Eswar Balla 2024201017

# PEER TO PEER DISTRIBUTED FILE SHARING SYSTEM

## Introduction

This project is an implementation of a peer-to-peer (P2P) distributed file-sharing system, where multiple users can share files with others in a group-based setting. The system is designed with fault-tolerance and scalability in mind,Users can upload files to groups they are a part of, and other members can download these files using a piece selection algorithm that enables parallel downloading from multiple peers, providing efficient bandwidth utilization.

## Implementation Details

### Tracker Module

The tracker is the central component in this system. It listens for incoming connections from clients on a specific port specified in the tracker_info.txt file. When a client connects, the tracker spawns a new thread to handle the client's requests independently. This architecture allows the tracker to handle multiple client connections simultaneously.

The tracker maintains three main types of data:

**User Data**: Tracks user accounts and authentication credentials.
**Group Data**: Manages group memberships, file ownership, and file sharing permissions.
**File Data**: Tracks files shared in each group, maintaining information on which clients are seeding each file.

By design, the tracker does not store actual file data but instead maintains metadata about the files and clients' IP addresses and ports, enabling direct peer-to-peer communication.

### Client Module

The client module is designed to interact with the tracker and perform all the necessary file-sharing functionalities. It connects to both trackers listed in tracker_info.txt and registers its presence. Upon connecting, the client can execute various commands, including but not limited to:

**create_user**: Registers a new user.
**login** and **logout**: Manages user sessions.
**create_group** and **join_group**: Allows users to create and join groups, respectively.
**upload_file** and **download_file**: Facilitates file-sharing between group members.

For downloads, the client contacts the tracker, which provides a list of peers sharing the requested file. The client then initiates direct connections with these peers to download the file chunks. The custom piece selection algorithm ensures the load is distributed across multiple peers by selecting file pieces randomly.

## Piece Selection Algorithm

The piece selection algorithm used in this system is an equally distributed random selection algorithm. It splits the file into multiple pieces and selects pieces from peers randomly, ensuring balanced bandwidth utilization across peers. The SHA1 hash is computed for each file to maintain data integrity, ensuring that downloaded pieces match the original file content.

## Conclusion

This project successfully implements a scalable, distributed file-sharing system with essential P2P features like parallel downloads, fault tolerance, and group-based sharing. The architecture leverages synchronized trackers to ensure seamless operation even if one tracker goes offline, ensuring high availability. The use of a random piece selection algorithm enables efficient bandwidth utilization and reduces file download times.

Future improvements could include implementing data encryption for secure file transfers, improving the piece selection algorithm to prioritize rarer pieces, and adding more robust error-handling and recovery mechanisms for enhanced reliability. Overall, the project provides a solid foundation for a distributed file-sharing system that can be extended and adapted to meet various peer-to-peer sharing needs.