COL334 Assignment 3

Kavya Chopra (2021CS10081), Manpreet Singh(2021CS10070)

October 15, 2023

1 Algorithm

In order to assemble the file from the server, we repeatedly send requests to the server starting from the first byte, i.e. offset being 0. Since the server can send at most 1448 bytes in a single response, we keep asking for 1448 bytes and incrementing the offset. The server sometimes skips requests, so we keep requesting that particular packet until we can successfully receive some data from the socket. We have kept a delay of 5000 micro seconds before sending another request to avoid getting a penalty, and used a select timeout of 5000 microseconds, so that we wait only 5000 microseconds when the server decides to skip the request. In future implementations, we wish to parallelise the reads and the writes using threads.

With these settings, we're able to assemble the data for 10000 lines in about 4.5 s, and for 100000 lines in about 23.2 s.
When we reduce the sleep time to 1000 microseconds, we see that we're sending the requests too quickly and the server, and the time taken increases to about 7.8 s for 10000 lines.
When we reduce the select time to 1000 microseconds, the total time taken is 4.4s. We don't see substantial change in the total time taken, because it represents the maximum amount of time the system will take to check if our socket is ready to write to/be read from, and we observe that the number of skipped requests is not very substantial.
Moreover, the number of requests sent versus the number of replies received ratio increases as the sleep time goes down (since the server starts squishing requests and penalising the client for sending requests at breakneck rates)

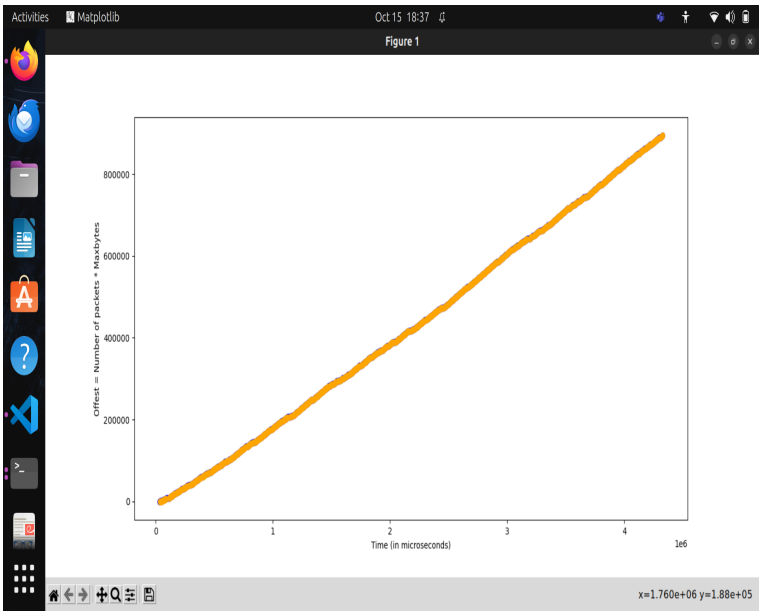
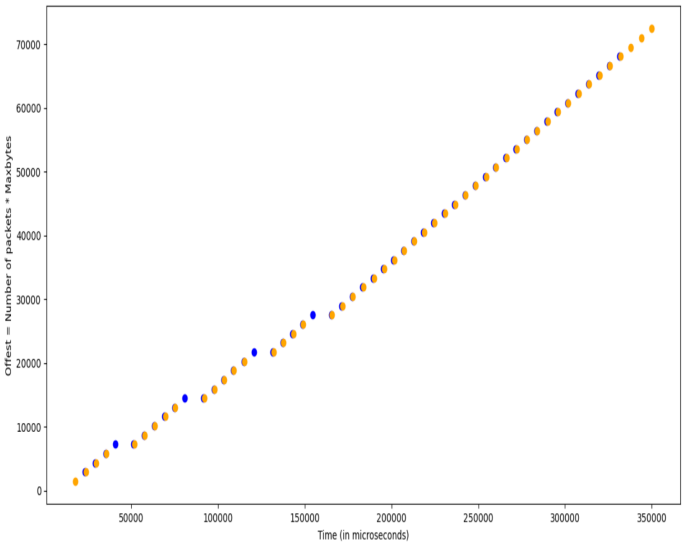


Figure 1: Data received vs time

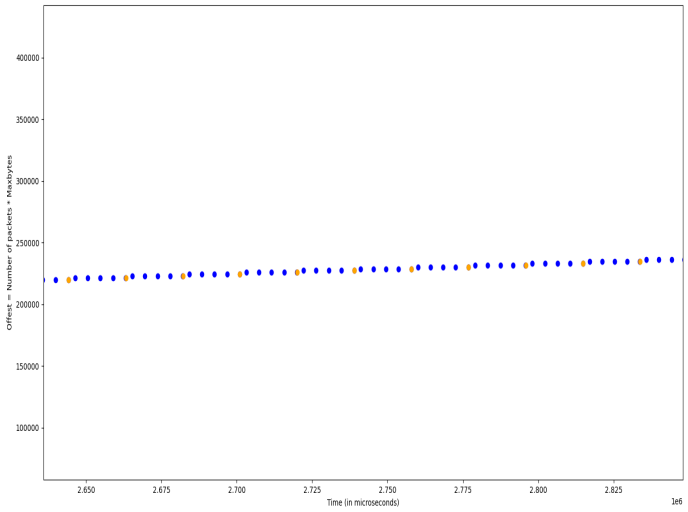
As we can see the graph is linear, which indicates that the server skipping requests is independent of which data bytes we're requesting.



Figure

Figure 2: Close Up, Blue represents data requests, Yellow represents server response. Sleep time = 5000 microseconds

In this graph, there is occasionally an extra blue dot, that represents the server skipping a request, so we ask for the same data bytes again. This is the case where we've kept our sleep time as 5000 microseconds. In the case where our sleep time is lower, the ratio of blue to orange dots becomes more significant, as evidenced by the below graphs



Figure

Figure 3: Close Up, Blue represents data requests, Yellow represents server response. Sleep time = 2000 microseconds

2 Acknowledgements

The code for the MD5 Hash library was taken from the internet, and the code for initialising the socket in C++ is a version of the one used in Beej's Socket Programming Guide. The main logic and the rest of the code is our own.