CS342: Computer Networks Laboratory

Deadline: 11:59 pm, 8 September 2024

Instructions:

- This assignment is to be done in groups.
- The programming questions can be written in C/C++/Java.
- Your code should have a readme file, a makefile, and it should be well commented. These will carry separate marks for each question.
- Submit a soft copy of the report, preferably in PDF format, together with your code in a zip file on teams. The name of the zip file should be like "Your_Groupno.zip" (example: "Group11.zip").
- Files submitted without proper naming format will not be evaluated.
- If your trace file size is larger than 2 MB, you are advised to provide the OneDrive/Google Drive/Dropbox link of the traces in your report.
- No extensions in submission are allowed. Delay in submission will lead to penalty in marks.
- Assignments submitted before the deadline will only be considered for evaluation.
- Please do not email your assignments separately to the TAs, it will not be considered for evaluation.
- Your code will be checked for plagiarism. Any kind of academic dishonesty, plagiarism, etc. will lead to penalties.
- No sharing of code between students, submission of downloaded code is allowed.
- The first instance of code copying will result in ZERO marks for the assignment. The second instance of code copying will result in a ` F' grade. Students may also be reported to the Students Disciplinary Committee, which can impose additional penalties.
- Total Marks: 50, 30(15x2, including 10 for programming and 5 for README/Makefile/comments) + 20 for VIVA

Q1) You are working for a tech company developing a sophisticated monitoring and control system for a fleet of autonomous drones. The system needs to support three types of communication:

1. **Control Commands:** These commands need to be sent from a central server to the drones with minimal delay to ensure real-time responsiveness.
2. **Telemetry Data:** Drones will periodically send telemetry data (e.g., location, speed, status) back to the central server. This data must be reliable and delivered in order, as it will be used for diagnostics and analysis.
3. **File Transfers:** Occasionally, drones will need to send large files (such as captured images or video data) to the central server. This requires a protocol that can handle efficient file transfer with low latency and high reliability, such as QUIC (Quick UDP Internet Connections).

Your task is to develop the client (drone) and server (control center) application. The system should handle multiple drones simultaneously. Additionally, implement a security feature where all messages are encrypted and decrypted using a simple XOR cipher.

Objective: Develop a tri-mode communication system within a single application:

- **Mode 1 (Control Commands):** A high-performance mode where the server sends control commands to drones with minimal latency.
- **Mode 2 (Telemetry Data):** A reliable mode where drones send telemetry data to the server. Ensure data is delivered accurately and in order.
- **Mode 3 (File Transfers):** Efficiently transfer large files from drones to the server using the QUIC protocol.

Q2) You are part of a team developing a sophisticated real-time weather monitoring system for a large city. This system includes a central server and multiple weather stations spread across the city. Each weather station continuously sends real-time weather data (e.g., temperature, humidity, air pressure) to the central server, which processes and displays this information.

The system must handle:

- **High Data Throughput:** Weather stations generate a substantial amount of data that must be managed efficiently to ensure timely updates without data loss.
- **Network Adaptability:** The system must handle varying network conditions and prevent any single weather station from overwhelming the central server or causing network congestion.

Objective: Develop the client (weather station) and server (central server) applications with the following requirements:

4. **Efficient Data Handling:** Implement a mechanism to ensure that the server can receive and process weather data from multiple weather stations without being

overwhelmed. The data transmission should be optimized to avoid overloading the server.

5. **Adaptive Data Transmission:** Implement a congestion control algorithm, specifically TCP Reno, at the weather stations to adjust their data transmission rate based on network conditions. This ensures that the network remains responsive and data is transmitted efficiently. For further details on TCP Reno, refer to this resource.
6. **Simulated Network Constraints:** Simulate a limited network environment where the server's capacity and available bandwidth are constrained. Your system should adapt to these limitations, managing data flow effectively under these conditions.
7. **Dynamic Data Compression:** Introduce a dynamic data compression feature. Weather data should be compressed before transmission and decompressed upon receipt. The compression should adapt based on data characteristics to balance between reducing data size and maintaining transmission speed.

****