Submission materials:

1. Generate a brief report following the report template in Canvas explaining:
    a. For each task, the experimental setup: indicate which part of the code you have modified
    b. For each task, the results: comment and include a screenshot of the result of the simulation and screenshots of the packet traces in Wireshark
    c. For each task, the conclusions: what did you learn?
2. In your submission, include also your modified .cc files properly named.

Remember: the laboratory assignments can be conducted either individually or in couples. If you want to work in couples, please enroll in a Lab Assignment teams in Canvas. Then, only one of the team members will need to submit the assignment.

Task 1 (30 points):

Modify the example seen in class, *first.cc*, to simulate the following network:

- 2 nodes
    - 1 network interface at each node
- Point-to-point link:
    - Data Rate: 10 Mbps
    - Delay: 2 ms
- IP address assignment:
    - 192.168.2.0/24
- Application:
    - UDP Echo Server on port 63
    - Packet size: 256 bytes
- Use the same values as in the example for the rest of the parameters

Compile and run the simulation. Visualize the packet trace file with Wireshark. Confirm that everything works as you would expect.

Task 2 (70 points):

Starting from *second*.cc in the Tutorials folder within your ns-3 installation, create and simulate a network with the following architecture:

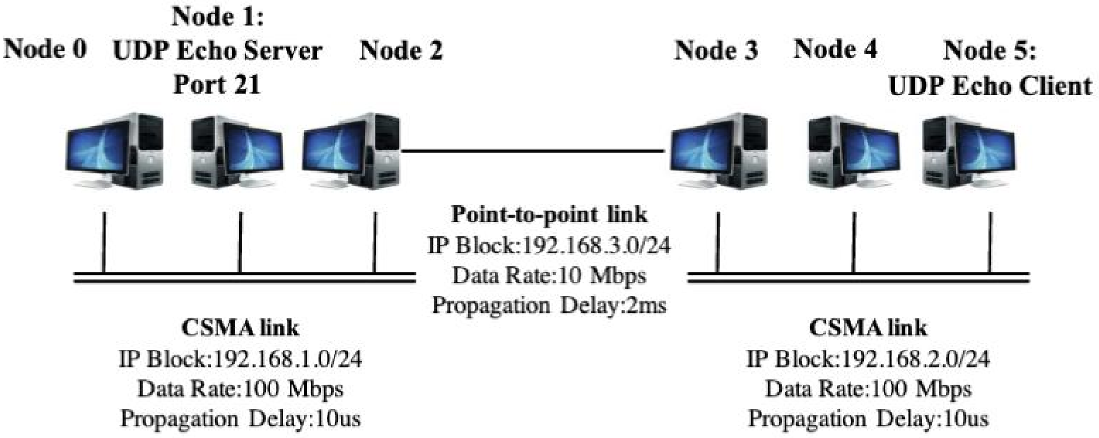

Figure 1: Network infrastructure

Detailed information:

- The network contains:
    - 3 nodes in the first shared bus operating under CSMA
    - 3 nodes in the second shared bus operating under CSMA
    - 2 nodes in the point-to-point link
        - Note that Nodes 2 and 3 in Figure 1 have two network interfaces, one for each link to which they are connected
- The applications running in the network are:
    - UDP Echo Server at Node 1:
        - Listening on port 21
    - UDP Echo Client at Node 5:
        - Sends 2 UDP Echo packets to the server at times 4s and 7s
- Enable packet tracing only in Nodes 2 and 4.

Verify that the network behavior is as expected, by capturing the packet traces and utilizing Wireshark to analyze them.