

COL334 Assignment - 3

Milestone-1 Report

October 15, 2023

Contents

1	Aim	2
2	Our Implementation	2
3	Graphs and Observations	2
4	Team Details	3

1 Aim

Milestone-1 of this assignment aims to ensure that we can transfer data reliably by sending UDP requests slowly and keeping track of the packets of data received and also the packets of the data that may have been dropped. In the assignment we send and receive UDP packets, and we build our implementation of reliable data transfer on top of it. After we receive all the data from the server, the data is reassembled and hashed, and sent back to the server for checking its correctness.

2 Our Implementation

For Milestone-1 we have implemented a simple method of data transfer on the top of UDP connections in order to ensure reliability. In our method, , we sequentially send requests with incremental offsets at constant time gap, then we wait for 0.02 seconds for receiving the response. If the response is received then we append the response to the hash map , and if no response is received then we move on to the next index in the iteration. After the 1st iteration , we repeat the same process for the values of offsets which are still empty in the hashmap again and again untill the set of offsets for which there is no response becomes empty. Thus we keep sending requests cyclically in a **round robin fashion** untill all the responses are received.

Now after the responses for all the subsets are received, we join all the responses together and calculate the hash which we need to send back to the server for checking the correctness. The above method **ensures reliability** as our process terminates only when the responses for each of the offset has been received.

3 Graphs and Observations

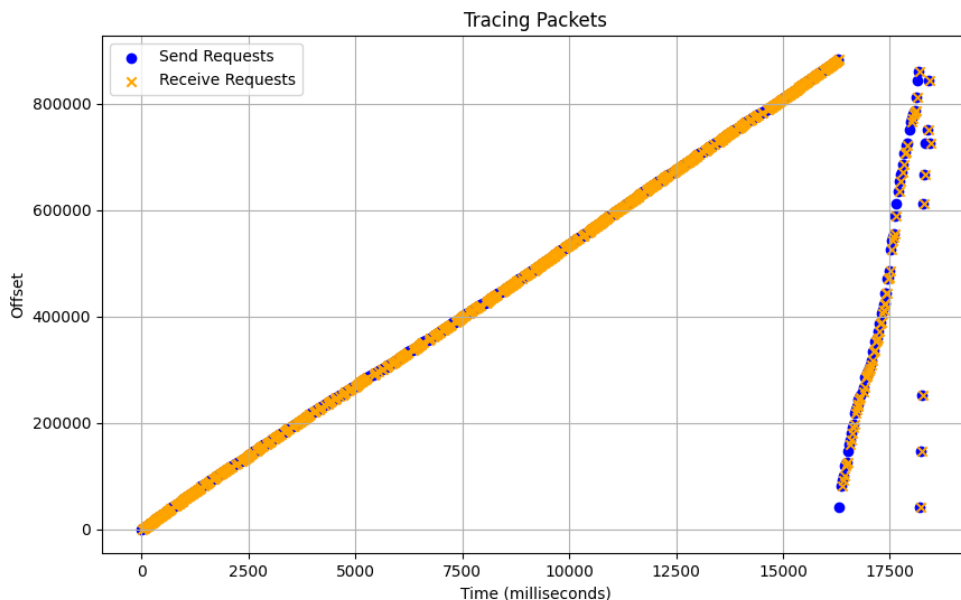


Figure 1: Graph of requests and response versus time from the start of the program

The above graph shows the timing of the requests sent to the server asking for the data packets in blue dots and the response received from the server as orange crosses. From the graph we can observe that in each iteration we have an almost linear plot, with significant gaps emerging after the first iteration and the plot almost becomes empty after the second iteration. By above observation , we can say that most of the data packets are received in the first two iterations and we send requests only for a few data packets in the third iteration.

The above graphs are obtained for a 0.02 second gap between every packet request, for which the penalty was observed to be 0.

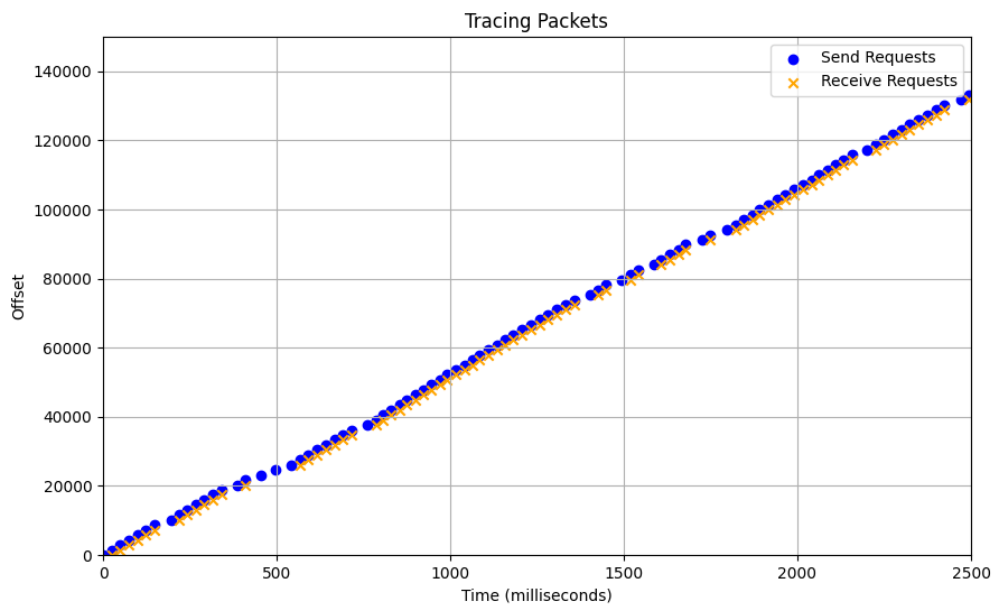


Figure 2: Zoomed in graph for requests and response versus time

The above graph shows us the zoomed in view of the previous graph (of the 0-2500ms part). You can also observe that the requests are spaced roughly 20ms apart, i.e. 0.02s. From the graph we can observe that most of the times the responses arrive almost immediately after the corresponding request for most packets. However, **some requests are dropped by the server** in a few cases for example the cluster of dots around the 500 millisecond mark in the graph.

4 Team Details

Team Members

- Utkarsh Sharma 2021CS10098
- Priyanshu Ranjan 2021CS10575