

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HỒ CHÍ MINH  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**MÔN: MẠNG 1  
BÁO CÁO BÀI TẬP LỚN 1**

**ĐỀ TÀI: ĐỊNH NGHĨA GIAO THỨC VÀ HIỆN THỰC  
ỨNG DỤNG CHIA SẺ FILE TRỰC TUYẾN**

**GVHD:**

**TS.PHẠM TRẦN VŨ  
THS.NGUYỄN HỒNG NAM**

**NHÓM 3:**

**HUỖNH QUANG BẢO – 51300225  
ĐẶNG QUỐC SƠN – 51303399  
LÊ HỮU TẤN – 51303576**

**---14/04/2016---**

## Mục lục

1. Giới thiệu .....	4
1.1 Thuật ngữ:.....	4
1.2 Tổng quát về cách thức hoạt động .....	4
2. Một số trường thông tin.....	5
a. <i>FileInfo</i> .....	5
b. <i>PeerInfo</i> .....	5
3. Giao thức Tracker – Peer .....	5
a. <i>Trường dữ liệu trao đổi giữa peer – tracker</i> .....	5
b. <i>Quá trình upload file</i> .....	6
c. <i>Keep alive connection (của uploader)</i> .....	7
d. <i>Quá trình yêu cầu danh sách file</i> .....	7
e. <i>Quá trình yêu cầu danh sách peer đang chia sẻ 1 file nào đó</i> .....	7
4. Giao thức Peer – Peer.....	8
a. <i>Trường dữ liệu từ downloader gửi sang uploader</i> .....	8
b. <i>Trường dữ liệu từ uploader gửi sang downloader</i> .....	8
c. <i>Quá trình handshaking và download dữ liệu</i> .....	8
5. Kết quả hiện thực .....	10
5.1 <i>common</i> .....	10
a. <i>Tag</i> .....	10
b. <i>FileInfo</i> .....	10
c. <i>PeerInfo</i> .....	10
d. <i>HashTable</i> .....	10
e. <i>Data</i> .....	11
f. <i>General</i> .....	11
5.2 <i>p2p</i> .....	11
a. <i>Downloader</i> .....	11
b. <i>DownManager</i> .....	12
c. <i>Uploader</i> .....	12
d. <i>UpManager</i> .....	13
e. <i>KeepConnection</i> .....	14
5.3 <i>tracker</i> .....	16
a. <i>Tracker</i> .....	16
b. <i>TrackerMonitor</i> .....	17



5.4	Giao diện chương trình .....	18
a.	Giới thiệu các component được sử dụng .....	18
6.	Đánh giá .....	19
6.1	Ưu điểm .....	19
6.2	Khuyết điểm .....	20

# BST protocol

## 1. Giới thiệu

### 1.1 Thuật ngữ:

- Peer: nút mạng tham gia vào chia sẻ file. Có thể đồng thời vừa là downloader và uploader.
- Swarm: là một mạng các peer đang tham gia chia sẻ cùng một file nào đó.
- Tracker: server trung tâm lưu giữ thông tin về các file đang được chia sẻ và danh sách các peer đang tham gia chia sẻ các file đó. Đóng vai trò như cánh cổng cho peer mới có thể tham gia vào swarm.
- Peer ID: là một chuỗi 6 byte dùng để định danh một peer. Peer ID là duy nhất trong một swarm. Peer ID được sinh ra ngẫu nhiên khi một peer được tạo.
- Hash: giá trị SHA1 hash dùng để xác định một file. Được tính toán từ dữ liệu trong file.
- Piece: khi chia sẻ hay download file, file được chia ra làm nhiều mảnh nhỏ (giá trị từ 64KB – 1MB) – piece.

### 1.2 Tổng quát về cách thức hoạt động

*BST Protocol bao gồm 2 giao thức riêng biệt:*

- Tracker – Peer protocol:** quy định các phương thức giao tiếp, trao đổi thông tin của peer và tracker về danh sách thông tin các file và danh sách các peer tham gia download hoặc cần upload một file xác định nào đó.
- Peer – Peer protocol:** quy định phương thức giao tiếp giữa các peer với nhau – quá trình download và upload dữ liệu các file.

*Để tham gia download 1 file, peer cần phải thực hiện lần lượt các bước sau:*

- Kết nối tới tracker, yêu cầu tracker gửi danh sách thông tin file đang được chia sẻ trên mạng.
- Khi có danh sách file từ tracker, peer chọn file cần download, gửi thông tin file đó lên tracker.
- Tracker sẽ trả về danh sách các peer đang tham gia chia sẻ file mà peer yêu cầu, peer download sẽ liên hệ với các peer trong danh sách và yêu cầu download dữ liệu file từ các peer đó.

*Để tham gia upload 1 file, peer cần phải thực hiện lần lượt các bước sau:*

- Kết nối tới tracker, yêu cầu upload file, đồng thời gửi thông tin file cần upload lên tracker.

- Trong quá trình upload, peer cần phải gửi các thông báo “keep alive” lên tracker, để tracker xác định được rằng peer vẫn còn upload file, sau những khoảng thời gian nhất định.

## 2. Một số trường thông tin

### a. FileInfo

*Thông tin về file (FileInfo): chứa thông tin cần thiết về một file để có thể trao đổi giữa các peer với nhau hoặc giữa peer với tracker, gồm:*

- Name: tên file.
- Size: dung lượng file.
- Hash: giá trị SHA1 của 1 file – dùng để xác định 1 file là duy nhất, hoặc có thể xác định được 2 file có tên khác nhau nhưng nội dung như nhau là cùng 1 file.

### b. PeerInfo

*Thông tin về peer (PeerInfo): chứa thông tin cần thiết về 1 peer để có thể trao đổi giữa các peer với nhau hoặc giữa peer với tracker, gồm:*

- ID: định danh của peer, là 1 số có 6 chữ số, được tạo 1 cách ngẫu nhiên khi chương trình chạy.
- IP: địa chỉ IP của máy tính mà chương trình đang chạy trên đó.
- Port: địa chỉ port mà peer đang đợi các peer khác kết nối tới để download file.
- Alive: khi 1 peer upload file, nó gửi thông tin cho tracker, sau 1 khởi thời gian Alive, nó phải gửi lại thông tin xác nhận vẫn còn muốn upload, sau khoảng thời gian Alive trên mà peer không gửi thông tin xác nhận thì tracker xóa peer đó ra khỏi danh sách các peer trong 1 swarm mà nó tham gia.

## 3. Giao thức Tracker – Peer

### a. Trường dữ liệu trao đổi giữa peer – tracker

PEER\_ID – 6 bytes

HASH – 40 bytes

FILE\_NAME - instance

Tên file không được xác định trước giá trị chuỗi byte cố định, khi trao đổi, tracker phải gửi độ dài tên file cho tracker chuẩn bị buffer.

FILE\_SIZE – 8 bytes

Dung lượng của bất kì file nào được quy định kiểu **long**. Dung lượng file được tracker gửi cho peer để peer xác định số byte cần đọc vào buffer khi nhận nội dung file (byte stream).

MESSAGE – 10 bytes

Tin nhắn trao đổi giữa peer và tracker, bao gồm: các yêu cầu tác vụ, xác nhận dữ liệu, yêu cầu kết thúc.

## b. Quá trình upload file

*Khi 1 peer muốn upload file cho các peer khác download, nó cần liên lạc với tracker để “đăng kí upload”.*

- Giao thức:
  - Peer kết nối tới tracker, khi kết nối thành công, peer và tracker bắt đầu việc trao đổi dữ liệu
  - Peer gửi thông tin của mình lên tracker, gồm:
    - Peer ID (PEER\_ID): peer gửi trực tiếp lên tracker.
    - Địa chỉ IP, địa chỉ port của peer: tracker lấy gián tiếp qua socket nó kết nối.
  - Peer gửi yêu cầu (MESSAGE) muốn upload file.
  - Tracker phản hồi (MESSAGE) yêu cầu của peer được chấp nhận thì peer gửi cho tracker các thông tin sau:
    - File: giá trị hash (HASH), tên file(FILE\_NAME), dung lượng file (FILE\_SIZE) của file cần upload.
    - Các thông tin về file được gửi lần lượt, sau mỗi lần gửi, peer đợi xác nhận (MESSAGE) của tracker mới gửi tiếp thông tin kế.
  - Khi gửi xong các thông tin cho tracker, peer đợi tracker xác nhận (MESSAGE) đã nhận đầy đủ thông tin, sau đó kết thúc kết nối.
  - Sau khi nhận được hết thông tin về file và peer, tracker gửi xác nhận tới peer upload và kết thúc kết nối.
- Tracker lưu giữ thông tin peer ứng file peer đó chia sẻ, thêm peer này vào swarm đã có sẵn hoặc tạo mới swarm cho file đó (nếu file này chưa được chia sẻ bởi bất kì peer nào).
- Khi peer được thêm mới vào danh sách, tracker sẽ lưu lại thuộc tính “alive” (thời gian sống của 1 peer – giảm dần giá trị qua thời gian) của peer, nếu sau khoảng thời gian nhất định, peer không có liên lạc gì với tracker để refresh thuộc tính “alive”, thì tracker sẽ xóa thông tin peer khỏi danh sách upload file.

### **c. Keep alive connection (của uploader)**

- Khi peer “đăng kí” với tracker upload file nào đó, peer upload phải đều đặn (sau 1 khoảng thời gian nào đó) gửi xác nhận lên cho tracker là nó vẫn còn “online” để cho phép các peer khác download dữ liệu.
- Giao thức:
  - Sau khoảng thời gian xác định (3 – 5p): peer gửi lại tracker yêu cầu upload file (tương tự như khi nó muốn upload file).

### **d. Quá trình yêu cầu danh sách file**

*Khi 1 peer muốn lấy danh sách các file đang được chia sẻ, peer phải qua các bước sau:*

- Kết nối tới tracker.
- Gửi yêu cầu (MESSAGE): cần danh sách các file đang được chia sẻ trên mạng.
- Tracker kiểm tra danh sách file đang được chia sẻ, gửi peer dung lượng file (FILE\_SIZE) lưu trữ danh sách các file.
- Nếu file sách file rỗng, tracker gửi dung lượng bằng “0” tới peer, và quá trình trao đổi dữ liệu kết thúc, ngắt kết nối.
- Nếu danh sách file không rỗng, tracker gửi dung lượng file lưu trữ danh sách file cho peer, sau đó gửi dữ liệu file lưu trữ danh sách file.
- Khi peer nhận được đầy đủ gói tin chứa file, gửi tín hiệu kết thúc (MESSAGE) tới tracker, quá trình yêu cầu danh sách file kết thúc.

### **e. Quá trình yêu cầu danh sách peer đang chia sẻ 1 file nào đó**

*Khi có được danh sách file, peer cần download chọn file cần download, sau đó kết nối tới tracker:*

- Peer gửi yêu cầu (MESSAGE) cần lấy danh sách peer đang download 1 file xác định nào đó.
- Tracker gửi yêu cầu giá trị hash (HASH) của file mà peer cần download.
- Peer gửi giá trị hash lên tracker:
  - Nếu file có giá trị hash peer gửi lên còn tồn tại, quá trình trao đổi dữ liệu tiếp tục:
    - Tracker gửi peer độ dài file lưu trữ thông tin peer đang chia sẻ file cần download (FILE\_SIZE).
    - Peer gửi xác nhận đã nhận được gói tin lên tracker (MESSAGE).
    - Tracker gửi nội dung file lưu giữ danh sách peer cho peer.
    - Khi gửi xong file, tracker tự động kết thúc kết nối, quá trình trao đổi kết thúc.

- Sau khi nhận được danh sách, peer kết thúc quá trình trao đổi với tracker và tiến hành kết nối tới các peer có trong danh sách và download file.
- o Nếu giá trị hash peer gửi lên không còn tồn tại, quá trình trao đổi dữ liệu kết thúc.

## 4. Giao thức Peer – Peer

Quy định các trường dữ liệu:

### a. Trường dữ liệu từ downloader gửi sang uploader

HASH	
MESSAGE	PIECE

### b. Trường dữ liệu từ uploader gửi sang downloader

PEER ID		
MESSAGE	PIECE	DATA

- HASH: 40 byte, giá trị Hash.
- MESSAGE: 10 byte, tin nhắn trao đổi giữa uploader và downloader.
- PIECE: 4 byte, giá trị piece mà downloader yêu cầu uploader gửi.
- PEER ID: 6 byte, giá trị peer ID.
- DATA: 128KB, dữ liệu được gửi đi mỗi lần.

Các tin nhắn trao đổi giữa downloader và uploader(MESSAGE), gồm:

- Xác nhận giá trị hash đúng/sai.
- Xác nhận giá trị peer ID đúng/sai.
- Tin nhắn yêu cầu nhận dữ liệu từ uploader.
- Tin nhắn kết thúc kết nối được.

**Ghi chú:** Với các tin nhắn từ downloader đến uploader như: xác nhận giá trị hash đúng/sai, xác nhận giá trị peer đúng/sai, tin nhắn kết thúc kết nối thì trường PIECE VALUE và DATA bỏ trống.

### c. Quá trình handshaking và download dữ liệu

Peer - uploader: peer cho peer khác download.



*Peer - downloader: peer download từ peer khác.*

### **Quá trình handshaking**

- Downloader kết nối tới uploader với địa chỉ port và ip từ PeerInfo
- Sau khi kết nối thành công, downloader gửi uploader giá trị hash (HASH VALUE) của file cần down (giá trị hash lấy từ FileInfo của file cần download).
- Uploader nhận giá trị hash từ downloader, so sánh với giá trị hash hiện có, có 2 trường hợp xảy ra:
  - Giá trị hash khớp:
    - Giá trị hash từ downloader khớp với giá trị hash của file uploader đang chia sẻ, thì uploader gửi downloader giá trị peer ID (PEER ID) của mình.
    - Downloader nhận giá trị peer ID của uploader, so sánh với các giá trị peer ID trong file PeerInfo, có 2 trường hợp xảy ra.
  - Giá trị hash không khớp:
    - Uploader gửi downloader tín hiệu sai giá trị hash.
    - Downloader gửi tín hiệu kết thúc kết nối.
- Nếu giá trị hash khớp, thì peer ID được gửi từ uploader sang downloader, downloader so sánh giá trị peer ID nhận được với danh sách peer ID đang giữ:
  - Nếu có giá trị khớp với peer ID từ uploader:
    - Quá trình handshaking thành công, chuyển sang download dữ liệu.
  - Nếu giá trị peer ID từ uploader không khớp:
    - Downloader gửi tín hiệu kết thúc kết nối.

### **Quá trình download dữ liệu**

Sau khi quá trình handshaking thành công, thì 2 peer tiến hành trao đổi dữ liệu:

- Downloader gửi yêu cầu download (MESSAGE) kèm theo giá trị piece (PIECE) của file cần download.
- Uploader nhận dữ liệu từ downloader:
  - Nếu MESSAGE từ downloader yêu cầu gửi dữ liệu, thì uploader gửi lại dữ liệu lại cho downloader gồm: tin nhắn xác nhận yêu cầu gửi (MESSAGE), giá trị piece dữ liệu được gửi (PIECE) và dữ liệu của file ứng với giá trị piece mà downloader yêu cầu (DATA).
  - Nếu MESSAGE từ downloader yêu cầu kết thúc kết nối thì uploader gửi xác nhận kết thúc quá trình trao đổi, sau đó ngắt kết nối với downloader.
- Quá trình gửi giá trị piece từ downloader, gửi dữ liệu từ uploader lặp đi lặp lại đến khi downloader không cần download từ uploader nữa, và gửi tín hiệu kết thúc download thay vì giá trị piece.

## 5. Kết quả hiện thực

*Các có 3 package chính:*

- common.
- p2p.
- tracker.

### 5.1 common

*Package common chứa các class dùng chung cho tracker và peer, gồm các class:*

#### a. Tag

*Chứa các hằng dùng trong chương trình:*

- MESSAGE – yêu cầu upload, yêu cầu kết thúc, ....
- Size của các trường dữ liệu như HASH, PEER\_ID,... , DATA (chứa dữ liệu file truyền đi).

#### b. FileInfo

*Chứa các trường dữ liệu cần thiết để mô tả thông tin về file, gồm:*

- Name: tên file.
- Size: dung lượng file.
- Hash: giá trị SHA1 của file.

#### c. PeerInfo

*Chứa các trường dữ liệu cần thiết để mô tả thông tin về peer, gồm:*

- peerID: ID của peer – giá trị định danh của peer trong 1 swarm.
- ipAddr: địa chỉ IP của peer.
- port: địa chỉ port của peer – port đang chờ kết nối từ các peer khác.
- alive: thời gian tồn tại tối đa (3 phút) của 1 peer khi nó kết nối tới tracker mà không gửi thông điệp “keep alive” tới tracker sau mỗi khoảng thời gian nhất định (nhóm hiện thực 2 phút).

#### d. HashTable

*Lưu trữ danh sách peer ứng với giá trị hash của file mà các peer này đang chia sẻ, gồm:*

- HashValue: giá trị hash của file đang được chia sẻ.
- List<PeerInfo> : danh sách các peer đang chia sẻ file.

### e. Data

*Dữ liệu trao đổi giữa các peer, gồm:*

- mssgType: yêu cầu (MESSAGE).
- pieceIndex: giá trị piece mà downloader yêu cầu từ uploader.
- byteArr: mảng các byte gồm 128KB, dữ liệu file mà uploader gửi cho downloader.

*Đối với dữ liệu mà downloader gửi sang uploader thì trường byteArr để trống.*

### f. General

- Không chứa trường dữ liệu.
- Chứa các phương thức thao tác lên dữ liệu: như đọc ghi file, tính giá trị hash, chuyển chuỗi byte sang class dữ liệu, ...

## 5.2 p2p

*Package p2p chứa các class dùng cho các peer: quản lý, kiểm soát quá trình download, upload file, kết nối và gửi yêu cầu lên tracker.*

### a. Downloader

*Class downloader thừa kế class Thread, có thể chạy multi – thread. Khi download 1 file, chương trình tạo nhiều thread downloader kết nối tới các peer khác để download từng đoạn file xác định.*

*Hoạt động:*

- Khởi tạo TCP socket kết nối tới 1 peer - địa chỉ IP và địa chỉ port của peer được cho trước.
- Khởi tạo các stream (I/O stream) để trao đổi dữ liệu với peer:
  - Input stream: đường dữ liệu từ peer gửi đến.
  - Output stream: đường gửi dữ liệu đến peer.
  - Khởi tạo stream đến file được chỉ định (RandomAccessFile) để ghi dữ liệu download vào file.

*Sau khi khởi tạo xong kết nối và các stream I/O, quá trình trao đổi dữ liệu bắt đầu:*

- Downloader gửi giá trị hash (40 bytes) của file cần download tới peer.
- Đọc từ input stream 14 bytes (14 bytes – MESSAGE + PIECE, vì giá trị piece trong bước này chưa dùng đến nên mặc định là 0), chuyển 10 byte đầu tiên thành chuỗi string:

- Nếu chuỗi string đó có nội dung xác nhận giá trị hash sai: downloader gửi 10 byte dữ liệu ra output stream (thông điệp kết thúc kết nối) tới peer, sau đó kết thúc kết nối.
- Nếu chuỗi string có nội dung xác nhận giá trị hash đúng: tiếp tục các bước kế tiếp.
- Downloader tiếp tục gửi thông điệp tới peer: 10 byte MESSAGE yêu cầu peer gửi giá trị PEER\_ID.
- Đọc 6 byte dữ liệu từ input stream: chuyển thành chuỗi, so sánh với giá trị PEER\_ID mong muốn (so sánh với giá trị PEER\_ID downloader lấy từ PeerInfo).
  - Nếu giá trị PEER\_ID không khớp, downloader gửi 10 byte MESSAGE kết thúc kết nối đến peer.
  - Nếu giá trị PEER\_ID khớp, tiếp tục quá trình.
- Downloader gửi tới output stream 14 byte dữ liệu – MESSAGE (yêu cầu download) và PIECE (vị trí đoạn dữ liệu của file mà downloader cần download).
- Đọc từ input stream 14 byte + 128KB dữ liệu – MESSAGE (thông điệp xác định đây là gói tin dữ liệu – 10 byte), PIECE (vị trí đoạn dữ liệu của file được gửi – 4 byte) và dữ liệu gửi (128KB).
- Khi lấy đoạn dữ liệu của file, thì ghi đoạn dữ liệu đó vào đúng vị trí của nó trong file stream (randomaccess file stream).
- Quá trình downloader gửi yêu cầu download và nhận dữ liệu từ peer lặp đi lặp lại cho đến khi downloader download đầy đủ các đoạn dữ liệu mà nó cần từ peer.
- Sau khi download xong tất cả các đoạn dữ liệu cần thiết, downloader gửi 10 byte dữ liệu vào output stream (MESSAGE – kết thúc kết nối).

#### b. DownManager

*Class DownManager cung cấp cơ chế quản lý các thread Downloader, mỗi thread download một số đoạn dữ liệu nhất định từ các peer khác nhau nhằm tối ưu hóa quá trình download.*

- Nhận danh sách peer đang chia sẻ file cần download từ tracker.
- Đếm số lượng peer có trong danh sách, từ đó chia nhỏ các phần của file ra tương ứng với số lượng peer.
- Tạo các thread Downloader kết nối tới các peer và tiến hành download.
- Xử lý ngoại lệ:
  - Nếu 1 peer bị mất kết nối, phần file được chia cho peer đó sẽ được chuyển sang 1 peer khác.
  - Nếu không có peer nào tồn tại, quá trình download thất bại.

#### c. Uploader

*Class uploader thừa kế class Thread, có thể chạy multi – thread. Khi upload 1 file, chương trình tạo nhiều thread uploader để có thể chia sẻ cùng lúc tới nhiều peer khác nhau.*

- Khởi tạo các stream I/O
  - Input stream: đường dữ liệu từ peer gửi đến.
  - Output stream: đường dữ liệu gửi đến peer.
  - Khởi tạo stream đến file được chỉ định (RandomAccessFile) để đọc dữ liệu upload cho các peer.
- Tính giá trị SHA1 của file đang upload (dùng để so sánh khi peer gửi giá trị hash đến).

*Uploader bắt đầu trao đổi dữ liệu với peer.*

- Đọc 40 byte từ input stream, chuyển thành chuỗi, so sánh với giá trị hash (giá trị SHA1 của file đang được upload).
  - Nếu giá trị vừa nhận khác giá trị hash của file: uploader gửi thông điệp tới peer (MESSAGE) xác nhận giá trị hash sai (gửi qua output stream).
  - Nếu giá trị vừa nhận giống giá trị hash của file: uploader gửi thông điệp tới peer (MESSAGE) xác nhận giá trị hash đúng (gửi qua output stream).
- Đọc tiếp 10 byte dữ liệu từ input stream (MESSAGE) chuyển thành chuỗi.
  - Nếu chuỗi là thông điệp yêu cầu gửi PEER\_ID thì uploader gửi đến output stream giá trị PEER\_ID của mình (6 byte), quá trình tiếp tục các bước sau.
  - Nếu chuỗi là thông điệp yêu cầu kết thúc kết nối (giá trị hash không khớp), uploader kết thúc kết nối.
- Sau khi gửi giá trị PEER\_ID, uploader lặp đi lặp lại quá trình sau cho đến khi peer gửi thông điệp (MESSAGE) kết thúc kết nối:
- Đọc 14 byte dữ liệu từ input stream (MESSAGE – yêu cầu gửi dữ liệu + PIECE – giá trị piece xác định vị trí đoạn dữ liệu trong file cần gửi).
  - Nếu 10 byte MESSAGE không phải là thông điệp kết thúc kết nối:
    - Lấy đoạn dữ liệu cần gửi từ file stream (RandomAccessFile).
    - Gửi tới peer 14 byte + 128KB (10 byte MESSAGE + 4 byte PIECE + 128KB DATA).
  - Nếu 10 byte MESSAGE là thông điệp kết thúc kết nối:
    - Uploader ngắt kết nối tới peer.

#### d. UpManager

- Class UpManager cung cấp cơ chế quản lý các thread Uploader. Khi peer upload 1 file, 1 instance UpManager được tạo ra, khi khởi động, UpManager mở port được định sẵn, chờ các peer khác kết nối tới. Mỗi khi có peer kết nối tới, UpManager tạo 1 thread Uploader để giao tiếp với peer cần download. Do vậy, khi peer upload 1 file, chương trình có khả năng phục vụ đồng thời nhiều yêu cầu download từ các peer khác nhau.
- Trong class UpManager, sử dụng class Timer để định thời, sau những khoảng thời gian nhất định (2 phút), UpManager sẽ gửi “Keep alive connection” lên tracker để thông báo sự tồn tại của mình.

#### e. KeepConnection

*Class KeepConnection cung cấp cơ chế giao tiếp giữa peer với tracker. Có 3 chức năng chính:*

- Đăng kí upload file.
- Keep alive connection.
- Yêu cầu danh sách file đang được chia sẻ.
- Yêu cầu danh sách peer để download file đang được chia sẻ.

#### *Hoạt động*

- Kết nối tới tracker.
- Khởi tạo các stream I/O
  - Input stream: đường dữ liệu từ peer gửi đến.
  - Output stream: đường dữ liệu gửi đến peer.
- Peer gửi thông điệp (MESSAGE – 10 byte) tới tracker – yêu cầu kết nối.
- Đọc 10 byte dữ liệu từ input stream, nếu có dữ liệu đọc được tức là tracker đã nhận yêu cầu và đang rảnh để đáp ứng yêu cầu của peer.
- Gửi tiếp giá trị PEER\_ID tới tracker.

*Tùy theo chức năng yêu cầu từ peer, sẽ có 3 giao thức để trao đổi dữ liệu với tracker.*

- Đăng kí upload file
  - Peer gửi tracker thông điệp (MESSAGE) yêu cầu “đăng kí” upload file.
  - Đợi tracker phản hồi (đọc 10 byte dữ liệu từ input stream): cho phép upload file.
  - Peer gửi tracker giá trị hash của file nó sẽ upload.
  - Đợi tracker phản hồi: đã nhận giá trị hash.
  - Peer gửi tracker độ dài tên file nó sẽ upload.
  - Đợi tracker phản hồi: đã nhận chiều dài tên file.
  - Peer gửi tracker tên file upload.
  - Đợi tracker phản hồi: đã nhận tên file.
  - Peer gửi tracker giá trị dung lượng file upload.

- Đợi tracker phản hồi: đã nhận giá trị dung lượng file upload.
  - Peer gửi tracker thông điệp (MESSAGE) kết thúc kết nối.
  - Quá trình trao đổi thông tin kết thúc.
- Keep alive connection
  - Giao thức hoàn toàn tương tự “đăng kí upload”. Khi peer bắt đầu upload file, thì nó sẽ “đăng kí upload” với tracker. Trong thời gian upload, nếu peer vẫn còn upload file thì phải đều đặn gửi “Keep alive connection” lên tracker để tracker theo dõi hoạt động của peer upload.
  - Khi gửi “Keep alive connection” lên tracker, tracker sẽ xem xét nếu trong cơ sở dữ liệu của nó đã tồn tại thông tin peer này (cùng 1 swarm) thì tracker sẽ update tình trạng của peer (thông qua thuộc tính alive của PeerInfo).
- Yêu cầu danh sách file đang được chia sẻ
  - Peer gửi yêu cầu (MESSAGE) tới tracker: yêu cầu danh sách file đang được chia sẻ.
  - Đọc độ dài file (1 byte) chứa danh sách file đang được chia sẻ. từ input stream (4 byte).
  - Gửi xác nhận (MESSAGE) đã nhận được độ dài file chứa danh sách file đang được chia sẻ. tới tracker.
  - Đọc 1 byte từ input stream: nội dung file chứa danh sách file đang được chia sẻ.
  - Gửi thông điệp (MESSAGE) kết thúc kết nối tới tracker.
  - Kết thúc quá trình trao đổi dữ liệu.
- Yêu cầu danh sách peer để download file
  - Peer gửi yêu cầu (MESSAGE) tới tracker: yêu cầu danh sách peer đang chia sẻ file cần download.
  - Đợi phản hồi từ tracker: đã nhận được yêu cầu của peer.
  - Gửi giá trị hash của file cần download tới tracker.
  - Đọc 10 byte từ input stream, chuyển thành chuỗi string:
    - Nếu chuỗi string là thông điệp (MESSAGE) kết thúc kết nối: nghĩa là file cần download hiện không còn peer nào chia sẻ.
      - Kết thúc kết nối tới tracker.
    - Nếu chuỗi string khác thông điệp (MESSAGE) kết thúc kết nối, tiếp tục các bước sau:
  - Chuyển 4 byte đầu tiên của chuỗi byte nhận được thành số nguyên – chính là dung lượng file chứa danh sách peer đang chia sẻ file cần download (1 byte).
    - Nếu  $l = 0$ : file mà peer yêu cầu download từ tracker không tồn tại (giá trị hash gửi lên tracker sai hoặc không còn peer nào trong mạng chia sẻ file đó).



- Nếu  $l \neq 0$ : file mà peer yêu cầu download từ tracker tồn tại, tiếp tục trao đổi dữ liệu với tracker.
- Gửi xác nhận tới tracker: đã nhận được giá trị dung lượng file chứa danh sách peer đang chia sẻ file cần download.
- Đọc 1 byte từ input stream: nội dung file chứa danh sách peer đang chia sẻ file cần download.
- Kết thúc kết nối tới tracker.

## 5.3 *tracker*

### a. Tracker

*Class tracker kế thừa class Thread cung cấp cơ chế giao tiếp với các peer.*

- Nhận các yêu cầu từ các peer và phục vụ.
- Quản lý danh sách các file đang được chia sẻ, danh sách các peer đang chia sẻ file.

#### *Hoạt động*

- Khi nhận được kết nối từ các peer, khởi tạo các stream I/O để trao đổi dữ liệu.
- Khi khởi tạo xong I/O stream, đọc 10 byte từ input stream, chuyển thành chuỗi string, đó là yêu cầu của peer gửi tới tracker.

#### *Đăng kí upload.*

Nếu thông điệp (MESSAGE) nhận được là yêu cầu cho phép đăng kí upload file từ peer.

- Gửi peer thông điệp (MESSAGE) sẵn sàng cho “đăng kí” upload file.
- Đọc 40 byte dữ liệu từ input stream – giá trị hash của file mà peer kết nối cần download.
- Gửi phản hồi tới peer: đã nhận giá trị hash.
- Đọc 4 byte dữ liệu từ input stream, chuyển thành kiểu số nguyên - độ dài tên file peer dự định upload (1 byte).
- Gửi phản hồi (MESSAGE) tới peer: đã nhận chiều dài tên file.
- Đọc 1 byte từ input stream, chuyển thành string – tên file mà peer upload.
- Gửi phản hồi tới peer: đã nhận tên file.
- Đọc 8 byte từ input stream, chuyển thành kiểu long - giá trị dung lượng file upload.
- Gửi phản hồi tới peer: đã nhận giá trị dung lượng file upload.
- Đọc 10 byte (MESSAGE) từ input stream – yêu cầu kết thúc kết nối (sau khi trao đổi xong tất cả các thông tin trên, peer sẽ gửi yêu cầu kết thúc kết nối).
- Quá trình trao đổi thông tin kết thúc.



*Yêu cầu danh sách file đang được chia sẻ.*

Nếu thông điệp (MESSAGE) nhận được là yêu cầu danh sách file đang được chia sẻ.

- Gửi độ dài file (l byte) chứa danh sách file đang được chia sẻ.
- Đọc xác nhận (MESSAGE) – đọc 10 byte dữ liệu từ input stream: đã nhận được độ dài file chứa danh sách file đang được chia sẻ từ peer.
- Gửi file chứa danh sách file đang được chia sẻ tới output stream.
- Khi peer nhận được đầy đủ file chứa danh sách file đang được chia sẻ, peer sẽ gửi phản hồi – yêu cầu kết thúc kết nối. Tracker đọc 10 byte dữ liệu từ input stream – thông điệp yêu cầu kết thúc kết nối từ peer.
- Kết thúc quá trình trao đổi dữ liệu.

*Yêu cầu danh sách peer đang chia sẻ 1 file nào đó.*

Nếu thông điệp (MESSAGE) nhận được là yêu cầu danh sách file đang được chia sẻ.

- Gửi phản hồi tới peer: đã nhận được yêu cầu của peer.
- Đọc giá trị hash được gửi tới từ peer (40 byte).
- Sau khi nhận được giá trị hash của file mà peer cần download, tracker tìm kiếm danh sách peer đang chia sẻ file đó trong cơ sở dữ liệu, sau đó gửi giá trị dung lượng file – l byte chứa danh sách peer chia sẻ file cần download.
  - Nếu trong cơ sở dữ liệu không tồn tại giá trị hash của peer gửi tới:
    - Tracker gửi peer giá trị l = 0.
    - Kết thúc trao đổi với peer.
  - Nếu trong cơ sở dữ liệu tồn tại giá trị hash của peer gửi tới:
    - Tracker gửi peer giá trị l.
- Đọc peer phản hồi (đọc 10 byte từ input stream – MESSAGE): đã nhận giá trị l.
- Tracker gửi l byte tới peer: nội dung file chứa danh sách các peer.
- Sau khi gửi hoàn tất, tracker kết thúc kết nối với peer.

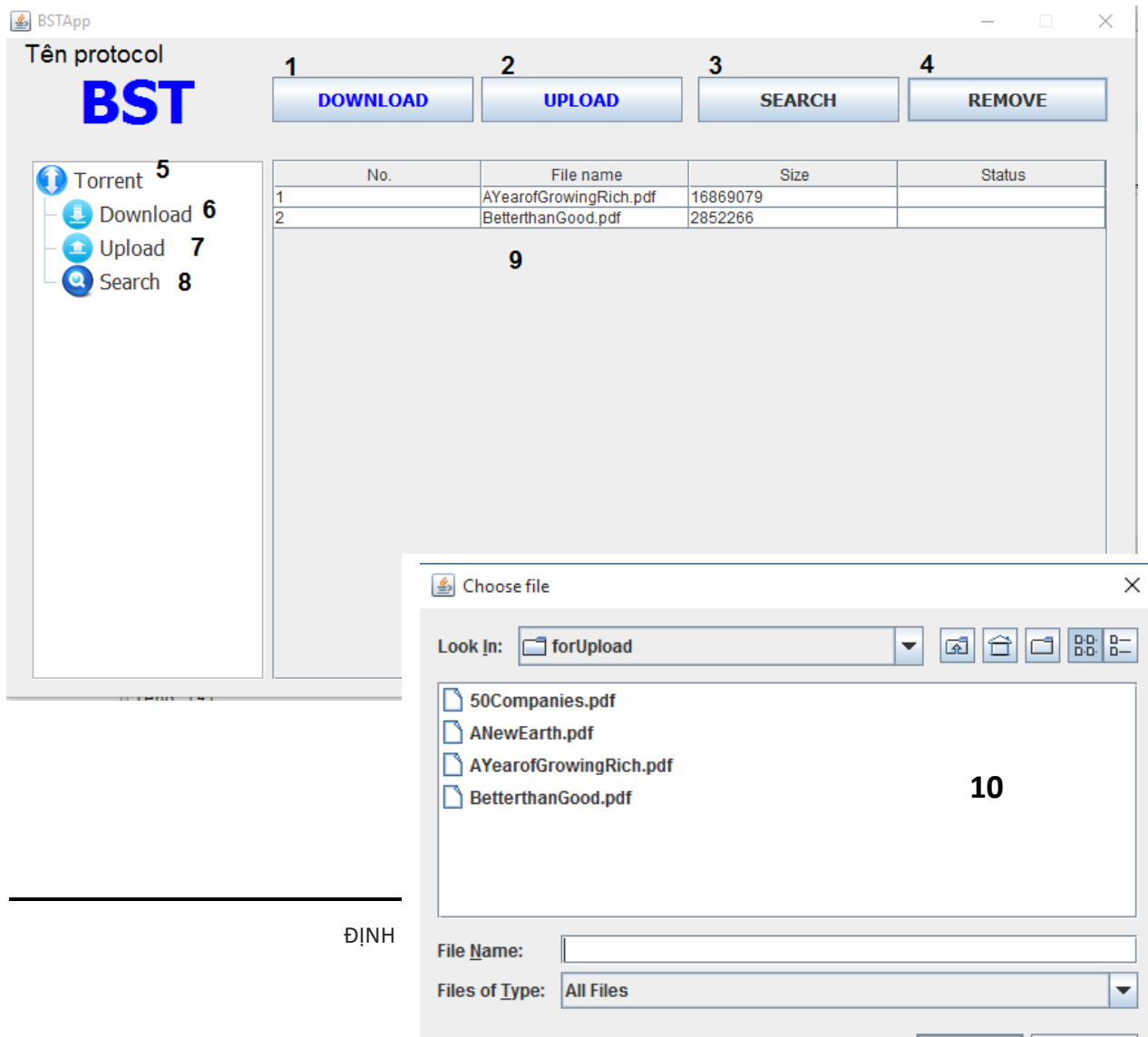
## **b. TrackerMonitor**

*Class TrackerMonitor quản lý các thread Uploader – chia sẻ đồng thời nhiều peer khác nhau, quản lý cơ sở dữ liệu – danh sách file đang được chia sẻ, danh sách peer đang chia sẻ các file có trong cơ sở dữ liệu, theo dõi tình trạng online của các peer upload.*

- Khởi động: tracker đợi các peer kết nối tới tại port xác định: 9876.
- Khi có peer kết nối tới, TrackerMonitor tạo ra 1 thread Tracker để làm việc với peer đó.
- Mọi thông tin có được từ các thread tracker về danh sách về danh sách file được chia sẻ, danh sách peer đang chia sẻ file được TrackerMonitor quản lý.
- Kiểm tra thuộc tính alive của peer sau mỗi phút để theo dõi tình trạng online của peer:
  - Sau mỗi lần kiểm tra, thuộc tính alive sẽ giảm đi 1 đơn vị.
  - Thuộc tính alive của peer tại thời điểm kiểm tra nếu bằng 0 thì peer đó sẽ bị xóa khỏi danh sách đang chia sẻ file.
- Kiểm tra danh sách file, file nào có danh sách peer đang chia sẻ nó trống thì file đó bị xóa khỏi danh sách các file đang được chia sẻ.

## 5.4 Giao diện chương trình

### a. Giới thiệu các component được sử dụng



## Giao diện của Peer

No.	Tên component	Chức năng
1	DOWNLOAD – JButton	Tiến hành download file được chọn trong JTable: kết nối tới tracker, yêu cầu danh sách peer đang chia sẻ file được chọn, kết nối tới các peer nhận được để download file.
2	UPLOAD – JButton	Tiến hành upload file được chọn trong JTable: kết nối tới tracker, đăng kí upload file được chọn (chọn qua JFileChooser), mở port chờ peer khác kết nối.
3	SEARCH – JButton	Kết nối tới tracker, yêu cầu danh sách file đang được chia sẻ.
4	REMOVE – JButton	Loại (dừng) upload file đang chia sẻ.
5	Torrent – JTree	Quản lý các JTree con: Download, Upload, Search.
6	Download – JTree	Chứa danh sách các file đang được download, hoặc đã download xong.
7	Upload – JTree	Chứa danh sách các file đang được upload.
8	Search – JTree	Hiển thị danh sách file đang được chia sẻ đã nhận từ tracker.
9	Display Panel – JTable	Hiển thị nội dung theo lựa chọn (Download, Upload, Search), gồm các trường hiển thị: No. (số thứ tự), File name (tên file), Size (dung lượng file – theo KB), Status (tình trạng file – đối với quá trình download).
10	ChooseFile- JFileChooser	Chức năng chọn file để upload.

## 6.Đánh giá

### 6.1 Ưu điểm

- Hiện thực chương trình theo đúng protocol đã định nghĩa.
- Tracker và peer có thể tương tác với peer, tracker có thể phục vụ nhiều peer cùng lúc.
- Tracker cập nhật đầy đủ, đúng về tình trạng online của các peer.



- Tracker cập nhật đầy đủ, đúng danh sách các file đang được chia sẻ.
- Hai peer có thể giao tiếp với nhau theo protocol định nghĩa.
- Peer có thể chia sẻ file cho các peer khác, có thể phục vụ cùng lúc nhiều peer.
- Peer có thể vừa download vừa upload file cho peer khác.
- Chương trình có giao diện để tương tác với người dùng.

## 6.2 Khuyết điểm

- Giao diện còn hạn chế, chưa đáp ứng được đầy đủ các tiện ích cho người dùng.
- Chưa xử lý các file có tên phức tạp: chứa dấu tiếng việt, khoảng trắng.
- Giá trị IP cho dùng cho tracker, peer khi demo còn đặt cứng trong code.