

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



MẠNG MÁY TÍNH

Assignment

DEVELOP A NETWORK APPLICATION

Advisor:	Bùi Xuân Giang	
Class:	L09	
SV thực hiện:	Hoàng Minh Hiếu	2210986
	Chắc Quang Minh	2212019
	Phan Duệ Triết	2213609
	Lê Nguyễn Tường Linh	2211850

HO CHI MINH CITY, Nov 2024



Contents

1	Giới thiệu đề tài	4
1.1	Tổng quan bài tập lớn	4
1.2	Mục tiêu và kết quả cần đạt được	4
1.3	Mô tả ứng dụng	5
1.3.1	Sản phẩm	5
1.3.2	Mô hình Mạng	5
1.3.3	Lưu trữ Tài liệu	5
2	Cơ sở Lý thuyết	6
2.1	Mô hình Mạng Peer-to-Peer Tập trung	6
2.1.1	Tổng quan về Mạng Peer-to-Peer (P2P)	6
2.1.2	Ưu điểm của Mô hình Mạng P2P	6
2.1.3	Mô hình P2P Tập trung và vai trò của Tracker	6
2.1.4	Các lợi ích chính của việc sử dụng Tracker trong mạng P2P bao gồm: . .	6
2.1.5	Hạn chế của Mô hình P2P Tập trung	6
2.2	Giao thức Truyền tải Dữ liệu	7
2.2.1	HTTP/HTTPS - Giao tiếp giữa Tracker và Peer	7
2.2.2	TCP/IP - Giao tiếp trực tiếp giữa các Peer	7
2.3	Các Mạng Đồng Đăng Hiện Nay và Phân Loại Theo Mức Độ Tập Trung	8
2.3.1	Mạng Đồng Đăng Tập Trung	8
2.3.2	Mạng Đồng Đăng Phi Tập Trung	8
2.3.3	Mạng Đồng Đăng Lai (Hybrid P2P)	9
2.4	Giao Thức BitTorrent	9
2.4.1	Cơ Chế Hoạt Động của BitTorrent	9
2.4.2	Lợi Ích của Giao Thức BitTorrent	10
2.4.3	Hạn Chế của BitTorrent	10
3	Yêu cầu Sản phẩm	11
3.1	Yêu cầu Chức năng	11
3.1.1	Yêu cầu Máy chủ (Tracker)	11
3.1.2	Yêu cầu Máy khách (Peer)	11
3.2	Yêu cầu Phi chức năng	12
3.2.1	Hệ thống	12
3.2.2	Hiệu suất hoạt động	12
3.2.3	Khả năng tương thích	12
3.2.4	Bảo mật	12
3.3	Yêu cầu giao thức	12
3.4	Yêu cầu giao diện	13
3.5	Yêu cầu về dữ liệu	13
4	Thiết kế Hệ thống	14
4.1	Tổng quan Hệ thống	14



5	Hướng dẫn cài đặt và Kiểm thử	15
5.1	Yêu Cầu Cơ Bản	15
5.2	Hướng Dẫn Cài Đặt	15
5.2.1	Clone Repository	15
5.2.2	Cài Đặt Các Phụ Thuộc	15
5.2.3	Chạy Tracker	15
5.2.4	Chạy Peer	16
5.3	Khắc Phục Sự Cố	16
5.3.1	Tracker Không Phản Hồi	16
5.3.2	Vấn Đề Tải Xuống	16
5.3.3	Ứng Dụng Bị Treo Hoặc Crash	16
6	Tài liệu kham khảo	17



Danh sách hình vẽ

1	Minh họa hệ thống chia sẻ thông tin	14
2	Kiểm tra kết nối tracker	15
3	Kiểm tra kết nối peer	16

1 Giới thiệu đề tài

Môn học Mạng máy tính đóng một vai trò quan trọng trong lĩnh vực công nghệ thông tin, nhờ môn học này chúng ta hiểu cách kết nối, chia sẻ giữa các thiết bị điện tử. Không những thế, môn học không chỉ giúp sinh viên nhận thức vai trò quan trọng của mạng máy tính trong cuộc sống hiện đại và trong phát triển của công nghệ thông tin.

Trong thời đại kỹ thuật số hiện nay, việc chia sẻ dữ liệu giữa các thiết bị ngày càng trở nên phổ biến và cần thiết hơn bao giờ hết. Để đáp ứng nhu cầu này mà không cần sự trung gian của máy chủ trung gian, việc phát triển ứng dụng chia sẻ file ngang hàng (P2P) trên mạng máy tính đang trở thành một ưu tiên quan trọng trong lĩnh vực công nghệ thông tin.

Báo cáo này sẽ tập trung vào quá trình thiết kế và triển khai của một ứng dụng P2P cho phép người dùng truyền tải dữ liệu trực tiếp giữa các thiết bị thông qua mạng máy tính. Mục tiêu của ứng dụng là tạo ra một môi trường truyền tải dữ liệu an toàn, nhanh chóng và hiệu quả, mà không cần sự can thiệp của máy chủ trung gian. Trên cơ sở những tính năng và yếu tố trên, báo cáo sẽ trình bày quy trình thiết kế và triển khai của ứng dụng P2P, cùng với các thách thức và giải pháp trong quá trình phát triển.

1.1 Tổng quan bài tập lớn

Bài tập lớn yêu cầu xây dựng một ứng dụng tương tự như Torrent dựa trên giao thức TCP/IP và hỗ trợ chuyển dữ liệu đa hướng (MDDT). Ứng dụng bao gồm hai loại host: tracker và node. Tracker đóng vai trò trung tâm, theo dõi các node đang kết nối đến và lưu trữ thông tin các phần dữ liệu có trên node. Node sẽ thông báo cho tracker về các tệp tin có trong khu lưu trữ cục bộ trên máy của node, nhưng không truyền dữ liệu tệp tin lên tracker. Khi một node cần một tệp nhưng không có trong kho lưu trữ, node có thể gửi request đến tracker. Từ đó, ứng dụng trả về các node khác nhau đang giữ các phần của file (pieces of file), node request sẽ có khả năng tải xuống đồng thời nhiều tệp hoặc các phần của tệp từ nhiều node khác nhau. Vì vậy, ứng dụng yêu cầu việc triển khai node code phải đa luồng.

1.2 Mục tiêu và kết quả cần đạt được

Mục tiêu của bài tập lớn là xây dựng một ứng dụng đơn giản tương tự như Torrent dựa trên tcp/ip protocol với khả năng truyền dữ liệu đa chiều (MDDT). Kết quả cần đạt được bao gồm:

- Xây dựng hai loại host : tracker và node
- Tracker quản lý được các node đã kết nối và thông tin các phần của file mà node đang giữ.
- Tracker có thể phản hồi các yêu cầu về thông tin các node đang giữ các phần của file hoặc về thông tin các file đó.
- Node có thể thông báo cho Tracker về các files đang được giữ trên local repository.
- Node có thể gửi yêu cầu cho Tracker để lấy thông tin của các node đang giữ các phần của file hoặc thông tin các file đó.
- Kết nối nhiều node với nhau để trao đổi nhiều files và các phần của file một cách đồng thời bằng đa luồng.
- Các node trao đổi file thành công.
- Ngoài ra, đảm bảo các tính năng người dùng khác.

1.3 Mô tả ứng dụng

1.3.1 Sản phẩm

Ứng dụng sẽ được thiết kế để người dùng có thể chia sẻ và tải xuống tài liệu qua mạng ngang hàng với hai thành phần chính:

- **Tracker (Máy chủ):** Đóng vai trò quản lý siêu dữ liệu về các tệp tin sẵn có và thông tin về các máy khách đang kết nối. Tracker giúp xác định các tài nguyên có sẵn và cung cấp danh sách các máy khách có thể chia sẻ tệp tin, nhưng không tham gia vào việc truyền tải tệp tin thực tế.
- **Node (Máy khách):** Mỗi máy khách lưu trữ và truyền tải tệp tin thực tế. Các máy khách sẽ giao tiếp với Tracker để đăng ký tài liệu và nhận thông tin về các peer khác đang lưu trữ tệp tin, đồng thời có khả năng kết nối trực tiếp với nhau qua TCP để chia sẻ tài liệu.

1.3.2 Mô hình Mạng

Ứng dụng sử dụng mô hình Mạng Peer-to-Peer Tập trung, trong đó có một máy chủ trung tâm (Tracker) quản lý các peer. Các peer chia sẻ và nhận tài liệu trực tiếp từ nhau, tối ưu hóa sử dụng băng thông và giảm tải cho Tracker. Các đặc điểm chính của mô hình này bao gồm:

- **Thông tin trung tâm:** Tracker duy trì thông tin siêu dữ liệu và danh sách các peer kết nối. Các peer sẽ kết nối với tracker qua giao thức HTTP/HTTps để lấy thông tin về danh sách peer đang chia sẻ để kết nối
- **Truyền tải tệp tin trực tiếp giữa các peer:** Các máy khách thực hiện truyền tải trực tiếp với nhau, sử dụng kết nối TCP/IP để tăng hiệu suất và giảm thời gian truyền, đảm bảo độ tin cậy và hiệu suất cao.

1.3.3 Lưu trữ Tài liệu

Tài liệu sẽ được lưu trữ cục bộ trên mỗi máy khách. Khi một máy khách muốn chia sẻ tệp tin, nó sẽ thông báo cho Tracker về sự có mặt của tệp tin đó, giúp các peer khác dễ dàng tìm kiếm và truy cập.

- **Siêu dữ liệu:** Tracker chỉ lưu trữ siêu dữ liệu về tệp tin, bao gồm thông tin về các node đang chia sẻ tệp, tên tệp và kích thước, không lưu trữ nội dung thực tế của tệp.
- **Lợi ích của lưu trữ cục bộ:** Cách lưu trữ này giúp tối ưu hóa dung lượng lưu trữ cho Tracker, tận dụng tốt hơn tài nguyên từ các máy khách và phân tán quá trình lưu trữ dữ liệu.

2 Cơ sở Lý thuyết

2.1 Mô hình Mạng Peer-to-Peer Tập trung

2.1.1 Tổng quan về Mạng Peer-to-Peer (P2P)

Mạng Peer-to-Peer (P2P) là một mô hình mạng phân tán, trong đó các máy tính trong mạng, được gọi là peer, hoạt động bình đẳng trong việc chia sẻ tài nguyên. Không giống như mô hình máy chủ - máy khách truyền thống, nơi một máy chủ chính quản lý tất cả các tài nguyên và phục vụ cho các máy khách, mỗi peer trong mạng P2P đều có thể đóng vai trò vừa là máy chủ vừa là máy khách. Do đó, các peer có thể gửi và nhận tài nguyên từ các peer khác trong mạng mà không cần phải dựa vào một máy chủ trung tâm để quản lý tất cả hoạt động truyền tải.

2.1.2 Ưu điểm của Mô hình Mạng P2P

Mô hình P2P giúp tăng cường tính phân tán, giảm tải cho máy chủ và tối ưu băng thông, vì mỗi peer có thể chia sẻ tài nguyên trực tiếp với các peer khác. Điều này làm cho mô hình P2P đặc biệt hiệu quả trong việc chia sẻ tệp tin hoặc tài nguyên trên quy mô lớn, như với các hệ thống chia sẻ tệp như BitTorrent.

2.1.3 Mô hình P2P Tập trung và vai trò của Tracker

Trong mô hình P2P Tập trung, một máy chủ trung tâm đóng vai trò là Tracker. Tracker quản lý một thư mục chứa thông tin về tài nguyên và các peer đang chia sẻ tài nguyên đó, giúp xác định vị trí của các tài nguyên trong mạng. Tuy nhiên, Tracker không tham gia trực tiếp vào việc truyền tải dữ liệu. Thay vào đó, khi một peer yêu cầu tài nguyên, Tracker chỉ cung cấp thông tin về các peer khác đang sở hữu tài nguyên đó, sau đó các peer sẽ trực tiếp kết nối với nhau để chia sẻ tệp.

2.1.4 Các lợi ích chính của việc sử dụng Tracker trong mạng P2P bao gồm:

- **Dễ dàng quản lý và giám sát tài nguyên:** Tracker giữ thông tin về tài nguyên và các peer, giúp dễ dàng xác định vị trí tài nguyên và quản lý tình trạng của các peer.
- **Tối ưu hóa tìm kiếm tài nguyên:** Người dùng có thể tìm kiếm tài nguyên mà không cần quét toàn bộ mạng.
- **Giảm thiểu tải trên máy chủ:** Máy chủ không tham gia vào truyền tải dữ liệu, do đó không gây ra nghẽn băng thông trên Tracker.

2.1.5 Hạn chế của Mô hình P2P Tập trung

Mô hình này có nhược điểm là phụ thuộc vào Tracker trung tâm. Nếu Tracker gặp sự cố hoặc bị quá tải, hệ thống có thể gặp khó khăn trong việc tìm kiếm và kết nối với các peer khác. Điều này làm giảm độ tin cậy và hiệu quả của hệ thống khi xảy ra lỗi hoặc tắc nghẽn tại máy chủ.

2.2 Giao thức Truyền tải Dữ liệu

Trong ứng dụng này, hai giao thức chính được sử dụng cho các mục đích truyền tải và quản lý thông tin giữa Tracker và các peer là HTTP/HTTPS và TCP/IP.

2.2.1 HTTP/HTTPS - Giao tiếp giữa Tracker và Peer

- **HTTP (Hypertext Transfer Protocol):** Là một giao thức tầng ứng dụng dùng để truyền tải dữ liệu siêu văn bản (như HTML) giữa các máy tính qua mạng. Trong hệ thống này, HTTP được sử dụng để thực hiện các thao tác truyền tải thông tin giữa Tracker và peer, ví dụ: peer gửi yêu cầu về danh sách các tệp tin có sẵn hoặc danh sách các peer có tệp tin cần tải.
- **HTTP** là giao thức cho phép trình duyệt yêu cầu và tải về các tài nguyên từ trang web, bao gồm tài liệu siêu văn bản HTML, hình ảnh, video và các dữ liệu khác. Hoạt động của HTTP dựa trên mô hình “yêu cầu - phản hồi” trong hệ thống “khách - chủ” (client-server). Khi người dùng truy cập vào một trang web, trình duyệt sẽ gửi yêu cầu HTTP đến máy chủ, máy chủ phản hồi bằng cách gửi lại dữ liệu trang web để trình duyệt hiển thị. Một phương pháp cho yêu cầu HTTP như GET, POST, HEAD, PUT, ...
- **HTTPS (Hypertext Transfer Protocol Secure):** Là một mở rộng của HTTP, bổ sung thêm khả năng bảo mật với giao thức mã hóa TLS (Transport Layer Security). Trong ứng dụng này, HTTPS có thể được sử dụng khi cần bảo vệ thông tin về tài nguyên hoặc các kết nối giữa máy khách và Tracker.

2.2.2 TCP/IP - Giao tiếp trực tiếp giữa các Peer

TCP/IP (Transmission Control Protocol / Internet Protocol) là tập hợp các giao thức chuẩn được sử dụng để kết các thiết bị trong mạng Internet và truyền dữ liệu giữa chúng. Đây là bộ giao thức quan trọng nhất cho các hoạt động của Internet, bao gồm hai giao thức chính :

- **TCP (Transmission Control Protocol):** Đảm bảo rằng dữ liệu được chia thành các gói nhỏ và được truyền tải một cách đáng tin cậy. TCP kiểm tra xem các gói nhỏ có được gửi đến đúng đích hay không và tái lắp ráp chúng thành dữ liệu gốc.
- **IP (Internet Protocol) :** Định tuyến các gói dữ liệu qua nhiều mạng khác nhau và đảm bảo rằng chúng đến đúng địa chỉ đích.

Hoạt động của giao thức TCP có thể được chia thành ba giai đoạn.

- **Thiết lập kết nối** là một quá trình bắt tay nhiều bước (thường là threeway handshake) để thiết lập kết nối trước khi truyền dữ liệu.
- **Truyền dữ liệu** cung cấp cơ chế báo nhận (Acknowledgement), cơ chế đánh số thứ tự gói tin (sequencing), có chế điều khiển luồng thích hợp (flow control)...
- **Chấm dứt kết nối** sẽ đóng kết nối và giải phóng tất cả các tài nguyên được phân bổ

Nhờ có TCP/IP, các thiết bị có thể giao tiếp với nhau trên quy mô toàn cầu một cách hiệu quả và chính xác.

2.3 Các Mạng Đồng Đẳng Hiện Nay và Phân Loại Theo Mức Độ Tập Trung

Mạng đồng đẳng (P2P) có thể được phân loại dựa trên mức độ tập trung của chúng. Các loại mạng P2P phổ biến hiện nay bao gồm:

2.3.1 Mạng Đồng Đẳng Tập Trung

Mạng đồng đẳng tập trung có một máy chủ trung tâm, thường gọi là Tracker, để duy trì danh sách các tài nguyên và các peer hiện có trong mạng. Tracker giúp cho quá trình tìm kiếm và kết nối giữa các peer trở nên dễ dàng và nhanh chóng hơn. Tuy nhiên, do phụ thuộc vào máy chủ trung tâm, mạng tập trung có thể gặp phải một số nhược điểm về độ tin cậy nếu Tracker bị lỗi hoặc bị tấn công.

2.3.1.1 Ưu điểm:

- Tìm kiếm tài nguyên nhanh chóng: Vì Tracker lưu trữ toàn bộ siêu dữ liệu và thông tin về các peer, người dùng có thể dễ dàng tìm kiếm và truy cập các tài nguyên cần thiết.
- Quản lý tài nguyên hiệu quả: Tracker có thể điều phối và giám sát tài nguyên, giúp quản lý mạng một cách hiệu quả.

2.3.1.2 Nhược điểm:

- Phụ thuộc vào máy chủ trung tâm: Nếu máy chủ trung tâm gặp sự cố, toàn bộ mạng sẽ bị ảnh hưởng.
- Dễ bị tấn công: Mạng có nguy cơ bị tấn công hoặc bị quá tải nếu có nhiều người dùng truy cập cùng một lúc.

2.3.1.3 Ví dụ điển hình : Napster, một trong những hệ thống chia sẻ nhạc đầu tiên, sử dụng mô hình mạng P2P tập trung với máy chủ lưu trữ thông tin về các bài hát và các peer.

2.3.2 Mạng Đồng Đẳng Phi Tập Trung

Mạng đồng đẳng phi tập trung không có máy chủ trung tâm mà tất cả các peer đều đóng vai trò bình đẳng. Mỗi peer có thể vừa là máy khách vừa là máy chủ, giữ thông tin về các peer khác và tự thực hiện quá trình tìm kiếm tài nguyên. Trong mạng phi tập trung, các peer được tổ chức thành một mạng lưới lớn và có thể liên lạc trực tiếp với nhau để truyền tải tài liệu.

2.3.2.1 Ưu điểm:

- Tính phân tán cao: Không phụ thuộc vào máy chủ trung tâm, nên mạng sẽ tiếp tục hoạt động kể cả khi một số peer bị lỗi.
- Khó bị tấn công: Vì không có điểm tập trung, các mạng phi tập trung khó bị tấn công toàn diện hơn.

2.3.2.2 Nhược điểm:

- Tìm kiếm tài nguyên chậm hơn: Vì không có một trung tâm quản lý tài nguyên, quá trình tìm kiếm tài nguyên có thể mất nhiều thời gian hơn.
- Phức tạp hơn về cấu trúc: Quản lý mạng phi tập trung cần các thuật toán phức tạp để duy trì thông tin giữa các peer.

2.3.2.3 Ví dụ điển hình : Gnutella, một mạng chia sẻ tệp phi tập trung, cho phép các peer tự động kết nối với nhau và tìm kiếm tài liệu mà không cần sự điều phối của máy chủ trung tâm.

2.3.3 Mạng Đồng Đẳng Lai (Hybrid P2P)

Mạng đồng đẳng lai kết hợp cả tính năng của mạng tập trung và mạng phi tập trung. Một số thành phần của hệ thống có thể tập trung, ví dụ như một máy chủ trung tâm lưu trữ một số siêu dữ liệu cơ bản và hỗ trợ tìm kiếm, nhưng quá trình truyền tải dữ liệu thực tế giữa các peer là phi tập trung.

2.3.3.1 Ưu điểm:

- Tìm kiếm tài nguyên nhanh chóng: Với một số thành phần tập trung, quá trình tìm kiếm có thể được tối ưu hóa.
- Tính phân tán cao trong truyền tải: Dữ liệu vẫn được chia sẻ trực tiếp giữa các peer, giúp tận dụng hiệu quả băng thông.

2.3.3.2 Nhược điểm:

- Phức tạp trong quản lý: Mô hình lai đòi hỏi thiết kế phức tạp hơn và quản lý chặt chẽ giữa các phần tập trung và phi tập trung.

2.3.3.3 Ví dụ điển hình : BitTorrent, một giao thức chia sẻ tệp lai, sử dụng máy chủ Tracker để lưu trữ thông tin về các peer có sẵn và tệp tin nhưng vẫn cho phép các peer chia sẻ dữ liệu trực tiếp với nhau. **Đây là hướng mà nhóm chọn để phát triển ứng dụng**

2.4 Giao Thức BitTorrent

2.4.1 Cơ Chế Hoạt Động của BitTorrent

BitTorrent là một trong những giao thức P2P nổi tiếng nhất, được thiết kế để phân phối và chia sẻ tệp tin trên mạng bằng cách tận dụng băng thông từ nhiều peer. BitTorrent kết hợp cả yếu tố tập trung và phi tập trung, sử dụng máy chủ Tracker để điều phối thông tin và quá trình chia sẻ trực tiếp giữa các peer.

BitTorrent sử dụng một số thuật ngữ và cơ chế quan trọng sau:

- **Tracker**: Là một máy chủ đóng vai trò trung tâm trong việc điều phối thông tin giữa các peer. Tracker không lưu trữ dữ liệu tệp, mà chỉ lưu thông tin về các peer đang chia sẻ tệp và cung cấp danh sách các peer để một peer có thể tìm được những peer khác trong mạng.
- **Seeder**: Là các peer đã tải xuống toàn bộ tệp và tiếp tục chia sẻ với các peer khác. Seeder đóng vai trò quan trọng trong việc duy trì sự sẵn có của tệp tin trong mạng.

- **Leecher:** Là các peer đang tải xuống tệp từ Seeder hoặc từ các Leecher khác. Khi hoàn thành quá trình tải xuống, một Leecher có thể trở thành Seeder để chia sẻ lại tài liệu.
- **Swarm:** Là toàn bộ các peer (bao gồm cả Seeder và Leecher) tham gia vào quá trình tải xuống và chia sẻ một tệp tin cụ thể.
- **.torrent File:** Đây là tệp siêu dữ liệu, chứa thông tin về các tệp cần chia sẻ (như tên, kích thước và các đoạn phân chia) và địa chỉ của Tracker. Tệp .torrent không chứa dữ liệu thực tế của tệp tin mà chỉ chứa các thông tin cần thiết để các peer có thể tải xuống và lắp ráp lại tệp từ các phần riêng biệt.

2.4.2 Lợi Ích của Giao Thức BitTorrent

- **Tận dụng băng thông tối đa:** Bằng cách tải xuống từ nhiều nguồn cùng một lúc, BitTorrent có thể đạt tốc độ tải xuống cao hơn so với tải trực tiếp từ một nguồn duy nhất.
- **Tăng tính sẵn có của tệp tin:** Các Seeder và Leecher cùng chia sẻ tệp tin, giúp duy trì và lan truyền tệp tin trên mạng mà không cần phụ thuộc vào một máy chủ duy nhất.
- **Tính phân tán và bảo mật cao hơn:** BitTorrent không lưu trữ tệp tin tập trung trên một máy chủ duy nhất, giúp tăng cường bảo mật và tránh được các cuộc tấn công từ chối dịch vụ (DoS).

2.4.3 Hạn Chế của BitTorrent

- **Phụ thuộc vào Seeder:** Khi không còn Seeder nào duy trì tệp tin, tệp sẽ không thể tải xuống được, gây gián đoạn trong quá trình chia sẻ tài liệu.
- **Nguy cơ tải xuống tệp không an toàn:** Vì các peer có thể chia sẻ bất kỳ tệp nào, các tệp không an toàn hoặc độc hại có thể lây lan qua BitTorrent, đặc biệt khi không có hệ thống xác thực nội dung.

3 Yêu cầu Sản phẩm

3.1 Yêu cầu Chức năng

3.1.1 Yêu cầu Máy chủ (Tracker)

- **Quản lý Máy khách:**
 - Chấp nhận kết nối từ các máy khách và duy trì danh sách các máy khách đã kết nối.
 - Gán định danh duy nhất cho mỗi máy khách khi kết nối.
- **Quản lý Siêu dữ liệu Tập tin:**
 - Lưu trữ và quản lý siêu dữ liệu về các tệp tin có sẵn trên mỗi máy khách, bao gồm tên tệp tin, kích thước .
 - Cập nhật thông tin về sự sẵn có của tệp tin khi máy khách xuất bản hoặc hủy xuất bản tệp tin.
- **Xử lý Yêu cầu:**
 - Phản hồi yêu cầu từ máy khách về:
 - * Danh sách các tệp tin có sẵn.
 - * Thông tin về các peer giữ các phần cụ thể của tệp tin.
 - Cung cấp cơ chế cho máy khách truy vấn sự sẵn có của tệp tin.
- **Giám sát Trạng thái:**
 - Theo dõi trạng thái của các máy khách kết nối.
 - Loại bỏ các máy khách khỏi danh sách khi ngắt kết nối hoặc hết thời gian chờ.

3.1.2 Yêu cầu Máy khách (Peer)

- **Xác thực Người dùng:**
 - Cho phép người dùng đăng nhập và đăng xuất một cách an toàn.
 - Xử lý quản lý phiên làm việc và duy trì trạng thái người dùng.
- **Xuất bản Tập tin:**
 - Cho phép máy khách xuất bản các tệp tin mà họ muốn chia sẻ bằng cách thông báo cho Tracker.
 - Hỗ trợ tạo siêu dữ liệu (ví dụ: tệp .torrent) cho các tệp tin chia sẻ.
- **Khám phá và Tải xuống Tập tin:**
 - Cho phép máy khách yêu cầu thông tin về các tệp tin và nhận danh sách các peer có chúng.
 - Hỗ trợ tải xuống tệp tin bằng cách kết nối đến nhiều peer đồng thời.
 - Ghép các phần tệp tin đã tải xuống thành tệp tin gốc.
- **Tải lên Tập tin:**

- Phản hồi các yêu cầu từ các peer khác để tải lên các phần tệp tin.
- Quản lý kết nối tải lên một cách hiệu quả.

- **Báo cáo Tiến trình:**

- Cập nhật cho Tracker về trạng thái của quá trình tải xuống, bao gồm các mảnh piece đang chờ và piece đã tải xuống.
- Thông báo cho Tracker khi các phần tệp tin mới trở nên sẵn có để chia sẻ.

- **Xử lý Lỗi:**

- Xử lý các lỗi mạng một cách nhẹ nhàng và thử lại kết nối khi cần.
- Cung cấp thông báo thông tin cho người dùng trong trường hợp thất bại.

3.2 Yêu cầu Phi chức năng

3.2.1 Hệ thống

- Hệ thống phải được triển khai trên nhiều host tham gia vào mạng.
- Tệp được truyền thành công qua các client và cần sự can thiệp của server

3.2.2 Hiệu suất hoạt động

- Tốc độ tải lên và tải xuống. Đảm bảo khả năng tải file từ nhiều nguồn lúc lúc (multi-peer download/upload) để tối ưu tốc độ và băng thông.

3.2.3 Khả năng tương thích

- Quản lý tài nguyên hiệu quả: Ứng dụng cần quản lý tốt bộ nhớ và CPU, đặc biệt là khi chạy đa luồng để tải từ nhiều nguồn.
- Hỗ trợ đa nền tảng cho các hệ điều hành chính (Windows, macOS, Linux)

3.2.4 Bảo mật

- **Tính Toàn vẹn Dữ liệu:**
 - Sử dụng checksum hoặc hash để xác minh tính toàn vẹn của các tệp tin đã tải xuống.
- **Xác thực:**
 - Triển khai các cơ chế xác thực cơ bản để ngăn chặn truy cập trái phép.

3.3 Yêu cầu giao thức

Ứng dụng sử dụng hoàn toàn giao thức HTTP ở tầng Application và TCP ở tầng transport để truyền/ nhận dữ liệu hệ thống.

3.4 Yêu cầu giao diện

- **Server**

- Giao diện quản lý: Server cần có một giao diện quản trị để quản lý các client, theo dõi trạng thái file và kiểm soát băng thông
- Thống kê: Hiển thị thông tin chi tiết về lưu lượng mạng, số lượng client đang kết nối, số lượng file và trạng thái tải xuống/tải lên.
- Bảng điều khiển: Cung cấp thông tin cấu hình server và điều khiển trạng thái như khởi động lại dịch vụ, quản lý peer.

- **Client**

- Bảng điều khiển tải xuống: Hiển thị thông tin về tiến trình tải, tốc độ tải lên/tải xuống và danh sách các phần file đã tải.
- Quản lý file: Hiển thị danh sách các file mà client đã tải xong và đang chia sẻ, cùng với thông tin chi tiết về từng file.
- Thông tin peer: Cho phép xem danh sách các peer đang kết nối và cung cấp tùy chọn kết nối hoặc dừng kết nối với peer nào đó.

3.5 Yêu cầu về dữ liệu

- **Lưu trữ tài liệu**

- Nơi lưu trữ: **Client**
- Mục đích: Lưu trữ tài liệu bản gốc, các tài liệu tải về từ hệ thống cần giảm dung lượng lưu trữ tại server. Kích thước tối đa của mỗi tài liệu là 1GB, tổng kích thước sẽ phụ thuộc vào ổ đĩa của người dùng
- Thời lượng lưu trữ: Mỗi người dùng đăng nhập vào hệ thống, ứng dụng sẽ kiểm tra và đăng tải các tài liệu lưu trữ local trên server. Nếu người dùng bị thoát khỏi hệ thống (do lỗi phát sinh) mà chưa kịp thoát phiên đăng nhập thì tài liệu sẽ local repo vẫn sẽ tồn tại.
- Tính khả dụng: Toàn bộ dữ liệu được lưu trữ local sẽ gắn với thiết bị, không phải phiên đăng nhập. Nếu người dùng vẫn còn khả dụng trên hệ thống thì tất cả các tài liệu được công bố sẽ được phép tải về.

- **Lưu trữ thông tin hệ thống**

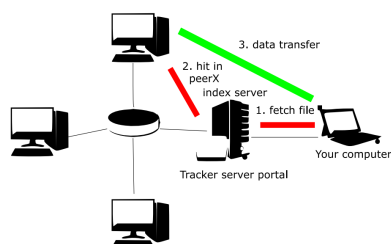
- Nơi lưu trữ : **Server**
- Mục đích: Lưu trữ thông tin về hoạt động trong hệ thống nhằm quản lý vận hành nhưng không trực tiếp lưu trữ tài liệu.
- Thời gian lưu trữ :Cơ sở dữ liệu tập chung sẽ được khởi tạo sau mỗi lần thiết lập server và sẽ được xóa đi khi tắt server. Hệ thống sẽ cập nhật thường xuyên về tính khả dụng của tài liệu và client trên hệ thống. Mỗi client sẽ được gắn với danh sách tài liệu được công bố nên nếu client thoát khỏi hệ thống (đăng xuất hoặc mất kết nối) thì hệ thống sẽ đánh dấu cả client và tài liệu đều không còn khả dụng.

4 Thiết kế Hệ thống

4.1 Tổng quan Hệ thống

Hệ thống được thiết kế như một mạng peer-to-peer tập trung bao gồm:

- **Tracker (Máy chủ):**
 - Đóng vai trò là điều phối viên trung tâm.
 - Một máy chủ trung tâm giữ thông tin về các máy khách đang kết nối tới các tệp tin.
 - Không lưu trữ nội dung tệp tin thực tế.
- **Node (Máy khách):**
 - Lưu trữ và chia sẻ tệp tin thực tế.
 - Giao tiếp với Tracker để lấy thông tin về tệp tin và các peer khác.
 - Truyền tải các phần tệp tin trực tiếp giữa nhau sử dụng kết nối TCP.



Hình 1: Minh họa hệ thống chia sẻ thông tin

5 Hướng dẫn cài đặt và Kiểm thử

5.1 Yêu Cầu Cơ Bản

Trước khi cài đặt BitTorrent, hãy đảm bảo bạn có các yêu cầu sau:

- **Python 3.7 trở lên:** Khách hàng được phát triển bằng Python, do đó yêu cầu Python 3.7+.
- **Kiến Thức Cơ Bản Về Terminal/Command Prompt:** Hiểu biết cơ bản về các lệnh dòng lệnh.

5.2 Hướng Dẫn Cài Đặt

5.2.1 Clone Repository

Đầu tiên, clone repository chứa mã nguồn về máy tính của bạn.

```
git clone https://github.com/qminh3/P2P-FILE-SHARING.git
```

5.2.2 Cài Đặt Các Phụ Thuộc

Đảm bảo bạn đã cài đặt pip. Sau đó, cài đặt các gói Python cần thiết.

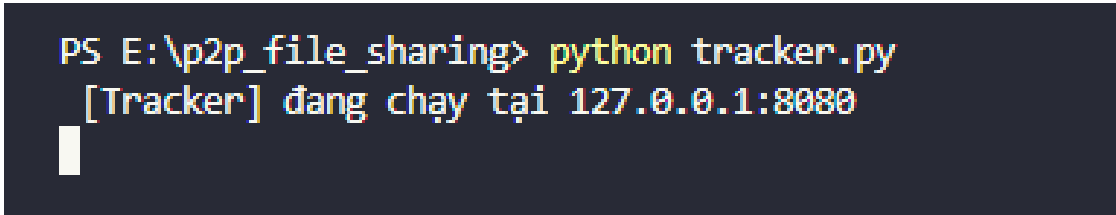
```
pip install bencodepy json base64 hashlib socket
```

5.2.3 Chạy Tracker

Tracker chịu trách nhiệm quản lý thông báo từ các peer và duy trì danh sách các peer đang hoạt động cho mỗi tệp.

Khởi Động Tracker:

```
python tracker.py
```



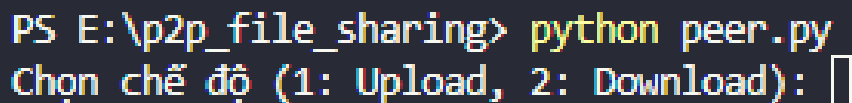
```
PS E:\p2p_file_sharing> python tracker.py  
[Tracker] đang chạy tại 127.0.0.1:8080
```

Hình 2: Kiểm tra kết nối tracker

- Tracker sẽ bắt đầu lắng nghe tại 127.0.0.1:8080.
- Đảm bảo rằng cổng 8080 được mở và không bị chặn bởi tường lửa của bạn.

5.2.4 Chạy Peer

python peer.py



```
PS E:\p2p_file_sharing> python peer.py
Chọn chế độ (1: Upload, 2: Download):
```

Hình 3: Kiểm tra kết nối peer

Chọn chế độ (1: Upload, 2: Download):

```
# Nhập port cho peer server:
# Nhập tên của file cần tải:
# Nhập piece_length:
# Nhập tên file để lưu:
# Chọn phương thức tải file:
# 1. Tải file từ một peer
# 2. Tải file từ nhiều peers
# Nhập lựa chọn (1/2):
```

5.3 Khắc Phục Sự Cố

Dưới đây là một số sự cố phổ biến và cách khắc phục:

5.3.1 Tracker Không Phản Hồi

- **Vấn Đề:** Không thể kết nối với Tracker.
- **Giải Pháp:**
 - Đảm bảo Tracker đang chạy (`tracker.py` đang hoạt động).
 - Kiểm tra cài đặt tường lửa để đảm bảo cổng 8080 được mở.

5.3.2 Vấn Đề Tải Xuống

- **Vấn Đề:** Nhập tên file sai hoặc piece-length lớn hơn tổng số mảnh file .
- **Giải Pháp:**
 - Xác minh rằng các peer seeding đang hoạt động và chia sẻ tệp.
 - Kiểm tra kết nối internet của bạn.
 - Đảm bảo cài đặt tường lửa cho phép các cổng cần thiết.

5.3.3 Ứng Dụng Bị Treo Hoặc Crash

- **Giải Pháp:**
 - Kiểm tra terminal để xem log lỗi.
 - Đảm bảo tất cả các phụ thuộc đã được cài đặt đúng cách.
 - Khởi động lại ứng dụng và thử lại.



6 Tài liệu tham khảo

- [1] BitTorrent Protocol Specification. “Giao thức BitTorrent quy định các khái niệm cốt lõi như Tracker, Seeder, và Leecher để hỗ trợ quá trình chia sẻ tệp hiệu quả.” *BitTorrent BEP 3*, 2008.
- [2] Peer-to-Peer Systems and Applications. “Mô hình P2P bao gồm các mạng tập trung, phi tập trung và lai, với các giao thức truyền tải dữ liệu đa dạng.” *Steinmetz & Wehrle*, 2005.
- [3] Computer Networking: A Top-Down Approach (7th Edition). “Giới thiệu nền tảng của giao thức TCP/IP và ứng dụng trong hệ thống P2P.” *Kurose & Ross*, 2017.