University of Dhaka

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Lab Report 3: Implementing File transfer using Socket Programming and HTTP GET/POST requests

Submitted By:

Name: Meherun Farzana

Roll No : 05

Name: Mohd. Jamal Uddin Mallick

Roll No : 07

Submitted On :

February 8, 2024

Submitted To :

Dr. Md. Abdur Razzaque

Dr. Md. Mamun Or Rashid

Dr. Muhammad Ibrahim

Redwan Ahmed Rizvee

1 Introduction

File transfer is a critical aspect of modern networking applications, facilitating the exchange of data between systems. This lab report focuses on implementing file transfer using two essential techniques: socket programming and HTTP GET/POST requests. Socket programming provides a foundational framework for establishing communication channels between client and server applications, while HTTP protocols offer standardized methods for data exchange over the web.

1.1 Objectives

- Develop a file transfer mechanism using socket programming to establish communication between client and server applications.
- Implement HTTP GET and POST requests to facilitate file retrieval and submission over the network.
- Evaluate the performance and scalability of the implemented file transfer system under varying network conditions and file sizes.

2 Theory

Socket programming is the foundation of network communication, facilitating the establishment of connections between client and server applications for data exchange. HTTP (Hypertext Transfer Protocol) serves as the backbone of web communication, enabling clients to request resources from servers using methods such as GET and POST. In file transfer implementations, socket programming enables connection setup and data transmission, while HTTP requests facilitate resource retrieval (GET) and data submission (POST) between client and server. Combining these techniques provides a robust framework for efficient and standardized file transfer over networks.

3 Methodology

3.1 Server

The server is initialized on a specific port and it listens for incoming requests. Whenever a client requests to connect, the server accepts the connection and provides necessary services.

In case of sockets, the server lets the client choose whether they want to get the list of available files, upload or download a file. The server can handle any type of file including text, audio, image, video etc. of all formats.
In case of HTTP, the server is an HTTP server, serving GET and POST requests from clients.

3.2 Client

The client side is a program requesting service from the server. It tries to connect to the server address mentioning the particular port on which the server is serving. When the server accepts the connection, it is provided with the options of service the client can avail. The client selects whether they want to see the list of available files, upload or download a file.
In case of HTTP, the client can make a *GET* request to the server for a specific resource and avail it. Also it can make a *POST* request for uploading new files into the server.

4 Experimental result

Some Snapshots of the Client Side queries can be seen in the following figures:

Figure 1: Content of Server

Figure 2: Client Side View after successful request

5 Experience

1. We had to learn how to enable the server to handle multiple client at once.

2. We had to see some examples of how to use HttpServer package in Python

References

[1] Multithreading in python. *GeeksforGeeks*, aug 3 2022. [Online; accessed 2023-01-25].

[2] File transfer using tcp socket in python. *GeeksforGeeks*, apr 17 2023. [Online; accessed 2023-01-25].

[3] Python Basics. Create a python web server. aug 31 2021.