



Ahmedabad
University

CSE 342: Computer Networks

Section 1 Winter Semester 2022

Faculty: Professor Shashi Prabh

Teaching Assistant: Dhruvil Dave & Arpit Vaghela

Submitted by

Name	Roll No
Sameep Vani	AU1940049
Kairavi Shah	AU1940177

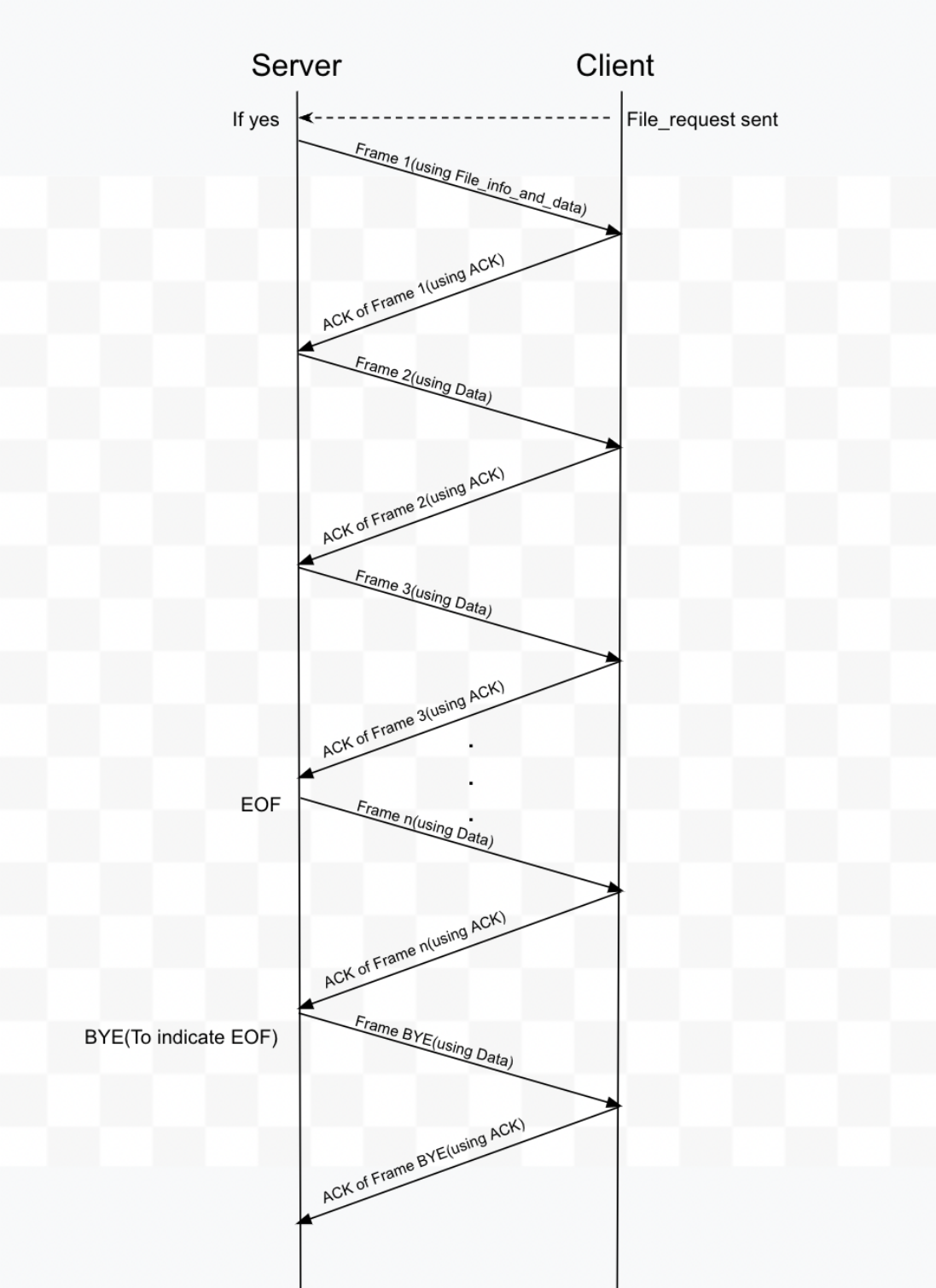
Lab 3: Reliable data transfer over UDP

This lab is an extension of the last lab. But this time we implemented a stop and wait protocol. The client sends the file request then the server verifies it and sends the frame if it has the file and if not then sends the file not found message. Each time the server sends the frame the client sends the ACK of it and if the ACK is not received the frame is sent again. So, this time we have to send message structures over socket such that in a given timeout the message should be sent and ACK should be received.

First of all we have declared all the structs which are File_request, ACK, File_info_and_data, Data and File_not_found which were all specified in the instructions.

As shown in the image below, the client sends the File_request struct through socket then if the server has that file then it moves ahead with sending the file otherwise the server sends File_not_found struct. If the server has the file then the first frame is sent through File_info_and_data which contains all the details of file as well as sequence_no of frame so that no two same frames are sent. Then next consecutive frames are sent from data struct which contains the block data and size plus sequence_no(for the same reason).

After each frame sent we perform XOR operation of the sequence_no in order to toggle it. The timeval is used for timeout and it is started as soon as the frame is sent on server side to ensure receiving ACK in given time. The timeval is executed using setsockopt function in which we have used SO_RCVTIMEO to set timeout for input which in our case is ACK. The timeout is set in such a way that the timeout is not too much or too less for ACK to reach the server. If the ACK is not received then the frame is sent again to receive ACK for the same. After the ACK is received, the process is repeated till the EOF is achieved and it is indicated to the client using BYE frame and client sends its ACK too.



Performance

Recv File Size	Sent File Size
1.5	1.5
6	6.1
9	9.1
11.3	11.4
16.5	16.7
23.3	23.6

Received File Size (MB) V/S Sent File Size (MB)



It can be inferred from the above table and graph that as we are following the reliable transmission of data, almost every single time the recv and sent file size remains the same. Sometimes as seen in table recv file size is little (like 1-3 Bytes) less than sent file size but that doesn't mean the frames are lost.