

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



Mạng Máy Tính

NETWORK APPLICATION P2P FILE SHARING

GVHD: Lê Bảo Khánh
SV: Nguyễn Tuấn Anh - 2252038
Dinh Hải Nam - 2252498
Ngô Ngọc Triệu Mẫn - 2212009
Trần Gia Quý - 2252694
Đương Thanh Tú - 2252879

TP. HỒ CHÍ MINH, THÁNG 11/2024

Mục lục

1	Tổng quan Bài tập lớn	2
1.1	Thông tin Bài tập lớn	2
1.2	Phạm vi	2
1.3	Phân công công việc	2
2	Cơ sở lý thuyết và Đề xuất giải pháp	2
2.1	Mô hình Centralized Peer-to-Peer(P2P)	2
2.1.1	Tổng quát	2
2.1.2	Giao thức truyền tải file	3
2.1.2.a	Transmission Control Protocol (TCP)	3
2.1.2.b	HyperText Transfer Protocol (HTTP)	4
2.2	Phát triển ứng dụng	5
2.2.1	Công nghệ sử dụng	5
2.3	Tóm tắt công nghệ sử dụng	6
3	Yêu cầu phần mềm	6
3.1	Dối tượng sử dụng	6
3.2	Yêu cầu về phần mềm	7
3.2.1	Yêu cầu về chức năng	7
3.2.2	Yêu cầu phi chức năng	7
3.3	Yêu cầu quản lý dữ liệu	8
3.3.1	Lưu trữ dữ liệu	8
4	Thiết kế hệ thống	9
4.1	Flow hoạt động của hệ thống	9
4.2	Sơ đồ Usecase (Usecase Diagram)	10
4.2.1	Sơ đồ Usecase cho toàn bộ hệ thống	10
4.2.2	Sơ đồ Usecase cho module <i>Thiết lập kết nối và tải file</i>	11
4.3	Sơ đồ Sequence (Sequence Diagram)	12
4.4	Sơ đồ khối (Class Diagram)	13
5	Kiểm thử	13
6	Hướng dẫn sử dụng cho người dùng	14
6.1	Hướng dẫn tải và cài đặt	14
6.2	Cách sử dụng hệ thống	17



1 Tổng quan Bài tập lớn

Trong bài tập lớn lần này, nhóm sẽ thực hiện một ứng dụng gần tương tự với BitTorrent, bao gồm các kết nối Peer-to-Peer (P2P) và tracker để hỗ trợ chia sẻ file giữa các người dùng. Ứng dụng sẽ bao gồm các thành phần chính như Tracker, đóng vai trò là máy chủ theo dõi và duy trì danh sách các peer (người dùng) đang tham gia chia sẻ file. Tracker sẽ cung cấp thông tin về các peer khác khi một peer mới tham gia để tải hoặc chia sẻ file. Mạng lưới P2P sẽ cho phép các peer kết nối trực tiếp với nhau, vừa tải xuống, vừa chia sẻ file mà không cần qua máy chủ trung gian, tối ưu hóa băng thông và hiệu suất của hệ thống. Thông tin về file chia sẻ sẽ được lưu trong một file torrent, bao gồm metadata về tên file, kích thước và thông tin về các mảnh nhỏ (piece) của file, cũng như địa chỉ của tracker. File được chia thành nhiều mảnh nhỏ để tăng tốc độ chia sẻ và đảm bảo phục hồi dữ liệu hiệu quả hơn khi xảy ra sự cố. Các mô tả của file sẽ được giải thích chi tiết ở những phần dưới đây.

1.1 Thông tin Bài tập lớn

- Tiêu đề Bài tập lớn: Ứng dụng mạng Peer-to-Peer trong chia sẻ tệp tin.
- Thời gian thực hiện: Từ ngày 11/10/2024 đến ngày 30/10/2024
- Số lượng thành viên: 5 người.
- Mục tiêu: Phát triển ứng dụng chia sẻ tài liệu peer-to-peer (P2P) bằng mô hình mạng tập trung và giao thức TCP/IP.

1.2 Phạm vi

- Sản phẩm: Phát triển ứng dụng chia sẻ tài liệu peer-to-peer (P2P) dựa trên mô hình mạng tập trung (Centralized Peer-to-Peer) và sử dụng giao thức TCP/IP.
- Mô hình mạng: Sử dụng mô hình P2P tập trung, trong đó máy chủ theo dõi (tracker) sẽ duy trì danh sách các peer đang tham gia vào mạng. Tracker đóng vai trò cung cấp thông tin về các peer khác khi một peer mới tham gia hoặc muốn chia sẻ/tải xuống tệp.
- Tài liệu lưu trữ: Hệ thống sử dụng MongoDB làm cơ sở dữ liệu để quản lý thông tin về tệp và người dùng. Database được lưu trữ trên một server tại vùng 1, giúp đảm bảo dữ liệu luôn hoạt động ổn định và sẵn sàng.

1.3 Phân công công việc

2 Cơ sở lý thuyết và Đề xuất giải pháp

2.1 Mô hình Centralized Peer-to-Peer(P2P)

2.1.1 Tổng quát

Mạng peer to peer (P2P) là một kiến trúc ứng dụng phân tán nhằm phân vùng nhiệm vụ hoặc khối lượng công việc giữa các peer. Trong khi ở mô hình client-server, các máy khách (client) phải gửi yêu cầu cho máy chủ (server) và server đáp ứng yêu cầu, thì các máy khách trong mô hình P2P (peer) là những thiết bị tham gia trong ứng dụng có đặc quyền như nhau. Chúng tạo



Họ và Tên	Student ID	Tasks
Nguyễn Tuấn Anh	2252038	Viết báo cáo phần 2.2, 3.3, 4.1, 4.3, hiện thực phần code cho torrent-peer
Dinh Hải Nam	2252498	Soạn báo cáo phần 2, code backend cho Tracker
Ngô Ngọc Triệu Mẫn	2212009	Soạn báo cáo phần 2.3, 3.2.1 và phần 6, Quay và chỉnh sửa video demo phần 5, Code frontend cho Tracker Hub, cấu hình Database
Trần Gia Quý	2252694	Viết báo cáo phần 1 và 3.1, 3.2.2, Code frontend cho Tracker Hub, hiện thực realtime cho Tracker Hub.
Dương Thanh Tú	2252879	Viết báo cáo phần 4.2, 4.4, Code backend cho Tracker

Bảng 1: Bảng phân chia công việc

thành một mạng lưới các node ngang hàng, có thể đóng vai trò là server hoặc client tùy thuộc vào tình huống.

Mạng P2P là sự đồng bộ giữa server và client, mạng lưới người dùng phân tán được tạo ra thông qua ứng dụng phần mềm. Mạng P2P tạo ra một mạng ảo, đóng vai trò trung gian kết nối các peer cùng một hệ thống, thông qua trung gian này các peer có thể chia sẻ dữ liệu. Khi người dùng muốn tìm và tải xuống một tệp tin, họ có thể gửi yêu cầu tìm kiếm đến các thiết bị khác trên mạng. Sau khi tải xuống, một tệp mạng ảo được nhận từ các máy tính khác nhau, tạo thành một quy trình truyền tải linh hoạt và phân phối dữ liệu một cách hiệu quả. Điều này giúp P2P hoạt động như một đường hai chiều, nơi công việc truyền tải tệp tin được phân phối và thực hiện một cách linh hoạt và tiện lợi.

P2P có các ưu điểm như tài nguyên và công việc được phân phối đều giữa các nút giúp tăng hiệu quả hệ thống hay tài nguyên được phân tán nên cả hệ thống vẫn có thể hoạt động nếu một số nút bị ngừng hoạt động. Tuy nhiên, vẫn còn đó những hạn chế như khó khăn trong quản lý độ trễ hoặc an ninh trong mạng P2P.

2.1.2 Giao thức truyền tải file

2.1.2.a Transmission Control Protocol (TCP)

a/ Khái niệm

TCP là giao thức định hướng kết nối (connection-oriented protocol) giúp trao đổi tin nhắn giữa các thiết bị khác nhau qua mạng. Giao thức vận chuyển cơ bản này có nhiều ứng dụng, bao gồm WWW, liên kết giữa máy chủ web và trang web, email, FTP (File Transfer Protocol), vỏ bảo mật (Secure Shell), phương tiện truyền phát trực tuyến và các giao thức có dạng peer-to-peer. TCP cùng với IP (Internet Protocol), giao thức giúp truyền tải các gói dữ liệu giữa các máy tính, tạo nên TCP/IP.

b/ Cách thức hoạt động

Bước 1: Thiết lập kết nối

Quá trình này được thực hiện thông qua quá trình bắt tay 3 bước (3-way handshake process):

- Client gửi một gói tin SYN đến server — yêu cầu kết nối từ cổng nguồn (source port) của client đến cổng đích (destination port) của server.
- Server phản hồi bằng một gói tin SYN/ACK, xác nhận yêu cầu kết nối.
- Client nhận gói tin SYN/ACK và trả lời bằng một gói tin ACK của chính nó.



Bước 2: Truyền tải dữ liệu

Sau khi kết nối được thiết lập thành công, trong mô hình TCP, dữ liệu được chia thành những gói nhỏ gọi là segment. Sau đó, các segment này được ráp lại thành dữ liệu ban đầu ở đầu bên kia, đảm bảo rằng mỗi thông điệp đến được vị trí mục tiêu một cách nguyên vẹn. Sau khi một tệp dữ liệu cụ thể được chia thành các segment, chúng có thể di chuyển qua nhiều tuyến khác nhau nếu một tuyến bị tắc nghẽn, nhưng đích đến vẫn giữ nguyên.

c/ Ưu và nhược điểm của TCP

Ưu điểm

- Cung cấp công cụ kiểm soát luồng.
- Đảm bảo rằng dữ liệu đến đúng địa điểm ở đúng thứ tự mà nó được gửi.
- Là một giao thức hỗ trợ cho nhiều dạng dữ liệu, tài liệu hóa tốt và được triển khai rộng rãi, được duy trì bởi các tổ chức uy tín như IETF (Internet Engineering Task Force).

Nhược điểm

- TCP được thiết kế cho Mạng điện rộng (WAN), do đó kích thước của nó có thể trở thành vấn đề cho các mạng nhỏ với tài nguyên hạn chế.
- Phải chạy qua nhiều lớp, dẫn đến làm chậm tốc độ của mạng.

d/ Các lớp của TCP/IP

TCP/IP được chia thành 4 tầng, mỗi tầng sử dụng các giao thức ở tầng dưới để đạt được mục đích của mình:

- **Lớp Network Access:** Xác định chi tiết về cách dữ liệu được truyền qua mạng bởi các thiết bị phần cứng giao tiếp trực tiếp với môi trường mạng, chẳng hạn như cáp đồng trục, cáp quang hoặc dây đồng xoắn đôi.
- **Lớp Internet:** Đóng gói dữ liệu thành các gói IP (Internet Protocol), chứa địa chỉ nguồn và địa chỉ đích (địa chỉ logic hoặc địa chỉ IP) được sử dụng để định tuyến các gói tin giữa các máy chủ và qua các mạng.
- **Lớp Transport:** Cho phép các thiết bị trên các máy chủ nguồn và đích trao đổi dữ liệu, đồng thời xác định mức độ dịch vụ và trạng thái kết nối được sử dụng cho việc truyền dữ liệu. Giao thức chính ở đây là TCP.
- **Application Layer:** Cung cấp các chức năng cho phép người dùng trao đổi dữ liệu ứng dụng qua mạng.

2.1.2.b HyperText Transfer Protocol (HTTP)

a/ Khái niệm

HTTP là giao thức tiêu chuẩn cho Internet, được sử dụng để trao đổi thông tin giữa máy chủ Web (Web server) và máy khách (client), là ứng dụng của giao thức TCP/IP. HTTP có các đặc tính giúp giao thức này tuy đơn giản nhưng lại hoạt động khá mạnh mẽ như tính đơn giản, dễ dàng mở rộng và đặc biệt là "Stateless, nhưng không sessionless".

b/ Cách thức hoạt động

HTTP được thiết kế dựa trên giao thức Client/Web server. Máy tính của người dùng hoạt động như các máy khách (client). Sau khi người dùng thực hiện một hành động, các máy khách sẽ gửi yêu cầu tới máy chủ và chờ phản hồi từ các máy chủ này.



c/ HTTP Request và Response

Việc giao tiếp giữa máy khách và máy chủ được thực hiện dựa trên cặp yêu cầu-phản hồi (request-response). Client gửi các yêu cầu, và Server phản hồi lại các yêu cầu này. HTTP Request và Response sẽ có cấu trúc gần giống nhau bao gồm Start Line, Header và Message Body.

HTTP Request

Một Request bao gồm 2 phần chính Start Line (hay ở đây là Request line) và Header.

Request line bao gồm 3 phần: Method, URI, và phiên bản HTTP. HTTP request có một số Method như HEAD, PUT, DELETE, OPTIONS, CONNECT, nhưng phổ biến nhất vẫn là GET và POST. Với GET, câu truy vấn sẽ được dính kèm vào đường dẫn của HTTP request. Trong khi đó với POST, câu truy vấn sẽ được gửi trong phần message body của HTTP request. URI là địa chỉ định danh của tài nguyên, và phiên bản HTTP chỉ định phiên bản HTTP đang được sử dụng.

Header cho phép máy khách gửi thêm thông tin về thông điệp yêu cầu HTTP và về chính máy khách. Một số trường Header phổ biến bao gồm: Accept; Accept-Encoding; Connection; Cookie; User-Agent.

HTTP Response

Cấu trúc HTTP response gần giống với HTTP request, chỉ khác Start Line là Status line. Và giống như Request line, Status line cũng có ba phần:

- HTTP-version: Phiên bản HTTP cao nhất mà server hỗ trợ.
- Status-Code: Mã kết quả trả về.
- Reason-Phrase: Mô tả về Status-Code.

Một số loại Status-Code thông dụng mà server trả về cho client:

- 1xx Information Message: Các status code này chỉ có tính chất tạm thời, client có thể không quan tâm.
- 2xx Successful: Khi đã xử lý thành công request của client, server trả về status dạng này (ví dụ: 200 OK — request thành công; 202 Accepted — request đã được nhận, nhưng không có kết quả nào trả về, thông báo cho client tiếp tục chờ đợi,...).
- 3xx Redirection: Server thông báo cho client phải thực hiện thêm thao tác để hoàn tất request (ví dụ: 301 Moved Permanently — tài nguyên đã được chuyển hoàn toàn tới địa chỉ Location trong HTTP response,...).
- 4xx Client Error: Lỗi của client (ví dụ: 404 Not Found — không tìm thấy tài nguyên,...).
- 5xx Server Error: Lỗi của server (ví dụ: 500 Internal Server Error — có lỗi trong quá trình xử lý của server,...).

2.2 Phát triển ứng dụng

2.2.1 Công nghệ sử dụng

- Quản lý Source Code và Quản lý phiên bản: Github
- Client (Peers)
 - Ngôn ngữ: Javascript
 - Môi trường: Node.js



- Tracker (Server)
 - Frontend:
 - * Pug: Template Engine dùng để render ra các file HTML.
 - * TailwindCSS: Một thư viện dùng để CSS các phần tử HTML trực tiếp bằng class thay vì ghi file .css.
 - * Javascript: Dùng để xử lý các sự kiện khi người dùng nhấn vào các phần tử HTML.
 - * Socket.io: Dùng để nhận thông báo truyền dữ liệu realtime từ server và xử lý chúng để hiển thị ra giao diện.
 - Backend:
 - * Express.js: Dùng để tạo server và giao tiếp thông qua phương thức HTTP và thực hiện routing các đường dẫn như /announce để các peer có thể gửi yêu cầu tới và thêm routing /home cho Tracker Hub
 - * Socket.io: Dùng để đạt được mục đích realtime và trả về dữ liệu cho client theo thời gian thực để hiển thị file vừa được chia sẻ lên tracker.
- Database
 - ORM: Mongoose
 - DBMS: Mongo DB
- Torrent File Cloud Storage: Cloudinary dùng để lưu trữ các file .torrent trên Cloud và giúp người dùng có thể tải dễ dàng.

2.3 Tóm tắt công nghệ sử dụng

Bảng 2: Tóm tắt công nghệ sử dụng

Objective	Solution
Connection Model	Centralized P2P
Data Transfer Protocol	HTTP, TCP
Version Control & Source Management	Git, GitHub
Front-end	Pug, TailwindCSS
Back-end	Node.js, Express.js, JavaScript, Mongoose
Database	MongoDB
File Storage	Cloudinary
Real-time Communication	Socket.io

3 Yêu cầu phần mềm

3.1 Đối tượng sử dụng

- Những người dùng cá nhân có nhu cầu chia sẻ tệp tin nhanh chóng.
- Các nhóm làm việc, dự án có nhu cầu chia sẻ tệp tin cùng nhau.



- Các cộng đồng có nhu cầu chia sẻ tệp tin với nhau.
- Các tổ chức nghiên cứu học thuật có nhu cầu chia sẻ tệp tin.

3.2 Yêu cầu về phần mềm

3.2.1 Yêu cầu về chức năng

Yêu cầu chức năng của Người dùng (Node):

- Đăng nhập/Đăng ký với hệ thống: Người dùng có thể đăng nhập hoặc đăng ký tài khoản để kết nối với máy chủ tracker.
- Cập nhật thông tin tệp tin: Sau khi kết nối người dùng thông báo với máy chủ tracker về các tệp tin sẵn sàng chia sẻ.
- Tải tệp từ các peer khác: Khi người dùng cần tải xuống một tệp không có sẵn trong kho lưu trữ, người dùng sẽ yêu cầu máy chủ tracker và nhận danh sách các peer khác để tải về.
- Kết nối và tải từ nhiều nguồn cùng lúc: Người dùng có thể kết nối với nhiều peer để tải các phần của tệp từ nhiều nguồn đồng thời.
- Chia sẻ tệp tin: Sau khi tải xuống người dùng có thể chia sẻ các tệp với các peer khác trong hệ thống.
- Xem tiến độ và thông kê: Người dùng có thể xem tiến độ tải lên/tải xuống cũng như các thông tin về tốc độ và danh sách các peer đang kết nối.

Yêu cầu chức năng của máy chủ theo dõi (Tracker):

- Quản lý danh sách người dùng và tệp tin: Máy chủ lưu trữ thông tin về những người dùng đang kết nối và những tệp tin mà họ đang chia sẻ.
- Cung cấp danh sách peer: Khi một người dùng yêu cầu tải tệp, máy chủ cung cấp danh sách các peer đang có tệp đó để giúp người dùng tải về.
- Thực hiện giao thức tracker: Máy chủ theo dõi các peer, tệp tin và quản lý các kết nối giữa các peer với nhau thông qua giao thức tracker.
- Xử lý yêu cầu từ người dùng: Máy chủ nhận và phản hồi lại các yêu cầu từ người dùng về các thông tin tệp và danh sách các peer thông qua giao thức HTTP.

3.2.2 Yêu cầu phi chức năng

- Hệ thống phải đảm bảo đạt hiệu suất khi upload và download cao, giảm thiểu độ trễ và quản lý băng thông hiệu quả.
- Hệ thống phải đảm bảo bảo mật thông tin của từng Node.
- Hệ thống phải đảm bảo tính nhất quán của dữ liệu, tính toàn vẹn khi tải tệp tin về máy.
- Hệ thống phải đảm bảo việc tiếp tục tải tệp ngay cả khi một số peers bị ngắt kết nối hoặc không còn hoạt động.
- Hệ thống phải cung cấp giao diện thân thiện, dễ sử dụng cho người dùng, cung cấp trình tải tệp tin cụ thể.



3.3 Yêu cầu quản lý dữ liệu

Trong Bài tập lớn lần này, như đã liệt kê công nghệ sử dụng ở trên, nhóm đã chọn mongoDB làm Database cho nhóm lần này. Để có thể đảm bảo Tracker luôn hoạt động ổn định và liên tục, nhóm đã host một Database ở server đặt tại nhà để có thể đảm bảo Database luôn được hoạt động liên tục (trừ trường hợp mất điện).

3.3.1 Lưu trữ dữ liệu

Trong Database, nhóm chia thành tổng cộng 3 Model:

1. downloadFile: Dùng để lưu các info để hiển thị ra giao diện trên trang tải file .torrent

- fileName: string
- infoHash: string
- seeders: array
- numOfPieces: number
- size: string
- link: string
- createdAt: Date
- updatedAt: Date

2. file: Dùng để lưu thông tin các file khi seeders thực hiện chia sẻ file.

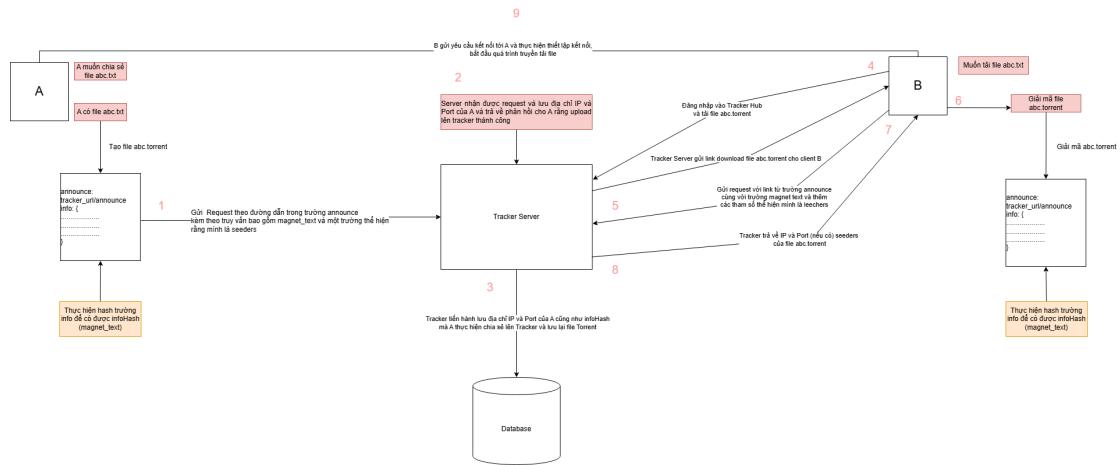
- fileName: string
- infoHash: string
- seeders: array
- numOfPieces: number
- size: string

3. user: Dùng để lưu thông tin người dùng khi người dùng tiến hành đăng ký tài khoản trên trang tải file .torrent

- fullName: string
- email: string
- password: string (đã được hash để đảm bảo bảo mật)
- token: string (dùng để kiểm tra danh tính người dùng)
- createdAt: Date
- updatedAt: Date

4 Thiết kế hệ thống

4.1 Flow hoạt động của hệ thống



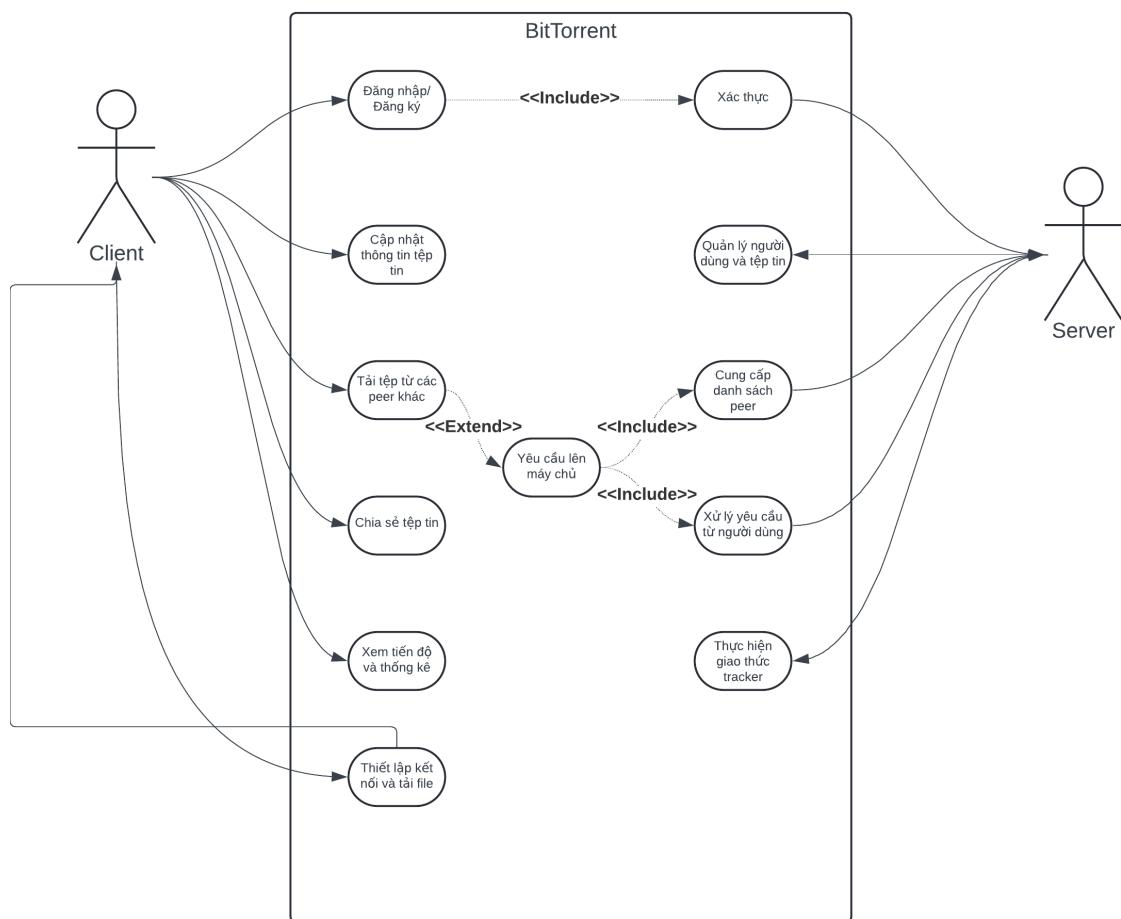
Hình 1: Flow hoạt động

Flow hoạt động tổng quát gồm 9 bước:

1. Gửi Request theo đường dẫn trong trường announce kèm theo truy vấn bao gồm magnet_text và một trường thẻ hiện rằng mình là seeders.
2. Server nhận được request và lưu địa chỉ IP và Port của A và trả về phản hồi cho A rằng upload lên tracker thành công.
3. Tracker tiến hành lưu địa chỉ IP và Port của A cũng như infoHash mà A thực hiện chia sẻ lên Tracker và lưu lại file Torrent.
4. Đăng nhập hoặc Đăng ký vào Tracker Hub và tải file abc.torrent.
5. Tracker Server gửi link download file abc.torrent cho client B.
6. Giải mã file abc.torrent.
7. Gửi request với link từ trường announce cùng với trường magnet text và thêm các tham số thẻ hiện mình là leechers.
8. Tracker trả về IP và Port (nếu có) seeders của file abc.torrent.
9. B gửi yêu cầu kết nối tới A và thực hiện thiết lập kết nối, bắt đầu quá trình truyền tải file.

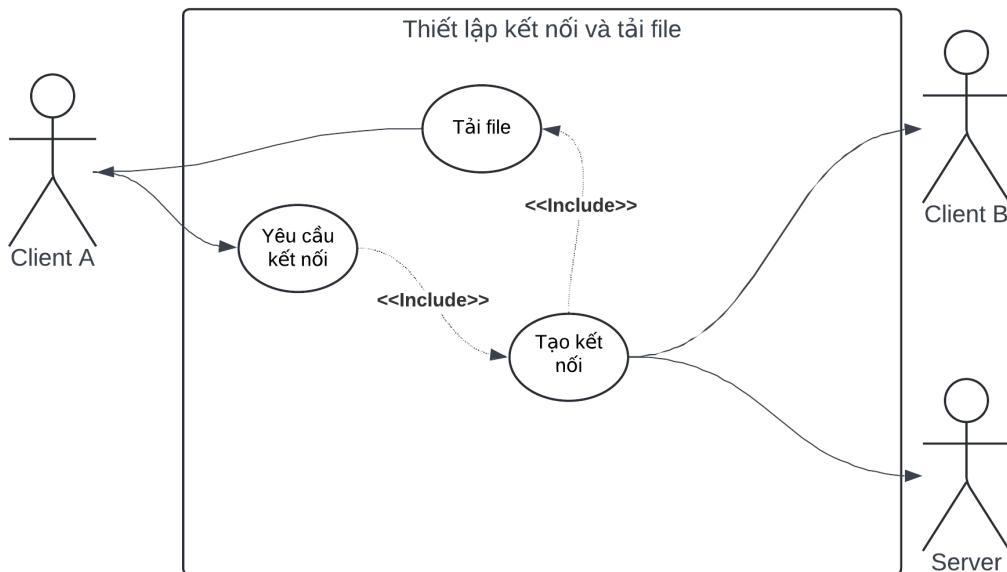
4.2 Sơ đồ Usecase (Usecase Diagram)

4.2.1 Sơ đồ Usecase cho toàn bộ hệ thống



Hình 2: Sơ đồ Usecase cho toàn bộ hệ thống

4.2.2 Sơ đồ Usecase cho module Thiết lập kết nối và tải file

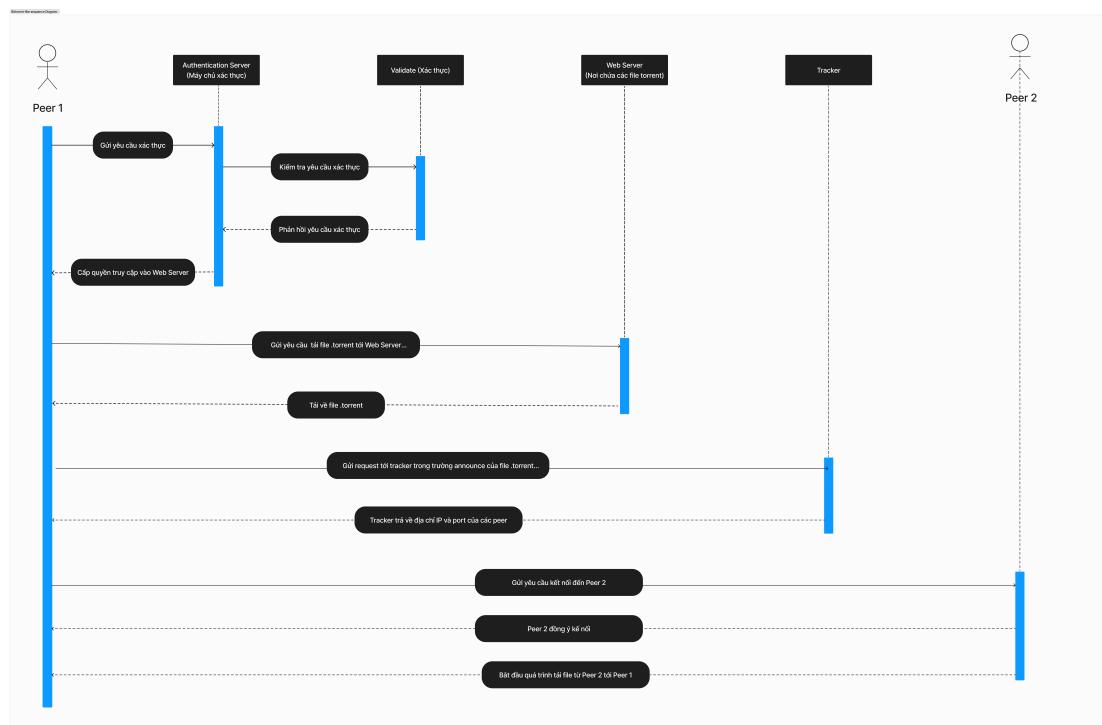


Hình 3: Sơ đồ Usecase cho module thiết lập kết nối và tải file

Use-case Name	Thiết lập kết nối và tải file
Use-case ID	TLKN&TF
Description	Tạo một cổng kết nối giữa 2 Client để cho việc tải file
Trigger	Khi client A yêu cầu tải file hoặc ngược lại
Pre-conditions	Cả hai client đều phải đang kết nối với Internet và file mà client A hoặc B yêu cầu phải tồn tại trên server
Post-conditions	Client A hoặc B tải được đúng file đã yêu cầu trước đó
Normal Flow	<ol style="list-style-type: none"> Client A (hoặc B) gửi yêu cầu tải file lên server. Server tìm được file torrent dựa trên magnet text mà client A(hoặc B) gửi lên server. Server tìm được magnet-text được lưu trong database và trả về địa chỉ IP của seeder (Client A hoặc B) cho Client nào muốn tải file. Bên mong muốn tải file (A hoặc B) gửi yêu cầu thiết lập kết nối TCP cho bên còn lại. Bên seeder phản hồi lại cho bên muốn tải file là việc tải file đã sẵn sàng và tải file theo yêu cầu của bên mong muốn tải file.
Exception Flow	<ol style="list-style-type: none"> Sẽ hiển thị ra lỗi cho client A. Quay lại màn hình yêu cầu tải file cho client A.
Alternative Flow	<ol style="list-style-type: none"> Ở bước 5, nếu một bên không chấp nhận tải file thì sẽ huỷ toàn bộ quá trình và quay lại màn hình yêu cầu tải file.

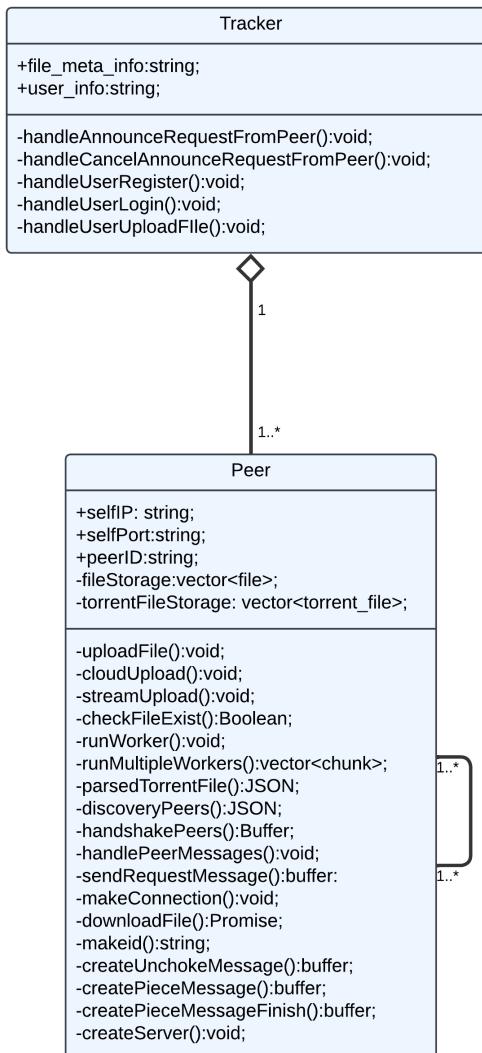
Bảng 3: Usecase Table cho module Thiết lập kết nối và tải file

4.3 Sơ đồ Sequence (Sequence Diagram)



Hình 4: Sequence Diagram

4.4 Sơ đồ khôi (Class Diagram)



Hình 5: Class diagram

5 Kiểm thử

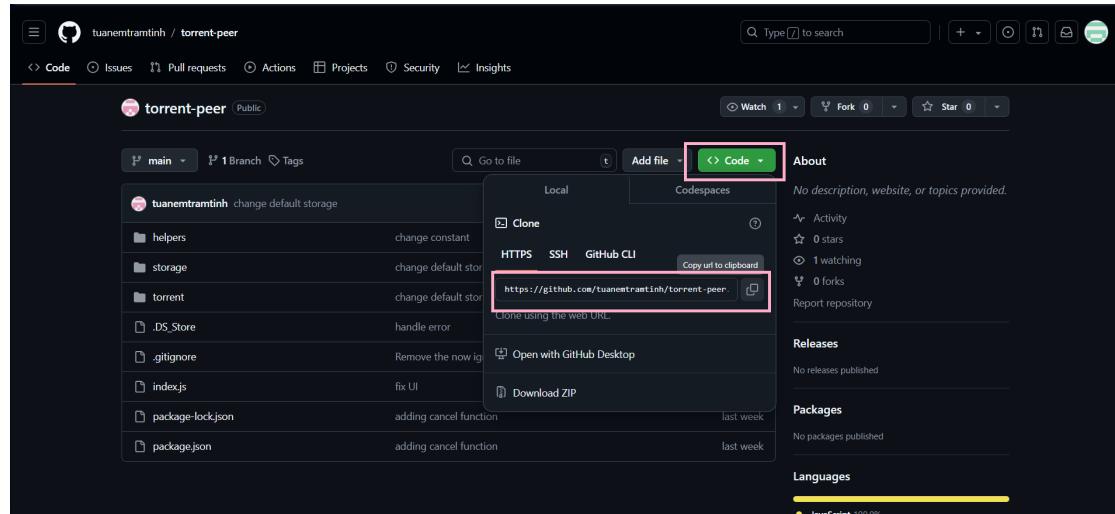
Bấm vào link này để chuyển hướng tới trang video demo sản phẩm: [Click vào đây](#)



6 Hướng dẫn sử dụng cho người dùng

6.1 Hướng dẫn tải và cài đặt

Bước 1: Đầu tiên vào <https://github.com/tuanemtramtinh/torrent-peer.git> sau đó bạn làm theo hình ảnh bên dưới:



Hình 6: Copy đường link để thực hiện bước tiếp theo

Bước 2: Bật terminal của máy bạn lên sau đó nhập theo hình ảnh bên dưới:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ACER\Desktop\btl mạng máy tính\test> git clone https://github.com/tuanemtramtinh/torrent-peer.git
Cloning into 'torrent-peer'...
remote: Enumerating objects: 714, done.
remote: Counting objects: 100% (158/158), done.
remote: Compressing objects: 100% (120/120), done.
remote: Total 714 (delta 89), reused 105 (delta 37), pack-reused 556 (from 1)
Receiving objects: 100% (714/714), 35.42 MiB | 6.04 MiB/s, done.
Resolving deltas: 100% (181/181), done.
PS C:\Users\ACER\Desktop\btl mạng máy tính\test> code .
```

Hình 7: Các dòng lệnh cần nhập

Bước 3: Vào IDE hoặc Code Editor, mở một folder bất kỳ mà bạn muốn chứa app. Sau đó, bật terminal lên và nhập theo hình ảnh bên dưới:



The screenshot shows a Visual Studio Code interface with a dark theme. The terminal tab is active, displaying the following command-line session:

```
PS C:\Users\ACER\Desktop\btl mạng máy tính\test>
PS C:\Users\ACER\Desktop\btl mạng máy tính\test> cd .\torrent-peer\
PS C:\Users\ACER\Desktop\btl mạng máy tính\test\torrent-peer> npm install
removed 2 packages, and audited 46 packages in 1s
12 packages are looking for funding
  run 'npm fund' for details
1 high severity vulnerability
Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
PS C:\Users\ACER\Desktop\btl mạng máy tính\test\torrent-peer> node index.js
-----Bạn muốn chạy server trên localhost hay giữa các LAN?-----
1.localhost
2. LAN
3. Thoát ra
-----Nhập vào lựa chọn của bạn: 
```

The terminal also shows a message from a character named Rikka asking for a choice.

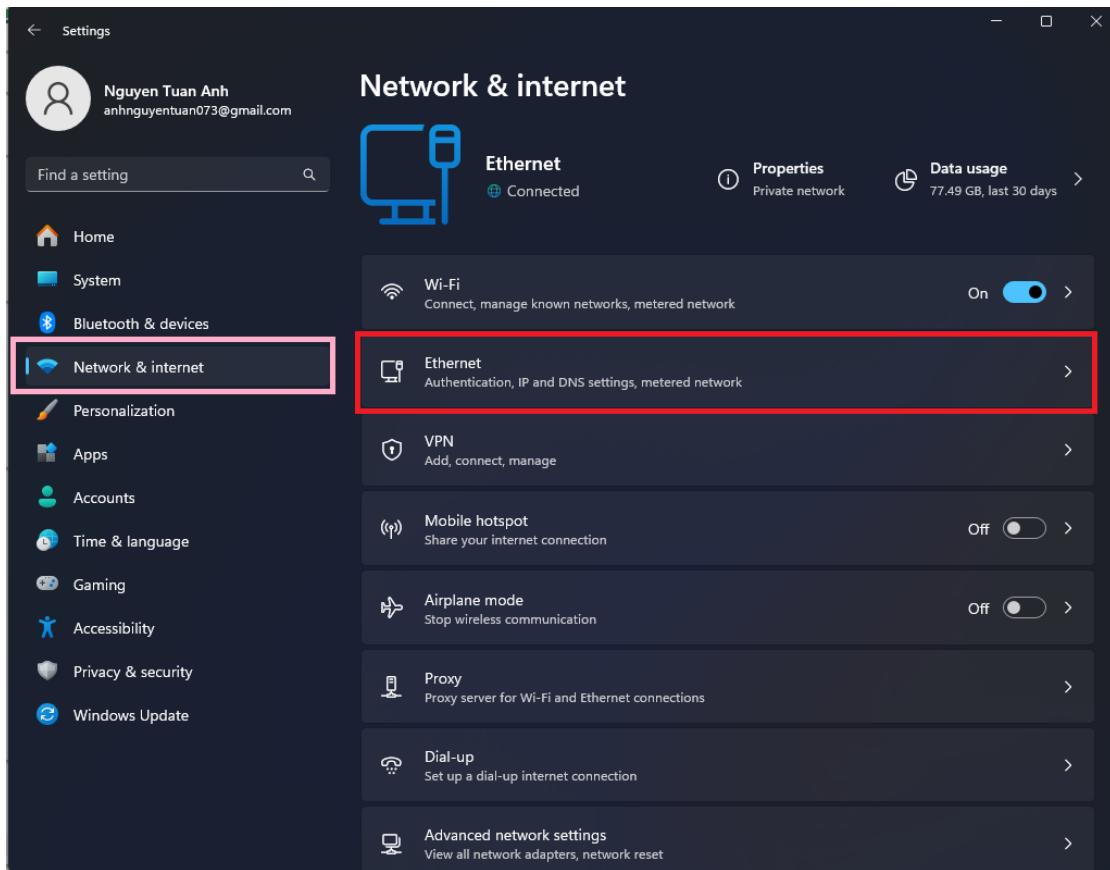
Hình 8: Các dòng lệnh cần nhập

Một số lưu ý của cấu hình riêng cho máy Window

Đối với máy Window, chúng ta cần phải cấu hình một số yếu tố sau để đảm bảo chương trình hoạt động ổn định. Trong Window, vào terminal, gõ câu lệnh sau:

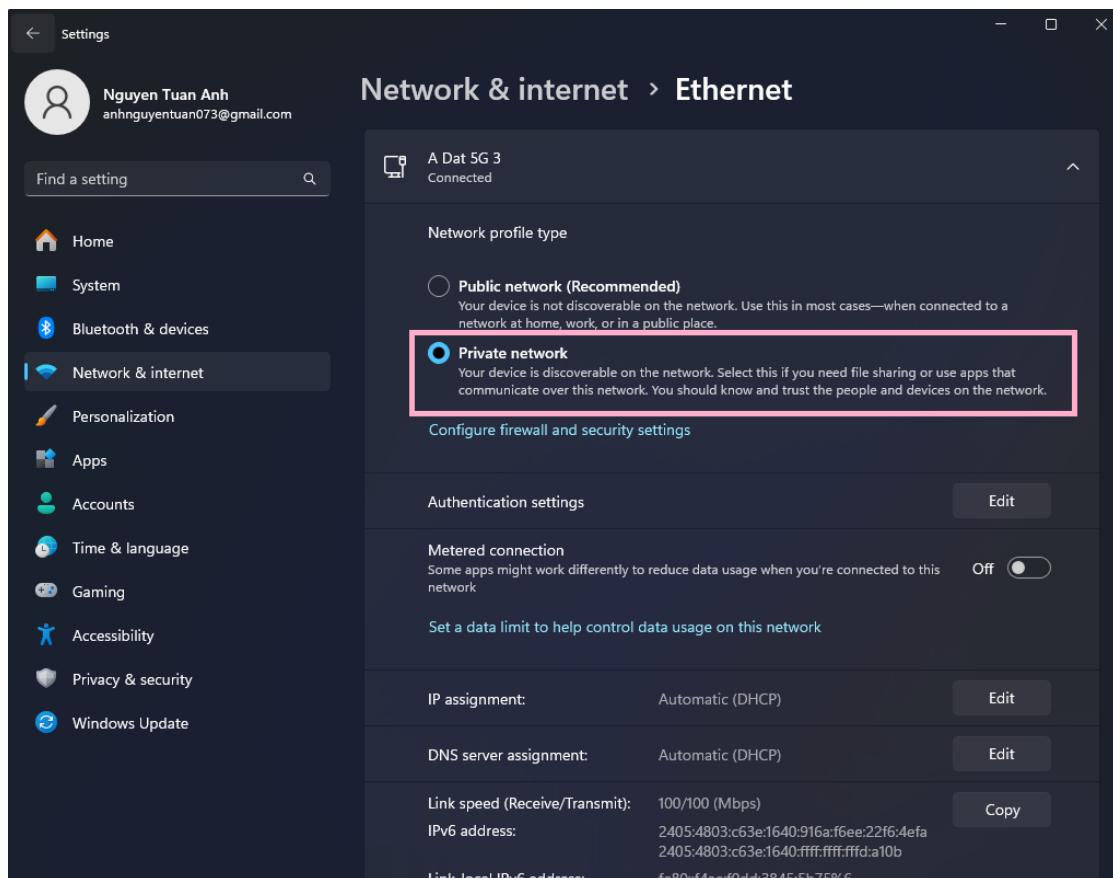
```
netsh advfirewall firewall add rule name="ICMPv4 Allow Ping Requests"
protocol=icmpv4:8,any dir=in action=allow
```

Ngoài ra, ta bấm vào nút **Window** trên bàn phím và ghi **Settings**, sau đó bấm enter để ra được màn hình như sau:



Hình 9: Bước 1 - Cấu hình hệ thống cho Window

Sau đó, chọn vào Ethernet, bấm vào **Private network** để thích hợp cho các ứng dụng chia sẻ file trong mạng nội bộ.



Hình 10: Bước 2 - Cấu hình hệ thống cho Window

6.2 Cách sử dụng hệ thống

Dựa trên Video của mục 5, ta có các bước như sau:

Bước 1: Đầu tiên vào <https://tracker-server-467x.onrender.com/user/register> ở đây bạn hãy tạo tài khoản của mình:



ATSH Tracker Login Register

Đăng ký tài khoản

Họ và tên
Le Van Man

Email
meomeocutedethuong@gmail.com

Mật khẩu

Đăng ký

Hình 11: Trang tạo tài khoản

Bước 2: Khi đăng nhập thành công, ta thấy 1 list danh sách các file của mọi người đã upload lên.

- Nếu bạn muốn upload thì bạn bấm vào nút Upload hiện trên thanh header của màn hình, lúc này bạn chỉ cần tải file bạn muốn chia sẻ (lưu ý: phải là file torrent).

ATSH Tracker Logout Upload

Trang đăng tải file

Nhập vào file của bạn

Chọn tệp | Không có tệp nào được chọn

Tải lên

Hình 12: Giao diện khi upload

- **Trường hợp không có file torrent:** Mở terminal lên và nhập các dòng dưới đây:



The screenshot shows the VS Code interface. The Explorer sidebar on the left displays the file structure of a project named 'BTL1.2'. Inside 'torrent-peer', there are folders 'helpers' and 'node_modules', and a 'storage' folder containing files like 'alice.txt', 'document.doc', 'image.jpg', 'pdf_document.pdf', 'pdf_sample.pdf', and 'video.mp4'. The terminal tab on the right shows the following output:

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC\Documents\Mạng máy tính\BTL1.2\torrent-peer>node index.js
-----Bạn muốn chạy server trên localhost hay giữa các LAN?-----
1. Localhost
2. LAN
3. Thoát ra

Nhập vào lựa chọn của bạn: 2
Server listening on 172.20.10.6:8081
```

This screenshot is similar to the one above, showing the VS Code interface with the same file structure in the Explorer sidebar. The terminal tab shows a different set of output, indicating a menu interaction:

```
=====MENU=====
1. Chia se file
2. Tai file
3. Thoat ra
=====MENU=====
Chon mot tuy chon: 1
Vui long nhap vao ten file muon chia se: pdf_sample.pdf
File co ton tai. Tien hanh chia se file len tracker
Upload len tracker thanh cong
```

Hình 13: Upload file

- Nếu bạn muốn Download file về thì bấm nút Download ở cột link ngay dòng chứa file bạn cần. Sau đó, lưu nó vào file torrent trong folder torrent-peer. Cuối cùng bạn chỉ cần mở terminal lên và nhập các dòng dưới đây:



```
PS C:\Users\ACER\Desktop\btl\mang may tinh\torrent-peer> node index.js
----Bạn muốn chạy server trên localhost hay giữa các LAN?----
1. Localhost
2. LAN
3. Thoát ra

Nhập vào lựa chọn của bạn: 2
Server listening on 172.20.10.3:8081
```

```
====MENU====
1. Chia se file
2. Tai file
3. Thoát ra
====MENU====
Chon mot tuy chon: 2
Vui long nhap vao ten file ban muon tai: pdf_sample.torrent
[ '172.20.10.6:8081', '172.20.10.3:8081' ]
{ numOfPeers: 2, numOfPieces: 115 }
```

Hình 14: Download file

Lúc này chỉ cần nó trả về thành công là hoàn thành download file

Tài liệu

- [1] Hoàng Vui. (02/02/2024). *Peer to peer là gì? Mô hình của peer to peer.* Truy cập từ <https://vietnix.vn/peer-to-peer-la-gi/#peer-to-peer-la-gi-p2p-la-gi>.
- [2] Bram Cohen. (10/01/2008). *The BitTorrent Protocol Specification.* Truy cập từ https://www.bittorrent.org/beps/bep_0003.html.
- [3] Codecrafters. *Build your own BitTorrent .* Truy cập từ <https://app.codecrafters.io/courses/bittorrent/overview>.