Data Communication and Computer Network Laboratory

Master of Computer Application

Second Year, Second Semester

Session: 2024-25

Assignment - I

Date: 08/08/2024

1 Basic TCP Client-Server Communication

Objective: To understand basic client-server communication using TCP sockets.

Tasks:

1. Server:
   (a) Create a TCP server that listens on a specified port.
   (b) Accept a connection from a client.
   (c) Receive a message from the client.
   (d) Send a response back to the client.
   (e) Close the connection.
2. Client:
   (a) Create a TCP client that connects to the server.
   (b) Send a message to the server.
   (c) Receive a response from the server.
   (d) Close the connection.

2 File Transfer

Objective: To implement a simple file transfer protocol using TCP sockets.

Tasks:

1. Server:
   (a) Create a TCP server that listens on a specified port.
   (b) Accept a connection from a client.
   (c) Receive a file name from the client.
   (d) Send the requested file to the client if it exists.
   (e) Close the connection.

2. Client:
   - (a) Create a TCP client that connects to the server.
   - (b) Send a file name to the server.
   - (c) Receive the file from the server and save it locally.
   - (d) Close the connection.

3 HTTP Server

Objective: To implement a simple HTTP server that can serve static web pages.

Tasks:

1. Server:
   - (a) Create a TCP server that listens on port 80 or 8080.
   - (b) Parse incoming HTTP requests.
   - (c) Serve static HTML files based on the request.
   - (d) Implement basic HTTP response headers.
2. Client:
   - (a) Use a web browser or a tool like `curl` to send HTTP requests to the server.

4 Basic TCP Client-Server Communication

Objective: To understand the differences between TCP and UDP by implementing simple UDP client-server programs.

Tasks:

1. Server:
   - (a) Create a UDP server that listens on a specified port.
   - (b) Receive messages from clients.
   - (c) Send responses back to the clients.
2. Client:
   - (a) Create a UDP client that sends messages to the server.
   - (b) Receive responses from the server.

Languages: Python, C, or Java