

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MẠNG MÁY TÍNH (CO3093)

Bài tập lớn

NETWORK APPLICATION

Giảng viên hướng dẫn: TS.Nguyễn Lê Duy Lai

Sinh viên: Trần Như Mai Anh (2210140)
Lê Duy Anh (2112762)

Thành phố Hồ Chí Minh, tháng 4 năm 2025



Mục lục

Danh sách thành viên và khối lượng công việc	2
1 Các chức năng của ứng dụng	3
1.1 Chia sẻ file và folder	3
1.2 Tải file và folder qua .torrent	3
1.3 Xem thông tin của mỗi torrent	3
2 Các protocol của ứng dụng	4
2.1 Giữa các Peer và Tracker Server	4
2.1.1 Tải lên	4
2.1.2 Tải về	5
2.1.3 Request /announce	5
2.1.4 Lấy thông tin scrape của file torrent	7
2.1.5 Định nghĩa các tham số trong /scrape Request	7
2.2 Giữa các Peer với nhau	8
2.2.1 2-way Handshake	8
2.2.2 Choke & Unchoke	9
2.2.3 Gửi Interest Uninterest	9
2.2.4 Gửi Bitfield	9
2.2.5 Gửi Pieces	10
3 Lược đồ Usecase	11
3.1 Đặc tả usecase	12
4 Lược đồ tuần tự	14
4.1 Share data	14
4.2 Download from torrent	15
4.3 Peer to Peer Protocol	16
5 Class Diagram	17
6 Hướng dẫn sử dụng	18
6.1 Khởi động ứng dụng	18
6.1.1 Từ source code	18
6.1.2 Từ Application file	18
6.2 Chia sẻ	18
6.3 Xem thông tin	20
6.4 Ngắt kết nối	20
7 Kiểm thử hệ thống	22
7.1 Share file	22
7.1.1 Tracker-CLient	22
7.1.2 Client-Client	22
7.2 Scrape server	23



Danh sách thành viên và khối lượng công việc

STT	Họ và tên	MSSV	Vấn đề	% hoàn thành
1	Trần Như Mai Anh	2210140	Hiện thực phần tracker, xây giao diện	100%
2	Lê Duy Anh	2112762	Hiện thực phần giao tiếp giữa các peer	100%

Bảng 1: Danh sách thành viên và khối lượng công việc



1 Các chức năng của ứng dụng

1.1 Chia sẻ file và folder

Một người dùng muốn chia sẻ một số file, hoặc một folder gồm nhiều file, khi nhấn nút "Share", cửa sổ File Explorer hiện lên để người dùng chọn các file và folder cần chia sẻ. Sau khi file/folder được tải lên, phần mềm sẽ xử lý và trả một file .torrent chứa các thông tin cần thiết cho người khác có thể dùng và đưa vào phần mềm để tải về các file hoặc folder đã được chia sẻ.

1.2 Tải file và folder qua .torrent

Người dùng có được file .torrent, nếu có nhu cầu muốn tải các file hoặc folder được chia sẻ thông qua file .torrent, thì mở ứng dụng và nhấn nút "Download". Lúc này cửa sổ Window Explorer sẽ hiện lên để người dùng chọn file .torrent chứa các file cần tải, sau đó tiến trình tải sẽ bắt đầu.

1.3 Xem thông tin của mỗi torrent

Nếu người dùng muốn xem thông tin các peer đang hoạt động (tải file/seed file) trong một torrent bất kì, thì nháy vào mục "Scrape", lúc này cửa sổ Window Explorer hiện lên để người dùng chọn file .torrent để xem các thông tin cần thiết.

2 Các protocol của ứng dụng

2.1 Giữa các Peer và Tracker Server

Mỗi khi người dùng khởi động ứng dụng, thì phiên hoạt động này sẽ trở thành một peer và được kết nối với Tracker Server (như mô hình Client-Server).

2.1.1 Tải lên

Khi người dùng tải lên một file hoặc một folder gồm nhiều file, phần mềm sẽ xử lý các file được tải lên và cho ra một file .torrent gồm các thông tin cần thiết để người khác có thể dùng và tải các file đã chia sẻ. Nội dung của file .torrent như sau.

Một file .torrent chứa các thông tin được bencoded dưới dạng dictionary, với các key chính sau:

1. info:
 - Mô tả: Dictionary chứa thông tin về file hoặc các file trong torrent.
 - Chi tiết:
 - Với torrent chứa một file (single-file torrent), chỉ có thông tin về file đó.
 - Với torrent chứa nhiều file (multi-file torrent), thông tin bao gồm cấu trúc thư mục và danh sách các file.
2. announce:
 - Mô tả: URL của tracker (chuỗi).
 - Vai trò: Tracker giúp kết nối các peers trong mạng P2P.
3. announce-list (tùy chọn):
 - Mô tả: Danh sách mở rộng URL tracker, dạng list of lists (mỗi tracker có thể có nhiều endpoint).
 - Lợi ích: Cung cấp tính tương thích ngược với các client cũ.
4. creation date (tùy chọn):
 - Mô tả: Thời điểm tạo file torrent, dạng UNIX epoch (số giây từ 01/01/1970 00:00:00 UTC).
 - Kiểu dữ liệu: Integer.
5. comment (tùy chọn):
 - Mô tả: Nhận xét tự do từ tác giả file torrent (chuỗi).
6. created by (tùy chọn):
 - Mô tả: Tên và phiên bản của chương trình dùng để tạo file torrent (chuỗi).
7. encoding (tùy chọn):
 - Mô tả: Định dạng mã hoá chuỗi được sử dụng trong phần pieces của dictionary info (chuỗi).
 - Ví dụ: UTF-8 hoặc các định dạng khác.



Sau khi file .torrent được tạo:

- Peer sẽ thông báo với Tracker Server rằng peer đang giữ file này và sẵn sàng để được leech bởi các peer khác nếu họ có nhu cầu, thông qua request /announce.
- Tracker Server sẽ thêm peer này vào các peer đang hoạt động trong torrent trên.
- Tracker Server gửi lại thông tin của các peer đang hoạt động trong torrent này cho peer này (có bao gồm chính nó).

2.1.2 Tải về

Người dùng tải lên một file .torrent để tải các file được chia sẻ từ torrent này. Lúc này quy trình sẽ như sau:

- Peer sẽ gửi thông báo tới Tracker Server được liệt kê trong file torrent, với yêu cầu thông qua /announce.
- Tracker Server:
 - Tìm các peer đang hoạt động trong torrent tương ứng.
 - Thêm thông tin của peer yêu cầu vào danh sách các peer hoạt động trong torrent này.
 - Gửi lại thông tin của các peer đang hoạt động trong torrent này cho peer đang cần tải (có bao gồm chính nó).

Sau khi nhận được thông tin này, peer sẽ thực hiện kết nối trực tiếp với các peer được liệt kê để tải file.

2.1.3 Request /announce

```
GET / announce?info_hash=b%DE%D6%C3y%8E1%00%3E%07%BE%9F
%9F6Z%FA%2A%0A%D1%FC&peer_id=-PY0001-
YYbuyNCwrD1y&ip=10.130.85.145&port
=64219&uploaded=0&downloaded=0&left=0&compact=
0&event=STARTED
HTTP /1.1
Host : tracker.example.com
```

Định nghĩa các tham số trong /announce request:

- info_hash:
 - Một chuỗi 20 byte SHA1 hash, URL-encoded, được lấy từ giá trị của khóa info trong file Metainfo (.torrent).
 - Giá trị của info là một dictionary được mã hóa bằng bencode.
- peer_id:
 - Một chuỗi 20 byte, URL-encoded, được sử dụng làm ID duy nhất cho peer.
 - Peer ID được tạo bởi client khi khởi động và có thể là bất kỳ giá trị nào, bao gồm dữ liệu nhị phân
 - Một số client thường mã hóa Peer ID theo các tiêu chuẩn riêng, ví dụ: -AZ2060-, -UT3530-, ở đây dùng -PY0001.



- ip:
 - Địa chỉ IP của peer, thường được sử dụng để xác định peer trên mạng.
 - Đây là một tham số tùy chọn. Nếu không được cung cấp, Tracker sẽ sử dụng địa chỉ IP từ kết nối hiện tại.
- port:
 - Số cổng mà client đang lắng nghe.
 - Các cổng phổ biến cho BitTorrent nằm trong dải 6881-6889.
 - Nếu không thiết lập được cổng trong dải này, client có thể chuyển sang cổng ngẫu nhiên.
- uploaded:
 - Tổng số byte đã tải lên kể từ khi sự kiện started được gửi tới Tracker, biểu diễn dưới dạng ASCII.
- downloaded:
 - Tổng số byte đã tải xuống kể từ khi sự kiện started được gửi tới Tracker, biểu diễn dưới dạng ASCII.
- left:
 - Số byte còn lại mà peer cần tải xuống để hoàn thành torrent, biểu diễn dưới dạng ASCII.
 - Giá trị này giúp Tracker biết peer đã hoàn thành việc tải file hay chưa.
- compact:
 - Nếu được đặt thành 1, client sẽ nhận phản hồi dạng compact (danh sách peers được thay thế bằng một chuỗi byte gồm 6 byte mỗi peer: 4 byte địa chỉ IP và 2 byte cổng).
 - Nếu được đặt thành 0, client yêu cầu phản hồi không compact (thường sẽ bị từ chối bởi một số Tracker để tiết kiệm băng thông).
- event:
 - Loại sự kiện mà peer gửi tới Tracker. Các giá trị có thể bao gồm:
 - * STARTED: Peer bắt đầu tải torrent.
 - * STOPPED: Peer dừng tải torrent.
 - * COMPLETED: Peer đã hoàn thành việc tải xuống toàn bộ torrent.

Tracker Server nhận yêu cầu /announce, phản hồi sẽ có cấu trúc

```
{
  "tracker_id": "TRACKER123",
  "info_hash": "abc123",
  "peers": [
    {
      "peer_id": "-AZ2060 -6 wfG2wk6wF00 ",
      "ip": "192.168.1.5",
      "port": 6881
    },
    {
      "peer_id": "-UT3530 -9 dbXwJgQ8Kmn ",

```



```
{
  "ip": "203.0.113.25",
  "port": 51413
}
]
```

- tracker_id: ID của Tracker Server.
- info_hash: Mã băm của torrent đang được yêu cầu.
- peers: Danh sách các peer đang hoạt động, bao gồm:
 - peer_id: Chuỗi định danh của peer
 - ip: Địa chỉ IP của peer.
 - port: Cổng kết nối của peer.

2.1.4 Lấy thông tin scrape của file torrent

Nếu người dùng không có nhu cầu tải hoặc chia sẻ file, mà chỉ muốn xem thông tin các peer đang hoạt động với một torrent bất kì, thì peer tương ứng với phiên đăng nhập của người dùng sẽ gửi scrape request của file .torrent đến Tracker Server.

Một yêu cầu /scrape cơ bản có thể có định dạng như sau:

```
GET / scrape?info_hash=%D1%FC%3E%07%BE%9F%9F6Z%FA%2A%0A HTTP /1.1
Host : tracker . example . com
Connection : close
```

2.1.5 Định nghĩa các tham số trong /scrape Request

- info_hash: Chuỗi định danh của peer
 - Một chuỗi 20 byte SHA1 hash, URL-encoded, đại diện cho torrent mà client muốn lấy thông tin.
 - Nhiều info_hash có thể được gửi trong một yêu cầu /scrape bằng cách phân tách các giá trị bằng ký tự &.

Tracker Server sẽ tìm kiếm thông tin các peer đang hoạt động với file torrent này và trả về kết quả như sau:

```
{
  "tracker_id": "tracker-12345",
  "info_hash": "b2d6c3798e6c003e07be9f9f365afa2a0ad1fc",
  "total_peers": 42
}
```

Với:

- tracker_id:
 - Chuỗi định danh duy nhất của Tracker Server đang phản hồi.



- `info_hash`:
 - Chuỗi 20 byte (SHA1 hash) xác định torrent mà yêu cầu liên quan đến.
- `total_peers`:
 - Số lượng peer đang hoạt động liên quan đến `info_hash`.

Peer nhận được thông tin từ scrape request, peer sẽ không làm gì khác, trong khi với announce request, peer sẽ kết nối với các peer được liệt kê trong thông tin trả về nếu là để tải file.

2.2 Giữa các Peer với nhau

Khi một peer đã nhận được danh sách các peer đang seed hoặc leech torrent từ tracker, bước tiếp theo là thiết lập kết nối trực tiếp với các peer này để trao đổi dữ liệu.

2.2.1 2-way Handshake

Handshake là thông điệp bắt buộc và phải là thông điệp đầu tiên được gửi giữa hai peer muốn kết nối. Thông điệp này xác minh rằng cả hai peer đang sử dụng cùng giao thức và cùng tham gia vào một torrent cụ thể.

Cấu trúc của thông điệp handshake:

$$\text{handshake} := \langle \text{pstrlen} \rangle \langle \text{pstr} \rangle \langle \text{reserved} \rangle \langle \text{info_hash} \rangle \langle \text{peer_id} \rangle$$

Với:

- `pstrlen`: Độ dài của chuỗi giao thức (`<pstr>`), được biểu diễn bằng một byte.
- `pstr`: Chuỗi định danh giao thức, thường là "BitTorrent protocol".
- `reserved`: 8 byte dự phòng, tất cả thường được đặt là 0.
- `pstr: info_hash`: Băm SHA-1 dài 20 byte của khóa info trong tệp .torrent.
- `peer_id`: Chuỗi dài 20 byte, là định danh duy nhất cho mỗi peer.

Sau khi nhận được thông điệp handshake từ một peer khác, peer cần kiểm tra `info_hash` để xác nhận rằng torrent được tham gia là cùng một torrent, trước khi tiến hành giao tiếp thêm.

Quy trình:

- Peer A gửi thông điệp handshake cho Peer B.
- Peer B nhận, kiểm tra `info_hash` và phản hồi bằng thông điệp handshake tương ứng.



2.2.2 Choke & Unchoke

Các thông điệp Choke và Unchoke điều khiển quyền truy cập của một peer để tải xuống dữ liệu:

- Choke ($\langle id=0 \rangle$): Thông điệp chặn, thông báo rằng peer sẽ không gửi dữ liệu cho peer

`choke : $\langle len=0001 \rangle \langle id=0 \rangle$`

khác. Thông điệp này không có payload.

- Unchoke ($\langle id=1 \rangle$): Thông điệp bỏ chặn, cho phép peer yêu cầu và nhận dữ liệu.

`unchoke : $\langle len=0001 \rangle \langle id=1 \rangle$`

Việc sử dụng choke và unchoke giúp các peer ưu tiên tài nguyên mạng cho những kết nối mang lại hiệu quả cao hơn.

Quy trình:

- Ban đầu, Peer A gửi Choke để từ chối yêu cầu từ Peer B.
- Khi sẵn sàng chia sẻ, Peer A gửi Unchoke.

2.2.3 Gửi Interest Uninterest

Các thông điệp interested và not interested được sử dụng để thể hiện trạng thái quan tâm hoặc không quan tâm đến việc nhận dữ liệu từ một peer.

- interested: Peer thông báo quan tâm đến dữ liệu của peer khác.

`interested : $\langle len=0001 \rangle \langle id=2 \rangle$`

- not interested: Peer không còn quan tâm đến dữ liệu.

`not interested : $\langle len=0001 \rangle \langle id=3 \rangle$`

Quy trình:

- Peer A gửi Interested nếu bitfield của Peer B chứa phần dữ liệu mà Peer A cần.
- Khi đã hoàn tất hoặc không cần thêm dữ liệu, Peer A gửi Not Interested.

2.2.4 Gửi Bitfield

Thông điệp bitfield cung cấp thông tin về trạng thái sở hữu các mảnh dữ liệu của một peer. Bitfield thường được gửi ngay sau khi hoàn tất handshake, với dạng là một chuỗi bit.

`bitfield : $\langle len=0001+X \rangle \langle id=5 \rangle \langle bitfield \rangle$`

- bitfield: biểu diễn trạng thái các mảnh:



- Bit cao nhất trong byte đầu tiên tương ứng với index=0.
- Bit được đặt giá trị 1 nếu mảnh tương ứng đã được tải xong, giá trị 0 nếu chưa tải.
- Các bit dư thừa được đặt giá trị 0.

Có thể hiểu

- Bit 1: Peer sở hữu phần dữ liệu tương ứng.
- Bit 0: Peer không sở hữu phần dữ liệu.

Nếu peer nhận được thông điệp bitfield không hợp lệ (sai độ dài hoặc chứa bit dư thừa không bằng 0), kết nối sẽ bị ngắt.

Quy trình:

- Sau khi handshake, Peer A gửi Bitfield để thông báo trạng thái dữ liệu.
- Peer B phân tích bitfield để xác định các phần dữ liệu cần tải xuống.

2.2.5 Gửi Pieces

Sau khi nhận được yêu cầu, peer gửi lại các khối dữ liệu được yêu cầu dưới dạng các thông điệp piece. Mỗi khối dữ liệu được truyền là một phần của mảnh, giúp đảm bảo tính linh hoạt và hiệu quả trong việc quản lý truyền tải.

Các thông điệp chính liên quan đến quá trình gửi một file bao gồm:

- Request (<id=6>):

request : <len=0013> <id=6> <index> <begin> <length>

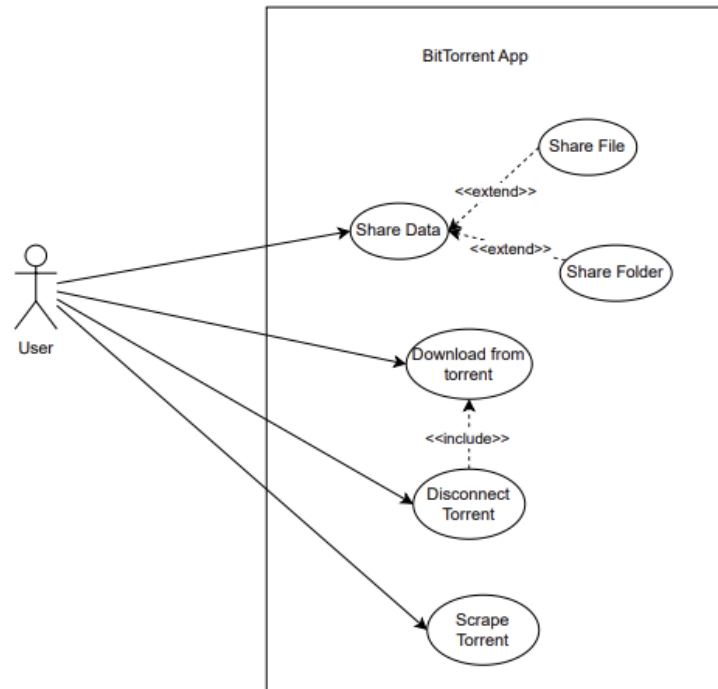
Peer gửi yêu cầu tải một phần dữ liệu, bao gồm:

- index: Vị trí của phần dữ liệu.
- begin: Offset trong phần dữ liệu.
- length: Kích thước khối dữ liệu cần tải.
- Piece (<id=7>): Peer phản hồi với khối dữ liệu được yêu cầu, gồm:
 - index, begin: Thông tin về vị trí và offset.
 - block: Dữ liệu thực tế.
- Cancel (<id=8>): Peer hủy yêu cầu tải xuống.

Quy trình:

- Peer A gửi Request đến Peer B.
- Peer B kiểm tra trạng thái Choke, nếu được phép, gửi Piece với khối dữ liệu tương ứng.
- Sau khi nhận đủ pieces, Peer A lắp ráp chúng thành file hoàn chỉnh.

3 Lược đồ Usecase



Hình 1: Usecase Diagram

Use Case ID	Tên Use Case	Mô tả
UC-1	Share data	Người dùng chia sẻ dữ liệu
UC-1.1	Share file	Người dùng chia sẻ file
UC-1.2	Share folder	Người dùng chia sẻ folder
UC-2	Download from torrent	Người dùng tải xuống từ một torrent file
UC-2.1	Disconnect Torrent	Người dùng ngừng tham gia vào một mạng torrent
UC-3	Scrape Torrent	Người dùng gửi yêu cầu lấy thông tin của torrent từ Tracker Server



3.1 Đặc tả usecase

Use case ID	UC-1
Use case name	Share data
Actors	Actor-1: Người dùng
Description	Người dùng chia sẻ dữ liệu
Trigger	Người dùng chọn chia sẻ dữ liệu
Preconditions	Người dùng đã đăng nhập vào hệ thống
Postconditons	Người dùng tham gia vào mạng torrent mà mình đã chia sẻ
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn chức năng chia sẻ ở transfer page.2. Hệ thống xử lý yêu cầu chia sẻ.3. Người dùng tham gia vào mạng torrent mà mình đã chia sẻ.
Alternative flow	<ol style="list-style-type: none">1a. Người dùng chia sẻ file1b. Người dùng chia sẻ folder

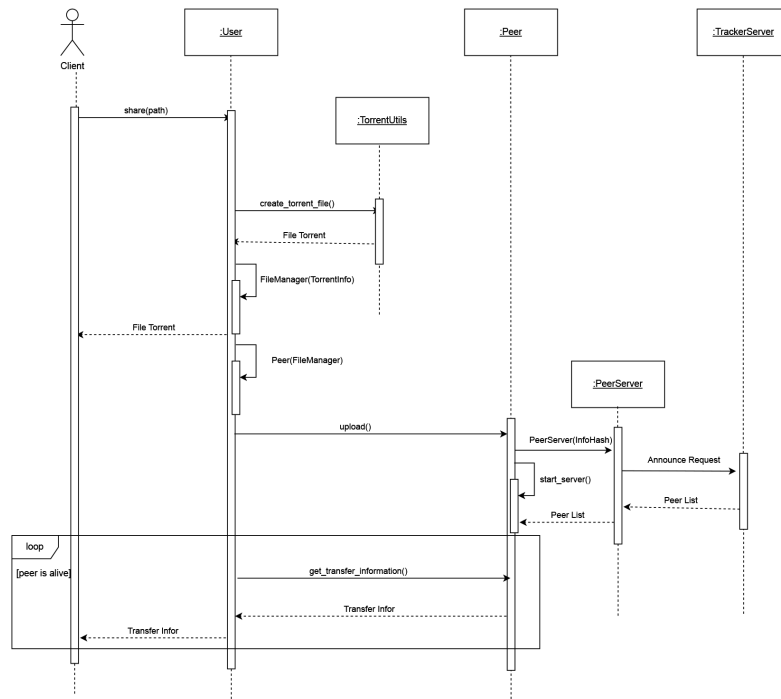
Use case ID	UC-2
Use case name	Download from torrent
Actors	Actor-1: Người dùng
Description	Người dùng tải xuống dữ liệu
Trigger	Người dùng chọn tham gia vào mạng torrent
Preconditions	Người dùng đã đăng nhập vào hệ thống
Postconditons	Người dùng tham gia vào mạng torrent
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn chức năng thêm torrent ở transfer page.2. Hệ thống xử lý yêu cầu tham gia vào mạng torrent.3. Người dùng tham gia vào mạng torrent, bắt đầu tải xuống và tải lên các piece.
Alternative flow	<ol style="list-style-type: none">2a. Hệ thống không tìm thấy torrent người dùng yêu cầu



Use case ID	UC-3
Use case name	Scrape data
Actors	Actor-1: Người dùng
Description	Người dùng yêu cầu xem thông tin của mạng torrent
Trigger	Người dùng chọn chức năng scrape torrent
Preconditions	Người dùng đã đăng nhập vào hệ thống
Postconditons	Hệ thống hiển thị thông tin của torrent
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn chức năng scrape torrent.2. Hệ thống xử lý yêu cầu .3.Hệ thống hiển thị thông tin của torrent.
Alternative flow	<ol style="list-style-type: none">2a. Hệ thống không tìm thấy torrent người dùng yêu cầu

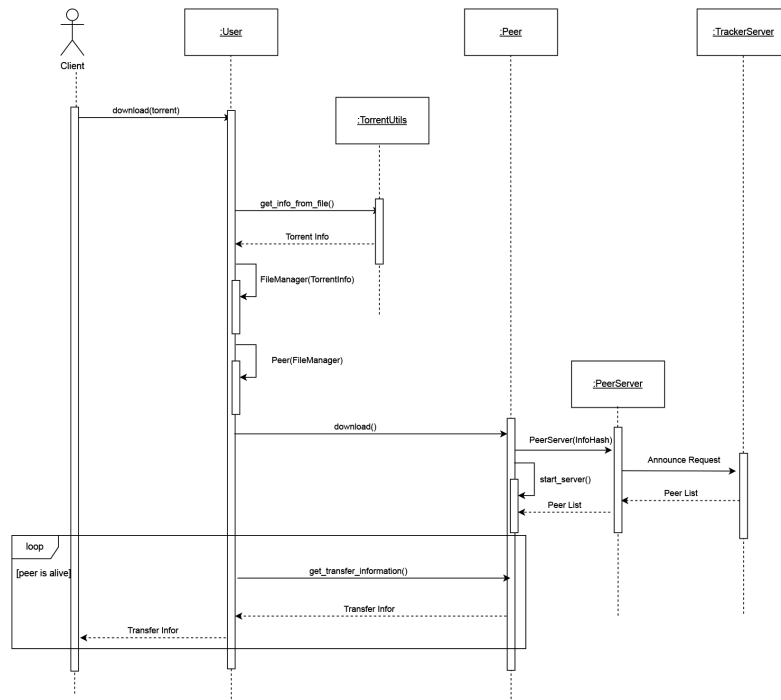
4 Lược đồ tuần tự

4.1 Share data



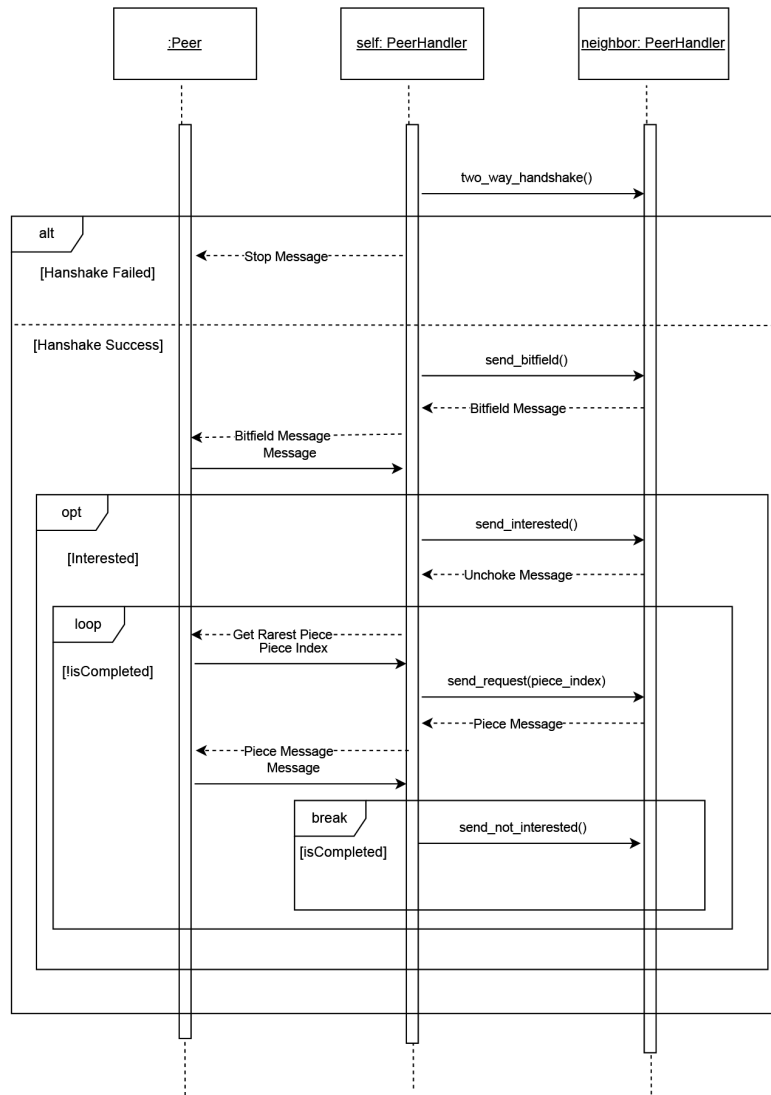
Hình 2: Sequence Diagram Share Data

4.2 Download from torrent



Hình 3: Sequence Diagram Download

4.3 Peer to Peer Protocol



Hình 4: Sequence Diagram Peer to Peer Protocol

5 Class Diagram



Hình 5: Class Diagram



6 Hướng dẫn sử dụng

6.1 Khởi động ứng dụng

6.1.1 Từ source code

Đầu tiên tải source code theo đường link Github được cung cấp bên dưới. Sau đó chạy lệnh trong terminal để tải các thư viện cần thiết

```
pip install -r requirements . txt
```

Để mở Tracker server, chạy lệnh



```
python TrackerServer.py
```

Để chạy client application, chạy lệnh

```
python app.py
```

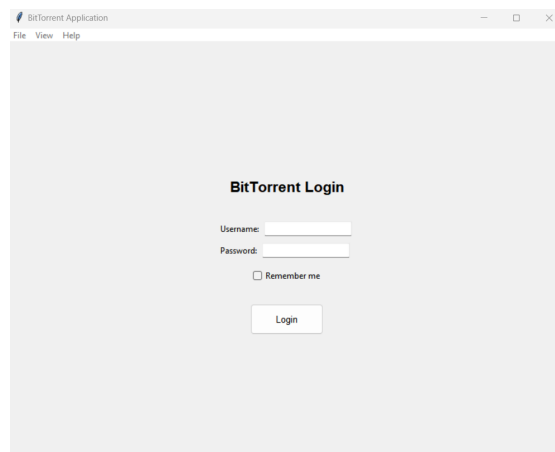
6.1.2 Từ Application file

Mở hai file Application trong folder dist là app và Tracker để chạy chương trình tương ứng

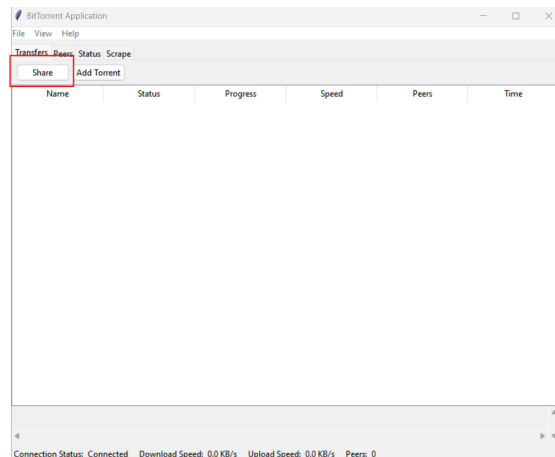
 app	4/21/2025 10:06 PM	Application	10,017 KB
 TrackerServer	4/21/2025 10:05 PM	Application	6,981 KB

6.2 Chia sẻ

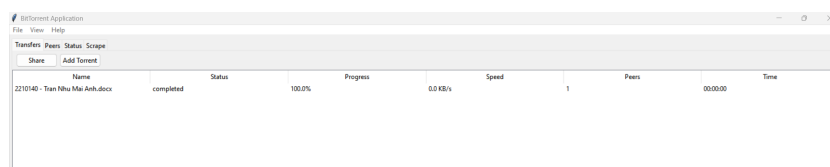
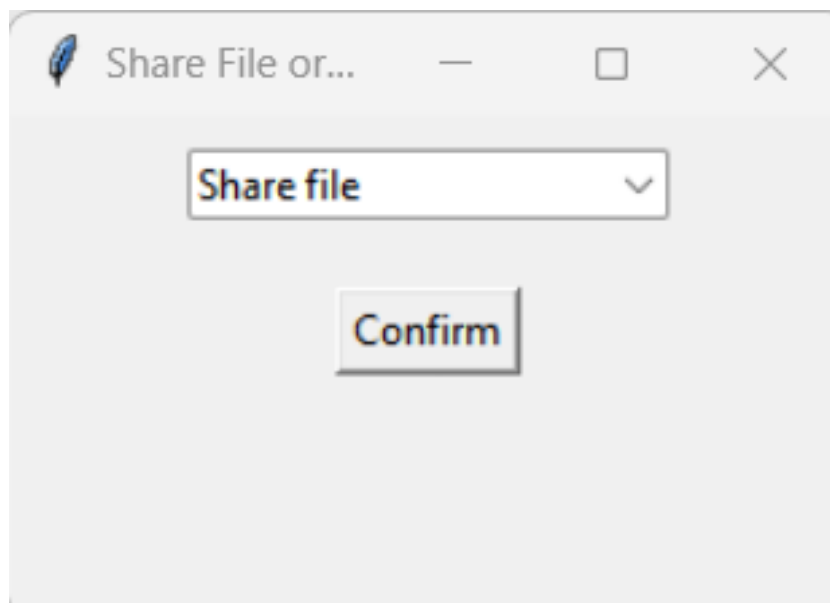
Khi vừa khởi động app bạn sẽ thấy giao diện đăng nhập như sau, hãy nhập username và password bất kỳ để vào trong giao diện chính.



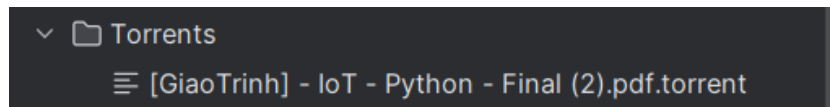
Nhấn vào share ở góc trái trên cùng để thực hiện chia sẻ.



Các bạn có thể chọn chia sẻ 1 file hoặc 1 folder.

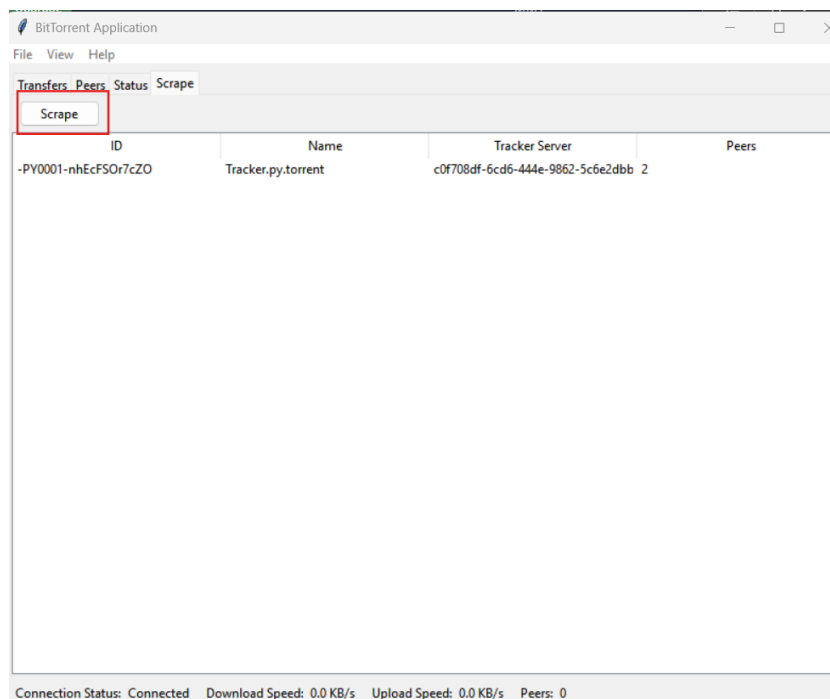


Bạn có thể thấy trong thư mục Torrent có một file torrent mới vừa được tạo ra như sau, chúng ta sẽ dùng file torrent này để tải xuống. Trên thực tế file torrent này sẽ phải được chia sẻ rộng rãi trên mạng Internet thông qua các website được sử dụng như các trang index để tìm kiếm nội dung



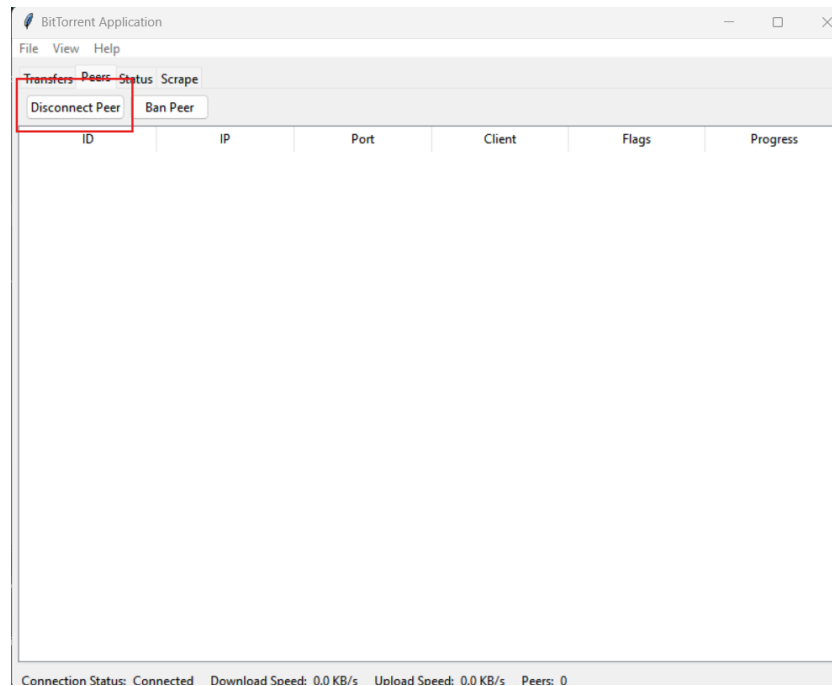
6.3 Xem thông tin

Bây giờ chúng ta sẽ thực hiện xem thông tin của file torrent thông qua chức năng scrape. Hãy tạo một process khác và thêm file torrent.



6.4 Ngắt kết nối

Sau khi đã tải xong bạn có thể ngừng tham gia vào mạng torrent này ở thu mục Peers chọn peer cần ngắt kết nối và chọn disconnect.



Source code để build sẽ được tìm ở github sau: <https://github.com/anhtran0427/CO3093>

7 Kiểm thử hệ thống

Do ứng dụng chưa hỗ trợ chạy trên hai máy khác nhau (Vì chưa cấu hình được tường lửa) nên chúng ta sẽ thực hiện bắt gói bằng chức năng loopback trong wireshake để bắt các gói tin localhost.

7.1 Share file

Tại phần kiểm thử này, ta sẽ thực hiện chia sẻ 1 video có dung lượng là 93.3MB giữa 2 process khác nhau

7.1.1 Tracker-CLient

[illegible]

Ta thấy tracker được gửi 4 lần announce request, 2 cho event start và 2 cho event complete

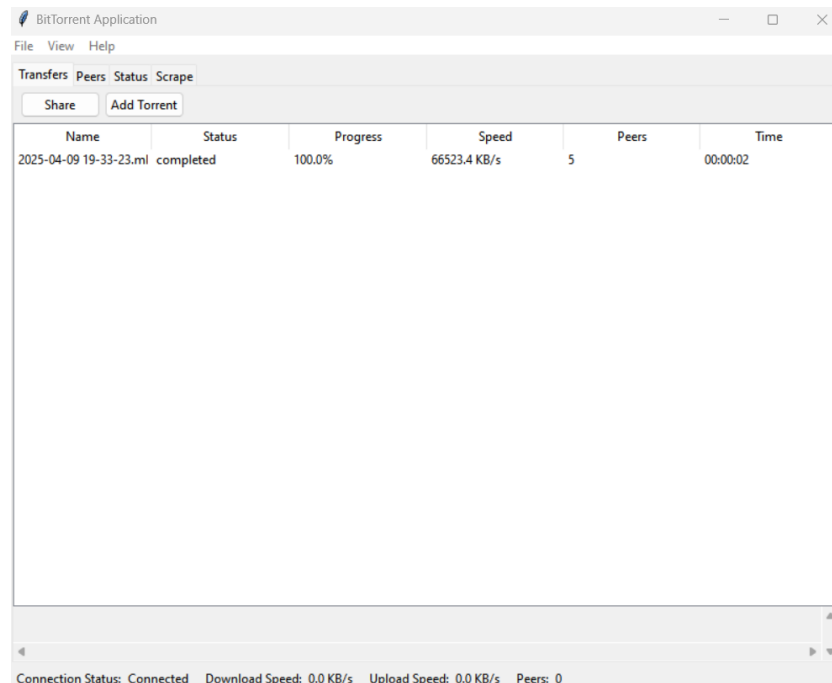
7.1.2 Client-Client

1227	36.187280	165.254.78.282	165.254.78.81	112	Handshake
1232	36.187590	165.254.78.282	165.254.78.81	112	Handshake
1234	36.188442	165.254.78.282	165.254.78.81	112	Handshake
1236	36.188651	165.254.78.282	165.254.78.81	112	Handshake
1238	36.188907	165.254.78.282	165.254.78.81	73	313field, len=818
1240	36.189091	165.254.78.282	165.254.78.81	73	313field, len=818
1242	36.189347	165.254.78.282	165.254.78.81	73	313field, len=818
1244	36.189592	165.254.78.282	165.254.78.81	73	313field, len=818
1246	36.189848	165.254.78.282	165.254.78.81	49	not interpreted
1248	36.190094	165.254.78.282	165.254.78.81	49	not interpreted
1250	36.190302	165.254.78.282	165.254.78.81	49	not interpreted
1252	36.190509	165.254.78.282	165.254.78.81	49	not interpreted
1254	36.191116	165.254.78.282	165.254.78.81	49	not interpreted
1256	36.191287	165.254.78.282	165.254.78.81	49	not interpreted
1258	36.191507	165.254.78.282	165.254.78.81	61	Request: Piece (idx=0, begin=0, len=80000)

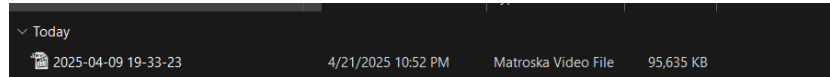
Dưới đây là các protocol bittorrent đã được giới thiệu ở phần trên. Các peer đồng loạt gửi Handshake cho nhau nếu thành công sẽ tiếp tục, thất bại sẽ hủy kết nối. Sau đó gửi Message BITFIELD để xác nhận các pieces. Sau khi xác nhận sẽ gửi thông báo Interested hoặc Not Interested. Nếu có peer Interested thì phía peer kia sẽ gửi thông báo Unchoke để bắt đầu gửi nhận Piece.

5838	31.405415	169.254.78.202	169.254.78... Bi...	49 Not Interested
------	-----------	----------------	---------------------	-------------------

Sau khi kết thúc sẽ thông báo, cho Not Interested cho peer còn lại và gửi Complete cho Tracker server. Thời gian để download video 93.3MB sẽ có tổng thời gian là 1,018207 Tuy nhiên khi ta tăng số peer lên 5, thời gian share file sẽ là 1,564677.



Đồng thời, file đã được lưu vào folder được chọn (mô phỏng việc upload giữa các máy)



7.2 Scrape server

