**Bittorrent Protocol Specification v1.0**

**Định danh (Identification)**

BitTorrent là một giao thức chia sẻ tệp ngang hàng (P2P) do Bram Cohen thiết kế. Bạn có thể truy cập trang của ông tại <http://www.bittorrent.com>. BitTorrent được thiết kế để hỗ trợ chuyển giao tệp **giữa nhiều peer qua các mạng không tin cậy.**

**Mục đích (Purpose)**

Mục đích của tài liệu này là ghi lại phiên bản 1.0 của đặc tả giao thức BitTorrent. Trang protocol specification page của Bram mô tả giao thức này ở mức độ khái quát và thiếu chi tiết hành vi trong một số khu vực. Hy vọng rằng tài liệu này sẽ trở thành một đặc tả chính thức, được viết bằng các thuật ngữ rõ ràng, không mơ hồ, và có thể được sử dụng làm cơ sở cho các cuộc thảo luận và triển khai trong tương lai.

Tài liệu này được duy trì và sử dụng bởi cộng đồng phát triển **BitTorrent**. Mọi người đều được mời đóng góp cho tài liệu này, với hiểu biết rằng nội dung ở đây đại diện cho giao thức hiện tại, đã được triển khai trong một số phiên bản client.

**Phạm vi (Scope)**

Tài liệu này áp dụng cho phiên bản đầu tiên (tức là phiên bản 1.0) của đặc tả giao thức BitTorrent. Hiện tại, nó áp dụng cho cấu trúc tệp torrent, giao thức ngang hàng (peer wire protocol), và đặc tả giao thức HTTP/HTTPS của Tracker. Khi các phiên bản mới của mỗi giao thức được định nghĩa, chúng nên được chỉ định trên các trang riêng biệt của chúng, không phải ở đây.

**Quy ước (Conventions)**

Trong tài liệu này, một số quy ước được sử dụng để trình bày thông tin một cách ngắn gọn và rõ ràng. Nó hỗ trợ các thuật ngữ sau:

* peer so với client: Trong tài liệu này, một peer là bất kỳ client BitTorrent nào đang tham gia tải xuống. Client cũng là một peer, tuy nhiên nó là client BitTorrent đang chạy trên máy cục bộ. Người đọc của đặc tả này có thể coi mình là client, kết nối với nhiều peer.
* piece so với block: Trong tài liệu này, một piece đề cập đến một phần của dữ liệu tải xuống được mô tả trong tệp metainfo, có thể được xác minh bằng SHA1 hash. Một block là một phần của dữ liệu mà client có thể yêu cầu từ một peer. Hai hoặc nhiều block tạo thành một piece, và có thể được xác minh.
* defacto standard: Các khối lớn của văn bản in nghiêng (italics) chỉ ra một thực tiễn phổ biến trong nhiều triển khai client BitTorrent đến mức nó được coi là một tiêu chuẩn defacto.

**Bencode (Mã hóa Bencode)**

Bencoding là một cách để chỉ định và tổ chức dữ liệu ở định dạng ngắn gọn. Nó hỗ trợ các loại sau: chuỗi byte (byte strings), số nguyên (integers), danh sách (lists), và từ điển (dictionaries).

**1. Chuỗi Byte (Byte Strings):** Các chuỗi byte được mã hóa như sau: <chiều dài chuỗi được mã hóa theo ASCII> : <dữ liệu chuỗi>. Lưu ý rằng không có ký tự bắt đầu và kết thúc cố định, và không có dấu kết thúc.

Ví dụ: 4:spam biểu diễn chuỗi "spam".

Ví dụ: 0: biểu diễn chuỗi rỗng "".

**2. Số nguyên (Integers):** Các số nguyên được mã hóa như sau: i<số nguyên được mã hóa ở cơ số 10 ASCII>e. Ký tự bắt đầu và kết thúc là i và e.

Ví dụ: i3e biểu diễn số nguyên 3.

Ví dụ: i-3e biểu diễn số nguyên -3.

Lưu ý: i0e là hợp lệ. Mọi mã hóa có số 0 ở đầu như i03e là không hợp lệ, ngoại trừ i0e, là số nguyên 0.

**3. Danh sách (Lists):** Danh sách được mã hóa như sau: l<giá trị được bencoded>e. Ký tự bắt đầu là l và ký tự kết thúc là e.

Ví dụ: l4:spam4:eggse biểu diễn danh sách hai chuỗi: ["spam", "eggs"].

Ví dụ: le biểu diễn danh sách rỗng [].

**4. Từ điển (Dictionaries):** Từ điển được mã hóa như sau: d<chuỗi bencoded> <phần tử bencoded> e. Ký tự bắt đầu là d và ký tự kết thúc là e. Các khóa phải là chuỗi và xuất hiện theo thứ tự sắp xếp.

Ví dụ: d3:cow3:moo4:spam4:eggse biểu diễn từ điển {"cow" => "moo", "spam" => "eggs"}.

**Metainfo File Structure**

Tất cả dữ liệu trong tệp metainfo đều được mã hóa bằng Bencode. Đặc tả cho Bencode đã được định nghĩa ở phần trên. Nội dung của tệp metainfo (tệp kết thúc bằng “.torrent”) là một từ điển mã hóa, chứa các khóa được liệt kê dưới đây. Tất cả các giá trị chuỗi ký tự đều được mã hóa theo UTF-8.

* **info:** Một từ điển mô tả các tệp của torrent. Có hai dạng có thể có: một dạng dành cho torrent “một tệp” không có cấu trúc thư mục, và một dạng dành cho torrent “nhiều tệp” (xem chi tiết bên dưới).
* **announce:** URL thông báo của tracker (chuỗi).
* **announce-list** (tùy chọn): Đây là một phần mở rộng của đặc tả chính thức, cung cấp khả năng tương thích ngược (danh sách của các danh sách chuỗi).
* **creation date** (tùy chọn): Ngày tạo của torrent, theo định dạng UNIX epoch tiêu chuẩn (số nguyên, giây kể từ ngày 1 tháng 1 năm 1970 00:00:00 UTC).
* **comment** (tùy chọn): Các nhận xét văn bản tự do của tác giả (chuỗi).
* **created by** (tùy chọn): Tên và phiên bản của chương trình được sử dụng để tạo tệp .torrent (chuỗi).
* **encoding** (tùy chọn): Chuỗi mã hóa chỉ định định dạng mã hóa được sử dụng để tạo phần **pieces** trong từ điển **info** của tệp .torrent (chuỗi).

**1. Từ điển thông tin (Info Dictionary)**

Phần này chứa các trường chung cho cả hai chế độ, “một tệp” và “nhiều tệp”.

* **piece length:** Số byte trong mỗi khối (số nguyên).
* **pieces:** Chuỗi bao gồm sự nối của tất cả các giá trị băm SHA1 20 byte, mỗi khối một giá trị băm (chuỗi byte, không bị giải mã).
* **private** (tùy chọn): Trường này là số nguyên. Nếu được đặt thành “1”, client **chỉ** được phép công bố sự hiện diện của mình với các peer khác **chỉ** thông qua các tracker được mô tả rõ ràng trong tệp metainfo. Nếu trường này được đặt thành “0” hoặc không có, client có thể nhận peer từ các phương thức khác, ví dụ: PEX (trao đổi peer) hoặc DHT (bảng băm phân tán). Lưu ý: Có nhiều cuộc tranh luận về việc hỗ trợ tracker riêng tư. Yêu cầu chính thức cho một thay đổi đặc tả có thể tìm thấy [ở đây](http://www.bittorrent.com). **Azureus** là client đầu tiên tôn trọng các tracker riêng tư, xem wiki của họ để biết thêm chi tiết.

**a. Chế độ tệp đơn (Info in Single File Mode)**

Trong trường hợp chế độ “một tệp”, từ điển info chứa cấu trúc sau:

* **name:** Tên tệp. Đây chỉ là tham khảo (chuỗi)
* **length:** Độ dài của tệp tính bằng byte (số nguyên).
* **md5sum** (tùy chọn): Chuỗi thập lục phân 32 ký tự tương ứng với tổng kiểm MD5 của tệp. Điều này không được sử dụng bởi **BitTorrent** nhưng được bao gồm bởi một số chương trình để tương thích tốt hơn.

**b. Chế độ nhiều tệp (Info in Multiple File Mode)**

Trong trường hợp chế độ “nhiều tệp”, từ điển info chứa cấu trúc sau:

* **name:** Tên của thư mục trong đó lưu trữ tất cả các tệp. Đây chỉ là tham khảo (chuỗi).
* **files:** Một danh sách các từ điển, một cho mỗi tệp. Mỗi từ điển trong danh sách này chứa các khóa sau:
* **length:** Độ dài của tệp tính bằng byte (số nguyên).
* **md5sum** (tùy chọn): Chuỗi thập lục phân 32 ký tự tương ứng với tổng kiểm MD5 của tệp. Điều này không được **BitTorrent** sử dụng nhưng được bao gồm bởi một số chương trình để tương thích tốt hơn.
* **path:** Danh sách chứa một hoặc nhiều phần tử chuỗi đại diện cho đường dẫn và tên tệp. Mỗi phần tử trong danh sách này tương ứng với tên thư mục hoặc (trong trường hợp của phần tử cuối cùng) tên tệp. Ví dụ: tệp "dir1/dir2/file.ext" sẽ gồm ba phần tử chuỗi: "dir1", "dir2", và "file.ext".

**2. Ghi chú (Notes)**

piece length chỉ định kích thước của khối, và thường là lũy thừa của 2. Kích thước của khối được chọn dựa trên tổng số dữ liệu tệp trong torrent, và bị giới hạn bởi thực tế rằng kích thước khối quá lớn sẽ gây ra sự không hiệu quả, trong khi kích thước khối quá nhỏ sẽ làm tệp .torrent trở nên lớn.

Thực hành tốt nhất hiện tại là giữ kích thước khối dưới 512KB hoặc nhỏ hơn, cho các torrent khoảng 8-10GB, ngay cả khi điều đó dẫn đến tệp .torrent lớn hơn. Điều này giúp tạo ra một swarm hiệu quả hơn để chia sẻ tệp. Các kích thước phổ biến nhất là 256KB, 512KB, và 1MB.

Mỗi khối có độ dài bằng nhau, ngoại trừ khối cuối cùng, khối này có thể không đủ. Số lượng khối được xác định bằng cách ceil(tổng chiều dài / kích thước khối).

Đối với ranh giới của khối trong chế độ nhiều tệp, xem dữ liệu tệp như một chuỗi liên tục dài, bao gồm sự nối của từng tệp theo thứ tự được liệt kê trong danh sách **files**. Số lượng khối và ranh giới của chúng được xác định tương tự như trong trường hợp của một tệp duy nhất. Các khối có thể chồng lấn ranh giới của tệp.

Mỗi khối có một giá trị băm SHA1 tương ứng với dữ liệu chứa trong khối đó. Các giá trị băm này được nối để tạo thành giá trị **pieces** trong từ điển **info** ở trên. Lưu ý rằng đây không phải là danh sách mà là một chuỗi duy nhất. Độ dài của chuỗi này phải là bội số của 20.

**Giao thức Tracker HTTP/HTTPS (Tracker HTTP/HTTPS Protocol)**

Tracker là một dịch vụ HTTP/HTTPS phản hồi các yêu cầu HTTP GET. Các yêu cầu này bao gồm các thông tin từ client giúp tracker theo dõi thống kê chung về torrent. Phản hồi bao gồm một danh sách peer giúp client tham gia vào torrent. URL cơ bản bao gồm "announce URL" như đã được định nghĩa trong tệp metainfo (tệp .torrent). Các tham số sau đó được thêm vào URL này, sử dụng các đối số chuẩn của CGI (ví dụ: dấu ? sau URL announce, tiếp theo là các chuỗi 'param=value' được phân cách bởi dấu &).

Lưu ý rằng tất cả dữ liệu nhị phân trong URL (đặc biệt là info\_hash và peer\_id) phải được mã hóa đúng cách. Điều này có nghĩa là bất kỳ byte nào không thuộc tập ký tự 0-9, a-z, A-Z, '.', '-', '\_', và '~' phải được mã hóa theo định dạng "%nn", với nn là giá trị thập lục phân của byte đó. (Xem RFC1738 để biết chi tiết.)

**1. Các tham số yêu cầu của Tracker (Tracker Request Parameters)**

Các tham số được sử dụng trong yêu cầu GET từ client đến tracker như sau:

* **info\_hash:** Giá trị băm SHA1 chưa mã hóa, dài 20 byte của khóa info từ tệp metainfo. Lưu ý rằng giá trị này sẽ là một từ điển bencoded, như đã được định nghĩa trong phần **info** ở trên.
* **peer\_id:** Chuỗi chưa mã hóa, dài 20 byte, được sử dụng làm ID duy nhất cho client, được client tạo ra khi khởi động. Chuỗi này có thể là bất kỳ giá trị nào, và có thể là dữ liệu nhị phân. Hiện tại không có hướng dẫn cụ thể để tạo peer ID này. Tuy nhiên, người ta có thể cho rằng nó ít nhất phải là duy nhất cho máy cục bộ của bạn, do đó có thể kết hợp thông tin quá trình xử lý và có lẽ là một dấu thời gian được ghi lại khi khởi động.
* **port:** Số cổng mà client đang lắng nghe. Các cổng dự trữ cho BitTorrent thường là từ 6881 đến 6889. Client có thể chọn bỏ qua nếu không thể thiết lập cổng trong phạm vi này.
* **uploaded:** Tổng số byte đã tải lên (kể từ khi client gửi sự kiện ‘bắt đầu’ đến tracker) ở dạng cơ số 10 ASCII. Dù không được chỉ định rõ ràng trong đặc tả chính thức, sự đồng thuận là đây nên là tổng số byte đã tải lên.
* **downloaded:** Tổng số byte đã tải xuống (kể từ khi client gửi sự kiện ‘bắt đầu’ đến tracker) ở dạng cơ số 10 ASCII. Dù không được chỉ định rõ ràng trong đặc tả chính thức, sự đồng thuận là đây nên là tổng số byte đã tải xuống.
* **left:** Tổng số byte còn lại cần tải xuống để hoàn tất torrent. Giá trị này phải được gửi ở dạng cơ số 10 ASCII.
* **compact:** Đặt giá trị này là 1 để chỉ định rằng client chấp nhận một phản hồi được nén gọn. Danh sách peer sẽ được thay thế bằng một chuỗi peer, mỗi peer là một chuỗi dài 6 byte. Bốn byte đầu là host (theo thứ tự byte mạng), hai byte cuối là cổng (cũng theo thứ tự byte mạng).
* **no\_peer\_id:** Chỉ định rằng tracker có thể bỏ qua từ điển peer ID. Tham số này bị bỏ qua nếu compact được bật.
* **event:** Nếu được chỉ định, phải là một trong số started, completed, stopped, hoặc (không có tham số nào sẽ chỉ định rằng đây là yêu cầu thực hiện theo định kỳ).
* **ip:** (Tùy chọn) Địa chỉ IP thực của client.
* **numwant:** (Tùy chọn) Số lượng peer mà client muốn nhận về từ tracker.
* **key:** (Tùy chọn) Một định danh bổ sung không chia sẻ với bất kỳ peer nào khác.
* **trackerid:** (Tùy chọn) Nếu thông báo trước đó chứa một giá trị tracker id, nó sẽ được đặt tại đây.

**2. Phản hồi của Tracker (Tracker Response)**

Tracker phản hồi bằng một tài liệu "text/plain" gồm một từ điển bencoded với các khóa sau:

* **failure reason:** Nếu có, sẽ không có khóa nào khác hiện diện. Giá trị là một thông báo lỗi có thể đọc được, giải thích lý do tại sao yêu cầu thất bại (chuỗi).
* **warning message:** (mới, tùy chọn) Tương tự như failure reason, nhưng phản hồi vẫn được xử lý bình thường. Thông báo cảnh báo được hiển thị giống như lỗi.
* **interval:** Khoảng thời gian tính bằng giây mà client nên đợi trước khi gửi các yêu cầu thường xuyên đến tracker.
* **min interval:** (tùy chọn) Khoảng thời gian thông báo tối thiểu. Nếu hiện diện, client không nên thông báo lại thường xuyên hơn khoảng thời gian này.
* **tracker id:** Chuỗi mà client nên gửi lại trong các thông báo sau. Nếu không có và thông báo trước đó đã gửi một tracker id, không xóa giá trị cũ; tiếp tục sử dụng nó.
* **complete:** Số lượng peer đã hoàn tất toàn bộ tệp, tức là seeders (số nguyên).
* **incomplete:** Số lượng peer chưa hoàn tất tệp, tức là leechers (số nguyên).
* **peers:** (mô hình từ điển) Giá trị là một danh sách các từ điển, mỗi từ điển có các khóa sau:
* **peer id:** ID tự chọn của peer, như đã mô tả ở trên cho yêu cầu tracker (chuỗi).
* **ip:** Địa chỉ IP của peer, có thể là IPv6 (dạng hex) hoặc IPv4 (dạng bốn số chấm) hoặc tên DNS (chuỗi).
* **port:** Số cổng của peer (số nguyên).
* **peers:** (mô hình nhị phân) Thay vì sử dụng mô hình từ điển được mô tả ở trên, giá trị peers có thể là một chuỗi bao gồm bội số của 6 byte.

**Quy ước về 'scrape' của Tracker (Tracker 'scrape' Convention)**

Theo quy ước, hầu hết các tracker hỗ trợ một dạng yêu cầu khác, truy vấn trạng thái của một torrent (hoặc tất cả các torrent) mà tracker đang quản lý. Điều này được gọi là "trang scrape" vì nó tự động hóa quá trình "lấy dữ liệu trang" (screen scraping) vốn tẻ nhạt từ trang thống kê của tracker.

URL của scrape cũng là một yêu cầu HTTP GET, tương tự như phương pháp được mô tả ở trên. Tuy nhiên, URL cơ bản là khác nhau. Để tạo URL scrape, sử dụng các bước sau: bắt đầu với URL announce. Tìm ký tự '/' cuối cùng trong URL đó. Nếu văn bản ngay sau ký tự '/' đó không phải là 'announce', thì nó sẽ được coi là một dấu hiệu rằng tracker không hỗ trợ quy ước scrape. Nếu có, thay thế 'scrape' cho 'announce' để tìm trang scrape.

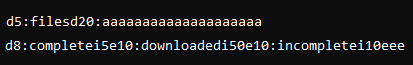
Ví dụ:

* http://example.com/announce → <http://example.com/scrape>
* http://example.com/x/announce → <http://example.com/x/scrape>
* http://example.com/announce.php → <http://example.com/scrape.php>
* http://example.com/a → (scrape không được hỗ trợ)
* http://example.com/announce?x=206644 → <http://example.com/scrape?x=206644>
* http://example.com/announce?x=2/4 → (scrape không được hỗ trợ)
* http://example.com/x%20644/announce → (scrape không được hỗ trợ)

Lưu ý đặc biệt rằng không được bỏ mã hóa thực thể (entity unquoting). Tiêu chuẩn này đã được Bram ghi chép trong kho lưu trữ danh sách phát triển **BitTorrent**: <http://groups.yahoo.com/group/BitTorrent/message/3275>.

URL scrape có thể được bổ sung thêm tham số info\_hash, một giá trị 20 byte như đã mô tả ở trên. Điều này giới hạn báo cáo của tracker chỉ cho torrent cụ thể đó. Nếu không, các thống kê cho tất cả các torrent mà tracker quản lý sẽ được trả về. Các tác giả phần mềm được khuyến khích mạnh mẽ để sử dụng tham số info\_hash bất cứ khi nào có thể, nhằm giảm tải và băng thông của tracker.

Bạn cũng có thể chỉ định nhiều tham số info\_hash cho các tracker hỗ trợ nó. Mặc dù đây không phải là một phần của đặc tả chính thức, nó đã trở thành một tiêu chuẩn de facto - ví dụ:



Phần mở rộng không chính thức cho 'scrape' (Unofficial extensions to scrape)