# *CS 558: Computer Systems Lab*
# *(January-May 2026)*
# *Assignment –2: Network Protocol Analysis Using Wireshark*

*Submission deadline: 11:55 PM, Thursday, 12th February, 2026*

Wireshark is a free and open-source packet sniffer and network protocol analyser tool. It helps to capture network packets and understand the structure of different networking protocols.

## Instructions

- The evaluation of this assignment will be conducted **either offline or viva voce during the lab session, during which** group members must clearly justify and explain their observations to the evaluator.
- Each group will be assigned a **specific application** (as per the Task Allocation Table). The group is required to explore the application's functionalities and capture relevant network traffic using Wireshark. Any application-specific requirements will be mentioned in the table.
- Only **one member per group** is required to upload the final submission using the provided form: **https://forms.gle/Z6kaxjAUTYcJH4GX7**
- Experiments must be performed under **varying network conditions**, such as different times of day and different locations (e.g., lab, hostel), to observe diverse traffic patterns.
- Responses should be strictly based on **observations from captured traces**. Wherever necessary, include screenshots of packet traces and clearly highlight important fields or patterns.
- In case of any ambiguity, missing information, or inconsistency in the problem statement, the **assumptions made must be explicitly stated** in the answers.
- Install **Wireshark** from www.wireshark.org and become familiar with packet capturing techniques and filtering mechanisms before starting the experiments.
- The final report must be submitted in **PDF format**, along with the captured trace files compressed into a single archive. The archive should be named according to the group number (e.g., Group_4.zip, Group_4.rar, or Group_4.tar.gz).
- If the trace files exceed 2 MB, include a cloud storage link (OneDrive, Google Drive, or Dropbox) to the traces in the report.
- **Plagiarism checks will be strictly enforced**, and any instance of unfair practices will result in negative marking equivalent to the maximum marks allotted for the assignment.
- Marks will be awarded **collectively to the entire group**, not on an individual basis.

## Questions

1. Using the captured packet traces, **identify the network protocols** observed at various layers of the protocol stack. Only include protocols that can be clearly inferred from the traces, and briefly outline their packet structures based on observations.
2. From the traces, **analyze important header fields** of the identified protocols. Discuss observed values such as source and destination IP addresses, port numbers, MAC addresses, protocol identifiers, and any other relevant fields.
3. For the assigned application, **describe the flow of message exchanges** for its core functionalities (e.g., upload, download, streaming, play, pause, etc.). Identify whether any handshake or setup phase exists and briefly explain the sequence of messages involved, if applicable.
4. Based on your observations, **justify the choice of protocol(s)** used by the application and explain how they contribute to the correct and efficient functioning of the application.
5. Perform experiments at **different times of the day** and compute key performance metrics from the collected traces, including:
    - Throughput
    - Round Trip Time (RTT)
    - Packet size distribution
    - Packet loss count
    - Number of TCP and UDP packets
    - Ratio of responses received per request sent

    Present the measured values clearly, preferably in **tabular form**.

6. Examine whether the application content is delivered from a **single server or multiple sources**. If multiple sources are involved, list the corresponding IP addresses and explain the possible reasons for such distribution.

| **Group No.** | | **Application** |
|---|---|---|
| 1 | 12 | Zoom |
| 2 | 13 | Live Sport Streaming |
| 3 | 11 | YouTube- uploading video |
| 4 | | YouTube- downloading and buffering |
| 5 | | NPTEL video lectures |
| 6 | | Twitch (live streaming video platform) or Hotstar video streaming |
| 7 | | MS-Teams |
| 8 | | P2P connectivity using Remote Desktop and<br>Software like TeamViewer |
| 9 | | Strong DC++ |
| 10 | | Online games |

## Note

- While analyzing **video or audio communication applications**, ensure that packet traces are collected for scenarios in which both communicating hosts are on the same network, as well as when one host is outside the local network.
- For experiments related to **video uploads or downloads**, the selected video content must be at least 20 minutes long to ensure meaningful traffic analysis.
- In the case of **online gaming applications**, perform gameplay sessions against opponents connected both within the local network and from external networks.
- To achieve **accurate and reliable measurements**, disable or minimize background traffic such as advertisements, auto-suggestions, or other auxiliary server communications.
- During packet capture, **avoid accessing any other internet-based applications or websites**, as they may introduce unwanted traffic into the trace files.
- Packet capture should be performed using **TCPDUMP with appropriate filters**, followed by detailed analysis in Wireshark. Direct capturing through Wireshark alone may result in packet loss due to memory constraints.
- Ensure that **Layer 2 protocols** are included and examined as part of the overall analysis.

---

## IMPORTANT

The submitted report must be **concise and analysis-focused**. Avoid generic explanations of protocol formats or theoretical functionality. Only include insights that are directly derived from the observed packet traces. Screenshots should be used **sparingly and purposefully**, strictly to support conclusions. The report must not consist solely of screenshots. All captured trace files should be submitted along with the report.