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:

1. Syed Mumtahin Mahmud, Roll: 50
2. Nazira Jesmin Lina, Roll: 55

Submitted On :

January 31, 2023

Submitted To :

Dr. Md. Abdur Razzaque

Md Mahmudur Rahman

Md. Ashraful Islam

Md. Fahim Arefin

1 Introduction

File transfer over a network can be done using either socket programming or HTTP requests.

Socket programming involves directly sending data over a network connection using low-level APIs. It requires manual handling of data transfer, data chunking, error handling, etc. It is typically faster than HTTP as it has lower overhead.

HTTP (Hypertext Transfer Protocol) is a higher-level protocol that is used to transfer data over the internet. HTTP GET and POST are two commonly used methods for sending data over HTTP. GET is used for retrieving data from a server, while POST is used for sending data to a server. The client-server model is a common architecture used in many networked systems, including the World Wide Web, email, and file sharing.

1.1 Objectives

- **Socket Programming:** The main objective is to transfer files between two or more systems over a network in a low-level, efficient, and fast manner.
- **HTTP GET/POST Requests:** The objective is to transfer files between a client and a server using the standard HTTP protocol. The goal is to utilize the built-in error handling, security, and data transfer abstractions provided by HTTP.

2 Theory

2.1 File Transfer via Socket:

File transfer via sockets refers to the process of transferring a file from one system to another over a network connection using socket programming. Socket programming is a low-level technique for sending data over a network, where the user has direct control over the transfer process.

In file transfer via sockets, a socket is created on each system, and a network connection is established between the two systems using the socket API. The file to be transferred is divided into smaller chunks, and each chunk is sent over the network connection. The recipient system receives the chunks and reassembles the original file.

File transfer via sockets offers more control over the transfer process compared to higher-level protocols like HTTP, but also requires manual handling of error handling, data chunking, and other aspects of the transfer.

Overall, file transfer via sockets is a fast and efficient way to transfer files over a network, but requires more manual work and a deeper understanding of network programming.

2.2 File transfer via HTTP:

File transfer via HTTP refers to the process of transferring a file from a server to a client or from a client to a server using the HyperText Transfer Protocol (HTTP). HTTP is a standard protocol for transmitting data over the web, and is widely used for web communication between a client (e.g. a web browser) and a server (e.g. a web server).

In file transfer via HTTP, the client sends a request to the server using either an HTTP GET or POST request. For a GET request, the server sends the requested file to the client, while for a POST request, the client sends the file to the server. Overall, file transfer via HTTP is a simple and convenient way to transfer files over a network, but may not be as fast or efficient as other methods, such as socket programming.

3 Methodology

3.1 Server

File transfer via socket and HTTP both involve a server as the recipient of the file transfer.

3.1.1 File transfer via socket

The server creates a socket and listens for incoming connections from clients using the socket API. When a client connects, the server accepts the connection and creates a new socket for communication with that client. The server receives data from the client in small chunks and reassembles the original file. The server stores the received file in a specified location and closes the socket when the transfer is complete.

3.1.2 File Transfer via HTTP

The server runs an HTTP server software, such as Apache or Nginx, that listens for incoming HTTP requests from clients. When a client sends an HTTP GET or POST request, the server sends or receives the file based on the request type. The server handles the transfer of the file using the HTTP protocol, which includes built-in error handling and data chunking. The server stores the received file in a specified location and sends a response back to the client indicating the success or failure of the transfer.

3.2 Client

The client in file transfer via socket and HTTP both initiate the transfer and send the file to the server.

3.2.1 File Transfer via Socket

The client creates a socket and connects to the server's socket using the socket API. The client divides the file into smaller chunks and sends each chunk to the server over the network connection. The client handles error handling, such as retransmitting lost packets, and manages the transfer process. The client closes the socket when the transfer is complete.

3.2.2 File Transfer via HTTP

The client sends an HTTP GET or POST request to the server, specifying the file to be transferred. The client handles error handling and any necessary authentication, such as providing a username and password. The client receives the file from the server or sends the file to the server, depending on the request type. The client receives a response from the server indicating the success or failure of the transfer.

In both cases, the client acts as the initiator of the file transfer, managing the transfer process and sending the file to the server. However, the implementation and functionality of the client differ between socket programming and HTTP.

4 Experimental Result

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

4.1 Task 1: File Transfer via Socket Programming

4.1.1 Server :

Server takes a request from the client.If client want to upload and download,server accepts that request and perform the desired operation and send back the result to client.'list' will return the available file which are currently available in server. 'upload (filename)' command will add a file in a server. 'download (filename)' will return the file client wants. Also server can handle multiple handle together and work simultaneously .

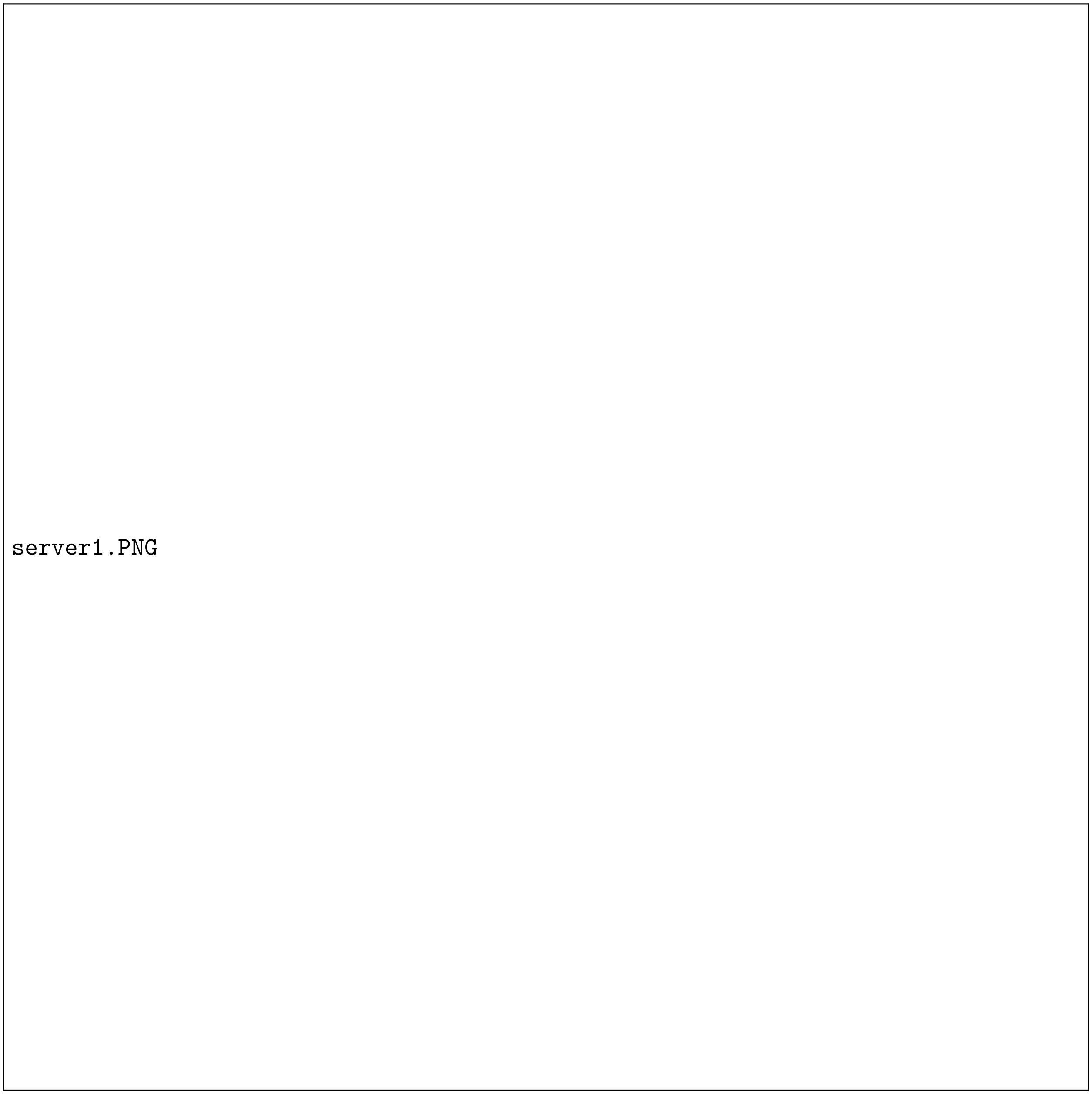

Figure 1: Content of Server for Task 1

4.1.2 Client :

The client can decide whether it wants to download or upload. It can send request to sever for a particular file download or upload.After server's response it can be able to do these operations. Also with the list command it will be able to see the content available in server. 'upload (filename)' command will add a file in a server. 'download (filename)' download the file from the server.

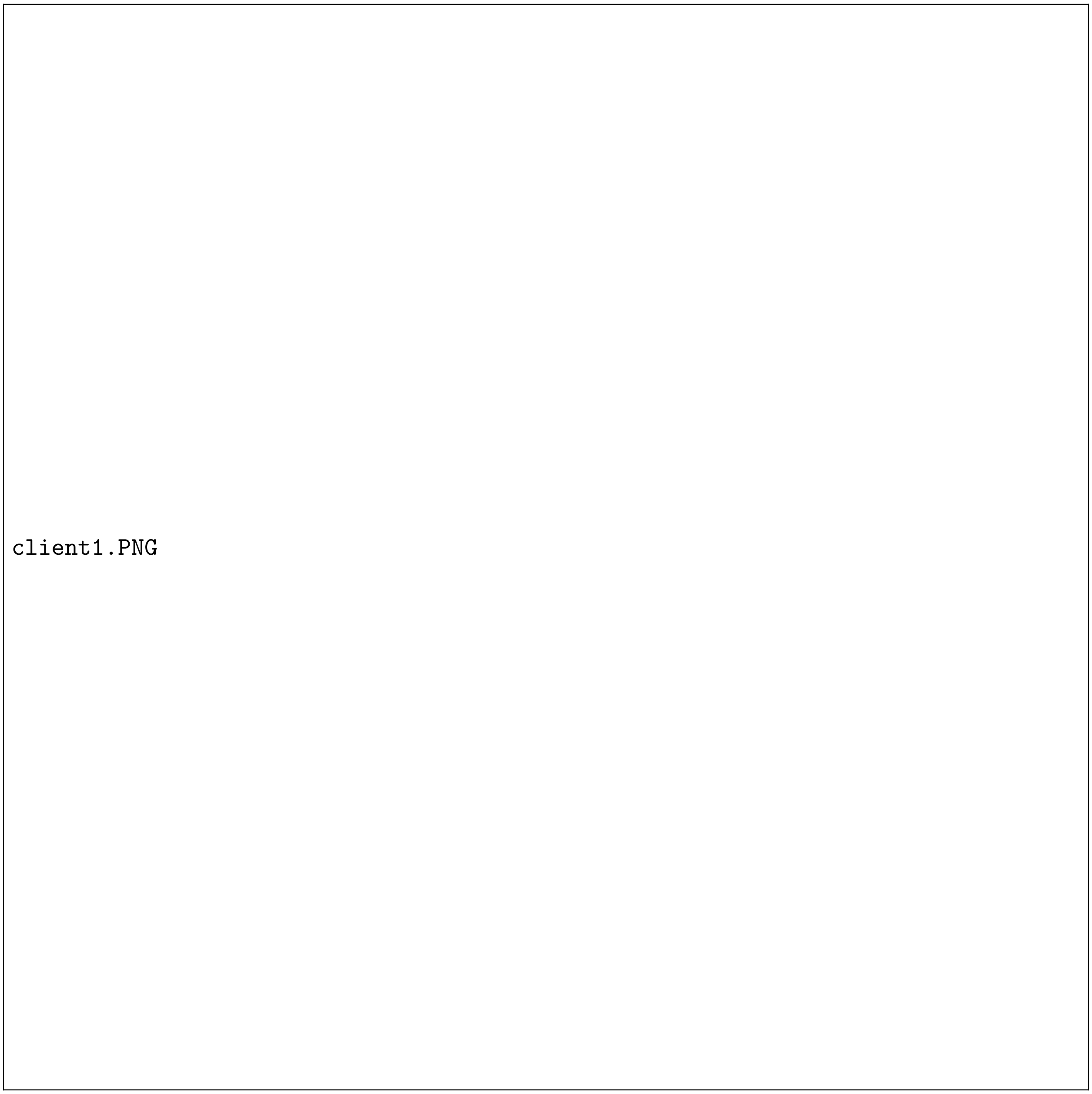

Figure 2: Content of Client for Task 1

4.2 Task 2: File Transfer via HTTP

4.2.1 Server :

HTTP file transfer server works by receiving file requests from clients, sending an HTTP response message with the requested file, and providing the necessary information for the client to properly display or handle the file.When a client requests a file from the server, the server sends an HTTP response message that includes the requested file. The response message has a status code, headers, and an optional message body that contains the file data. The headers contain information about the file such as its size, type, and encoding.

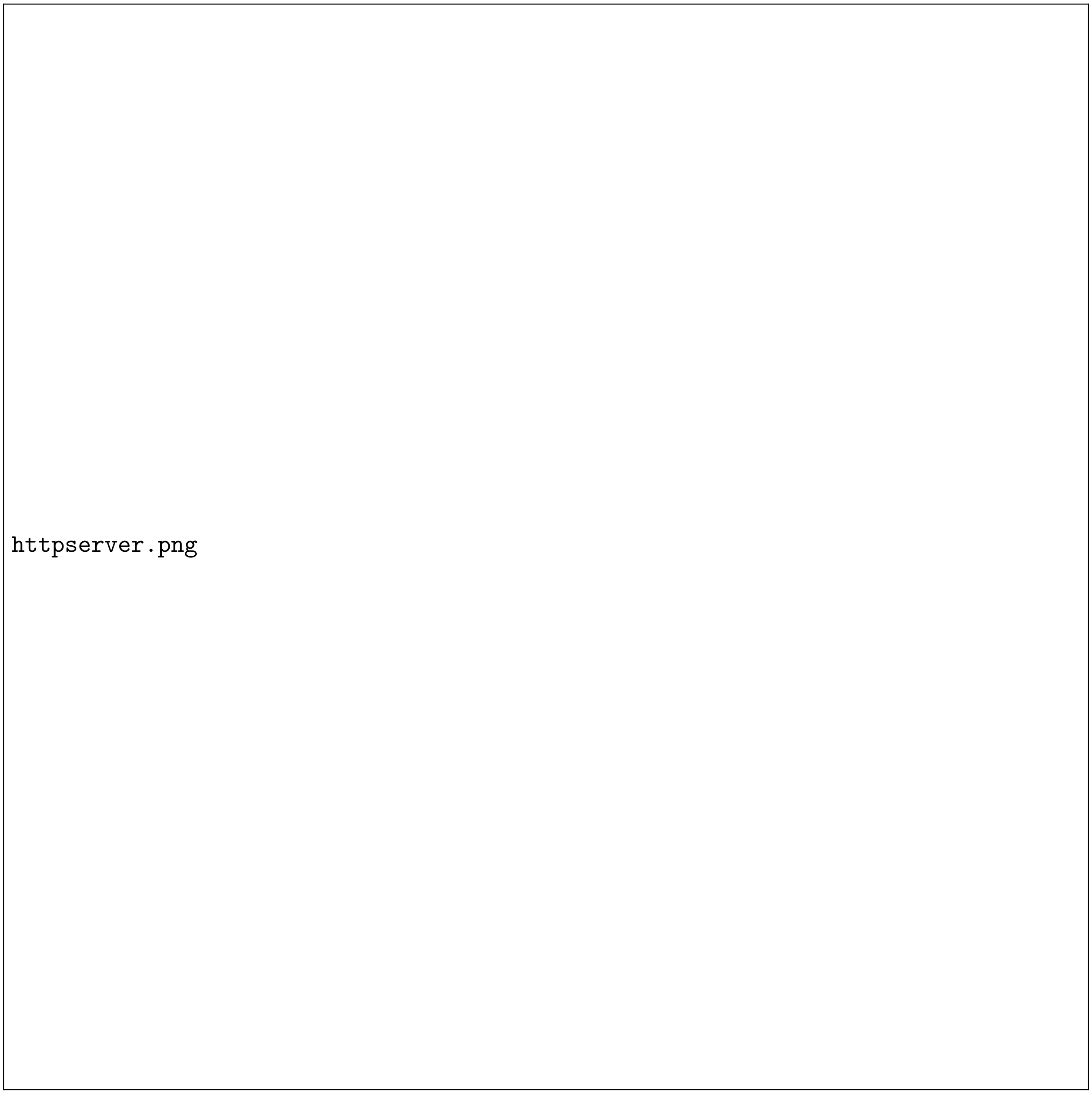

Figure 3: Content of Server for Task 2

Here server can handle both GET/POST from multiple clients simultaneously .The GET method is used to retrieve data from a server and is the most commonly used request method. It is used to request resources such as HTML pages, images, and scripts. The request and response data are passed in the URL, so the data size is limited.The POST method is used to submit data to a server for processing, such as when submitting a form or uploading a file. The data is passed in the body of the request and is not limited in size, making it suitable for larger data transfers.

4.2.2 Client :

The client side in an HTTP communication refers to the entity that initiates the request for data from the server. The client can be a web browser, a mobile app, or any other software that needs to retrieve data from an HTTP server. On the client side, an HTTP request message is generated and sent to the server. This message includes information such as the URL of the requested resource, the HTTP method (e.g. GET or POST), and any necessary data or headers.In conclusion, the client side in an HTTP communication is responsible for initiating requests to the server, processing the response from the server, and rendering or further processing the data as necessary.

Here in the client side, the user can ask any file through the URL. For this client needs the servers IP and port.

References

[mak, 2017] (2017). How to make an http client program in python. *GeeksforGeeks*. [Online; accessed 2023-01-25].

[pyt, 2017] (2017). Python network programming - server client file transfer. *GeeksforGeeks*. [Online; accessed 2023-01-25].

[201, 2017] (2017). Socket programming in python. *GeeksforGeeks*. [Online; accessed 2023-01-25].

[Python, 2022] Python, R. (2022). Socket programming in python (guide). *Real Python*. [Online; accessed 2023-01-25].

httpclient.png

Figure 4: Content of Client for Task 2