

cmsc 417 programming assignment #4: BitTorrent part 1

(Updated March 31)

Goal: Download (only) a file through bittorrent.

1 Deadline

DUE: Friday, April 11. The late deadline will be Monday, April 14; you have only one late deadline to play with.

2 Partner!

You MAY work with one partner for this assignment. For turnin, BOTH partner students must turn in a working implementation by the deadline. (this will simplify accounting.) Please include a text file “partners” in your turnin listing both names.

3 Description

Write a program that takes a url for a .torrent file as its first (only required) argument, then writes to a file “downloaded” all blocks of the downloaded file. When the file is completely downloaded, exit.

The (single) test will be whether the md5sum and file size of the “downloaded” file is correct for the torrent. Do not crash, do not err.

The protocol is well-documented at: <http://wiki.theory.org/BitTorrentSpecification>. A high-level overview is in the textbook page 711.

4 Danger!

Some network ISPs and network devices (NATs/firewalls) may interfere with bittorrent communication. We will run a tracker and a seeder on campus with a few test files for you to play with; they are not guaranteed to work. Like many protocols, BitTorrent evolves more than is standardized: be liberal in what you accept, conservative in what you send.

5 Future

The next deadline (4/25) will involve serving files, and the final deadline (5/9) the combination of the two with class-specific extensions. Do not delay.

The class-specific extensions to this client will incorporate multicast neighbor discovery from early programming assignments.

6 Notes

The mainline client will crash if you ask it for a block too large or for a full block off the end of the file. It doesn't seem like I can run a seeder for your leecher. A tracker is no problem: <http://scriptroute.cs.umd.edu:11417/announce>

I found the protocol relatively easy to implement, my trouble was in opening the file so that it would allow reading and writing, without truncating the file.

300 lines of Ruby. `Digest::SHA1.digest(data)`

Single-file torrents only.

I may test your client with different piece sizes, an already-present file (don't download again), and a sparse file (which I don't handle yet, but I know it's a bug).

You may download a block more than once, but don't make too much of a habit of it.

A simple torrent is at:

- <http://www.cs.umd.edu/class/spring2008/cmsc417/Jay.jpg.torrent>

That torrent describes the file:

- <http://www.cs.umd.edu/class/spring2008/cmsc417/Jay.jpg>

Which you may be able to seed by placing the file in the downloads folder of your own bittorrent client, even if my seeder dies. If this is not so exciting, create your own torrents or find those advertised by others. This one is small and should finish in less than a minute.