A Report On Peer To Peer Distributed File Sharing System:

File upload-

A file is read in pieces of size 512 kB. The parts are encrypted using Openssl envelope library, the key for encryption and decryption is hardcoded in the program. Then that encrypted piece is hashed into 20 bytes using SHA1 from Openssl sha.h library. After that the sha pieces of the chunks of the file are saved with different peers in the group, in a **Round Robin** manner. The pieces are indexed so that they can be put together in order while downloading.

File download-

The pieces are retrieved one by one from the peers in the group (no particular order followed to optimize the download process). After getting the pieces they are decrypted. Since the pieces were indexed, they can be put back together by arranging the pieces according to its index. C++ sort() function is used to sort the pieces with respect to their piece number. Then they are put back together and saved in the path given by the user, with the same name as the input file.

All the groups and their respective users are saved in a map data structure.
Multiple such map data structures are used in the tracker program to store the session information about connected peers. This information is stored in session only.

Tracker.cpp-

It accepts connection from multiple peers/clients. The users can send commands to the tracker. The tracker program reads the tracker_info.txt file to fetch the ip address and port number used by the client.

The tracker uses multithreading to handle multiple client requests.
The tracker processes the commands received from each peer and gives appropriate output.
I designed the tracker in a way that you can run multiple tracker programs in parallel on different ports (you just have to define the port number in the program itself).Multiple trackers can take care of the peers connected to it.

Client.cpp-

It connects to the tracker (can independently connect to other peers as well). The client can send commands to the tracker (can send messages or files too, to another client connected to it). In a peer to peer system, no one peer or client is the server. Here, every peer can be file uploader or consumer. The client can only access files in the group it is in, and the client can upload a file in a group it is in. Since the client.cpp is a standalone programme, it also employs multithreading to handle multiple peers.
Every peer's IP address and port number will be saved in the tracker_info.txt file.

Remarks-

The scope of this assignment lies in the application, that it increases bandwidth uitilization in real-time file sharing system. But, given the skillset I have and based on my flawed understanding of the assigned task, only certain level has been achieved. I look forward to finesse this assignment for a real world application.

References-

The references taken for this assignment are as follows-

1. https://youtu.be/X50uv51SP5I?si=0RGihlPJyk9k80qN
2. https://github.com/abhishek0508/peer-to-peer-file-sharing-system