

DATE: 10/25/2013  
COURSE: CSCI-P538  
YEAR: Fall 2013  
NAME: Prachi Shah(pracshah)  
PROJECT: Lab 3: BitTorrent Client Implementation

- 1) Developed by Bram Cohen, the bit torrent application enables peer-to-peer file transfer.
  - 2) It solves the problem of keeping centralized data into one location. It enables hosts to download and upload data amongst each other.
  - 3) Hosts communicate directly with each other.
  - 4) Here, hosts are both client and servers.
  - 5) Bit torrent allows downloaders to download data from much smaller number of file uploaders.
  - 6) The project focuses on implementing a 1-seeder 1-leecher BitTorrent Client.
  - 7) Clients (hosts) are downloaders/uploaders that exchange pieces of a file
  - 8) The information of the file to be downloaded will be retrieved from the Metainfo file (torrent file). The client will perform downloading/uploading of pieces of a file based on the information it has received from the metainfo file. peers can use this information to track status of pieces of a file.
  - 9) The port ip address and details of where the peices of a file are stored fare received rom the command-line arguments.
  - 10) A client will get a request form another peer directly in a peer-to-peer fashion.
  - 11) - BitTorrent clients are of 3 types:
    1. Seeder: A client that is only uploading a file.
    2. Leecher: A client that is only downloading a file.
    3. Peer: A client that is connected to other clients and is either downloading and/or uploading data.
- Swarm: All seeders, leechers, and peers actively downloading/uploading files are called Swarm.
- 12) Working:
  1. A file originator(seeder) who is willing to start sharing a file generates a metainfo file (torrent file) which contains file details.
  2. Here, the parameters of the command-line arguments will specify the leecher details (ip address and port). The leecher reads the information in metainfo file and get connection details to connect to seeder.
  3. Next, the leecher establishes a connection with the seeder, both peers do a handshake, seeder sends the details of the pieces it has to the leecher, the leecher then requests for pieces, the seeder uploads data; the leecher starts downloading the pieces until the entire file is downloaded from the seeder.
  4. Once a piece is downloaded, the leecher will announce to other peers in the swarm that it has a piece which is available for downloading to other peers.
- 13) Connection state: The client protocol maintains connection state.
- 14) Handshake:
  - Both peers perform a 3-way TCP symmetric handshake before they transfer data.
- 15) Messages: Once handshake is done, both peers share some messages with each other. The messages are choke, unchoke, interested, not interested, have, bitfiled, request, piece, cancel.
- 16) Logging: The logging functionality will log all the activities of the bit torrent application.
- 17) SHA1 Encryption is done of the data of the torrent file at both the seeder and the leecher side.

Team-spirit:

Prachi Shah:

Parsing in message sending and serialize and deserialize message data

Message sending and receiving

File uploading

File downloading

README file

Oliver Lewis:

1 seeder- 1 leecher connection

Handshake

Calculate SHA1 and necessary parsing of the node of the torrent file INFO

Logging functionality

Credits:

1) P538 Lab 3 Document.

2) Computer Networking: A Top-Down Approach (6th edition), by Jim Kurose and Keith Ross

3) TCP/IP Sockets in C: A Practical Guide for Programmers (2nd Edition), by Michael J. Donahoo and Kenneth L. Calvert

4) OpenSSL HMAC Hasing Example

<http://www.askyb.com/cpp/openssl-hmac-hasing-example-in-cpp/>

5) Glossary of BitTorrent terms:

[en.wikipedia.org/wiki/Glossary\\_of\\_BitTorrent\\_terms](http://en.wikipedia.org/wiki/Glossary_of_BitTorrent_terms)

6) BitTorrentSpecification

<https://wiki.theory.org/BitTorrentSpecification>

7) \*BitTorrent

<http://en.wikipedia.org/wiki/BitTorrent>