

## Final Project: BitTorrent

*Assigned: November 28**Due: December 14*

## 1 Description

For this project you need to implement a BitTorrent client. Successful implementations need to interoperate with commercial/open-source BitTorrent clients. Your project will be graded on its download performance compared to the reference client. Your client should have download performance comparable to or better than the reference client. You need to devise experiments to demonstrate that your client's performance is 'fast enough' and 'stable' in comparison to the reference BitTorrent client.

You will be working in groups, which you may form yourselves. Keep group sizes to 4 students). If you cannot find enough students for your group, email the TAs. When you have established your group, one member must send an email to the TAs (CC'ing other group members) which includes the full names and **directory ID** of all members. The email subject should be "[cmsc417] project group". We will create a new Git repository for your group, and reply with information on how to access it.

Along with your implementation, you must submit a report that details:

1. List of supported features
2. Design and implementation choices that you made
3. Problems that you encountered (and if/how you addressed them)
4. Known bugs or issues in your implementation
5. Contributions made by each group member

## 2 Features

### 2.1 Core Features

1. Communicate with the tracker (with support for compact format)
2. Download a file from other instances of your client
3. Download a file from official BitTorrent clients

## 2.2 Extra Credit

If you successfully implement the core features of this project, you may optionally implement one or more features below for extra credit:

1. Implement support for UDP-tracker protocol.
2. Implement optimistic unchoking (see “Choking and Optimistic Unchoking” in [1]).
3. Implement the rarest-first strategy (see “Piece Downloading Strategy” in [1]).
4. Implement an endgame mode (see “End Game” in [1]).
5. Implement an optional BitTyrant mode (see [2]).
6. Implement PropShare [4] and design experiments to compare performance to the official client.

## 3 Resources

### 3.1 Specification Information

You can find information on the BitTorrent specification in [1, 3].

### 3.2 Libraries

You are *not* allowed to make use of third-party libraries except for a bencoder/bdecoder library of your choosing. See “Implementations” in [1] for possible bencoder/bdecoder libraries.

If you use C, you can use any data structures or libraries you like, so long as they are “equivalent” to the libraries in Python in terms of power. It’s a good idea to ask, but ideally you’d have “equal power” no matter what language you choose.

You may use the SHA-1 hashing functions we provided for Assignment 4, which can be found in the ‘materials’ repository.

### 3.3 Reference Client

You can download a reference BitTorrent client for Windows/MacOS/Linux at:

<https://www.transmissionbt.com/>

You can use this client to help understand the protocol (e.g., via Wireshark/tcpdump packet captures) and for comparison in your experiments.

On Debian, you should be able to install it by running: `sudo apt-get install transmission`

### 3.4 Other Resources

1. Torrent metafile details: [https://fileformats.fandom.com/wiki/File\\_format](https://fileformats.fandom.com/wiki/File_format)
2. Sample metafiles: <https://github.com/webtorrent/webtorrent-fixtures/tree/master/fixtures>

## 4 Grading

At the end of the semester, each group will meet with the TAs to demonstrate their BitTorrent client implementation. Additionally, each group will discuss the information contained in their report (e.g., design choices) during this meeting. The TAs will make a post on Piazza with more details about scheduling the meetings closer to the deadline.

## 5 Additional Requirements

1. Your code must be submitted as a series of commits that are pushed to the origin/master branch of your team's Git repository. We consider your latest commit prior to the due date/time to represent your submission.
2. You may implement the project in any language of your choosing, minding the restrictions on libraries in §3.2.
3. You are not allowed to copy code from any source.
4. Your report must be provided as a PDF file named 'report.pdf' and placed inside the root of the repository as well.

## References

- [1] BitTorrentSpecification - Theory.org Wiki. <https://wiki.theory.org/BitTorrentSpecification>.
- [2] BitTyrant. <http://bittyrant.cs.washington.edu/>.
- [3] The BitTorrent Protocol Specification. [http://www.bittorrent.org/beps/bep\\_0003.html](http://www.bittorrent.org/beps/bep_0003.html).
- [4] Dave Levin, Katrina LaCurts, Neil Spring, and Bobby Bhattacharjee. BitTorrent is an Auction: Analyzing and Improving BitTorrent's Incentives. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 243–254. ACM, 2008.