# Assignment – 1

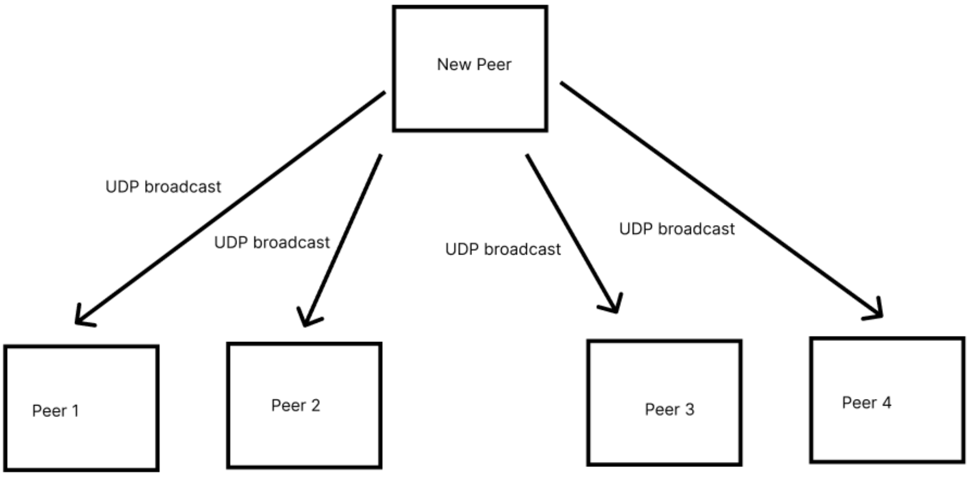
Satyaswaroop Nayak

M23CSE021

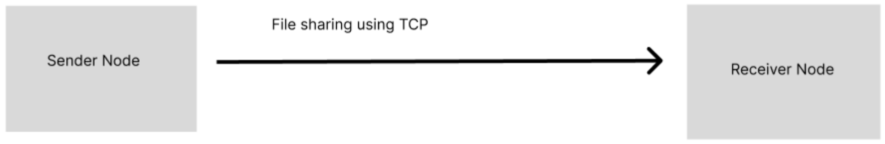
## 1. Introduction

The Peer-to-Peer (P2P) file sharing system described in this report aims to enable users to share files directly with each other over a decentralized network. This report outlines the architecture of the system, describes how peers will connect, communicate, and share files, discusses challenges related to peer discovery, presents the implementation details, and addresses considerations for error handling.

## 2. Architecture of the P2P System



Peer Discovery



## 2.1. Network Topology

The P2P system follows a decentralized architecture, where all peers are considered equal and communicate directly with each other without relying on a central server. Peers are interconnected in a mesh topology, allowing them to form direct connections with multiple other peers.

## 2.2. Peer Roles

Peers in the network have two primary roles:

Seeder: Peers with files available for sharing.

Downloader: Peers seeking files for download.

## 3. Peer Connection, Communication, and File Sharing

## 3.1. Peer Connection

Peers connect to network by broadcasting their unique id and port on which they are available to accept connection.

## 3.2. Communication

Communication between peers is facilitated through a custom P2P protocol. Peers can exchange information about available files, request file transfers, and keep each other updated on their status.

## 3.3. File Sharing

File sharing happens when sender node asks user, which receiver the file is to be sent, once it is provided by the user, the file is sent to the receiver.

## 4. Handling Challenges: Peer Discovery

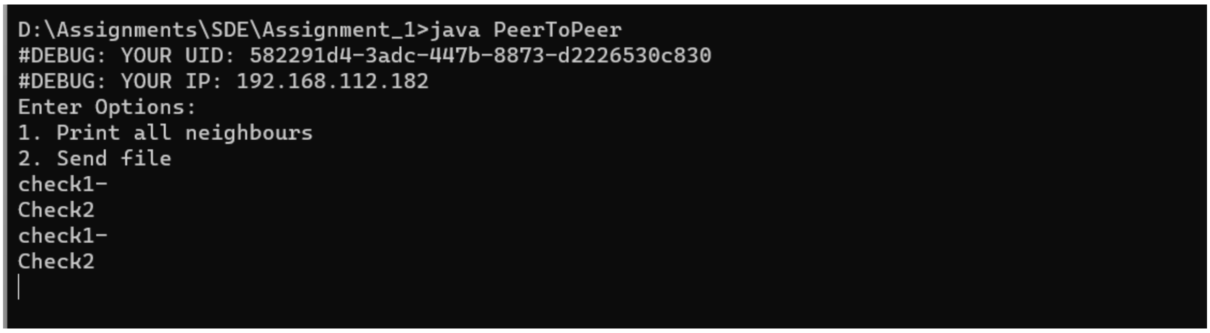
Peer discovery is managed by broadcasting a message using User Datagram Protocol by the new node and receiving the reply from all other nodes.

## 5. Implementation of the P2P System

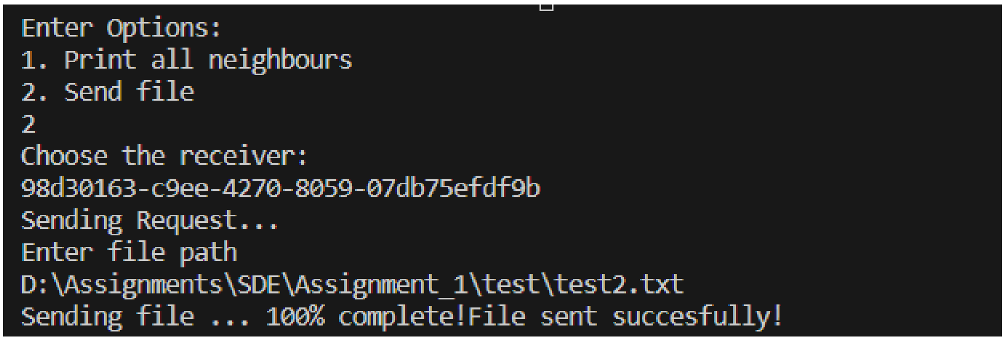
The P2P system has been implemented using java, leveraging libraries for network communication. The system's codebase is modular, making it extensible and easy to maintain.

## 6. Error Handling and User-Friendly Interactions

Error handling in the system includes addressing connection issues, file transfer failures. User interactions are managed through a user-friendly command-line interface (CLI) that provides clear feedback on actions and progress.



```
D:\Assignments\SDE\Assignment_1>java PeerToPeer
#DEBUG: YOUR UID: 582291d4-3adc-447b-8873-d2226530c830
#DEBUG: YOUR IP: 192.168.112.182
Enter Options:
1. Print all neighbours
2. Send file
check1-
Check2
check1-
Check2
```



```
Enter Options:
1. Print all neighbours
2. Send file
2
Choose the receiver:
98d30163-c9ee-4270-8059-07db75efdf9b
Sending Request...
Enter file path
D:\Assignments\SDE\Assignment_1\test\test2.txt
Sending file ... 100% complete!File sent succesfully!
```

## 7. System Performance

### 7.1. Scalability

To ensure system performance as the number of peers increases, the system optimizes its resource utilization and bandwidth management.

## 8. Conclusion

The architecture, implementation, and main components of a peer-to-peer (P2P) file sharing system have been described in detail in this study. A decentralized solution to file sharing is provided by the system, which also addresses issues with peer discovery, security, scalability, and quick file transfers. It offers users in an expanding peer network a smooth experience with user-friendly interactions and effective error management. It is a dependable option for P2P file sharing thanks to effective content finding and download algorithms that guarantee excellent efficiency.