**PROJECT 1**

# Quy định

* Số lượng thành viên 1 nhóm: tối đa 3 sinh viên, không cần đăng ký trước.
* Ngôn ngữ sử dụng: Java
* IDE: Netbean, Eclipse

# Yêu cầu

Sinh viên cài đặt các chức năng sau:

* Một ứng dụng Bittorent-like theo kiến trúc peer-to-peer để download và upload các phần của một file
* Cơ chế đảm bảo tính tin cậy cho giao thức tầng ứng dụng
* [Nâng cao] Cơ chế kiểm soát tắt nghẽn
* [Nâng cao] Cơ chế cân bằng tải để tối ưu hoá thời gian truyền file.

# Cách nộp bài

* Một sinh viên trong nhóm đại diện nhóm nộp bài trực tiếp trên moodle theo link “BT2-Link nộp bài – Lần [x]”
* Lưu bài với tên BT2\_Lan[x]\_MSSV1\_MSSV2\_MSSV3.[zip|rar]
* Cấu trúc bài nộp:
  + **Bao Cao**: chứa báo cáo về bài làm của mình
  + **Release**: chứa file thực thi của chương trình
  + **Source**: chứ source code của chương trình

# Tiêu chí đánh giá

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên kết quả** | **Tỉ lệ điểm** | **Ghi chú** |
| 1 | Gởi gói tin broadcast |  | Mức 1: gởi n gói tin  Mức 2: dùng cây khung nhỏ nhất |
| 2 | Tìm vị trí lưu các chunks |  |  |
| 3 | Download các chunks |  | Mức 1: download tuần tự các chunks  Mức 2: download đồng thời các chunks  Mức 3: phục hồi download 1 chunk nếu máy chứa chunk đó bị disconnected |
| 4 | Đảm bảo tính tin cậy |  |  |
| 5 | Kiểm soát tắt nghẽn |  |  |

# Nội dung

Cài đặt một ứng dụng truyền file **Bittorent-like** dựa trên giao thức truyền file Bittorent Peer-to-Peer. Ứng dụng này thực thi trên nền giao thức UDP. Sinh viên phải cài đặt tính tin cậy tương tự giao thức TCP cho giao thức tầng ứng dụng.

Ứng dụng này cho phép một máy có thể download đồng thời nhiều phần [chunk] của một file hay nhiều file từ các máy khác nhau.

**Mô tả sơ nét Bittorent**

Bittorent chia một file thành nhiều phần [chunk] khác nhau. Một chunk có thể lưu ở một hay nhiều máy khác nhau và client không biết trước vị trí lưu trữ của một chunk ở đâu. Khi client muốn download một file, client gởi một câu truy vấn để xác định vị trí lưu trữ file và chọn 1 nơi để gởi yêu cầu download file. Các chunk của 1 file có thể được download độc lập với nhau và khi client download đủ các chunk thì sẽ tái lắp ghép lại thành file ban đầu.

Bittorent sử dụng 1 “tracker” tập trung để lưu thông tin các chunk của một file được lưu ở máy nào. Khi một client muốn download 1 file, client phải download được file .torrent, trong đó liệt kê thông tin mỗi chunk của file. Mỗi chunk được định danh bằng 1 giá trị băm trên nội dung của nó. Để download 1 chunk, client kết nối trực tiếp đến server chứa chunk đó và thực hiện download. Sau khi client download 1 chunk, client tính toán giá trị băm trên chunk down được với giá trị băm lưu trữ trong file .torrent để kiểm tra client download đúng hay không.

Để download 1 chunk, client nhận được từ tracker danh sách các máy chứa chunk đó sau đó client kết nối trực tiếp đến một trong các máy chứa chunk đó để download.

**Một số yêu cầu đặc biệt trong Bittorent-like:**

* Không dùng tracker. Client sẽ gởi broadcast để xác định thông tin các chunk lưu trữ ở đâu. Sinh viên cài đặt thuật toán broadcast gói tin dựa trên thông tin topology của mạng.
* Sử dụng giao thức UDP để truyền/nhận các chunk của file và cài đặt cơ chế đảm bảo tính tin cậy cho giao thức tầng ứng dụng

**Ghi chú:**

* Mỗi peer biết được thông tin các peer khác trong hệ thống. Thông tin này, sinh viên tổ chức dưới dạng 1 file và peer tự động load thông tin này tại thời điểm thực thi hoặc có thể update thông tin này khi cần.

# Gợi ý

# Cấu trúc file lưu trữ

File chứa thông tin các node [nodes.map]: gồm N + M + 1 dòng

* Dòng đầu tiên lưu số N: số node trong mạng
* N dòng tiếp theo: mỗi dòng là thông tin của 1 peer theo cấu trúc [id] [ip address] [port]
* M dòng tiếp theo lưu mối quan hệ giữa các node theo cấu trúc [id\_node\_1] [id\_node\_2] [cost] cho biết node 1 và node 2 có thể gởi dữ liệu cho nhau và tốn chi phí là cost

File lưu thông tin các chunks của 1 file [filename.chunk]: gồm N dòng tương ứng với file có N chunks. Mỗi dòng lưu thông tin chunk thứ i của file và giá trị băm của chunk i là hash\_value: [i] [hash\_value]

# Cấu trúc các gói tin

Cấu trúc chung của 1 gói tin trao đổi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Unuse | Header Length | Packet Length | checksum |
| Sequence Number | | | Acknowledge Number | |
| Option | | | | |
| Data | | | | |

Trong đó:

|  |  |  |
| --- | --- | --- |
| **Trường** | **Số byte** | **Ghi chú** |
| Type | 1 | Loại gói tin |
| Unuse | 1 | Không dùng |
| Header Length | 2 | Kích thước header |
| Packet Length | 2 | Kích thước gói tin |
| Checksum | 2 | Kiểm tra tính đúng đắn của dữ liệu |
| Sequence Number | 4 | Vị trí byte dữ liệu đầu tiên (dùng cho kiểm tra tính tin cậy) |
| Acknowledge Number | 4 | Báo vị trí byte cuối cùng đã nhận (dùng cho kiểm tra tính tin cậy) |
| Option |  | Dùng để chèn thêm thông tin nếu cần |
| Data |  | Dữ liệu của gói tin  Nếu là gói tin điều khiển, thì phần này lưu thông tin của gói tin điều khiển  Tham khảo: ICMP |