

**EE4204 Computer Networks (Part 1)**  
**Socket Programming Assignment (Sem2, 2024-25)**  
Lecturer : Assoc. Prof. Mohan Gurusamy, [gmohan@nus.edu.sg](mailto:gmohan@nus.edu.sg)

---

**Instructions:**

This lab assignment focuses on implementing a client server socket program with UDP transport protocol for transferring messages using a flow control protocol. Problems 1-3 are for your practice. Problem 4 is the assignment problem. Choose appropriate values for parameters such as data unit size and error probability. Repeat the experiment several times and plot the average values in a report with a brief description of results. The details of lab schedule, demo, and rubrics to be used for assessment will be announced separately.

The solution and program must be yours based on your own effort. Our university views any form of cheating and plagiarism (like copying from other students or reports or web sources) very seriously which will result in severe penalty.

**Practice Problems:**

1. Develop a socket program in UNIX/Linux that uses (i) TCP as the transport protocol and (ii) UDP as the transport protocol for transferring a short message between a client and server. The client sends a string (input by the user) to the server and the server prints the string on the screen after receiving it.
2. Develop a TCP-based client-server socket program for transferring a large message. Here, the message transmitted from the client to server is read from a large file. The entire message is sent by the client as a single data-unit. After receiving the file, the server sends an ACK message to the receiver. Verify if the file has been sent completely and correctly by comparing the received file with the original file (“diff” command could be used). Measure the message transfer time and throughput.
3. Develop a TCP-based client-server socket program for transferring a large message. Here, the message transmitted from the client to server is read from a large file. The message is split into short data-units which are sent one by one without waiting for any acknowledgement between transmissions of two successive data-units. Verify if the file has been sent completely and correctly by comparing the received file with the original file. Measure the message transfer time and throughput for various sizes of data-units.

**Assignment Problem**

4.

Develop a UDP-based client-server socket program for transferring a large message. Here, the message transmitted from the client to server is read from a large file. The message is split into short data-units which are sent by using stop-and-wait flow control. Also, a data-unit sent could be damaged with some error probability. Verify if the file has been sent completely and correctly by comparing the received file with the original file. Measure the message transfer time and throughput for various sizes of data-units. Also, measure the performance for various error probabilities and for the error-free scenario.

Choose appropriate values for parameters such as data unit size and error probability. You can simulate errors according to the frame error probability. You are free to implement the ARQ in your own way, but with stop-and-wait. For example, you may want to avoid TIMEOUTs and handle retransmissions in some other way (you may want to choose negative acknowledgement). You may also want to simulate errors with a certain probability by generating a random number. For example, to simulate error probability 0.1, generate a random number in the range between 0 and 999; if this number falls within the range 0 to 99, then assume there is an error in the data unit received, otherwise there is no error. Repeat the experiment several times and plot the average values in a report with a brief description of results, assumptions made, etc. Choose at least six values for data unit size in the range between 200 and 1400 bytes. Choose at least six values for error probability in the range between 0.0 and 0.40. Include the following performance figures in your report:

- 1) Transfer time vs error probability (2 graphs; size = 500 and size = 1000 bytes)
- 2) Throughput vs error probability (2 graphs; size = 500 and size = 1000 bytes)
- 3) Transfer time vs data unit size (2 graphs; error probability = 0, error probability = 0.1)
- 4) Throughput vs data unit size (2 graphs; error probability = 0, error probability = 0.1)

**[ALL THE BEST]**