CS331: Computer Networks Assignment 1
Submission Date: Before 15-09-2025, 11:59 PM

Assignment Guidelines

- Teamwork: The assignment must be completed in pairs of two students. Only one team member (Member 1) must submit the assignment on behalf of the group.
- Submit a GitHub link to the project (public repository). The GitHub project repository should contain all the source code files, make files, and a final report (PDF).
- Include a README.md file with clear instructions for running your code.
- Code must be well-commented and follow good programming practices.
- Choose the appropriate **X.pcap file** where X = (Sum of last 3 digits of both team members) % 10. Find the pcap files here: **PCAPs**
- Late submissions will not be evaluated! Wrong PCAP selection will result in a zero score for the first question.

Task-1: DNS Resolver (80 Marks)

The purpose of this task is to understand and implement packet parsing and processing logic, specifically a custom DNS resolution support through the following:

a. Implement a client and a server with the following functionalities:
   a. The client should take input as a PCAP file, parse the PCAP file, and filter out the DNS query packets.
   b. Client will then update these DNS query packets with custom headers and send them to the server for DNS resolution.
      i. Custom header addition: A value of the current date-time and id, use timestamp in format "HHMMSSID" where
         a. HH- hour in 24-hour format
         b. MM- minute
         c. SS- second
         d. ID- Sequence of DNS query starting from 00
            E.g. first DNS query has id 00, the 6th will have id 06
      ii. Add this custom header on top of dns packet in the given format:

Custom header value(8 bytes)	Original DNS packet captured from pcap

      ii. When the server receives this message, it should extract the value from the custom header, then match it with the **predefined rules**.
   c. The final resolved address for each DNS that the client will receive as a response will be logged and added to the report.

b. For the report, submit the table containing queries, Custom header value, and their resolved IP addresses. Check the rules doc for a better understanding.

Task-2: Traceroute Protocol Behavior **(20 Marks)**

The purpose of this task is to understand how the traceroute utility works in different operating systems. Experiment on any of the two OSes (Windows, Linux, Mac).

Steps:

1. Run the following commands to trace the route to a given destination (e.g., www.google.com):
    a. On Windows: tracert www.google.com
    b. On Linux: traceroute www.google.com
2. Capture the network traffic during both executions using **Wireshark** or **tcpdump**.
3. Websites to trace any from the **document.**

Answer the following based on your observations:

1. What protocol does Windows tracert use by default, and what protocol does Linux traceroute use by default?
2. Some hops in your traceroute output may show ***. Provide at least **two reasons** why a router might not reply.
3. In Linux traceroute, which field in the probe packets changes between successive probes sent to the destination?
4. At the final hop, how is the response different compared to the intermediate hop?
5. Suppose a firewall blocks UDP traffic but allows ICMP — how would this affect the results of Linux traceroute vs. Windows tracert?

Instructions:

For all questions, provide screenshots or packet captures from your traceroute runs, and highlight relevant fields or outputs that support your answers. Include brief explanations where needed**.**