Computer Networks Lab

Project Abstract

Project Idea

Bit-Torrent Client Implementation

Group Members:

Manav Vagrecha (AU1841022)

Devam Shah (AU1841044)

Shreyansh Shah (AU1841046)

Under Guidance Of

Prof. Jitendra Bhatia

(jitendrabbhatia@gmail.com)

What is Bit-Torrent ?

Bit-Torrent is a peer-to-peer(P2P) content sharing protocol which works in a sort of decentralized fashion, consuming less bandwidth than traditional file sharing protocols. Earlier Bit-Torrent followed Pure P2P structure but due to various issues, the structure was changed to Hybrid P2P where the central entities are necessary for providing parts of the offered network services.

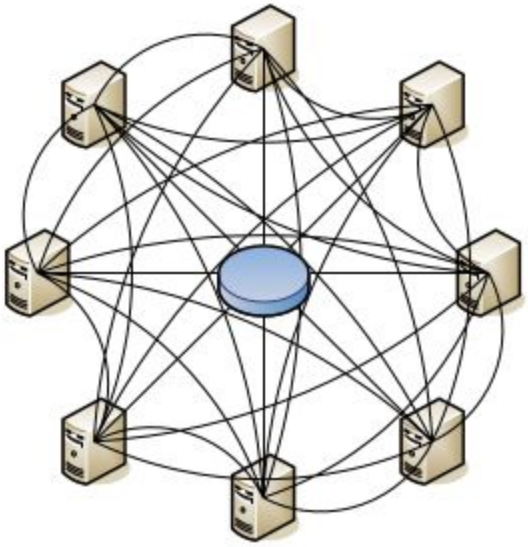

Fig 1: Structure of Bit-Torrent following hybrid P2P concept where the centralized entity is a tracker and other entities can be considered as peers

What does Bit-Torrent include ?

The Architecture of Bit-Torrent mainly includes :

- Torrent File - a static metainfo file
- Tracker - The tracker is a special server that keeps track of the connected computers. It also keeps track of where file copies reside on peer machines, which ones are available at time of the client request, and helps coordinate efficient transmission.
- Seeder - User who has a complete copy of all the files in the torrent and are connected to BitTorrent swarm for contributing so that others can download the file are known as seeder
- Peers / Leecher - User downloading from a bit-torrent swarm

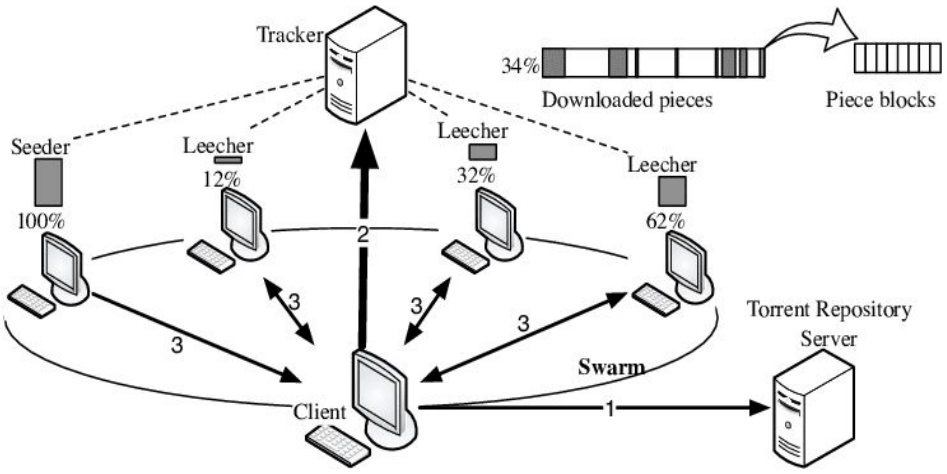

Fig 2: Architecture of BitTorrent

The concepts can be categorized into :

- Binary Encoding and Decoding
- Trackers for assisting and tracking P2P requests and responses
- Scraper for UDP or HTTP trackers
- Piece Selection Algorithms (Piece Management)
- Handling Peers for piece transfers (Peers Management)
- Peer to Peer Communications for file (Requests and Response Management)

How Bit-Torrent actually works?

- A computer joins Bit-Torrent swarm by loading a .torrent file into any bit-torrent client. Now Bit-Torrent contacts the trackers, which are specified in .torrent file and the tracker shares their IP addresses with other Bit-Torrent clients in the swarm, allowing them to connect to each other.
- Now, the Bit-Torrent client downloads bits of the files in the torrent in small pieces, downloading all the data it can get. Once the Bit-Torrent client has some data, it can then begin to upload that data to other BitTorrent clients in the swarm. This way, everyone downloading torrent is also uploading some data at the same time.
- This speeds up everyone's download speed and it doesn't put a lot of stress on a central server.
- One important thing here is that Bit-Torrent clients never actually download files from the tracker itself. The tracker participates in the torrent only by keeping track of the BitTorrent clients connected to the swarm, not actually by downloading or uploading data.

Process Flow

- Decoding the binary encoded content in the torrent file
- Separating the content from the torrent file
- Creating the file structure as mentioned in the torrent file
- HTTP / UDP Scrapping as per the links mentioned in the announcement lists. (both trackers uses different connection protocols and a different request and response format)
- The trackers responses as per the requests made by the peer.
- If any peer is found with the requested piece and ready to upload the piece then the peer-to-peer communication starts as stated in Fig 3.
    - Connecting Peers
    - Handshaking
    - There are 11 types of Message Passing before starting the file transfer : keep-alive, choke, unchoke, interested, not-interested, have, bitfield, request, piece, cancel, and port.
    - Requests by the Leecher
    - Responses by the Seeder

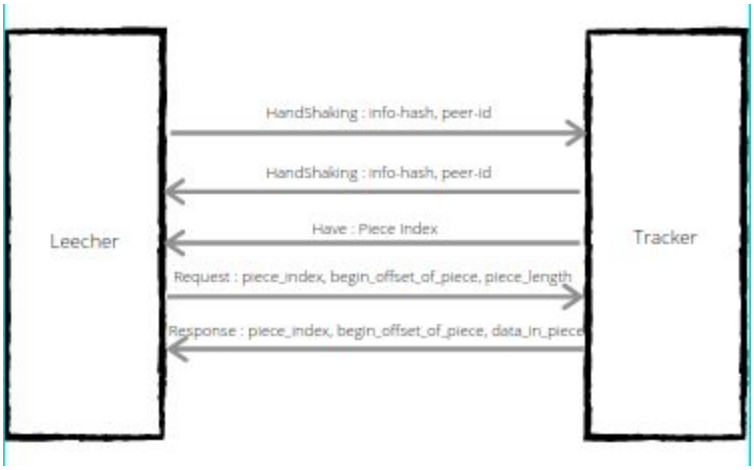

Fig 3: Tracker Communications with Leecher

Language Specification

Python

References

[1] https://wiki.theory.org/BitTorrentSpecification

[2] Implementation and analysis of Bit-Torrent Protocol

[3] Understanding BitTorrent Protocol