

RFC For 🌋🌊 (LitTorrent) V1

1 Introduction

This RFC specifies a standard for the Emoji Internet Community. Hosts on the Emoji Internet that choose to implement the LitTorrent protocol are expected to adopt and implement this standard.

LitTorrent is a mass file sharing protocol, which is kept as *LIT*eral as possible!

1.1 Address Discovery

Discovery is done through UDP Multicast.

All servers SHOULD subscribe to address 224.69.69.69 on port 42070.

(1) Client posts packet containing "🔊" (U+1F4E1) to multicast group.

(2) All clients should announce at least every 5 seconds, so that other LitTorrent applications can know when they disconnect

1.1 Ports

LitTorrent servers should listen for connections on TCP port 42069.

1.2 Request Header Format

First line of request header consists of two or more emojis, followed by carriage return and line feed. Header should be encoded in UTF-8.

(1) Protocol Version. (This document is currently on version 1)

📢 (U+1F945) = V1

📢 (U+1F948) = V2 reserved

📢 (U+1F949) = V3 reserved

(2) Method

📋(U+1F4C1) = LIST all shared files available

📁(U+1F4E5) = REQUEST a file. Followed by string containing filename

📁(U+1F485) = request a THUMBNAIL of a file. Followed by string containing filename. Server SHOULD respond with a 48x48 JPG image in an UPLOAD response

📁(U+1F4E4) = SEND a file. Followed by string containing filename - a server receiving a SEND request should create a new connection to the client's server, and REQUEST the file specified in the SEND request

🗑️ (U+1F5D1) = REMOVE a file. Followed by a string containing filename

👉 (U+1F590) = request IDENTIFIER (i.e. the name)

On further lines, zero or more parameters may be specified, each consisting of an emoji (key) followed by data (value).

Other parameters can be invented

Header MUST be terminated with another carriage return and line feed

1.3 Response Header Format

First line of response header consists of one or more emojis, followed by carriage return and line feed. Header should be encoded in UTF-8.

(1) Protocol Version, or version not supported.

🏆 (U+1F947) = V1

🏆 (U+1F948) = V2 Reserved

🏆 (U+1F949) = V3 Reserved

🚫 (U+1F6A7) = Protocol not supported. No other emojis should follow, and after sending this header the server MAY close the connection

(2) Status Code

👉 (U+1F44C) = OK.

🚫 (U+1F6B8) = NOPE. Request is rejected.

😡 (U+1F620) = ANGRY. Request is illegal.

🙄 (U+1F937) = CONFUSED. Request not understood.

🌀 (U+1F92F) = CORRUPTED or SIZE MISMATCH.

On further lines, zero or more parameters may be specified, each consisting of an emoji (key) followed by data (value).

📏 (U+1F4C0) = Size of body in bytes - Must be provided for all messages with a body (BEWARE: this is size one, when others are two)

NOTE: size should be formatted as string, e.g. "200" for 200 bytes length.

Other parameters can be invented

Header MUST be terminated with a carriage return and line feed, followed by another carriage return and line feed.

1.4 Body format

1.4.0 LIST

Single line per file:

```
"<filename>\0<size in bytes>\0<last modified>"
```

Where <last modified> is the number of milliseconds since the epoch (00:00:00 GMT, January 1, 1970).

Where "\0" is the character '\0', as opposed to "\\0"

1.4.1 IDENTIFIER

A string for the name

1.4.2 SEND

File or file segment in binary

Other commands SHOULD contain an empty body.