ASSIGNMENT REPORT

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of:

Prof. Aaditeshwar Seth
Assot. Professor

CSE Department

Submitted by:

Karan Agrawal (2023MCS2479)
Annanya Mehta (2023MCS2484)

INDIAN INSTITUTE OF TECHNOLOGY, DELHI

Hauz Khas, New Delhi -110016, India

Detailed Report on the Provided Python Code: Client-Server Communication

Introduction

This report delves into a Python script that serves as a client in a client-server communication system. The script's primary purpose is to interact with a remote server via the User Datagram Protocol (UDP). This communication involves the retrieval of data from the server in smaller segments, the assembly of these segments into a complete dataset, the computation of an MD5 hash for integrity checking, and the subsequent submission of the data back to the server. Moreover, this script is equipped with a feature to plot request and reply times along with corresponding offsets using the Matplotlib library. The subsequent sections offer a detailed examination of the code.

Code Overview

Server Information

The script initiates by defining essential server information. This includes the IP address and port number to establish a connection with the server. Additionally, it sets a timeout duration for socket operations, ensuring that the client can handle the server's response within a reasonable timeframe.

Server Host: "127.0.0.1"
Server Port: 9801
Timeout: 1 second
These parameters are fundamental for the client to locate and communicate with the server.

Lists for Request and Reply Time & Offset

This section initializes lists that will be used to record and analyze timestamps and offsets during the client's communication with the server. These lists are crucial for the subsequent plotting of request and reply times.

Request Time: This list collects timestamps corresponding to the moment when the client initiates a request.
Reply Time: This list captures timestamps associated with the reception of server replies.
Request Offset: Records the offset when a request is made, aiding in tracking where a specific segment of data is requested.
Reply Offset: Records the offset when a reply is received, helping analyze the position of each segment in the final dataset.
These lists are instrumental in understanding and visualizing the timing and sequencing of

client-server interactions.

Server Request Function

This function constitutes the core of the client's communication with the server. It is responsible for sending requests to the server and receiving responses using UDP sockets. Additionally, it provides an option to record timestamps and offsets for request and reply operations.

The function operates as follows:

It sends a request message to the specified server using a UDP socket.
Simultaneously, it records the current time in milliseconds, thereby capturing the request time.
If the offset parameter is supplied, it records the request offset.
Upon receiving a response, it captures another timestamp, denoting the reply time.
If an offset is specified, it records the reply offset.
Finally, it returns the server's response.
This function is the workhorse behind the client's communication with the server.

Connect Server Function

The connect_server function orchestrates the complete communication process between the client and the server. It includes the following key steps:

It initializes a UDP socket for communication.
Sends an initial "SendSize" command to the server, which informs the server to expect a data transfer.

Receives the server's response, which contains information about the size of the data to be retrieved.

Calculates the number of requests required to retrieve the entire dataset and initializes a data list to store the segments.
Requests data segments from the server in a loop, ensuring that each segment is received correctly.

Assembles the received segments into a complete dataset.
Computes an MD5 hash of the complete dataset to ensure data integrity.
Submits the computed hash and the data to the server using a "Submit" command.

Utilizes Matplotlib to create a plot that displays the request and reply times along with offsets, providing a visual representation of the communication timeline.

The connect_server function is the heart of the client's interaction with the server, responsible for data retrieval, assembly, integrity checking, and submission.

Script Execution
The script is executed by starting with the main function, which serves as the entry point. This function calls the connect server function to commence the client-server communication process.

```
if __name__ == '__main':
    main()
```

This conditional block ensures that the main function is executed when the script is run.

Running Time Graphs

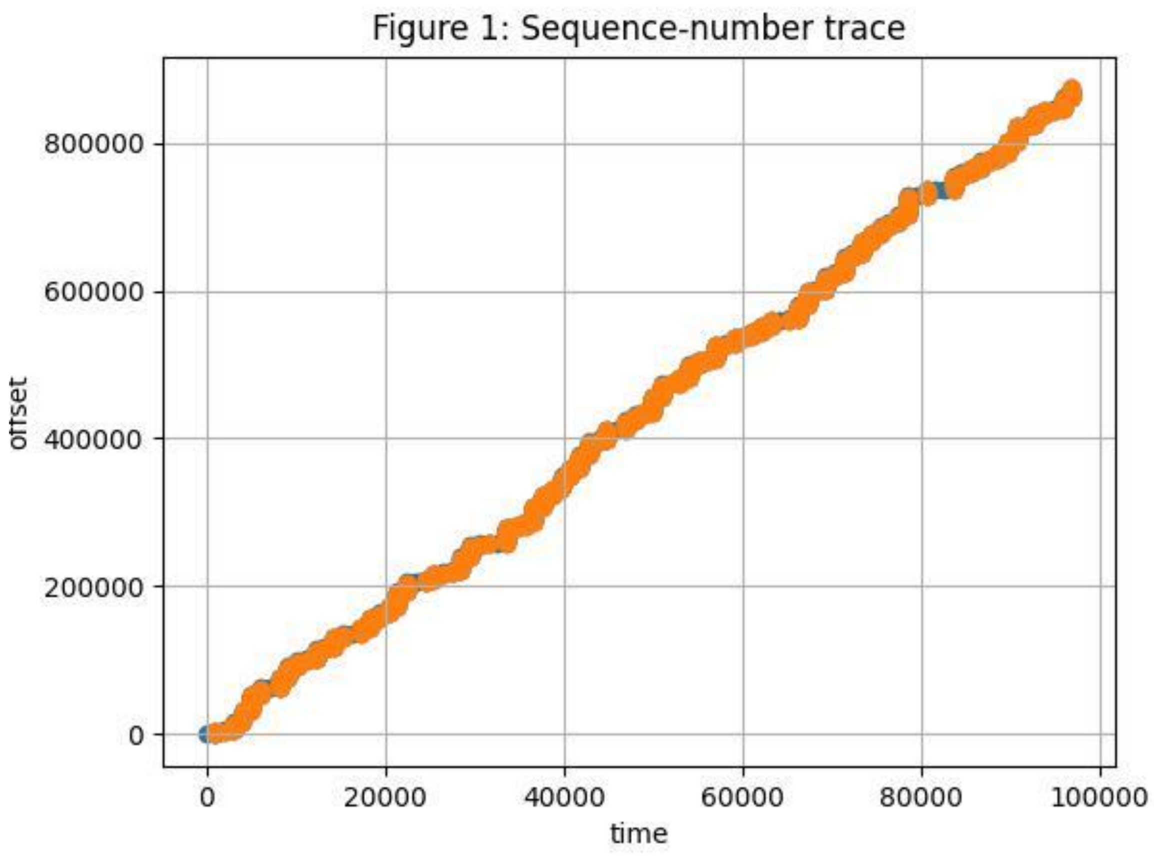
Figure 1: Sequence-number trace
offset
time