ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

MÔN: MẠNG MÁY TÍNH NÂNG CAO

ĐỀ TÀI: Xây dựng một ứng dụng Bittorent-like theo kiến trúc peer-to-peer để download và upload các phần của một file

SINH VIÊN 1: 1352010 – Lê Tất Đăng

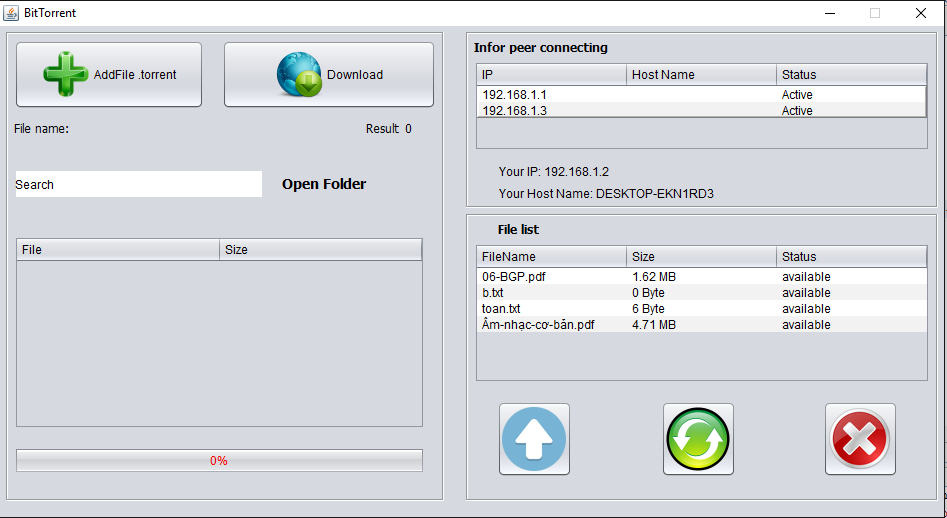
SINH VIÊN 2: 1312552 – Nguyễn Ngọc Thiện

SINH VIÊN 3: 1412558 – Phạm Quốc Toàn

GVLT: Lê Ngọc Sơn

GVTH: Lê Giang Thanh

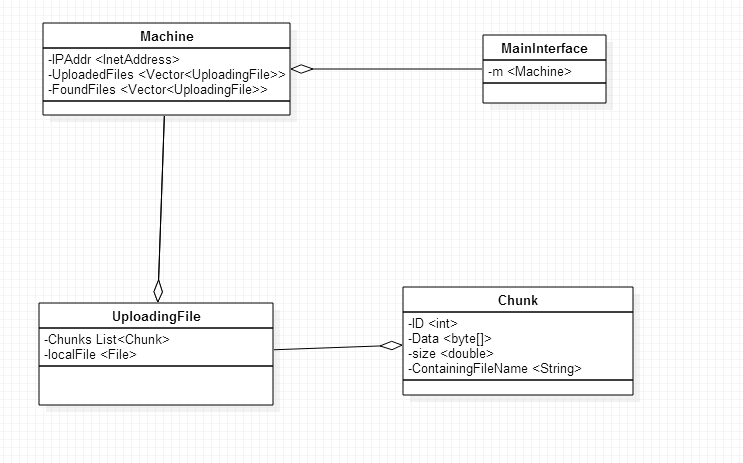
1. **Giới thiệu chương trình**



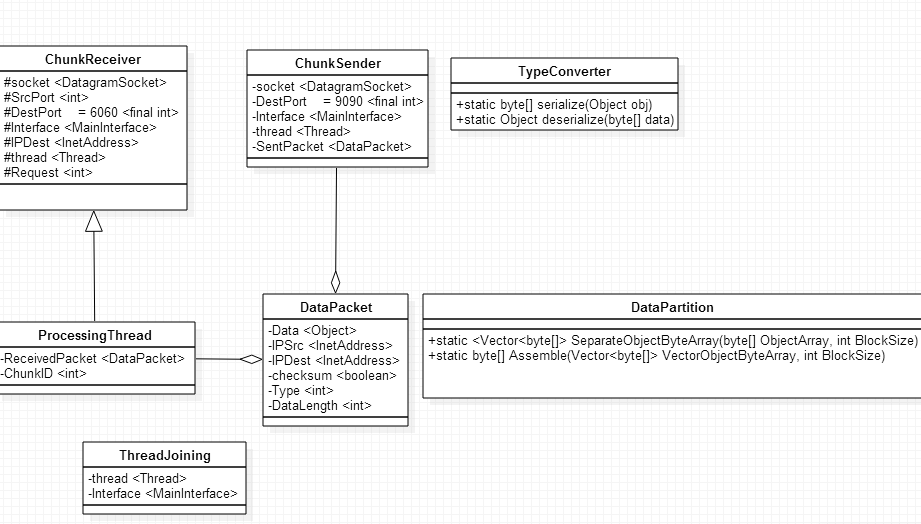
Phần mềm được tạo và vận hành theo cơ chế Peer-to-Peer (P2P), mỗi máy sẽ là một Peer và được kết nối tới các máy còn lại thông qua Socket. Phần mềm gửi nhận gói tin với giao thức UDP (truyền không tin cậy).

Phần mềm có các chức năng : upload/download file, xóa file tại local, refresh để tạo lại torrent, tìm kiếm file theo tên và theo torrent.

1. **Thiết kế chương trình (mô hình lớp hướng đối tượng)**
2. **Lõi chương trình**



1. **Xử lý thuật toán**



1. **Cơ chế Bittorrent**
2. **Request (tín hiệu quy ước)**

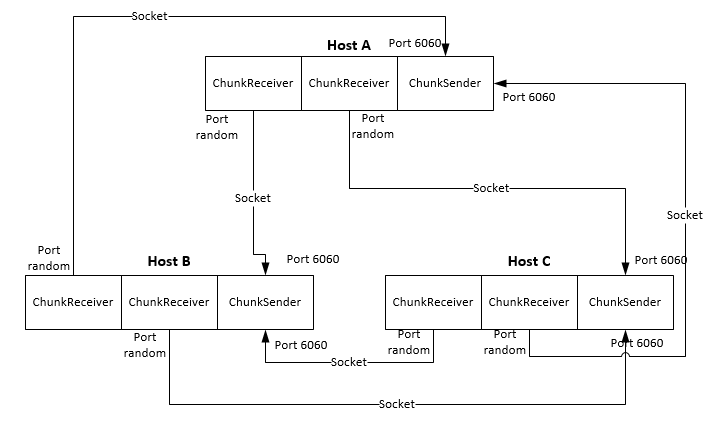
Request : yêu cầu (tín hiệu) mà máy này gửi đến máy khác để thực thi một tác vụ. Các yêu cầu này ở dạng số nguyên, và được quy ước sẵn giữa các máy.

* 1: tìm kiếm file theo tên
* 2: tìm hiếm file theo mã torrent
* 3: download chunk

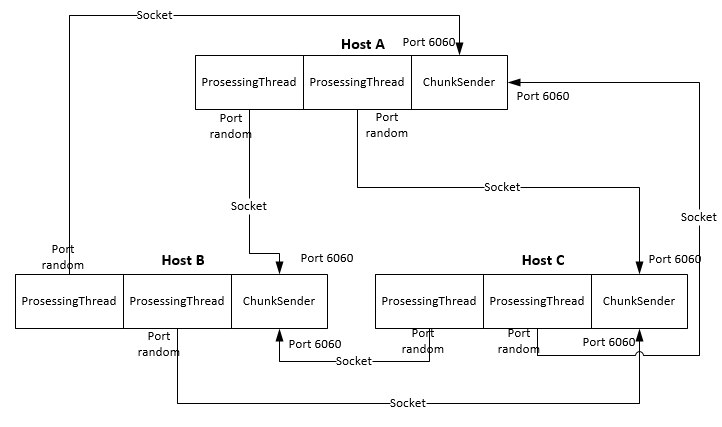
1. **Thuật toán kết nối P2P**

**ChunkSender** : nhận và thực hiện request từ ChunkReceiver/ProcessingThread, sau đó gửi trả lại kết quả cho ChunkReceiver/ProcessingThread của máy gửi request.

**ChunkReceiver** : gửi request 1 hoặc 2 cho ChunkSender của máy khác, nhận lại kết quả tìm kiếm từ ChunkSender rồi sau đó cập nhật lên bảng Process của giao diện.

 *Minh họa kết nối giữa 3 máy khi thực hiện request 1 và 2*

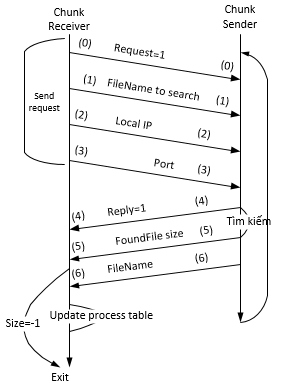
**ProcessingThread** : kế thừa từ ChunkReceiver, gửi request 3 rồi sau đó nhận lại chunk.



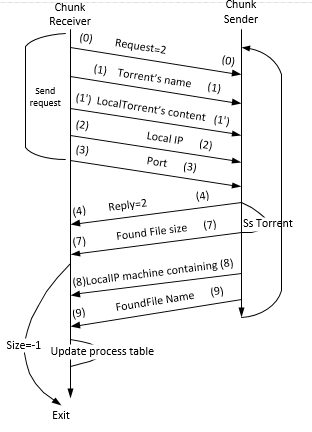
*Minh họa kết nối giữa 3 máy khi thực hiện request 3*

1. **Nguyên lý chương trình**

* Quy trình cho thực hiện request 1

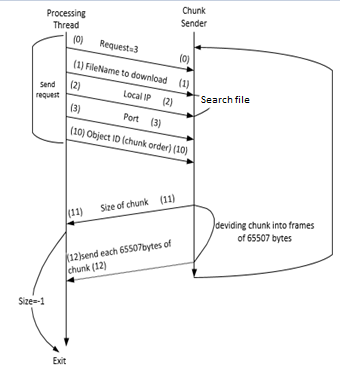


*Hình minh họa khi 1 ChunkReceiver thực hiện request 1*

* Quy trình thực hiện request 2

*Hình minh họa mỗi ChunkReceiver thực hiện request 2*

* Quy trình thực hiện request 3

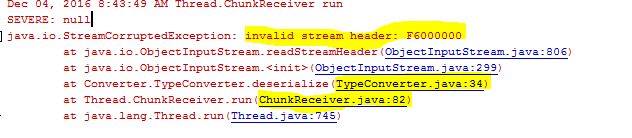


*Hình minh họa mỗi ProcessingThread thực hiện request 3*

1. **Khó khăn đã trải qua**

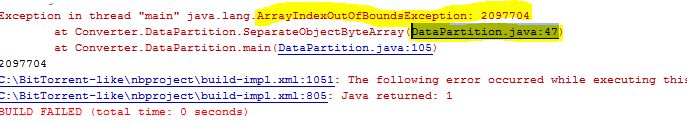
Sau đây là các lỗi bí ẩn nhất mà nguyên nhân của nó chỉ được phỏng đoán hoặc không thể tìm ra, chứ không được khẳng định chắc chắn. Do đó, đối với chúng, nhóm chỉ có thể lập trình theo cách tránh đi chứ không thể khắc phục ngay tại nguyên nhân. Cũng chính vì vậy mà bản thiết kế UML bị thay đổi khá nhiều so với ban đầu, và có những chi tiết không thể triển khai được như : lớp DataPacket (đóng gói dữ liệu một chunk để gửi đi) trong UML xử lý thuật toán không được dùng vào trong đồ án.

* **Lỗi Invalid stream header**

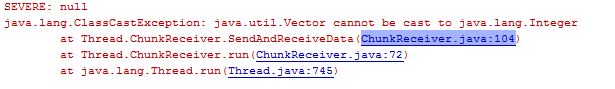


* Hoàn cảnh : máy này gửi qua máy kia một Object là Chunk, hoặc Vector<UploadingFile>. Object này phải được chuyển thành mảng byte để chuyển sang máy có yêu cầu request.
* Nguyên nhân : Dòng dữ liệu (mảng byte) từ máy này sang máy kia không còn nguyên vẹn ở máy nhận (1 hay nhiều byte đã bị thay đổi giá trị).
* Giải pháp : nhận biết được khi gửi một object do người LT tạo ra, thì dữ liệu luôn bị thay đổi ở máy nhận nên nhóm quyết định không gửi những object do mình tạo ra nữa.

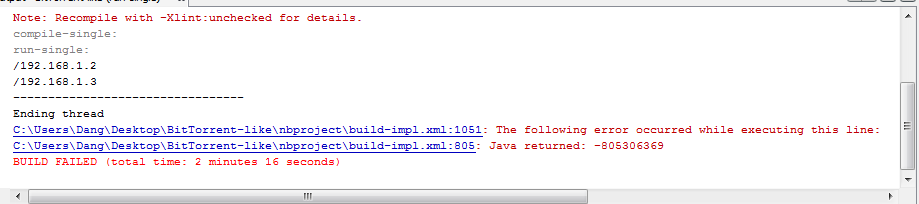
* **Lỗi ArrayIndexOutOfBoundsException : 2097704**



* Hoàn cảnh : khi phân tách một mảng byte[] lớn thành các mảng byte[] nhỏ, có 1 block là 1024 bytes *(Vector<byte[]> SeparateObjectByteArray(byte[] ObjectArray)).*
* Nguyên nhân : được cho là khi phải phân tách mảng lớn với hơn 2 triệu bytes, người LT đã không cẩn thận để truy xuất vào vùng nhớ bên ngoài mảng byte (out of bound).
* Giải pháp : Viết lại toàn bộ hàm *SeparateObjectByteArray* cẩn thận hơn và sử dụng hàm System.arraycopy có sẵn của hệ thống để sao chép nội dung mảng thay vì phải tự làm tính năng này.
* **Lỗi casting Vector to Integer**



* Hoàn cảnh : Khi máy thật muốn tìm kiếm hay download file từ máy ảo. Tuy nhiên máy ảo thực hiện các tác vụ này thì không hề xảy ra lỗi trên.
* Nguyên nhân : không rõ.
* Giải pháp : chỉ cho máy thật chạy chương trình lên nhưng không được phép thực hiện tác vụ nào trong 3 request. Chỉ có máy ảo gửi request đến máy thật và các máy ảo khác.
* **Lỗi Java returned : -805306369**



* Hoàn cảnh : Sau khi tìm kiếm file bằng torrent thành công và nhấn nút Download.
* Nguyên nhân : không rõ.
* Giải pháp : đợt thừ thất bại, nên tắt đi chạy lại.

1. **Việc chưa hoàn thành và lỗi còn vấp phải**
2. **Việc chưa hoàn thành**

* Mức 3: Phục hồi download chunk nếu đứt kết nối.
* Đảm bảo tính tin cậy: chưa xử lý khi có trường hợp gói tin bị mất, và bên nhận không nhận được.
* Kiểm soát tắc nghẽn.
* Tối ưu hóa chương trình: bỏ những chi tiết hay xử lý thừa và gây nhập nhằng trong mã nguồn, cải tiến xử lý trong mã nguồn.

1. **Lỗi**

* Có lần chạy trương trình thì download file thành công, nhưng có lần download không thành công. Nguyên do được cho là các gói tin (65507 bytes / frame) được gửi quá nhanh và liên tục nên bên nhận không nhận được kịp. Và cơ chế UDP không giúp tránh khỏi điều này.