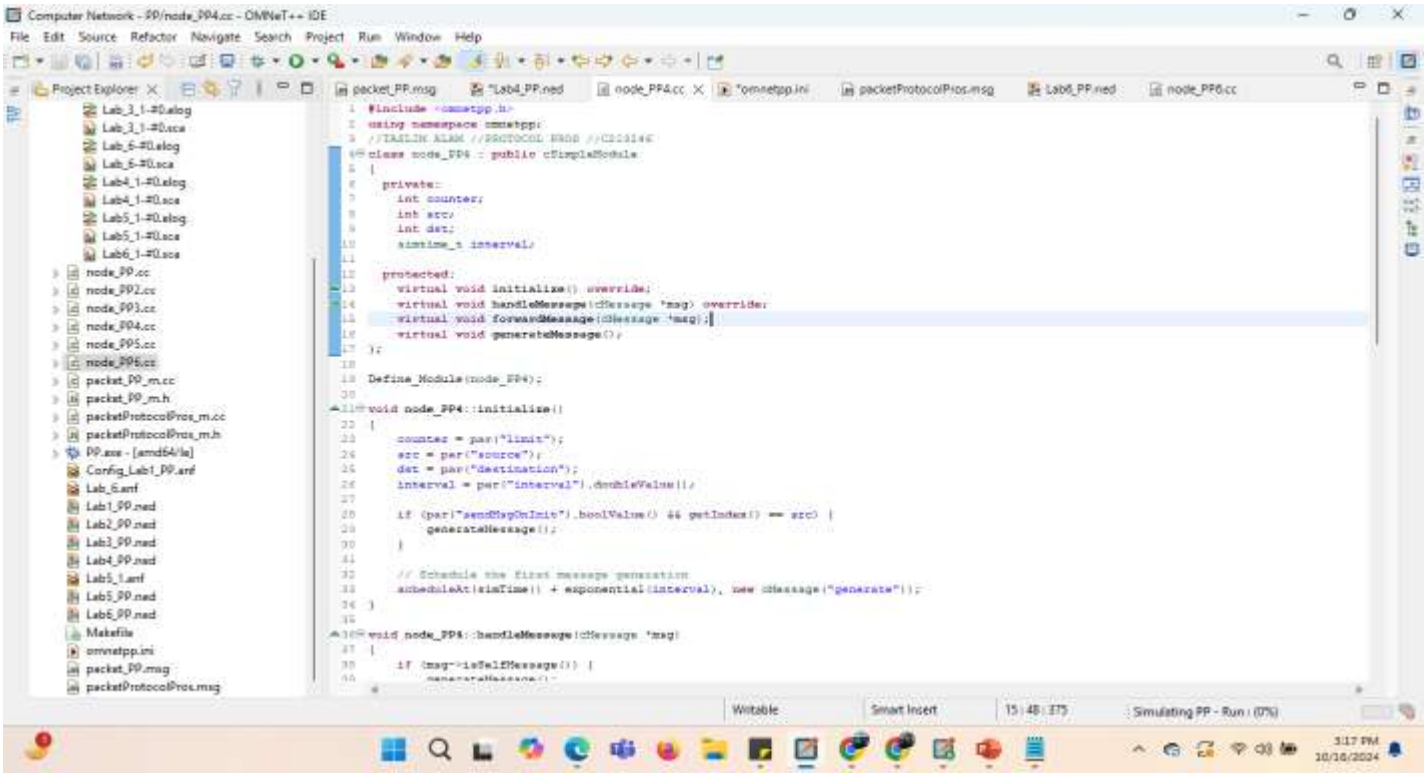

Figure 2.1: Laptop view of cc file

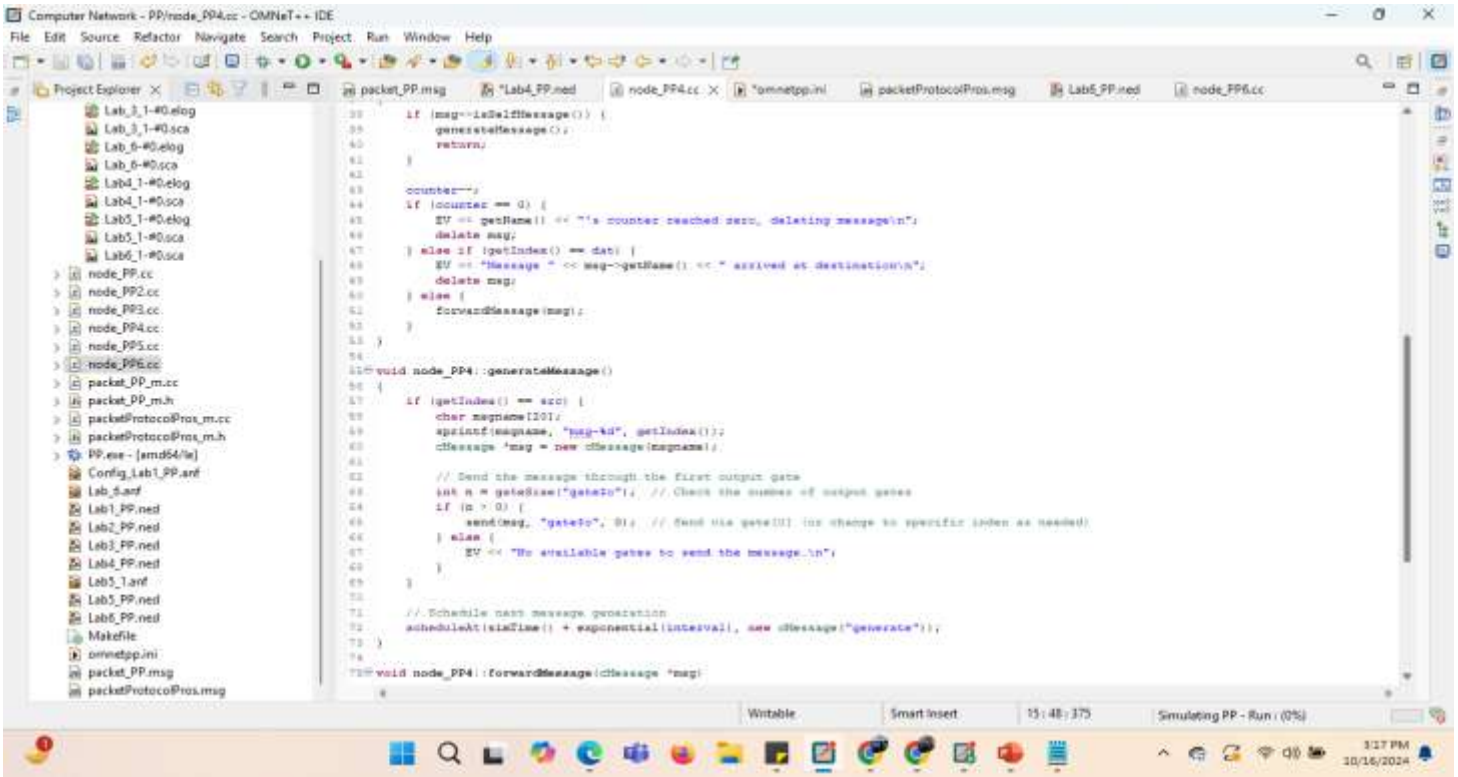

Figure 2.2: Laptop view of cc file

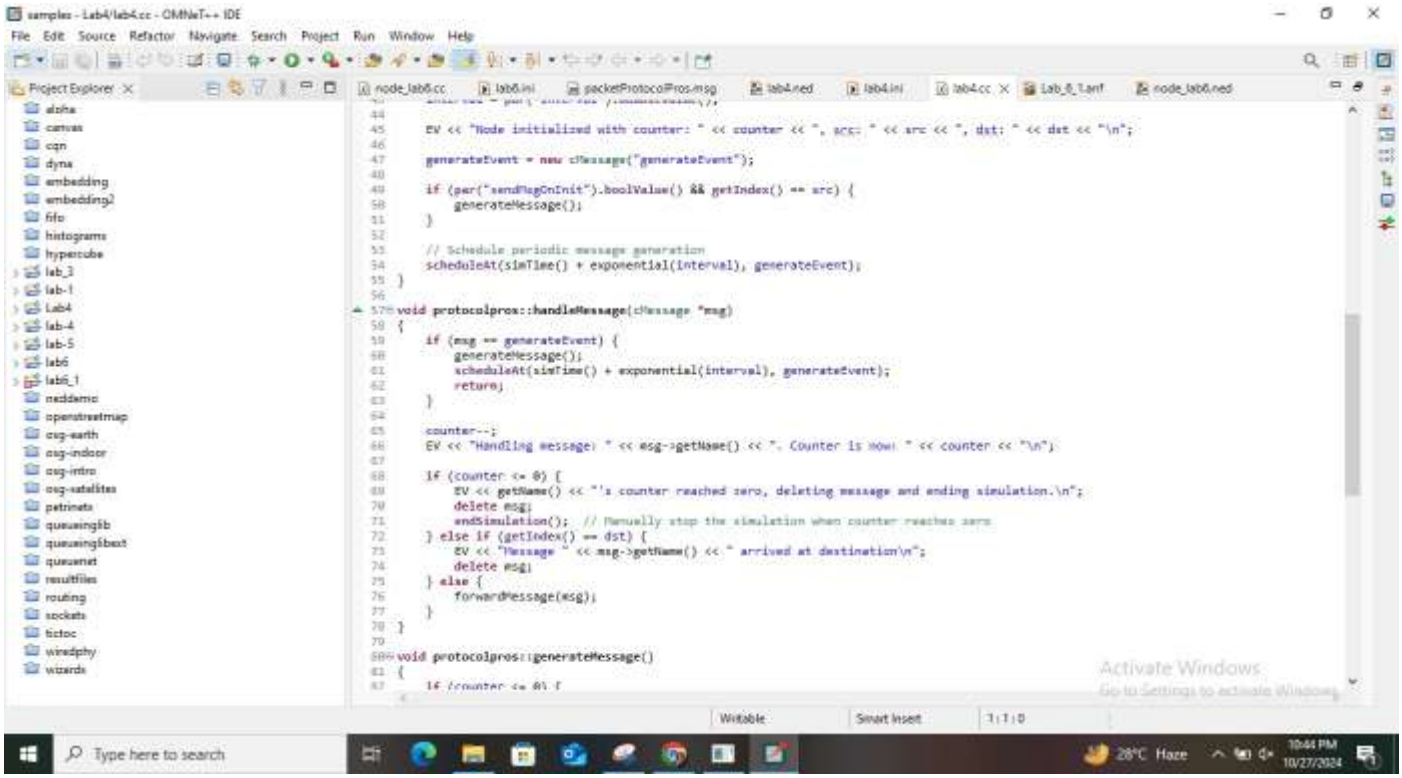

Figure 2.3: Laptop view of cc file

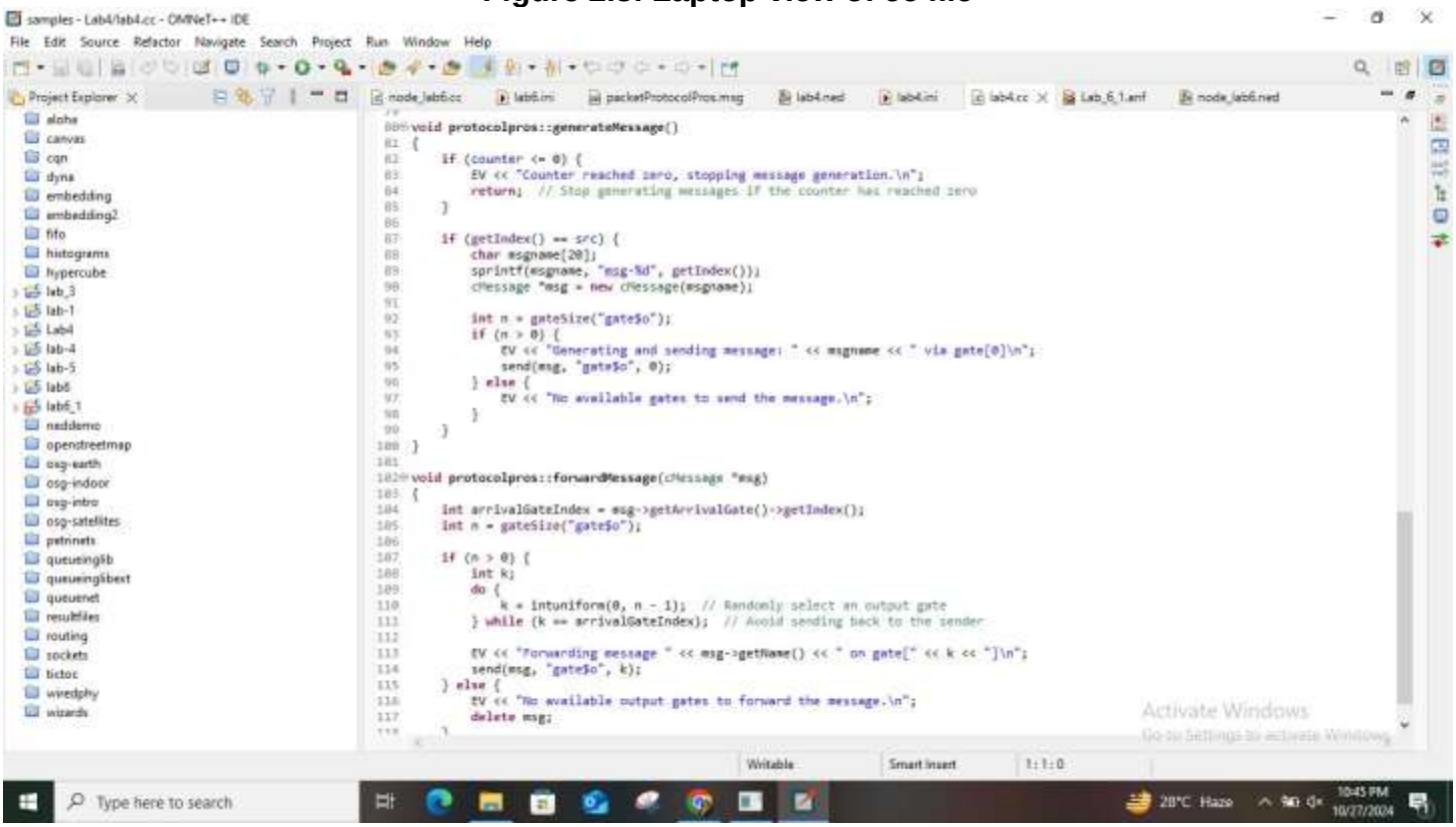

Figure 2.4: Laptop view of cc file

4.Initializing simulation(ini file):

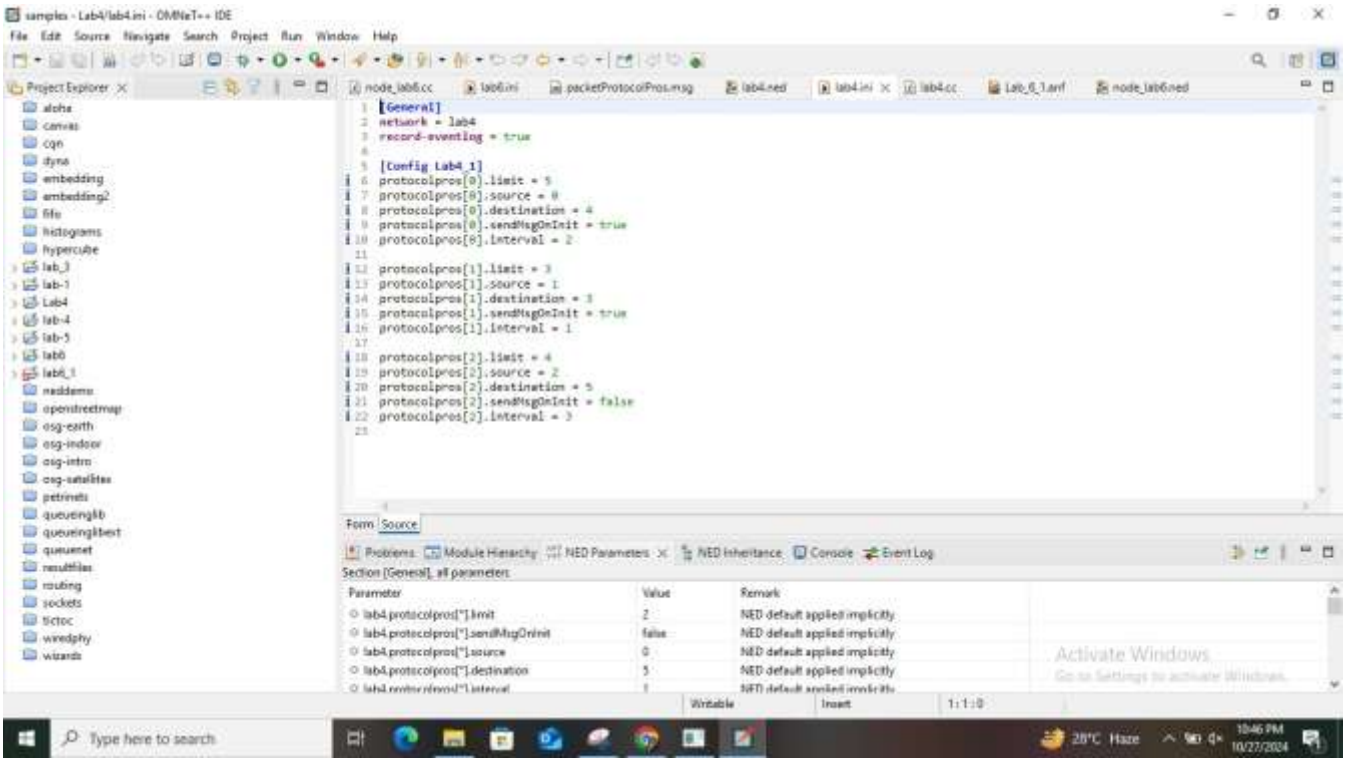

Figure 3: Laptop view of ini file

5.Experiment:

In this experiment, we observe the message routing behavior in the network after implementing the periodic message generation and improved forwarding logic. The experiment is conducted by running the simulation and visualizing the paths that messages take to reach their destination.

Steps to Conduct the Experiment

- Configure the simulation parameters in the INI file.
- Run the simulation and observe the routing of messages between nodes.
- Record data on the number of messages generated, average travel time, and any observed routing loops.

Expected Outcome:
With the improved routing logic, messages should avoid bouncing back to the previous node, reducing redundant transmissions. Additionally, the periodic message generation should simulate a more realistic network traffic pattern.

6.Result and Analysis:

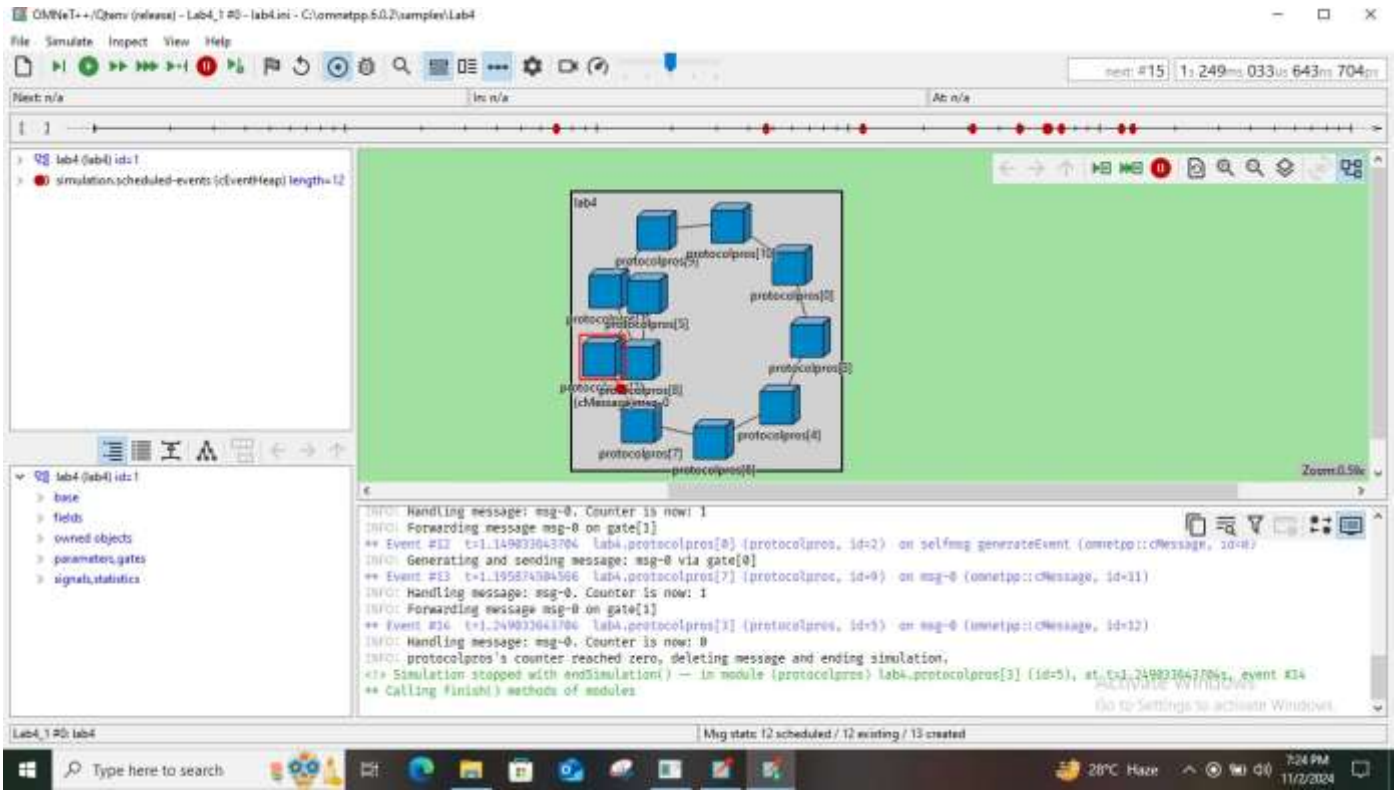

Figure 5.1: Laptop view of simulation and result

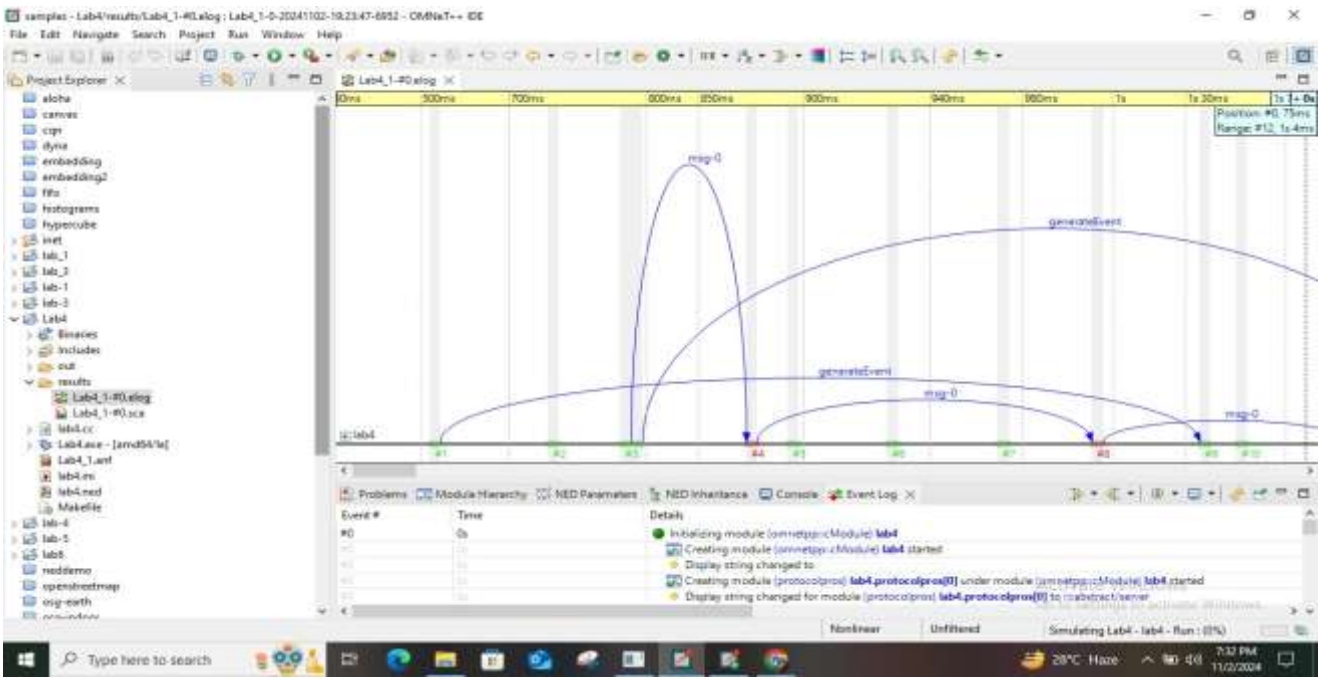

Figure 5.2: Laptop view of Line chart

Result and Analysis:

Observations
During the simulation, messages were successfully forwarded from the source to the destination without unnecessary bouncing between nodes. The changes made in the forwarding logic improved routing efficiency, as shown by reduced message delays and decreased network congestion.

Analysis
The periodic message generation created a steady flow of messages, demonstrating a continuous network load. This setup allows us to examine how each node handles multiple messages in sequence and reveals how network latency and congestion can be influenced by periodic traffic patterns. The analysis shows that the improved routing logic successfully reduced message travel time and minimized redundant transmissions.

Results Summary

- Reduced packet bouncing between nodes.
- Efficient routing with fewer delays.
- Consistent message generation with minimal congestion.

7.Conclusion:

In conclusion, this lab demonstrates the importance of efficient routing in network communication. By implementing message forwarding logic that prevents redundant bouncing and adding periodic message generation, we created a more realistic and optimized network model. These improvements allow for more accurate simulations of network traffic, which is beneficial for analyzing and optimizing real-world networks.

Through this lab, we learned how routing efficiency can impact network performance and gained hands-on experience with OMNeT++ in designing and testing network protocols. Future work could explore more complex routing algorithms and further optimize network performance under various traffic loads.