ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

MẠNG MÁY TÍNH (CO3093)

Báo cáo bài tập lớn

GVHD: Hoàng Lê Hải Thanh
Sinh viên: Võ Văn Hiếu - 2252219
Võ Lý Đắc Duy - 2252125
Phan Thảo Vy - 2252930
Đỗ Trường Khoa - 2252345
Lớp: CN01

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2024

Mục lục

1. Mô tả chức năng

1.1. Tracker

Về chức năng, tracker là "trạm" dùng để hoàn thành các tác vụ như lưu trữ thông tin của các peer. Những thông tin gồm các file được lưu trữ dưới dạng vị trí của các peers đang nắm giữ file đó, và làm công tác phản hồi liên quan đến việc seeder đăng kí thông tin của nó, leecher yêu cầu danh sách các peer. Bên cạnh đó cũng phản hồi cho việc thay đổi thông tin.

Check torrent info

- Kiểm tra và cập nhật danh sách torrent hiện tại, nhằm đảm bảo rằng thông tin về mỗi torrent đều được ghi nhận và lưu trữ một cách đầy đủ.
- Điều kiện: khi nhận được yêu cầu từ client, hệ thống sẽ lấy thông tin info hash từ yêu cầu đó để tiến hành kiểm tra.
- Hoạt động: hệ thống sẽ đối chiếu info hash từ client với danh sách các torrent đã có. Nếu info hash chưa tồn tại, tức đây là một torrent mới, hệ thống sẽ bổ sung thông tin này vào danh sách torrent để lưu trữ thông tin chi tiết về torrent đó. Sau khi hoàn tất việc kiểm tra và cập nhật danh sách torrent, hệ thống sẽ tiếp tục thực hiện quy trình kiểm tra thông tin client bằng cách gọi hàm `CHECK_CLIENT_INFO()` nhằm xử lý thông tin của peer mong muốn kết nối.

Check client info

- Kiểm tra và cập nhật danh sách các peer, giúp hệ thống nhận diện và quản lý thông tin của từng peer mong muốn kết nối đến tracker qua một tệp torrent cụ thể.
- Điều kiện: khi hệ thống nhận thông tin từ một peer muốn kết nối, bao gồm info hash và các thông tin như ip, port..., hàm sẽ được kích hoạt để tiến hành xử lý.
- Hoạt động: hệ thống sẽ kiểm tra danh sách các peer đã đăng ký trước đó với info hash cụ thể. Nếu đây là một peer mới, chưa có trong danh sách, hệ thống sẽ lưu trữ thông tin của peer, bao gồm địa chỉ IP, port, và các thuộc tính khác. Nếu peer đã tồn tại trong danh sách, tức là đã từng đăng ký với tracker trước đây, hệ thống sẽ cập nhật lại các thông tin như thời gian cuối cùng kết nối (lastseen), trạng thái kết nối (status), và các thông tin khác cần thiết để đảm bảo dữ liệu luôn chính xác và mới nhất.

Announce Response

- Xử lý và phản hồi các yêu cầu kết nối của các peer, nhằm giúp duy trì sự liên lạc và cập nhật thông tin giữa các peer và tracker.
- Điều kiện: khi nhận được yêu cầu `ANNOUNCE_TO_TRACKER()` từ peer, hàm này sẽ được gọi để thực hiện phản hồi phù hợp dựa trên mục đích của peer gửi yêu cầu.
- Hoạt động: hệ thống sẽ bắt lấy và phân loại các tín hiệu yêu cầu từ các peer dựa trên vai trò của chúng (Seeder, Leecher, hay Peer yêu cầu cập nhật):
  - Seeder muốn đăng ký: hệ thống sẽ thực hiện các hàm `CHECK_TORRENT_INFO()` và `CHECK_CLIENT_INFO()` để kiểm tra và cập nhật thông tin của torrent và client. Sau khi đăng ký thành công, hệ thống sẽ gửi phản hồi xác nhận đã hoàn tất quá trình đăng ký cho Seeder.
  - Leecher muốn tải xuống: hệ thống sẽ gọi hàm `CHECK_TORRENT_INFO()` để kiểm tra thông tin torrent, sau đó lấy danh sách các peer đang chia sẻ tệp tin này và chuyển danh sách đó sang dạng nhị phân để gửi lại cho Leecher. Phản hồi giúp Leecher kết nối với các peer khác để tải xuống dữ liệu cần thiết.

- Peer gửi yêu cầu cập nhật thông tin: đối với peer chỉ muốn cập nhật thông tin, hệ thống sẽ thực hiện các hàm `CHECK_TORRENT_INFO()` và `CHECK_CLIENT_INFO()` để kiểm tra và ghi nhận các thay đổi từ peer. Sau khi cập nhật thành công, hệ thống sẽ gửi phản hồi xác nhận quá trình thay đổi đã hoàn tất.

1.2. Client

1.2.1. Connection

Announce to tracker

- Peer kết nối với tracker để thực yêu cầu.
- Điều kiện: khi client kết nối với tracker.
- Hoạt động: gửi các yêu cầu kết nối khác nhau như: đăng ký seeder, leecher tải xuống, cập nhật thông tin.

Connect to peer

- Sau khi nhận được danh sách peer từ tracker, peer sẽ chọn một hoặc nhiều peer để kết nối. Thông thường, peer sẽ chọn ngẫu nhiên từ danh sách hoặc dựa trên một số tiêu chí nhất định (như tốc độ tải lên/tải xuống).
- Điều kiện: đã có danh sách peer và lựa chon peer để kết nối.
- Hoạt động: Peer sử dụng socket để thiết lập kết nối TCP với peer khác. Kết nối này thường sử dụng địa chỉ IP và cổng mà peer đã nhận từ danh sách peer.

Sau khi quá trình kết nối thành công, các peer có thể bắt đầu trao đổi dữ liệu.

1.2.2. Download

Upload torrent file

- Khi một leecher muốn tải một tài liệu dựa trên tệp torrent, nó sẽ thực hiện tải tệp .torrent từ thiết bị lên client. Tệp này chứa "info hash" (một mã định danh duy nhất của tệp), danh sách các phần (piece) của tệp cùng kích thước mỗi phần, và địa chỉ của các tracker.
- Điều kiện: đã có tệp .torrent trên thiết bị.
- Hoạt động: client sử dụng địa chỉ tracker trong tệp .torrent để kết nối với tracker bằng ANNOUNCE_TO_TRACKER() để thông báo sẽ trở thành leecher. Đồng thời, tracker gửi lại một danh sách các peers khả dụng mà leecher có thể kết nối để tải tệp.

Download flow

- Chuẩn bị Danh sách Công việc: Lấy danh sách các mảnh dữ liệu (pieces) cần tải và xác định danh sách các peers có thể tải dữ liệu.
- Tải dữ liệu từ nhiều nguồn:
  - Chọn một piece cần tải.
  - Kết nối với một peer để tải piece này về. Nếu tải thất bại (do lỗi mạng hoặc dữ liệu không khớp), chuyển sang peer khác để thử lại.
  - Mỗi piece có một giới hạn số lần thử, đảm bảo không lặp lại vô hạn nếu có lỗi.
- Xác minh Dữ liệu Tải về:
  - Sau khi tải một piece, tính toán mã băm và so sánh với mã băm gốc từ file torrent để xác minh dữ liệu có chính xác hay không.
  - Nếu dữ liệu hợp lệ, piece sẽ được đánh dấu hoàn thành và lưu trữ. Nếu không, thử tải lại từ peer khác.
- Xử lý Khi Tải Thất Bại:

- Nếu không tải thành công sau khi thử từ nhiều peers, piece đó sẽ bị đánh dấu thất bại và không tiếp tục thử nữa.
- Lặp Lại cho Đến Khi Hoàn Thành:
  - Quá trình tiếp tục cho đến khi toàn bộ các pieces trong danh sách đã được tải thành công hoặc đạt giới hạn thử thất bại.

Create handshake
- Tạo thông điệp handshake và gửi cho seeder, một bước quan trọng để thiết lập kết nối giữa hai peer.
- Điều kiện: đã thiết lập kết nối bằng socket.
- Hoạt động: khởi tạo các thông tin handshake và gửi qua socket. Sau đó đợi phản hồi HANDSHAKE_RESPONSE() từ seeder.

Interested
- Gửi thông báo rằng leecher sẽ tiến hành trao đổi thông tin với seeder.
- Điều kiện: nhận được HANDSHAKE_RESPONSE() từ seeder.
- Hoạt động: kiểm tra thông tin từ phản hồi của seeder. Nếu thông tin đúng gửi thông báo INTERESTED(). Nếu không đúng thông báo lỗi.

Request Piece
- Gửi yêu cầu tải mảnh.
- Điều kiện: nhận được thông báo UNCHOKE() và BITFIELD() từ seeder.
- Hoạt động: dựa vào thông tin trong bitfield để gửi yêu cầu tải mảnh cần thiết.

Handle Piece
- Xử lý dữ liệu của mảnh nhận được từ seeder.
- Điều kiện: nhận được SEND_PIECE() từ seeder.
- Hoạt động: xử lý và kiểm tra với hash.

Handle File
- Kết nối và lưu tệp về máy.
- Điều kiện: Đã tải tất cả các mảnh cần thiết.
- Hoạt động: tổng hợp các mảnh lại thành một file và kiểm tra. Nếu file hợp lệ sẽ tiến hành tải về máy.

1.2.3. Upload

Chức năng upload là chức năng tải file lên để tạo torrent và thông báo cho tracker, trở thành seeder giúp chia sẻ tài nguyên với những người dùng khác trong hệ thống. Để thực hiện việc này thì trải qua các giai đoạn:

Generate torrent file
- Tạo file .torrent cho các tài liệu cần chia sẻ.
- Điều kiện: Người dùng chọn mục "Upload file".
- Hoạt động: khi người dùng tải file lên client sẽ tạo file .torrent và lưu vào trong bộ nhớ của thiết bị, đồng thời lúc này client sẽ trở thành seeder thông báo với tracker bằng ANNOUNCE_TO_TRACKER() để tiến hành seeding.

Handshake response
- Thiết lập kết nối giữa seeder và leecher bằng cách phản hồi yêu cầu handshake. Điều này đảm bảo rằng cả hai bên đều có thể giao tiếp và xác nhận thông tin cần thiết.
- Điều kiện: Đã kết nối với leecher bằng socket và nhận được HANDSHAKE_REQUEST().

- Hoạt động: Kiểm tra thông tin từ yêu cầu của leecher. Nếu thông tin đúng sẽ gửi lại một handshake cho leecher. Nếu không đúng thông báo lỗi.

Send Bitfield
- Sau khi hoàn tất quá trình handshake và xác nhận rằng seeder đã thiết lập kết nối với leecher, seeder cần gửi thông điệp bitfield để thông báo về trạng thái của các phần (pieces) mà nó đang sở hữu. Dưới đây là quy trình chi tiết
- Điều hiện: sau khi đã handshake và nhận được thông báo INTERESTED().
- Hoạt động: tạo thông điệp bitfield và gửi qua socket.

Send Unchoke
- Sau khi hoàn tất quá trình handshake và nhận được thông điệp bitfield từ leecher, seeder cần gửi thông điệp "unchoke" để thông báo rằng nó đã sẵn sàng cung cấp dữ liệu cho leecher.
- Điều kiện: sau khi đã handshake và đã gửi bitfield.
- Hoạt động: tạo thông điệp unchoke và gửi qua socket.

Send piece
- Khi một leecher yêu cầu một mảnh dữ liệu cụ thể từ seeder, seeder cần gửi mảnh đó nếu nó có.
- Điều kiện: khi leecher yêu cầu một mảnh nào đó.
- Hoạt động: khi nhận được yêu cầu seeder sẽ tạo dữ liệu mảnh đó và gửi dữ liệu bằng socket.

2. Protocols

2.1. Tracker

CHECK_TORRENT_INFO
- Bước 1: kiểm tra trong torrent list đã có thông tin của file torrent tương ứng với infoHash chưa, nếu chưa, lưu thông tin info_hash đó vào list.
- Bước 2: hệ thống sẽ tiếp tục thực hiện quy trình kiểm tra thông tin client bằng cách gọi hàm CHECK_CLIENT_INFO() nhằm xử lý thông tin của peer mong muốn kết nối.

CHECK_CLIENT_INFO
- Bước 1: Kiểm tra peer trong danh sách tương ứng với infoHash để xác định xem peer này đã tồn tại chưa dựa trên ip và port của nó.
- Bước 2:
  - Nếu peer chưa tồn tại: Thêm peer vào mảng peers của infoHash với các thông tin như sau:
    - peer_id: ID duy nhất của peer.
    - ip: Địa chỉ IP của peer.
    - port: Cổng mà peer sử dụng.
    - status: Trạng thái kết nối hiện tại của peer (ví dụ: "connected").
    - lastSeen: Thời gian hiện tại để đánh dấu peer vừa được thêm vào.
  - Nếu peer đã tồn tại: Cập nhật thông tin của peer, bao gồm:
    - lastSeen: Thời gian hiện tại để làm mới thời điểm kết nối.
    - Status: Cập nhật trạng thái của peer nếu cần.

- Bước 3: sau khi cập nhật, thông tin về peer trong torrents của infoHash đã đầy đủ và chính xác, cho phép tracker theo dõi peer một cách hiệu quả. Có thể lấy danh sách peer để phản hồi cho yêu cầu download của leecher.

ANNOUNCE_RESPONSE
- Bước 1: nhận yêu cầu ANNOUNCE_REQUEST() từ peer.
- Bước 2: phản hồi cho peer theo yêu cầu.
  - HTTP/HTTPS Response
  - Gồm các tham số:
    - Failure reason: Nếu có, là thông điệp lỗi giải thích lý do yêu cầu thất bại (chuỗi).
    - Interval: Khoảng thời gian tính bằng giây mà client nên chờ giữa các yêu cầu gửi đến tracker.
    - Tracker ID: Chuỗi mà client nên gửi lại trong các thông báo tiếp theo. Nếu không có và thông báo trước đó đã gửi một tracker ID, không được bỏ qua giá trị cũ; tiếp tục sử dụng nó.
    - Complete: Số lượng peers đã có toàn bộ tệp, tức là seeders (số nguyên).
    - Incomplete: Số lượng peers không phải seeders, còn gọi là "leechers" (số nguyên).
    - Peers: là danh sách các từ điển, mỗi từ điển có các khóa sau:
      - Peer ID: ID tự chọn của peer (chuỗi).
      - IP: Địa chỉ IP của peer, có thể là IPv6 hoặc IPv4 hoặc tên miền DNS.
      - Port: Số cổng của peer (số nguyên).

2.2. Client

2.2.1. Connection

ANNOUNCE_TO_TRACKER
- Bước 1: Peer gửi một yêu cầu HTTP GET đến tracker dựa vào tracker url trong file torrent:
  - HTTP/HTTPS GET
  - Tham số:
    - info_hash: Mã hash duy nhất đại diện cho torrent.
    - peer_id: Một mã định danh duy nhất cho peer.
    - port: Cổng mà peer đang lắng nghe cho các kết nối đến.
    - uploaded, downloaded, left: Thông tin về số lượng dữ liệu đã tải lên, tải xuống và còn lại.
    - event: Thông báo về trạng thái của peer (chẳng hạn, started, stopped, completed).
- Bước 2: đợi phản hồi từ tracker.

SOCKET_CONECTION
- Bước 1: tạo kết nối TCP bằng socket
  - Socket (TCP)
  - Tham số:
    - Port: cổng kết nối.
    - IP: địa chỉ IP của peer.
- Bước 2: tiến hành trao đổi thông tin dựa trên Peer Wire Protocol:
  - Handshake: xác định xem hai peer có chia sẻ cùng một tệp torrent hay không.
  - Gửi thông điệp (Messages): mỗi thông điệp bao gồm: 4 byte đầu tiên: cho biết kích thước của thông điệp; 1 byte tiếp theo: xác định ID thông điệp và cuối là nội dung

thông điệp. Sau khi handshake thành công, các peer có thể gửi nhiều loại thông điệp khác nhau:
  - Choke: Thông báo rằng peer không muốn nhận dữ liệu từ peer kia.
  - Unchoke: Cho phép peer gửi dữ liệu.
  - Interested: Thông báo rằng peer muốn nhận dữ liệu từ peer kia.
  - Not Interested: Thông báo rằng peer không quan tâm đến dữ liệu từ peer kia.
  - Have: Thông báo rằng peer đã tải xuống một phần tệp.
  - Bitfield: Cung cấp thông tin về các phần mà peer đã tải xuống.
  - Request: Yêu cầu một phần dữ liệu cụ thể từ peer.
  - Piece: Gửi một phần dữ liệu mà peer đã yêu cầu.
- Bước 3: kết thúc trao đổi thông tin. Đóng socket.

2.2.2. Download

UPLOAD_TORRENT_FILE

- Bước 1: Chọn "upload torrent" và tải lên file torrent từ thiết bị
- Bước 2: Kết nối với tracker bằng ANNOUNCE_TO_TRACKER() để lấy danh sách các peer.
- Bước 3: Tiến hành thiết lập kết nối SOCKET_CONNECTION() với peer trong danh sách peer.

CREATE_HANDSHAKE

- Bước 1: đã thiết lập kết nối với seeder.
- Bước 2: gửi handshake đến seeder.
  - Gồm các tham số sau:
    - pstrlen:Độ dài của chuỗi giao thức. Luôn là 19 (đối với chuỗi "BitTorrent protocol").
    - pstr: Chuỗi giao thức. Giá trị: "BitTorrent protocol" (19 byte).
    - reserved: 8 byte dành cho các mục đích mở rộng (thường là 0 trong các phiên bản hiện tại).
    - info_hash: Mã hash SHA-1 của tệp torrent mà peer đang chia sẻ. Kích thước: 20 byte.
    - peer_id: Định danh duy nhất của peer, thường được tạo ngẫu nhiên. Kích thước: 20 byte.
  - Gửi thông điệp handshake bằng socket.
- Bước 3: chờ phản hồi HANDSHAKE_RESPONSE()

INTERESTED

- Bước 1: Nhận được HANDSHAKE_RESPONSE() từ seeder.
- Bước 2: Gửi thông điệp interested
  - Thông Điệp Interested: Thông điệp "interested" là một tín hiệu để cho biết rằng leecher sẵn sàng nhận dữ liệu từ seeder.
  - Cấu Trúc Thông Điệp: Thông điệp "interested" thường không cần bất kỳ tham số nào, chỉ cần một chỉ báo đơn giản để xác nhận rằng leecher có hứng thú.
  - Sử dụng socket để gửi thông điệp "interested" đến seeder.
- Bước 3: chờ thông điệp từ seeder.

REQUEST_PIECE

- Bước 1: nhận được bitfield và thông điệp seeder đã unchoke.
- Bước 2: gửi yêu cầu mảnh dữ liệu
  - Gồm các tham số:

  - pieceIndex: Chỉ số của mảnh dữ liệu mà leecher yêu cầu.
  - begin: offset bắt đầu.
  - requestLength: độ dài của mảnh.
  - Sử dụng socket để gửi thông điệp yêu cầu mảnh đến seeder.
- Bước 3: chờ phản hồi từ seeder.

HANDLE_PIECE

- Bước 1: Nhận và xử lí dũ liệu block: Nhận dữ liệu block từ peer và sao chép vào đúng vị trí trong piece đang được tải.
- Bước 2: Theo Dõi Tiến Trình Tải Xuống: Theo dõi lượng dữ liệu đã tải cho piece hiện tại và hiển thị tiến trình (theo phần trăm) trên console.
- Bước 3: Kiểm Tra Hoàn Thành Piece: Khi tất cả các block của piece đã được tải về, kiểm tra tính toàn vẹn của dữ liệu bằng cách so sánh mã băm của piece với mã băm đã lưu trong file torrent.
- Bước 4: Xác Minh Tính Toàn Vẹn và Lưu Piece:
  - Tính toán và so sánh mã băm SHA-1 của piece với mã băm gốc để xác minh tính toàn vẹn.
  - Nếu piece hợp lệ, lưu vào tệp và báo thành công; nếu không, báo lỗi do không khớp mã băm.
- Bước 5: Yêu cầu thêm block: Nếu piece chưa tải xong, yêu cầu thêm các block chưa nhận từ peer để hoàn thành tải xuống.

HANDLE_FILE

- Bước 1: Khởi tạo quá trình tải:
  - Xử lý bất đồng bộ khi thực hiện quá trình tải tệp từ các peer trong torrent.
  - Khởi tạo các worker (mỗi worker tương ứng với một peer) để tải các phần tệp song song.
- Bước 2: Theo dõi quá trình tải: Mỗi 5 giây, in ra tiến độ tải của các phần tệp đã hoàn thành bằng cách tính toán tỷ lệ phần trăm so với tổng số phần tệp
- Bước 3: Chờ hoàn thành tải: chờ tất cả các worker hoàn thành công việc tải tệp. Sau khi tất cả các worker hoàn thành, dừng việc theo dõi tiến trình.
- Bước 4: Kiểm tra tải hoàn tất: Kiểm tra xem tất cả các phần tệp đã được tải về đầy đủ chưa. Nếu không, ném ra lỗi.
- Bước 5: Ghép và lưu tệp:
  - Ghép tất cả các phần đã tải về thành tệp hoàn chỉnh.
  - Trích xuất tên tệp từ dữ liệu torrent, làm sạch tên tệp và lưu tệp vào thư mục đích.
- Bước 6: Xử lí kết quả:
  - Nếu tải xuống thành công, trả về thông tin về tệp đã tải xuống.
  - Nếu có lỗi, dừng theo dõi tiến trình và báo lỗi.

2.2.3. Upload

GENERATE_TORRENT_FILE

- Bước 1: Người dùng chọn file cần chia sẻ bằng cách nhấn vào nút “Upload file”.
- Bước 2: Tạo File .torrent:
  - Khi file được tải lên, client sẽ tạo một file .torrent cho tài liệu được chọn.
  - File .torrent sẽ chứa các thông tin sau:
    - info_hash: Mã hash duy nhất đại diện cho tài liệu.
    - file_size: Kích thước của file.
    - tracker_url: URL của tracker mà peer sẽ thông báo.

  - piece_length: Kích thước mỗi phần trong torrent.
  - file_name: Tên file cần chia sẻ.
- Bước 3: Lưu File .torrent: File .torrent được lưu vào bộ nhớ của thiết bị.
- Bước 4: Trở Thành Seeder
  - Thông Báo Cho Tracker: Sau khi tạo và lưu file .torrent, client sẽ trở thành seeder. Để thực hiện điều này, client cần gửi yêu cầu đến tracker bằng hàm ANNOUNCE_TO_TRACKER().

HANDSHAKE_RESPONSE
- Bước 1: Seeder đã kết nối với leecher qua socket và nhận được yêu cầu HANDSHAKE_REQUEST().
- Bước 2: Kiểm Tra Thông Tin Yêu Cầu
  - Phân Tích Yêu Cầu: Đọc thông tin trong yêu cầu handshake từ leecher. Các thông tin cần kiểm tra có thể bao gồm:
    - info_hash: Mã hash duy nhất của torrent.
    - peer_id: Mã định danh duy nhất của leecher.
    - protocol: Phiên bản giao thức hoặc thông tin khác liên quan đến kết nối.
  - Kiểm Tra Tính Hợp Lệ:
    - So sánh thông tin info_hash và peer_id nhận được từ leecher với thông tin mà seeder đang chia sẻ.
    - Nếu thông tin không khớp, từ chối kết nối và không gửi phản hồi.
- Bước 3: Gửi Phản Hồi Handshake
  - Tạo Phản Hồi Handshake: Nếu thông tin hợp lệ, seeder sẽ tạo một phản hồi handshake chứa:
    - info_hash: Để xác nhận lại mã hash của torrent.
    - peer_id: Để xác nhận lại mã định danh của seeder.
    - trạng thái kết nối: Có thể là thông tin về trạng thái hoặc quyền truy cập vào dữ liệu.
  - Sử dụng socket để gửi phản hồi handshake lại cho leecher.
- Bước 4: đợi thông điệp từ leecher.

SEND_BITFIELD
- Bước 1: Seeder nhận được thông điệp INTERESTED().
- Bước 2: Gửi thông điệp Bitfield
  - Seeder sẽ tạo một mảng bit (bitfield) để đại diện cho trạng thái của từng phần trong file torrent. Mỗi bit trong mảng này sẽ tương ứng với một phần trong file:
    - 1: Seeder đang sở hữu phần đó.
    - 0: Seeder không sở hữu phần đó.
  - Chuyển đổi mảng bit thành định dạng mà leecher có thể hiểu (ví dụ: buffer hoặc chuỗi).
  - Sử dụng socket để gửi thông điệp bitfield đến leecher.
- Bước 3: tiếp tục để unchoke cho cho leecher.

SEND_UNCHOKE
- Bước 1: đã hoàn tất handshake và gửi bitfield.
- Bước 2: gửi thông điệp Unchoke
  - Thông Điệp Unchoke: Thông điệp "unchoke" là một tín hiệu để cho biết rằng seeder sẽ bắt đầu gửi dữ liệu cho leecher. Điều này cho phép leecher bắt đầu tải xuống các phần (pieces) của file từ seeder.

- Cấu Trúc Thông Điệp: Thông điệp "unchoke" thường không cần bất kỳ tham số nào, chỉ cần một chỉ báo đơn giản để xác nhận rằng seeder đã sẵn sàng.
- Sử dụng socket để gửi thông điệp "unchoke" đến leecher.

- Bước 3: đợi thông điệp yêu cầu từ leecher.

SEND_PIECE

- Bước 1: nhận được yêu cầu từ leecher để tải một mảnh dữ liệu cụ thể.
- Bước 2: Tạo mảnh dữ liệu
  - Từ các thông tin của REQUEST_PIECE(): pieceIndex, begin và requestLength, tính toán và lấy ra đúng phần dữ liệu.
- Bước 3: seeder sẽ tiến hành gửi mảnh dữ liệu đến leecher
  - Hàm này có các tham số cần thiết như sau:
    - pieceIndex: Chỉ số của mảnh dữ liệu mà leecher yêu cầu.
    - begin: offset bắt đầu.
    - pieceData: Dữ liệu của mảnh mà seeder sẽ gửi.
  - Sử dụng Socket để gửi dữ liệu.
- Bước 4: đợi thông điệp yêu cầu từ leecher.