COL334 Assignment 3(Milestone 1)

October 15, 2023

Aastha Rajani (2021CS10093), Disha Nitin Shiraskar (2021CS10578)

1 Approach

- The code in focus establishes a UDP-based client-server communication protocol. At its core, the client's operation is divided between two concurrently executing threads. The first, governed by the 'send packets' function, methodically sends out data requests to a server. These requests target specific chunks of data identified by offsets, and the client uses a dynamic set called 'pending packets' to determine which chunks have yet to be retrieved.
- Simultaneously, a second thread, steered by the 'receive packets' function, stands ready to process incoming data from the server. As data packets are received, the client catalogs them and updates its internal tracking by removing the offset of the successfully received chunk from the 'pending packets' set. Throughout this procedure, a logging mechanism meticulously records every request sent and response received, including the specific offset and the exact timestamp of the interaction.
- After all anticipated data chunks are successfully collected, the client compiles them into a cohesive dataset. It then calculates an MD5 hash of the combined data to assure its integrity. This is followed by a final acknowledgment sent to the server, signaling the successful completion of the data retrieval process and sharing the computed hash value.

2 Graphs

2.1 Constant Rate Server

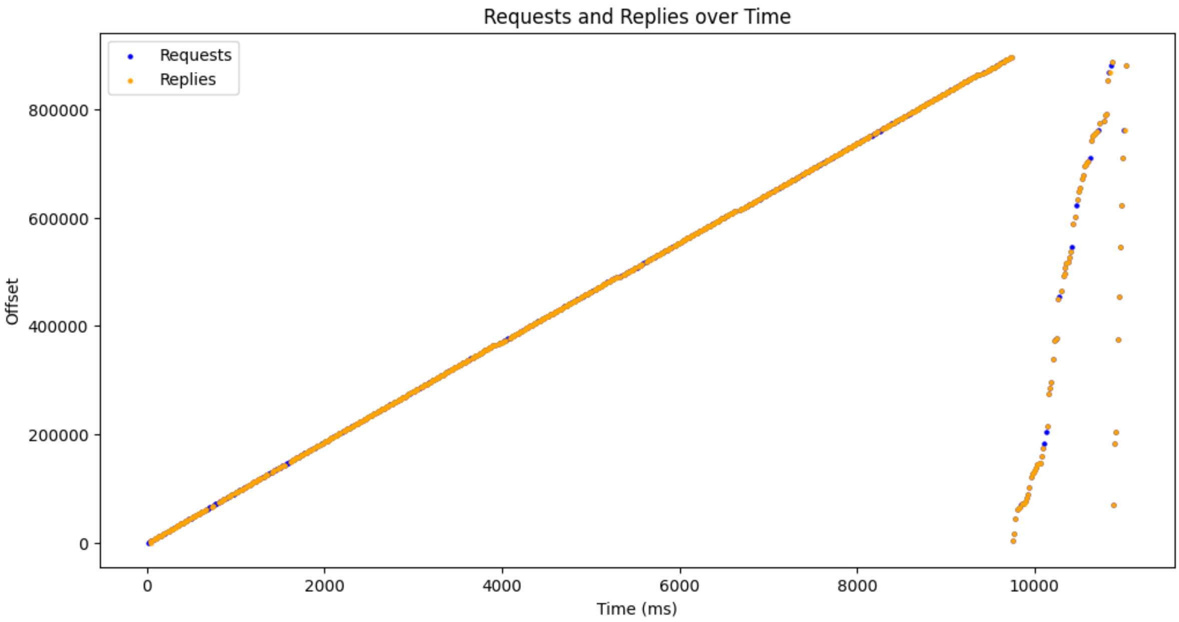

Figure 1: Zoomed-0ut View of constant rate server

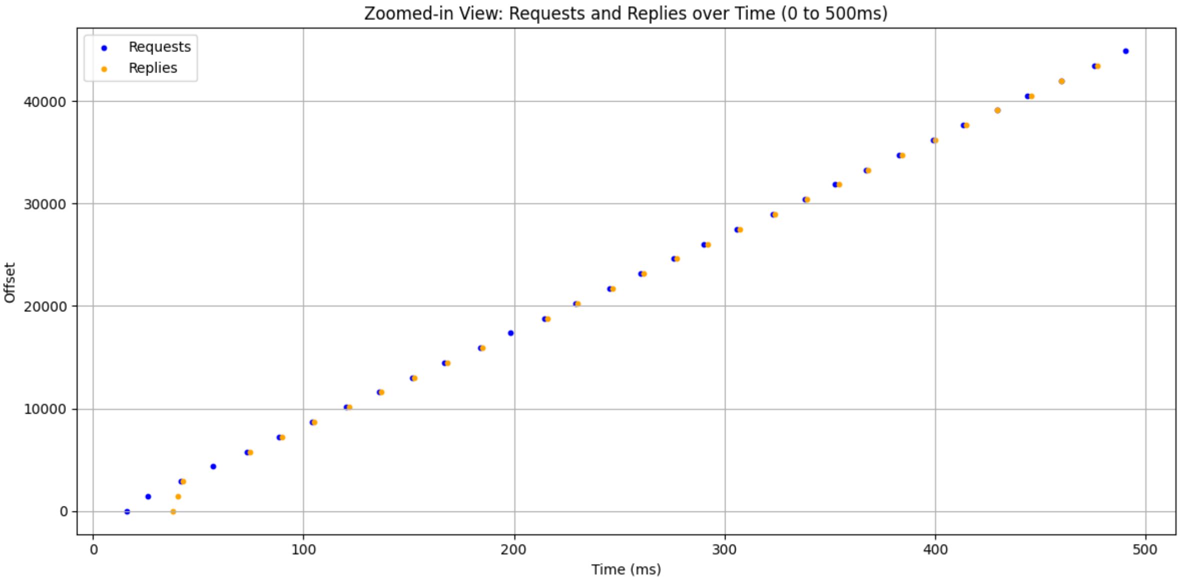

Figure 2: Zoomed-In View of constant rate server

2.2 Variable Rate Server

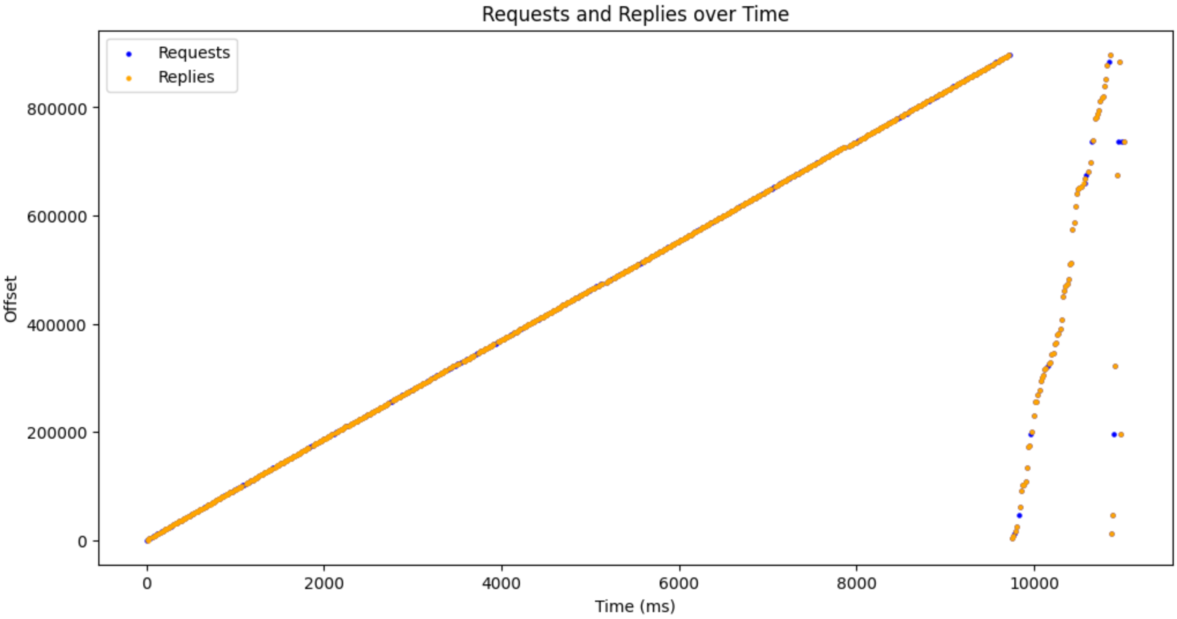

Figure 3: Zoomed-0ut View of variable rate server

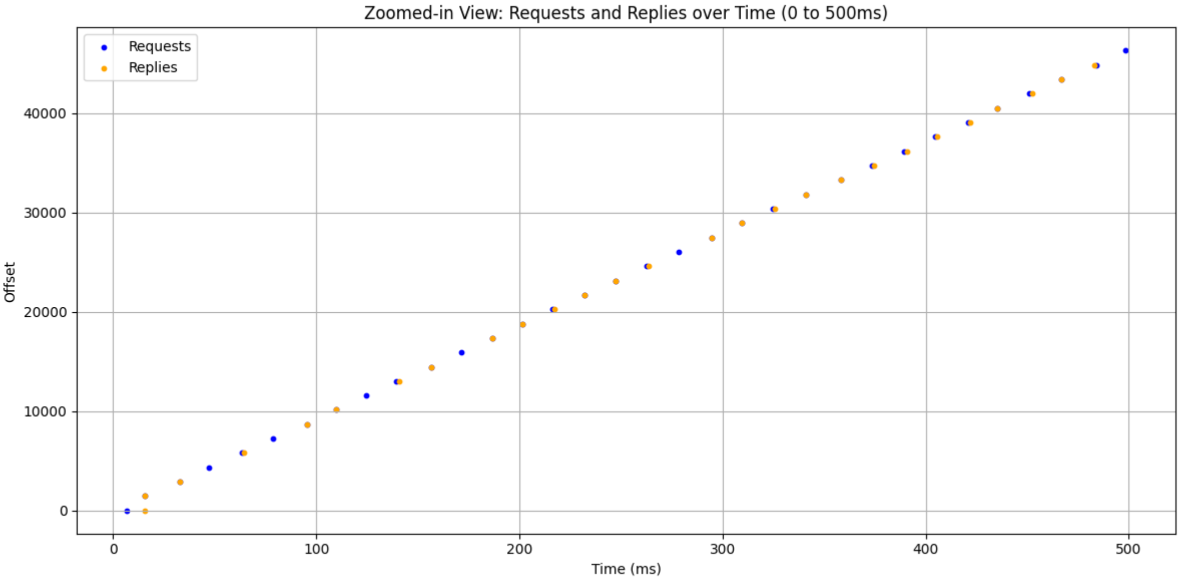

Figure 4: Zoomed-In View of variable rate server

2.3 Observations

- In Zoomed-Out View, each point shows the offset and the time for requests sent and replies received. Requests are shown in blue and replies in orange. Due to the scale of the axes, the replies appear to almost overlap with the requests. The client is programmed to run through the entire request space monotonically, then make another pass to request for offsets that were not received in the first round, then another, and so on, until all offsets are received. This is visible in the graph as the first pass from 0 until approximately 9800 ms, the second pass from 9800 ms to approximately 10200ms, etc.
- Zoomed-In View, shows the zoomed in view of duration 0 to 500 milliseconds of requests being sent and replies being received. Discrete requests and replies are visible in this graph.