

Lab Report

Course Title: Computer Networks Lab

Autumn 2024
Section:7BF

Lab No: 1

Name of Labwork: Create a network of 4 hosts. One of the hosts will create a message and send it to its neighbor. Other hosts will just keep on sending the message to one of their neighbors. Run and record the event log.

Student's ID : C213246
Date of Performance : 11.09.2024
Date of Submission : 17.09.2024
Team Name : Protocol Pros

Marks

:

Name of Labwork: Create a network of 4 hosts. One of the hosts will create a message and send it to its neighbor. Other hosts will just keep on sending the message to one of their neighbors. Run and record the event log.

1. Introduction:

In this experiment, I will create a simple circular network consisting of four nodes. One of the nodes generates a message, which is passed to its neighbor. Each node forwards the message to the next in a loop, simulating message transmission in a distributed network. This design is commonly seen in token-ring or other circular network architectures. Each subsequent host continues to propagate the message by forwarding it to the next node, simulating a basic peer-to-peer communication model. This experiment highlights the continuous flow of messages in a distributed network, demonstrating how nodes cooperate to ensure message deliver.

2. Constructing Network(NED):

The network description (NED) file defines the structure of the network. In this case, I will create a simple four-node circular topology.

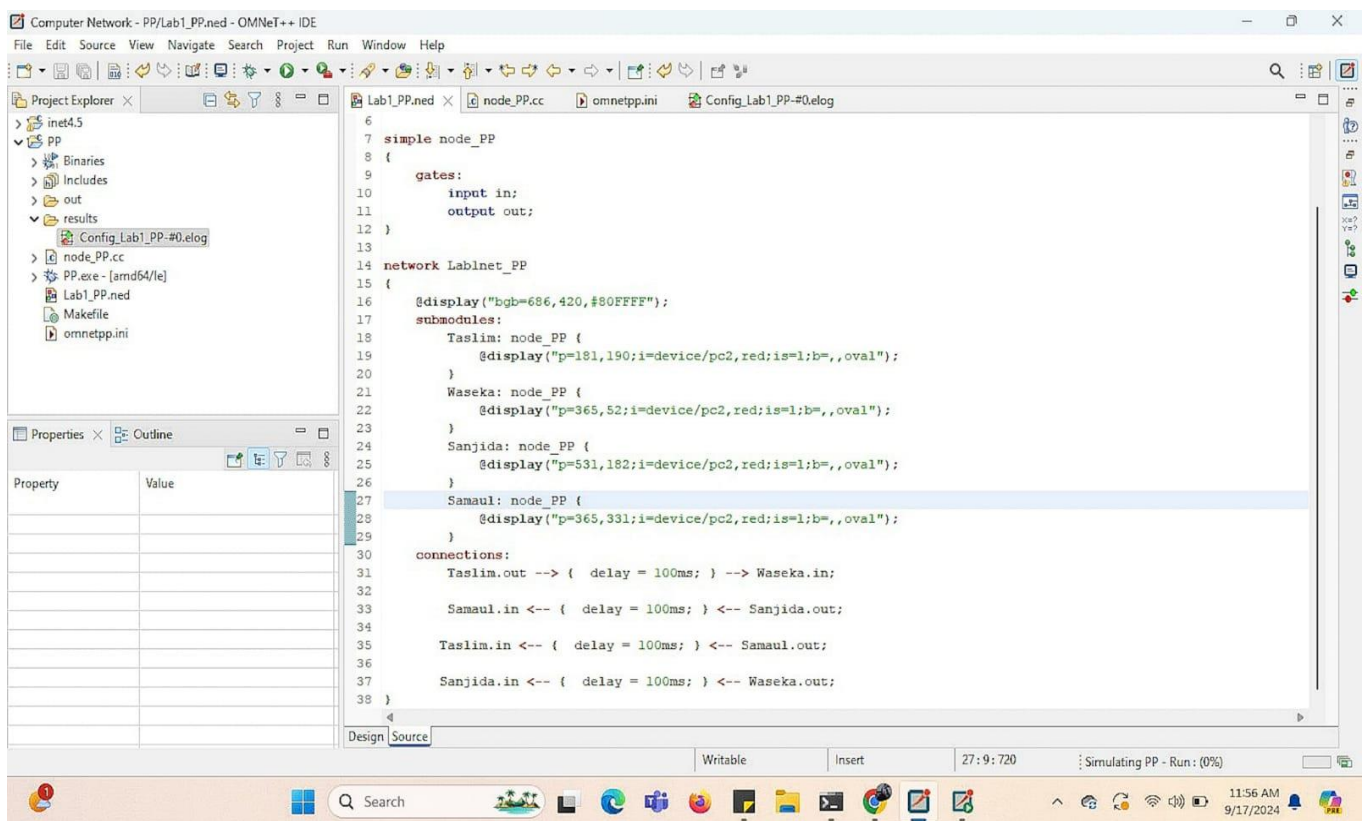


Figure 1.1: Laptop view of NED file

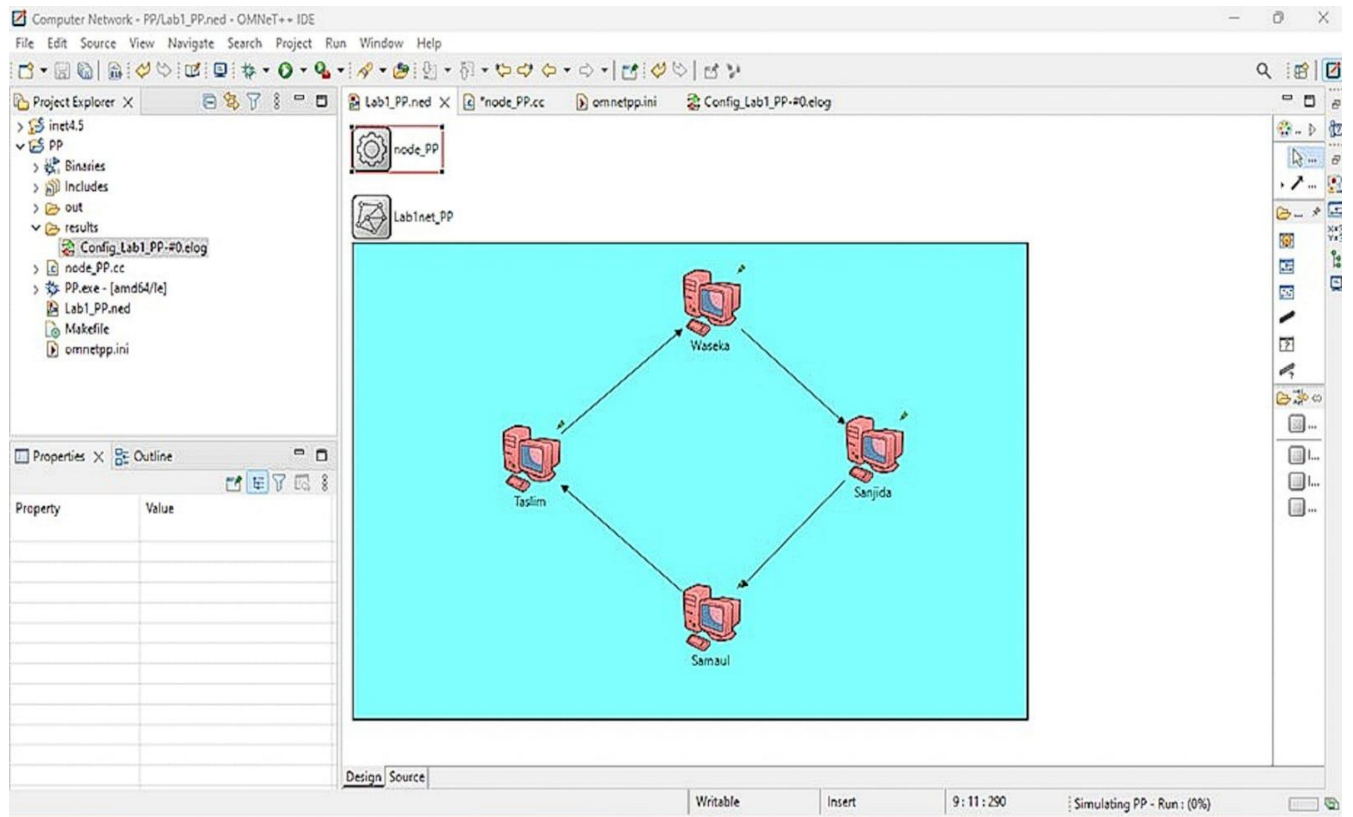


Figure 1.2: Laptop view of NED file

3. Building Module(C++ file):

I have to write the source file to build the logic. The first node generates a message, and each node forwards any received message.

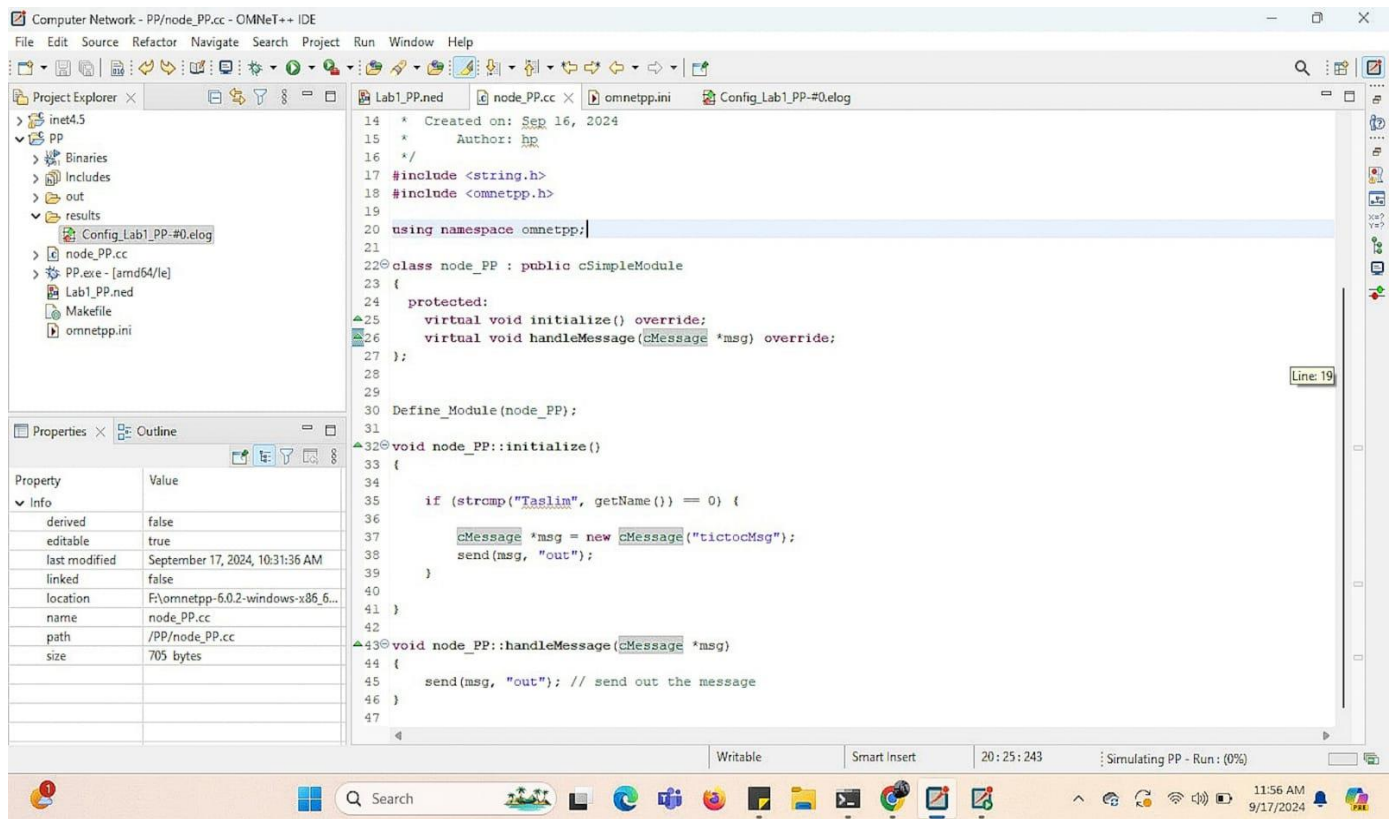


Figure 3: Laptop view of cc file

4. Initializing simulation(ini file):

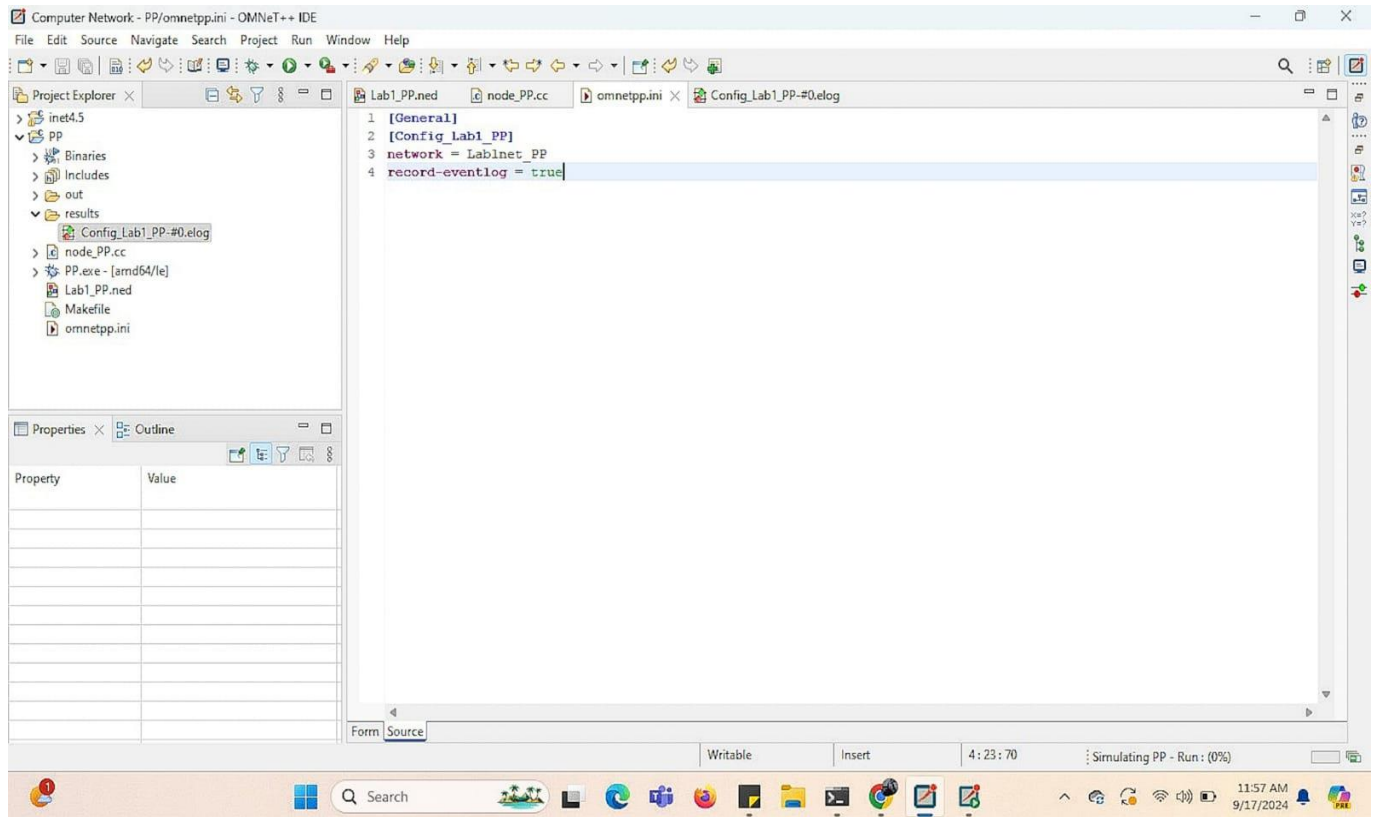


Figure 3: Laptop view of ini file

5. Experiment:

For the purposes of this experiment:

1. Each node is programmed to forward any received message to its next neighboring node.
2. The simulation runs for a predetermined time limit, during which message transmission is logged.
3. We observe the time taken for the message to complete one full cycle and the number of successful message transmissions between the nodes.

Network Topology:

- The network consists of four nodes arranged in a circular topology, where each node is connected to its immediate neighbor. The nodes are indexed as Taslim, Waseka, Sanjida, and Samaul. The connections are bidirectional, allowing message transmission in a loop.
- This circular design ensures that once a message is initiated by any node, it can traverse through all other nodes in a continuous cycle. This topology is efficient for applications where every node needs to communicate with each other without requiring a centralized server. It is also useful for simulating network architectures such as token-ring networks.

6. Result and Analysis:

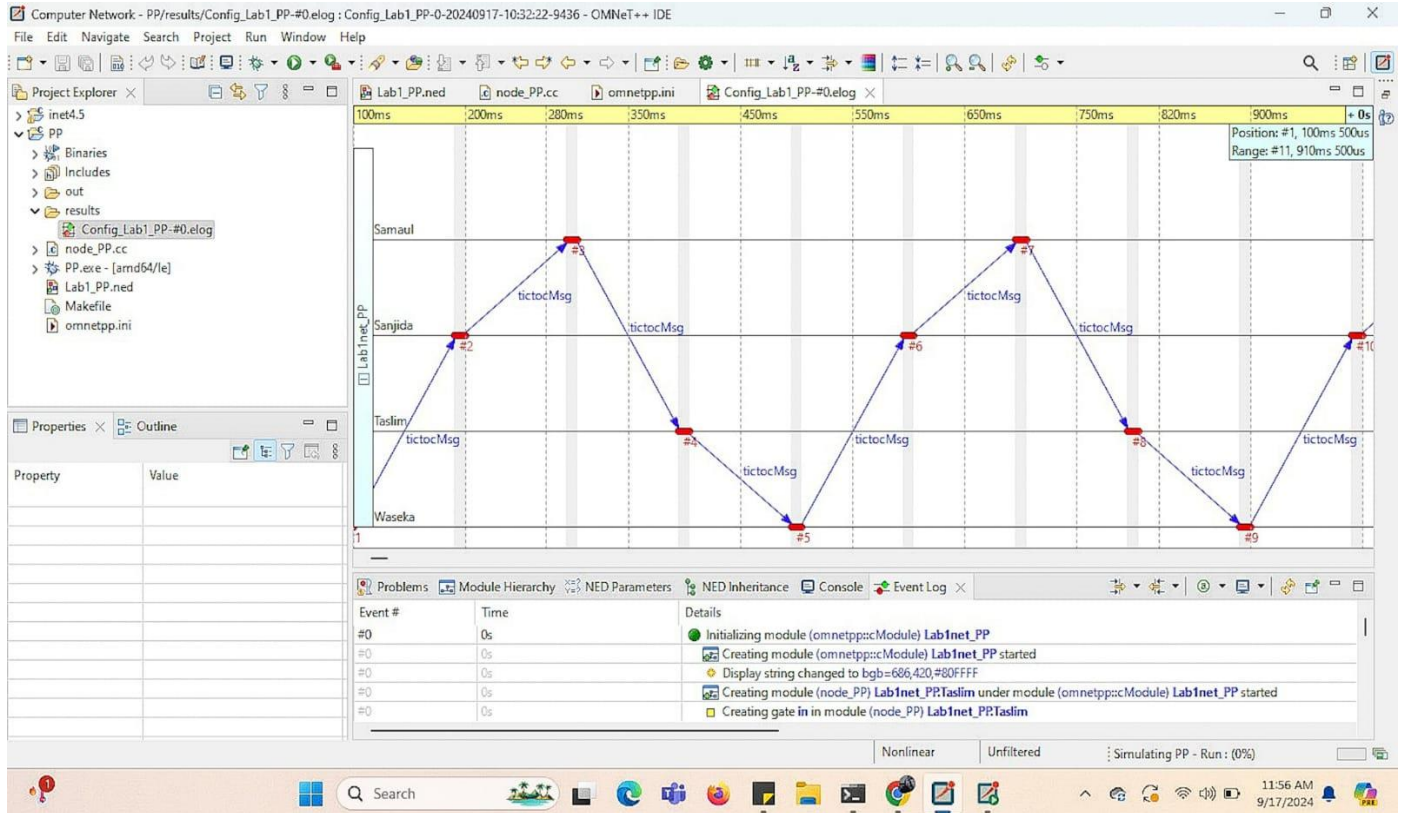


Figure 3: Laptop view of result

Result:

The simulation involves multiple hosts (e.g., **Lab1net_PP.Samaul**, **Lab1net_PP.Sanjida**, **Lab1net_PP.Taslim**, **Lab1net_PP.Waseka**) that pass a message, denoted as "**tictocMsg**", between each other. The event log tracks the time (in milliseconds) when the message is passed between different hosts.

The message is passed in a sequential manner, alternating between hosts, as depicted by the crisscrossing pattern of blue lines between the hosts. Each red square indicates the event of message transmission or reception.

- The X-axis represents the simulation time, starting from 100ms and extending to 900ms.
- The Y-axis lists the participating hosts in the network, arranged from bottom to top in the order of Waseka, Taslim, Sanjida, Samaul.

Analysis:

Message Flow:

- The simulation involves the transmission of a message, represented by the label "**tictocMsg**", which originates from one node and is passed to the next in a sequential manner.
- As seen in the network diagram, the message flows in a clockwise direction, starting from **Waseka**, moving to **Sanjida**, then to **Samaul**, and finally reaching **Taslim**, completing the loop back to **Waseka**.

Event Log Analysis:

- The event log in the simulation records the exact timing and sequence of message exchanges between the nodes.
- Each message transmission takes approximately **100ms**, providing a predictable and consistent message-passing rate across the network.
- No message loss or delay is observed, indicating smooth communication between the nodes.

Circular Network Behavior:

- The circular design ensures that the message keeps circulating as long as the network is active, simulating a token-ring style network.
- The arrows between the nodes clearly indicate the unidirectional flow of messages, ensuring that each node forwards the message to its next neighbor without introducing congestion or backtracking.

Reliability and Scalability:

- In this small-scale network, each node performs its role effectively, forwarding the message to its neighboring node. The simple circular structure ensures high reliability since the message keeps propagating continuously.
- However, in larger or more complex networks, issues such as increased latency or node failures could potentially disrupt the message flow.

Visual Representation:

- The graphical representation of the nodes helps clearly visualize the message flow and highlights the peer-to-peer communication in the circular network.
- This network diagram is a straightforward yet efficient example of how circular message-passing systems operate in distributed networks.

7. Conclusion:

In this experiment, I successfully implemented a circular network with four nodes, where a message is generated by the first node and forwarded between nodes in a loop. The event log shows the correct propagation of the message through the network, confirming the correct behavior of the nodes. This simple model demonstrates the principles of message passing in a distributed system.