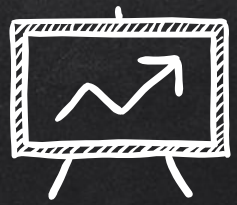


Peer To Peer File Transfering

HELLO! WE ARE FROM GROUP 9

Nguyen Khanh Nam

Nguyen Trong Son

Do Anh Tu

Do Thi Minh Ngoc

Nguyen Trung Dung

Pham Viet Minh Duc

Content

1. Introduction
2. Objectives
3. Existing system
4. Method
5. Evaluation
6. Demo

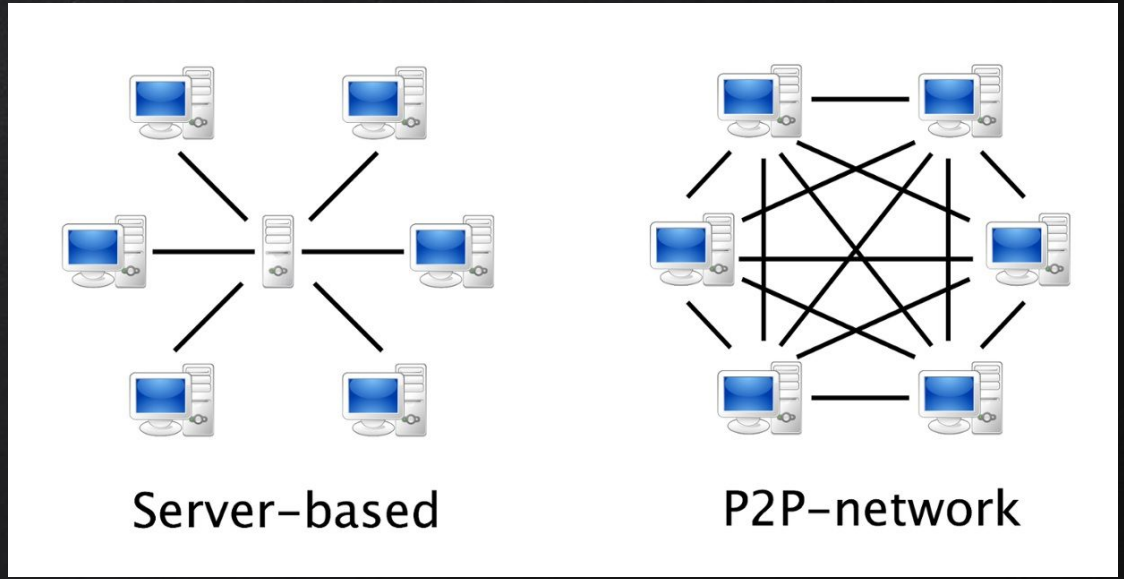
1 Introduction

- What is p2p?

Peer-to-peer, or P2P in its abbreviated form, refers to computer network using a distributed system.

- Why p2p?

It allows the computers to connect with each others to receive and send files simultaneously.



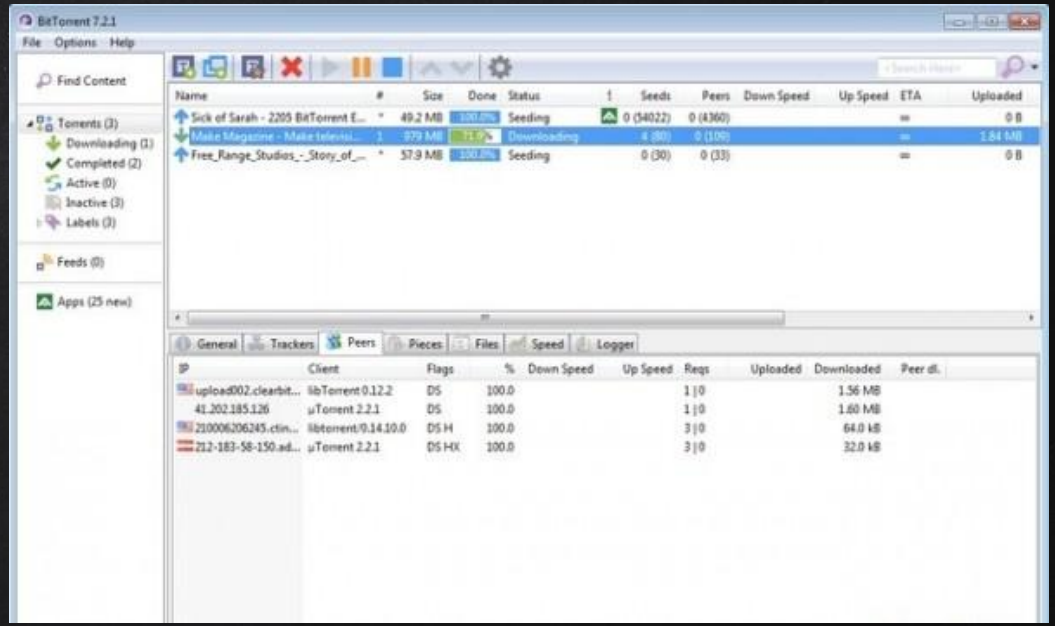
Objectives

- ✘ Know how the p2p network architecture
- ✘ Apply it into a simple file transferring application
- ✘ Chatting and Discussing
- ✘ Use for education purposes

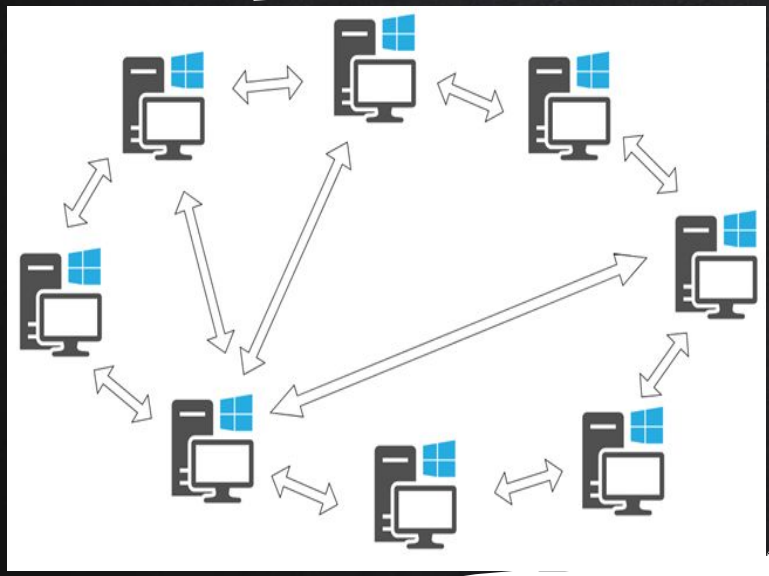
3 Existing System

Peer to peer networks

- Bittorrent
- Utorrent



4 Method

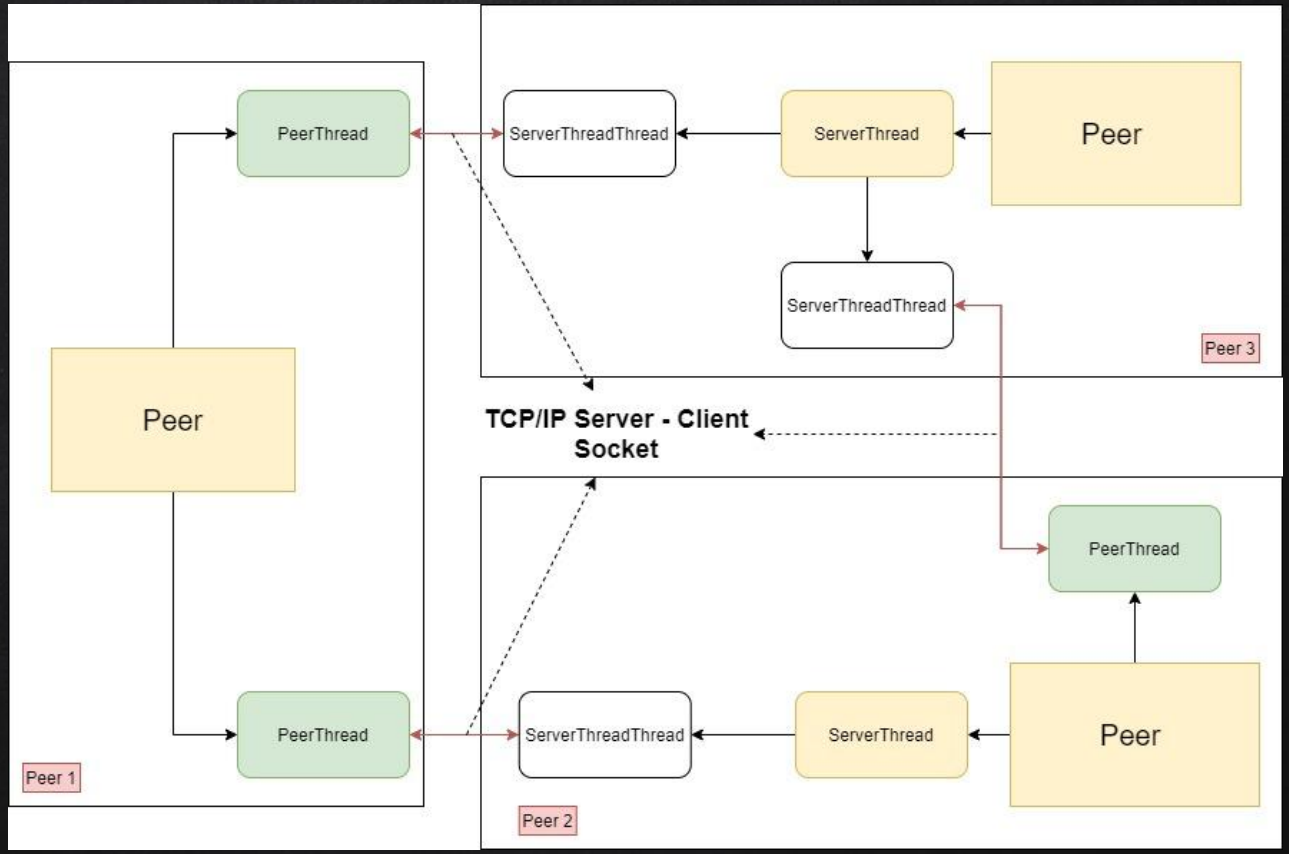


System Architecture Advantage:

Each peer can be a server for another

Connection Example

4



Method

✘ Peer ✘ PeerThread ✘ ServerThread ✘ ServerThreadThread

```
public class Peer {

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter username and port for this peer: (ex: Calar 101)");
        String[] setupValues = bufferedReader.readLine().split(" ");
        ServerThread serverThread = new ServerThread(setupValues[1]);
        serverThread.start();
        new Peer().updateListentoPeers(bufferedReader, setupValues[0], serverThread);
    }
```

How do we implement our system ?

Method

- ✘ Peer
- ✘ PeerThread
- ✘ ServerThread
- ✘ ServerThreadThread

```
public class PeerThread extends Thread{
    private BufferedReader bufferedReader;
    private FileOutputStream fos;
    private BufferedOutputStream bos;
    private InputStream is;
    private String status = "chat";
    public PeerThread(Socket socket) throws IOException {
        this.is = socket.getInputStream();
        bufferedReader = new BufferedReader(new InputStreamReader(is));
    }
```

How do we implement our system ?

Method

- ✘ Peer
- ✘ PeerThread
- ✘ ServerThread
- ✘ ServerThreadThread

```
public class ServerThread extends Thread {
    private ServerSocket serverSocket;
    private Set< ServerThreadThreads > serverThreadThreads = new HashSet< ServerThreadThreads >();
    public ServerThread(String portNumb) throws IOException {
        serverSocket = new ServerSocket(Integer.valueOf(portNumb));
    }
```

How do we implement our system ?

Method

- ✗ Peer
- ✗ PeerThread
- ✗ ServerThread
- ✗ ServerThreadThread

```
public class ServerThreadThreads extends Thread{
    private ServerThread serverThread;
    private Socket socket;
    private PrintWriter printWriter;
    private String filePath = null;
    public ServerThreadThreads(Socket socket, ServerThread serverThread) {
        this.serverThread = serverThread;
        this.socket = socket;
    }
```

How do we implement our system ?

5 Evaluation

	Avg. 1st Msg Response Time	Avg. 2st Msg Response Time	Avg. 3st Msg Response Time	Avg. 4st Msg Response Time	Avg. 5st Msg Response Time
Case 1	467.1 ms	5.5 ms	4.6 ms	6.55 ms	4.64 ms
Case 2	108.3 ms	2 ms	3.67 ms	2.67 ms	2.33 ms
Case 3	35 ms	3 ms	1 ms	1 ms	1 ms

Demo!


Thank you
For listening.