

Final Project: BitTorrent

*Assigned: April 10**Due: May 8*

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 official client. Your client should have download performance comparable to or better than the official client. You need to devise an experiment to demonstrate that your client's performance is 'fast enough' and 'stable' in comparison to the official BitTorrent client.

You will be working in groups, which you may form yourselves. Groups may be no larger than four students. Please register your group on ELMS. We will create a new Git repository for your group, which will be called `git@gizmonic.cs.umd.edu:bittorrent/<group_name>`. You can obtain the initial files in the repository `git@gizmonic.cs.umd.edu:assignments417/bittorrent`.

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 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 written in C. You may use the SHA-1 hashing functions in the `assignments417/bittorrent` repository.

3.3 Official Client

You can download the official BitTorrent client at <http://www.bittorrent.com/>. This is the client you should use as a protocol reference (via Wireshark/tcpdump packet captures) and for comparison in your experiments.

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 ELMS with more details about scheduling the demos.

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 installed on the VM.
3. You must provide a Makefile that is included along with the code that you commit. We will run ‘make’ in the root of the repo, which must produce a ‘client’ executable also located in the root of the repo directory.
4. Your report must be provided as a PDF file named ‘report.pdf’ and placed inside the root of the repository as well.
5. You must submit code that compiles in the provided VM, otherwise your assignment will not be graded.

6. You are not allowed to copy code from any source.

References

- [1] BitTorrentSpecification - Theory.org Wiki. <https://wiki.theory.org/index.php/BitTorrentSpecification>.
- [2] BitTyrant. <http://bittyrant.cs.washington.edu/>.
- [3] The BitTorrent Protocol Specification. 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.