Report Project

Lucas Fernandez

1. Presentation

For the project I choose to build a peer-to-peer file sharing system using the BitTorrent protocol. The working is simple, the client download a torrent file on a website, the client connect to a tracker or server thanks to information contains in the torrent then the tracker send to the client a list of peers who got pieces of the file and then the client connect with peers and start downloading the file but with the BitTorrent protocol the peer or client who download pieces of a file is also uploading to other peers pieces that he already got.

2. Goal

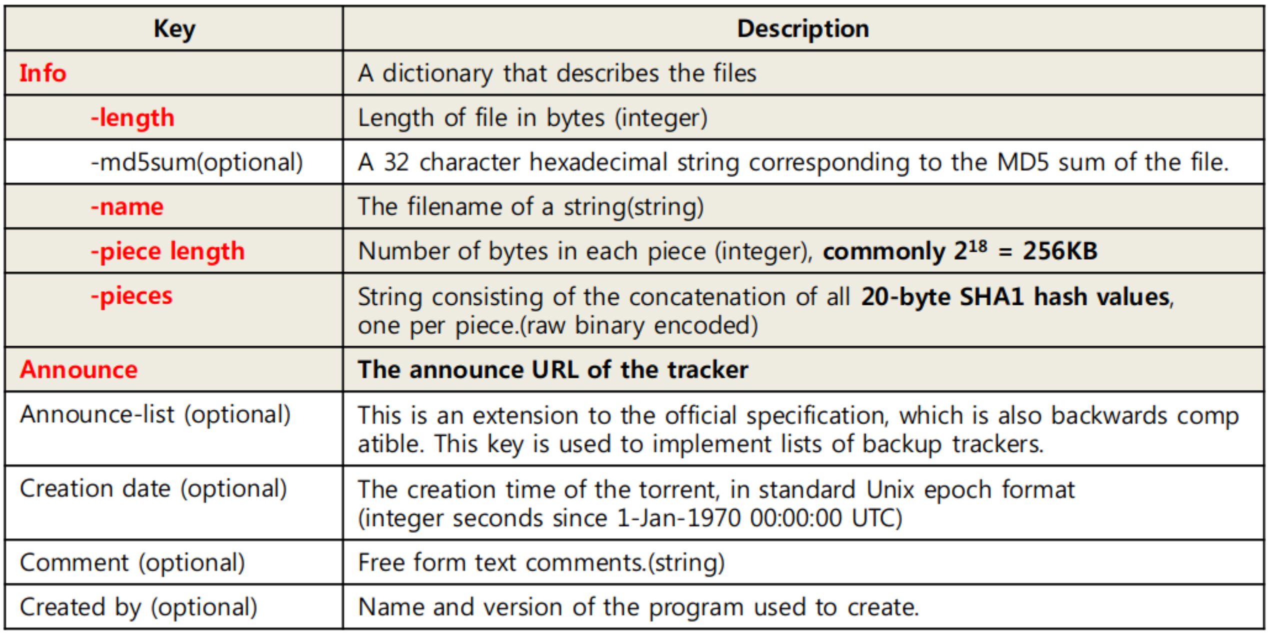
My purpose was to create a simple version of a BitTorrent client (like BitLord or µTorrent) using the BitTorrent protocol were a client could either create a new torrent from a existing file and send it to the tracker or just download a file. The project is coded in java for the graphics par I would have used the library JavaFx but due to a lack of time I could not do that last part.

3. Motivation

Why do I choose to build a peer-to-peer network ? Cause peer-to-peer represents approximately 43% to 70% of all internet traffic and is something that I use almost every to download some files and I found the technology behind this very interesting so it was a good way for me to learn more about that cause it something that we have seen a lot in classes so it was a way to develop a technology that we have overview in classes.
Why using BitTorrent protocol for creating this application and not an other one instead ? Well BitTorrent is a protocol specialize and also the most common to distributing large files and data, it also represents 3.35% of all bandwidth and it used in most of the application in file sharing.

4. Architecture

The project is divided in four main part :
- First part : The Torrent, the program create a torrent from an existing file. A Torrent is a file that contains all information about the file. Here is a picture of what can contains a torrent file. Attributes in red are those contain in the project.



Key	Description
Info	A dictionary that describes the files
-length	Length of file in bytes (integer)
-md5sum(optional)	A 32 character hexadecimal string corresponding to the MD5 sum of the file.
-name	The filename of a string(string)
-piece length	Number of bytes in each piece (integer), **commonly 2^{18} = 256KB**
-pieces	String consisting of the concatenation of all **20-byte SHA1 hash values**, one per piece.(raw binary encoded)
Announce	**The announce URL of the tracker**
Announce-list (optional)	This is an extension to the official specification, which is also backwards compatible. This key is used to implement lists of backup trackers.
Creation date (optional)	The creation time of the torrent, in standard Unix epoch format (integer seconds since 1-Jan-1970 00:00:00 UTC)
Comment (optional)	Free form text comments.(string)
Created by (optional)	Name and version of the program used to create.

- Second part : The tracker, is the server responsible of keeping track of each torrent and who have that pieces, for doing that I implement a class call 'Torrent_track' which take a torrent and a map of an array of byte representing the piece and a list of peers. Hence the tracker have a map of an array byte representing the SHA-1 hash of the filename and a torrent_track. It also have a port where or a port, since we are in local else it would take a URL, for listening and responding to the client request. We use the TCP protocol and Socket in java to connect the tracker and a client.

- Third part : The client, is use to connect to the tracker and collecting peers. The tracker chose randomly 5 pieces and send to the client the concatenated list of all peers without the duplicate.

- Fourth part : Peers, are the people who are uploading and downloading the file. When the client got the list of peers from the tracker, a new peer is created with the same id and port that the client and connect with peers contains in the list. I used Socket to do that, each peer have a client and server side to receive and sent the data. A peer become a seeder when he has downloaded all the pieces of the file.

5. Conclusion

For this project I used the Socket in java, that something that I never used before. The project still a couple of thing to be a real BitTorrent client like the Mainline DHT which is used to find peers in a purpose to have decentralized BitTorrent protocol and a graphical part so it can be more readable for a normal person but also some minor improvements like instead of using HTTP for connect the client and the tracker like it says in the documentation of BitTorrent or instead of choosing pieces randomly most of the client in the market choose pieces with the 'rarest first' technique.

6. Resources

Github project : https://github.com/lucas101199/P2Pa
Documentation of BitTorrent protocol : https://www.bittorrent.org/beps/bep_0000.html
Source of the image and helpful documentation : https://dandylife.net/docs/BitTorrent-Protocol.pdf
Wikipedia page of BitTorrent protocol : https://en.wikipedia.org/wiki/BitTorrent