Peer-to-Peer Distributed File Sharing System

1. Introduction

This project implements a peer-to-peer file sharing system, where users can share and download files within groups they belong to. The system supports multi-tracker synchronization, allowing redundancy and efficient file sharing. Clients download files in pieces from multiple peers using a custom piece selection algorithm, and file integrity is verified using SHA1 hashing.

2. Architecture Overview

The system is built using the client-server model, with trackers maintaining information about clients and shared files. Clients register, login, create groups, share files, and download files. The key components of the system are:

- **Trackers**: Maintain information about connected clients and shared files.

- **Clients**: Interact with trackers and peers to share and download files.

3. Features Implemented

- **Multi-Tracker System**: Supports synchronization between trackers to ensure availability.

- **Account Creation and Login**: Users must create an account and log in to access the system.

- **Group Management**: Clients can create and join groups, and only group members can share and download files.

- **File Sharing and Downloading**: Clients can share and download files within groups.

- **Parallel Downloading**: Files are downloaded from multiple peers simultaneously.

- **Piecewise SHA1 Hashing**: Ensures file integrity during the download process.

4. Workflow

1. **Account Creation**: The client creates a new account using the `create_user` command.

2. **Login**: The client logs in using the `login` command.

3. **Group Creation/Joining**: The client can create or join a group to share and download files.

4. **File Sharing**: The client shares files within the group using the `upload_file` command.

5. **File Downloading**: The client requests to download a file, and the tracker provides peer information for parallel downloading.

6. **Integrity Check**: The downloaded file pieces are verified using SHA1 hashing.

5. Custom Piece Selection Algorithm

The system employs a custom piece selection algorithm for downloading files from multiple peers. The client requests different pieces of the file from different peers to optimize download speed and network utilization. Each piece is verified for integrity using its SHA1 hash, and only valid pieces are added to the final file.

6. Tracker Synchronization

The system supports multiple trackers to ensure availability and redundancy. Trackers synchronize their data, so if one tracker goes offline, the client can seamlessly connect to another tracker. This ensures that file sharing information is consistent across all trackers.

7. Assumptions

1. At least one tracker will always be online.

2. Clients must register and log in to participate in the network.

3. Files are divided into 512KB pieces, and each piece is verified using SHA1 hashing.

4. Clients can only download files from groups they are a part of.

8. Conclusion

This peer-to-peer file sharing system implements a robust, multi-tracker architecture that ensures high availability and parallel file downloading. By incorporating features such as group-based sharing, piecewise SHA1 integrity checks, and tracker synchronization, the system provides an efficient and secure file sharing experience.