

1. Created a peer-to-peer network of multiple nodes which are able to communicate with one-another by exchanging messages (there are a total of 6 Nodes A, B, C and D, a client node and a server node).
2. Extended the capabilities of that peer-to-peer network to transfer a file of moderate size from one node to another. Performed an application level framing of the file in order to transfer it from one node to another. Fix the size of the frames to be 100 KB or you can take frame size as input from the user (in KBs).
3. Let us say when the authentication is successful, the client needs to upload a file (like assignment submission) to the server. The server however directs all the frames from a user (authenticated client) to the node which maintains the username and password for that particular user.
4. For the purpose of more system reliability, we use two server nodes in the network. Both the servers are all the same except for their own IP addresses and port numbers. In addition, these servers also cause random delays (the minimum delay is the RTT between client and server nodes) in the forwarding of frames to the destination nodes. The client knows about both the servers and for each message/frame, it chooses one of the servers alternatively.
5. Further, to ensure an error free delivery of the frames, I implemented a checksum based error detection method for the frames. The checksum should be transmitted with each frame. In case, the received frames get corrupted during the transmission, such frames must be retransmitted. This is called Stop-and-Wait ARQ protocol.

Code Link: <https://github.com/saurav8683/File-Transfer-System>