**Experimental Report**

First, Latest NS3 package is downloaded and installed (using the instructions mentioned in the official site) in the mac and to build and run installed CMake to compile the build plan. And installed Wireshark with can be downloaded and installed on the mac.

TASK 1:

Task1 is like the first.cc example in the ns3 package. For this task, two nodes are used, one for the client node and another for server node. And a PointToPoint connection is established with data rate as 10Mbps with a delay of 2 milliseconds. And changed IP Address to 192.168.2.0. And changed Server port from 9 to 63 along with package size from 1024 to 256 bytes. And to get pcap files as output it is required to enable pcap files using P2P.EnablePcapAll(“Task1”); before running the simulation, this will result in generating .pcap files for this simulation which are used to trace the packets flow.

Below are the changes made on first.cc for mentioned for task 1:

**­Graphical user interface, application

Description automatically generated**

Below is the output generated for task 1: It is clear from the picture that 256 bytes of data sent from client to server which was running in IP 192.168.2.2 on port 63 and the same way server is sending same data as a reply to the client which was running on port 49153. This clearly explains that client is communicating with server without any issues using p2p connection.

**Text

Description automatically generated**

Below are the packet traces in **wire shark** for task 1:

Our server is started 1sec earlier than the client sending a request to the server. So, the request is successful and able to deliver message to the server. From the picture it is evident that the packet is being sent from client to the server through assigned channel and destination address. And if we are sending more data than expected through the channel, the data is fragmented into multiple frames and the same happens from server to client as well.

**Table

Description automatically generated**

TASK 2:

In this task I have simulated network with 2 nodes on point-to-point link using PointToPointHelper class with a speed of 10Mbps and a delay of 1ms and 6 nodes on a shared bus operating under CSMA using CsmaHelper class with a speed of 100Mbps and a delay of 10 microseconds. And to install protocol stack I used InternetStackHelper. And to assign IP address to our device interfaces I have used Ipv4AddressHelper for both point-to-point and csma devices with 192.168.1.0 and 192.168.2.0 respectively. I created client using UdpEchoClientHelper and server application using UdpEchoServerHelper by providing port number 64 as the default parameter. Communication between different routes will happen using global route manager which can be created by using Ipv4GlobalRoutingHelper::PopulateRoutingTables(); which is included before running simulation. The two client nodes send UDP echo packets to the server at times 3s, 5s from Node 0 and at 2s, 4s and 6s from Node 1 to the server and the simulation results are as follows:

Text

Description automatically generated

We can clearly see from the output image that client sent 256bytes from Node 1 (192.168.2.1) to the server at 2 sec, 4sec and 6sec respectively and another client which is node 0 (192.168.1.1) sent 128 bytes to the server at time 3sec and 5sec respectively and server has received these responses and replied the same packets back to the client.

Below is the analysis from the pcap files using Wireshark:

Even if we send packets from the different nodes at the same time, the server can handle those messages one after the other without discarding them due to collision. And the data transfer also has no issues as the node 0 sends data to the node 1 and node 1 itself sending data to the server.

Graphical user interface

Description automatically generated

From the logs of pcap extraction list the first call is a broadcast call with in the ethernet which is an address resolution protocol where it is asking for destination in the csma “Who has 192.168.2.6? Tell 192.168.2.1” and the response of ARP is “192.168.2.6 is at 00:00:00:00:00:08”. Once the node resolution is done it started transferring the data to the server. The data transfer is done using the UDP protocol. And rest are the actual data transfer between the clients and server and how data is transferred at the time that we defined and how server has handled those.

Finally, I learned how to create a network using NETWORK SIMULATOR (NS3). How can I initiate transaction between multiple nodes in the network, what generally happens in the real network when we transfer larger data and how each layer is responsible in data transmission and the way data transferred between nodes using different protocols, source to destination routing with IP routing tables. Also, I learned how to analyze and trace a packet using Wireshark which can also help in capturing live traffic from my device.