

ĐẠI HỌC BÁCH KHOA TP.HCM
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BẢN BÁO CÁO
BÀI TẬP LỚN I MẠNG MÁY TÍNH I

ĐỀ TÀI:
CHƯƠNG TRÌNH CHIA SẺ FILE TRỰC TUYẾN

THÀNH VIÊN:

Nguyễn Văn	Quân	50902143
Đinh Trần Thái	Sơn	50902263
Nguyễn Duy	Tài	50902316
Phạm Minh	Thành	50902476
Đặng Ngọc	Vũ	50903332



Mục lục

I. Giới thiệu chương trình.....	3
1. Mô tả chương trình.....	3
2. Mô hình xây dựng chương trình.....	4
2.1 SERVER	4
2.2 CLIENT.....	7
3. Hoạt động của chương trình.....	11
3.1. Sơ đồ hoạt động của Client.....	11
3.2. Sơ đồ hoạt động của Server.....	12
II. Các Protocol giao tiếp giữa các chương trình.....	13
1. Giao thức giữa Client và Server.....	13
2. Giao thức giữa các Client.....	14
III. Giao diện người dùng và sử dụng chương trình.....	16
1. Giao diện người dùng.....	16
2. Sử dụng chương trình.....	16
3. Giao diện quản lý của Server.....	18
IV. Tổng kết.....	19

I. Giới thiệu chương trình.

1. Mô tả chương trình

BitCrazy là một chương trình chia sẻ file trên mạng theo mô hình lai giữa mô hình Client-Server và mô hình P2P.

Chương trình gồm 2 phần: Client và Server.

➤ Server làm nhiệm vụ:

- o Quản lý thông tin về file.
- o Quản lý thông tin người dung
- o Quản lý người dung kết nối tới
- o Hiện trạng các file trên máy người dung.

➤ Client làm nhiệm vụ

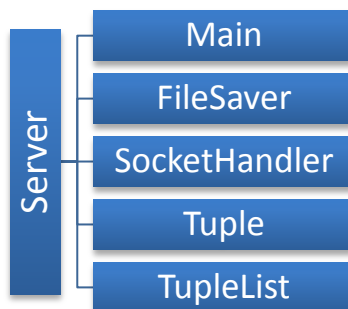
- o Kết nối tới Server.
- o Lấy danh sách những người đang chia sẻ file.
- o Tải các phần của file về máy của mình từ những người dùng khác.

❖ Tính năng của chương trình:

- Có khả năng chia sẻ nhiều file đồng thời.
- User có khả năng tải về nhiều file đồng thời.
- Chương trình có khả năng khôi phục lại việc tải file nếu bị ngắt giữa chừng.
- Chương trình có khả năng hạn chế tốc độ tải lên và tải xuống

2. Mô hình xây dựng chương trình

2.1 SERVER



Hình 1.0 *Cấu trúc các Class trong Server*

❖ **File Server** gồm những class và chức năng chính như sau:

- **SocketHandler.java** : Chứa giao thức Client-Server giúp Client-Server có thể giao tiếp với nhau và Các hàm để lấy thông tin từ máy client Seed lên server

STT	Method	Chức Năng
1	public void run()	Lưu LogFile của Server lên Disk
2	public void start()	Tạo Thread mới
3	public void stop()	Tắt Thread

- **FileSaver.java**: Xử lý đa tiến trình và lưu thông tin lên file để phòng Server bị crash.

STT	Method	Chức Năng
1	private void close() throws IOException	Đóng Socket của Server
2	private void doMain() throws IOException	Thực hiện giao tiếp giữa Server-Client
3	private String getGuestIP()	Lấy địa chỉ IP của máy khách
4	private int getGuestPort()	Lấy địa chỉ Port của máy khách
5	private String getStringElem(String _string, int _n)	Loại bỏ khoảng trắng khỏi chuỗi _string
6	private void open() throws IOException	Tạo Input Stream và Output Stream gắn vào Socket
7	private String readSocket() throws IOException	Đọc dữ liệu từ Socket
8	private void writeSocket(String msg) throws IOException	Ghi dữ liệu vào Socket

- **Tuple.java:** Định nghĩa cấu trúc file “persist.bin” chứa thông tin về Client được Server lưu trữ.

STT	Method	Chức Năng
1	public void addIP(String _ip)	Thêm IP khách vào dữ liệu của Server
2	public String getAnIP()	Lấy IP từ dữ liệu của Server

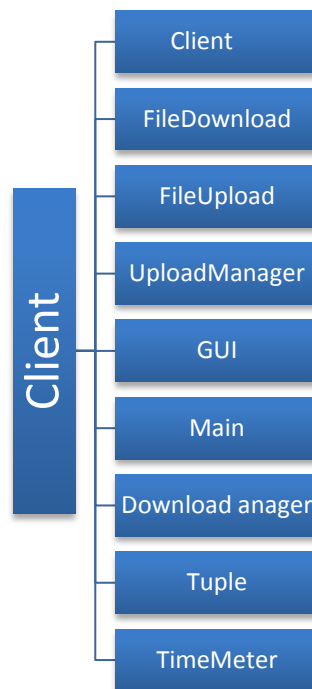
3	public long getFileSize()	Lấy Kích thước của File
4	public int getID()	Lấy ID của File
5	public int getIPCount()	Lấy Số lượng IP mà Server nắm giữ
6	public void removeAllIP()	Xóa tất cả các IP ra khỏi IP List của Server
7	public void removeIP(String _ip)	Xóa IP chỉ định ra khỏi IP List của Server
8	public void addIP(String _ip)	Gắn thêm IP vào IP List của Server

➤ **TupleList:** Chứa List các Tuple được định nghĩa trong Tuple.java

STT	Method	Chức Năng
1	public int add(String _ip, long _fileSize)	Thêm IP và kích thước file vào dữ liệu của Server
2	public void add(int _id, String _ip)	Thêm IP và ID và dữ liệu của Server
3	public Tuple getByID(int _id)	Lấy trường dữ liệu Tuple của Server thông qua ID
4	public void remove(String _ip)	Xóa tất cả thông tin trên Server của máy khác theo IP
5	public void remove(int _id, String _ip)	Xóa tất cả thông tin trên Server của máy khác theo IP,ID
6	public void removeAll()	Xóa thông tin của tất cả Client

- **Main.java:** Mở Port 33333 trên Server lắng nghe và đáp trả các thông tin từ Client gửi lên bằng cách sử dụng các hàm trong những file còn lại.

2.2 CLIENT



Hình 1.1 *Cấu trúc các class trong Client*

❖ **Client:** Chứa bộ giao thức Client-Server giúp Client giao tiếp với Server:

- **UploadManager:** quản lý serversocket và tạo các tiến trình upload.

STT	Method	Chức Năng
1	UploadManager(int port)	Thiết lập các tiến trình lắng nghe kết nối

2	private void createSocket()	Tạo serversocket để lắng nghe kết nối.
3	public void listen(GUI UI)	Thiết lập chế độ lắng nghe cho serversocket xử lý các kết nối tới., gồm các công việc: + tạo tiến trình FileUpload mới để truyền data + đưa Thread trên vào dãy ArrayList để quản lý.
4	public void close()	Tắt ServerSocket
5	public boolean stopThread(int hash)	Dừng Thread đang upload file có ID trùng với thông số ID

- **FileUpload:** Thread được gọi khi có 1 Connect từ Socket khác. Chứa chức năng Upload file cho Socket đang muốn lấy File về. Chứa phần còn lại của giao thức Client-Client.

STT	Method	Chức Năng
1	FileUpload(Socket sk,GUI UI)	Hàm khởi tạo. Tạo Thread

2	public void run()	Gồm các công việc + Khởi tạo SpeedController + Tạo các Buffer, Stream truyền dữ liệu ra và đọc dữ liệu vào. + giao tiếp với downloader từ các client khác, lấy offset, thông tin file. + đọc thông tin nhận được và gửi nó cho downloader
3	public void stop()	Tắt tiến trình quản lý tốc độ và tiến trình upload

- **FileDownload:** Thread được gọi khi Add ID trên Gui và gửi Request đến server sau đó lấy IP từ Server để Connect đến đối tượng đang giữ File cần down và lấy File về. Chứa 1 phần giao thức Client-Client.

STT	Method	Chức Năng
1	FileDownload(int ID, GUI UI,String IP, boolean isResume)	Khởi tạo các tham số cần thiết
2	public void stop()	Ngừng thread đang chạy
3	private synchronized void notice()	Thông báo cho class DownloadManager.java là thread đã ngừng chạy và gọi class đó thoát khỏi trạng thái wait (notify())

4	public int getID()	Trả về ID của thread hiện tại
5	public boolean stillAlive()	Kiểm tra xem thread còn sống hay không
6	public void run()	Tạo socket tới seeder, gửi ID của file cần down cho seeder, kiểm tra trạng thái là down mới hay khôi phục lại download, thực hiện giới hạn tốc độ download (nếu có) và cuối cùng là thực hiện download file từ input stream và ghi file ra đĩa

- **DownloadManager:** Quản lý tất cả các file đang hiện download

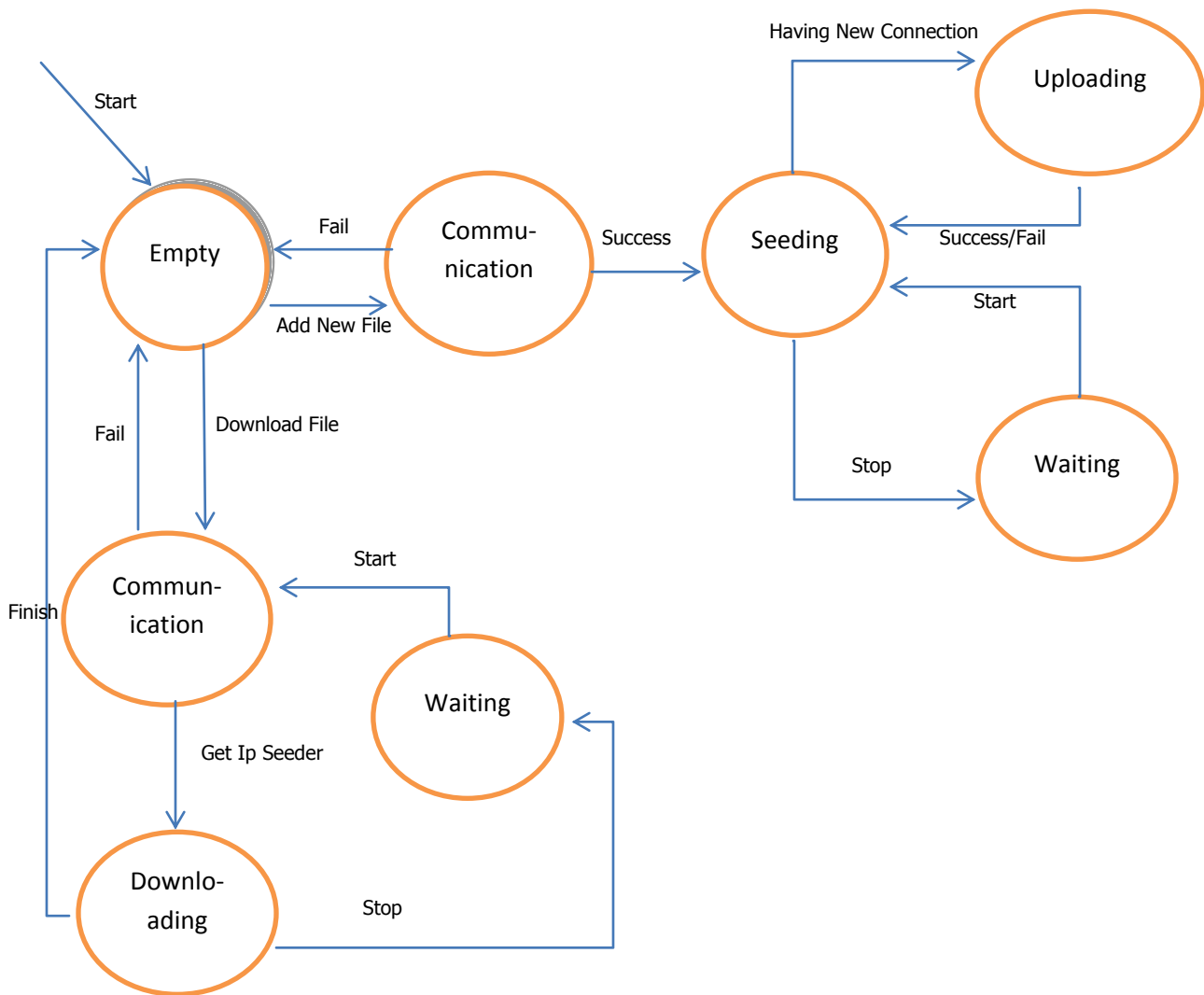
STT	Method	Chức Năng
1	public synchronized boolean stopThread(int id)	Tắt một thread với số thứ tự là id nếu thread này còn tồn tại
2	public synchronized boolean stopThread(int id)	Add một thread với số thứ tự là id vào trong danh sách các thread
3	public synchronized void waiting()	Sau khi stopThread thì chạy hàm này để chờ thread đó tắt trong thời gian 100ms

- **GUI:** Chứa toàn bộ giao diện chính của chương trình

- **Main:** Khởi tạo chương trình chính của Client và tạo ServerSocket ở Port 22222.

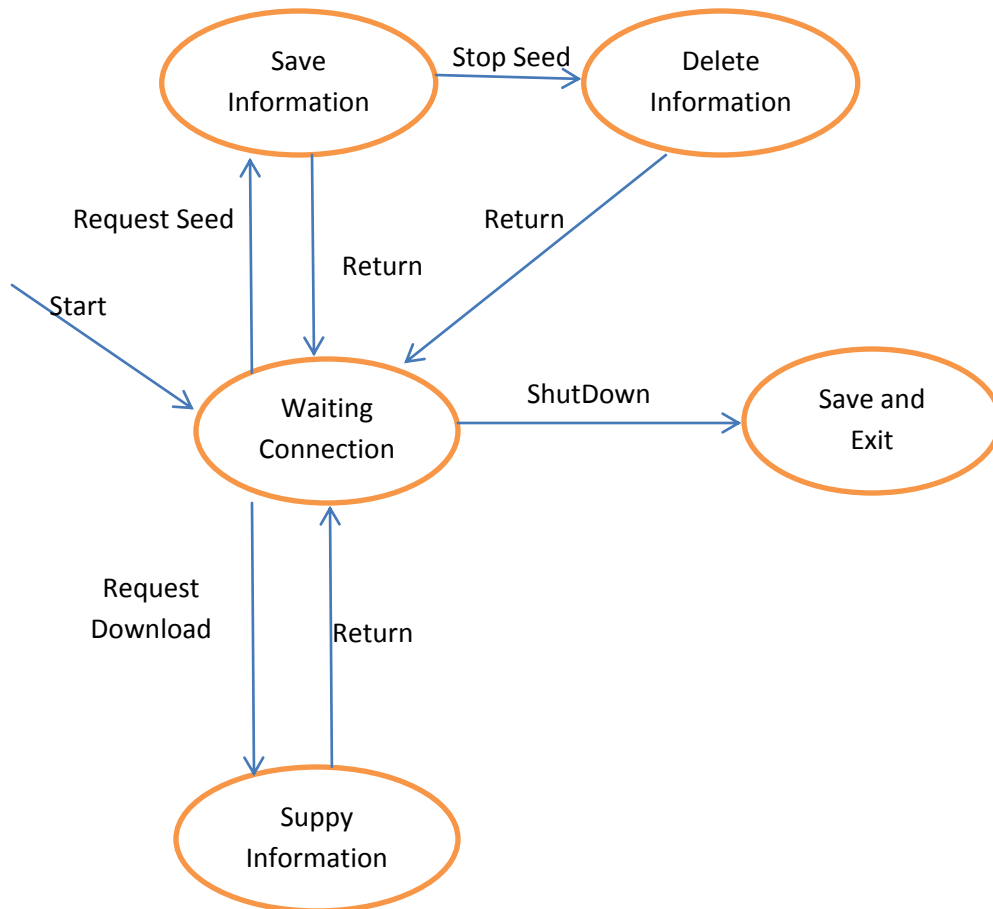
3. Hoạt động của chương trình

3.1 Sơ đồ hoạt động của Client



Hình 3. SƠ ĐỒ STD HOẠT ĐỘNG CỦA CLIENT

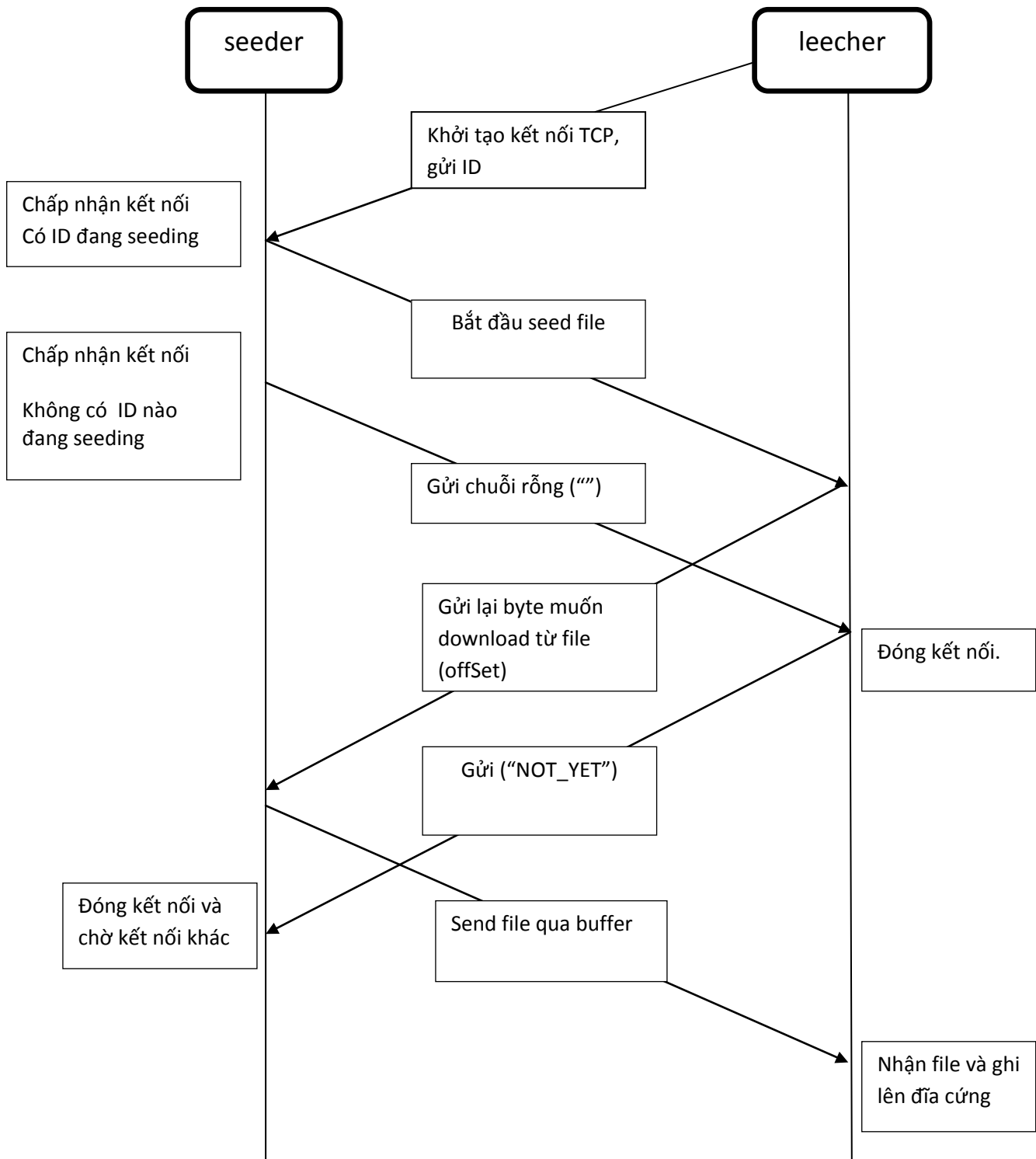
3.2 Sơ đồ hoạt động của Server



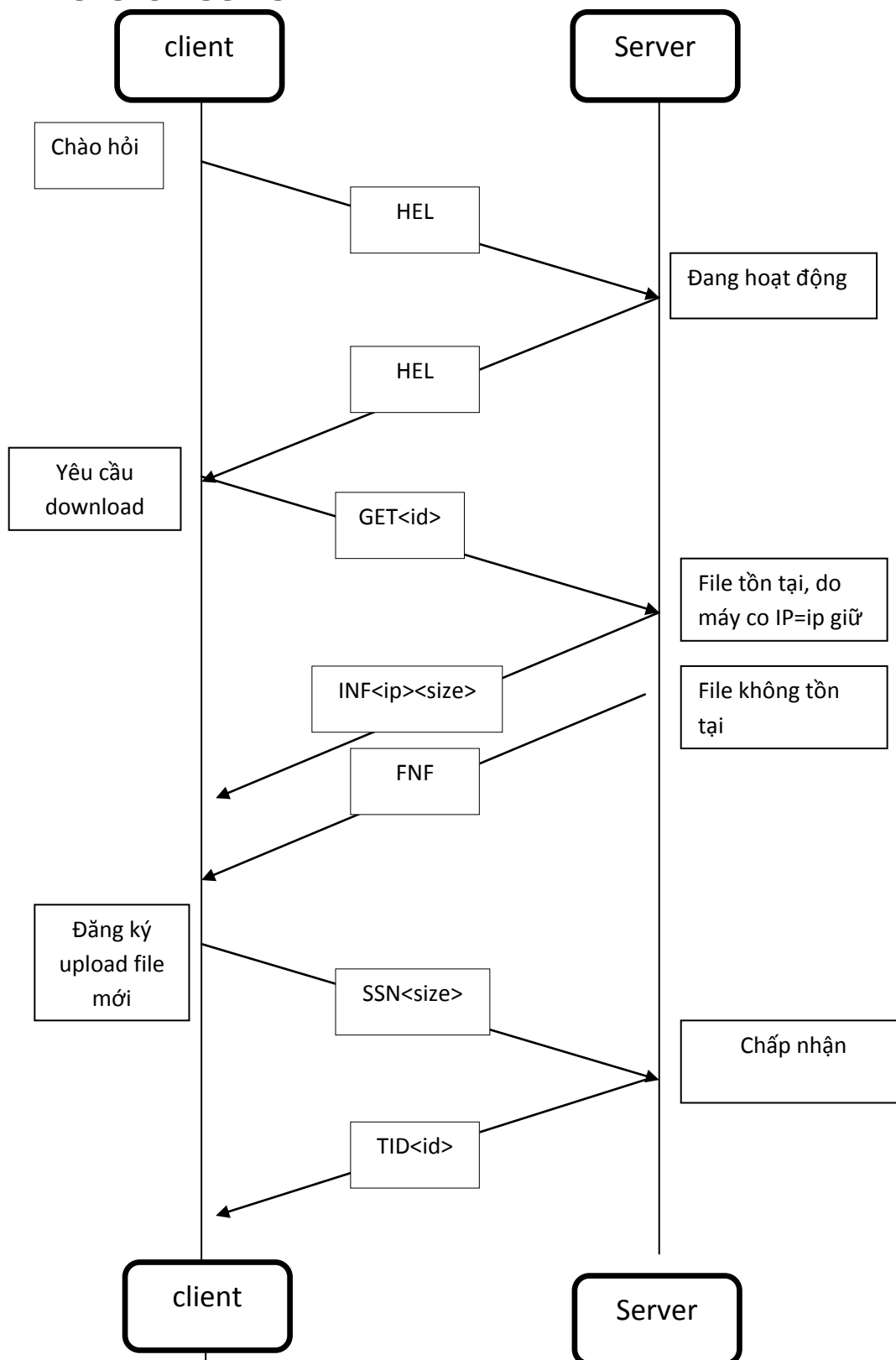
Hình 4. SƠ ĐỒ STD HOẠT ĐỘNG CỦA SERVER

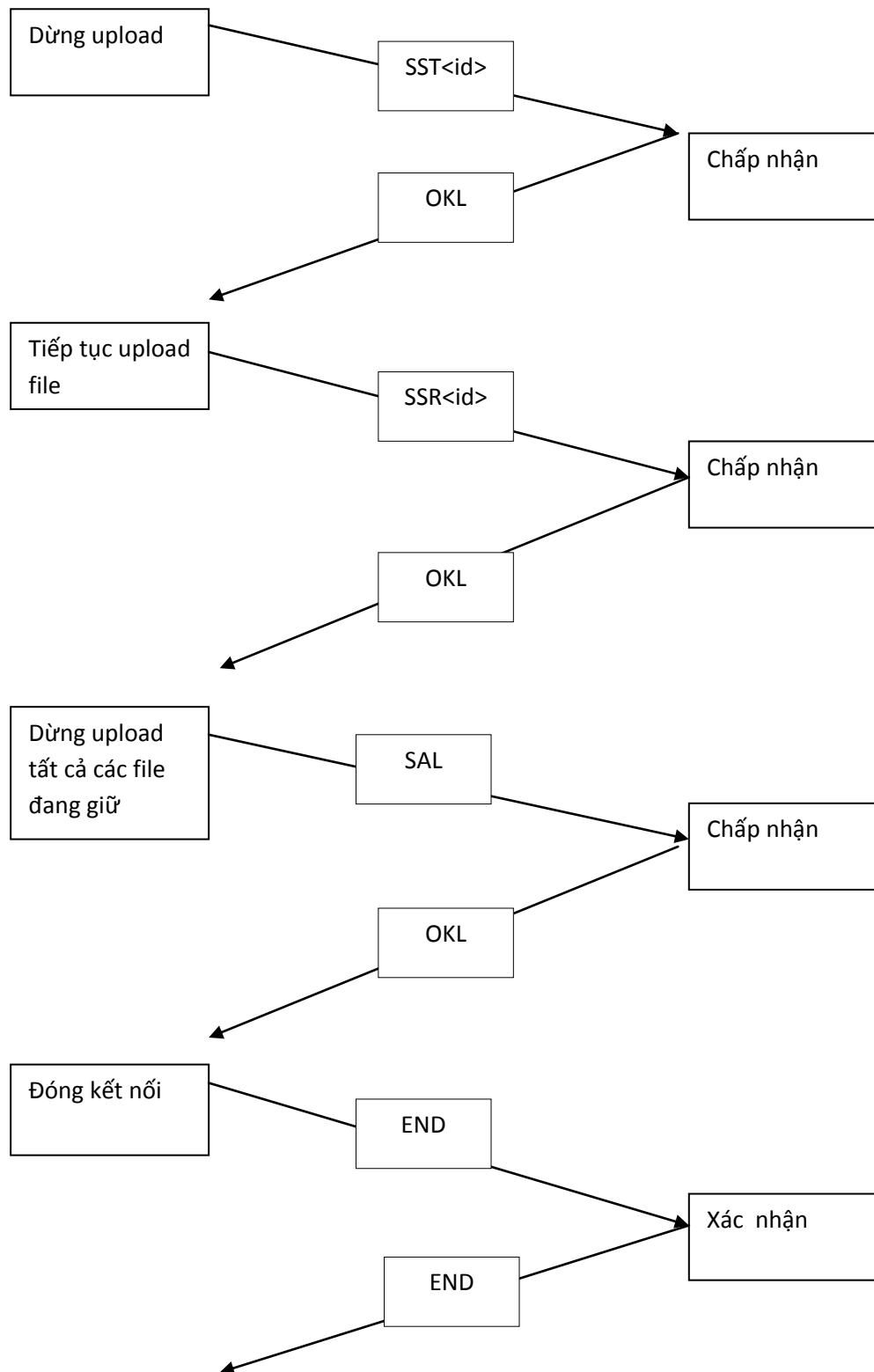
II. Protocol giao tiếp giữa các chương trình

1. Client – Client (seeder – leecher)



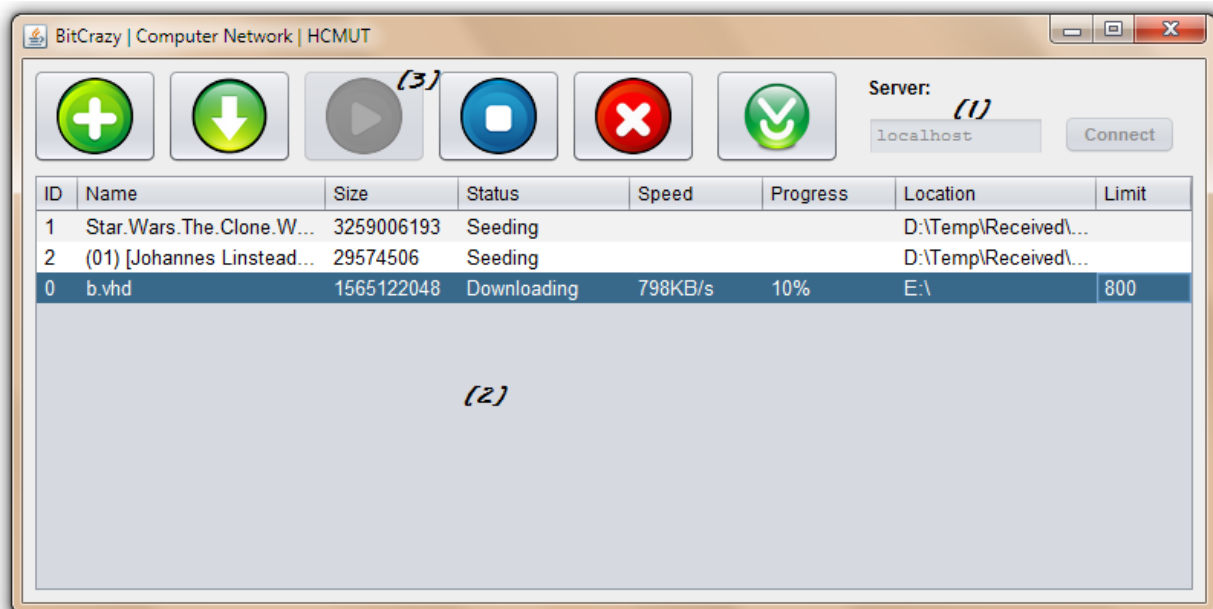
2. Client – Server





III. Giao diện người dùng và sử dụng chương trình

1. Giao diện người dùng



2. Sử dụng chương trình.

[1]:







- ❖ Nhập IP của Server sau đó nhấn **SET** để kết nối với Server. Kết nối thành công Sẽ trả về **Dialog Successful**. Kết nối không được sẽ trả về **Dialog Server Doesn't Exist**.

[2]:

- ❖ Khung giao diện chính của chương trình thông báo cho người dùng biết các thông tin về file **ID , Name, Size, Status, Speed, Progress, Location, Limit**. Đặc biệt là **ID** của file đó chính là **ID** được server cấp phát và sử dụng để down từ các máy client khác.

[3]:

❖ Các nút chức năng chính của chương trình

- **Upload:** 
Đăng ký upload một file với Server.
- **Download:** 
Nhập ID của file muốn down để download file đó về.
- **Start:** 
Bắt đầu download/upload một file hoặc resume lại file đã stop.
- **Stop:** 
Dừng download/upload một file.
- **Remove:** 
Xóa file đó ra khỏi trình quản lý của chương trình download.
- **Limit:** 
Giới hạn tốc độ download một file.

❖ Cách thức sử dụng chương trình:

Trước khi sử dụng chương trình phải config Server IP

- **DownloadFile:**
Nhấn Button **Download**→Nhập **ID** vào → Nhấn **OK**→ Chọn Nơi **lưu trữ File** và chọn **tên File**→**Save**
- **SeedFile:**
Nhấn Buton **Upload**→ Chọn **File muốn Seed**→ Nhấn **OK**→ Nhấn **Button Start** để bắt đầu SeedFile lên Server.

- **Stop File đang Download:**
Nhấn **Button Stop**
- **Start lại File đã bị Stop:**
Nhấn **Button Start**
- **Remove File ra khỏi danh sách DownLoad:**
Nhấn **Button Remove**
- **Để Limit tốc độ:**
Nhấn vào **cột limit** của File cần limit → **Sửa** giá trị → Nhấn vào **Button Limit**

3. Giao diện quản lý của Server

```

D:\Temp\Received\bitcrazy\bitcrazyServerv2.5.exe
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49825,localport=33333]
127.0.0.1:49825 >> SSN 29574506
127.0.0.1:49825 << TID 2
127.0.0.1:49825 >> END
127.0.0.1:49825 << END
Socket closed.
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49846,localport=33333]
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49847,localport=33333]
127.0.0.1:49847 >> GET 0
127.0.0.1:49847 << INF 127.0.0.1 1565122048
127.0.0.1:49847 >> END
127.0.0.1:49847 << END
Socket closed.
Error communicating with client: null
Socket closed.
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49858,localport=33333]
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49859,localport=33333]
127.0.0.1:49859 >> GET 0
127.0.0.1:49859 << INF 127.0.0.1 1565122048
127.0.0.1:49859 >> END

```

Server sẽ hiện ra tất cả các kết nối tới và các giao thức được sử dụng trong chương trình Đặc biệt sẽ cấp cho File 1 ID để các máy khác có thể dowload

Với cú pháp TID <id>

```
Accepted new incoming socket: Socket[addr=/127.0.0.1,port=49825,localport=33333]
127.0.0.1:49825 >> SSN 29574506
127.0.0.1:49825 << TID 2
127.0.0.1:49825 >> END
127.0.0.1:49825 << END
```

Server sẽ quản lý tất cả các File, người dùng, người dùng đang kết nối, hiện trạng tập tin bằng chính ID được Server cấp phát cho các máy.

IV. Tổng kết

Bitcrazy là chương trình download đơn giản, tuy nhiên nó đã mang đầy đủ các tính năng của chương trình chia sẻ file P2P, việc hiện thực nó giúp cho các thành viên trong nhóm có thêm các kiến thức về lập trình Java UI, Socket, kỹ năng Làm việc nhóm. Đặc biệt là hiểu sâu hơn về Socket cách truyền tải file qua mạng bằng giao thức TCP.