

Assignment 5 — BitTorrent

Assigned: Apr. 5

Due: April 19, 11:59:59 PM

1 Description

For this assignment you will implement a BitTorrent client. In this section we describe the required client functionality as well as extra bonus features that you could add to your project for bonus points. You can implement your project in language of your choice. You can get ideas from other open implementations **but do not copy code**. You should cite *code, ideas, material* that is not yours. We will run a **tracker** for you at a well known address. Along with your code you will submit a report detailing the design choices that you made, problems that you encounter, the bugs that your code has, and the features that it supports. Consider the report as an important task of this project and submit it with your code. Section 1.4 provides information about available libraries that you can use as well as pointers to websites that will help you through the assignment.

1.1 Required Functionality

Your client should support the following features:

- Communicate with a tracker in compact form. See Section 1.3 for details about the tracker.
- Downloading a file from other instances of the same client.
- Downloading a file from other teams' clients.
- Downloading a file from third party BitTorrent clients.

1.2 Bonus Functionality

You may choose to support one or more of the following features. They are listed in a difficulty-order. For each one of them you can find more information on the links of the resources Section 1.4.

1. Rarest-first strategy.
2. End-game mode.
3. Optimistic unchoking.
4. Prioritization.
5. Peer exchange.
6. Tracker-less mode

1.3 Tracker & Seeder

We will run a tracker for you. The tracker server will run at 128.8.126.105:9001. We will also provide you with three torrents, each one of them for files of size 5, 20, 102 MB respectively. We will run two seeders, one within the linux-lab and one outside the UMD firewall.

In the case your operating system is Unix-based and you can run your own tracker following the steps below:

- Install the python-twisted package. You can install via a package manager. The package name is "python-twisted".
- Download the BitTorrent source code from <http://download.bittorrent.com/dl/archive/BitTorrent-5.2.0.tar.gz>
- In order to create a torrent of a file you can use the "maketorrent-console.py" utility that you will find when you decompress the BitTorrent source.
- Execute the following command

```
nohup python bittorrent-tracker.py --port 6969 --dfile dstate --logfile tracker.log &
```

1.4 Libraries & Online Resources

There are numerous resources you can find and use for this project.

- BitTorrent Specifications Websites
 - <http://wiki.theory.org/BitTorrentSpecification>
 - http://www.bittorrent.org/beps/bep_0003.html
 - http://wiki.vuze.com/w/Main_Page. Beware that Azureus implements some extra messages that don't exist in the default version of the protocol.
- Libraries
 - You may use SHA-1 libraries as well as library implementations for bencoder and bdecoder.
 - You may **NOT** use not use a Torrent file parser that parses out the entire torrent for you. However, you can a bdecoder library in your torrent file parser.
- Open source implementations.
 - Transmission (C implementation): <http://www.transmissionbt.com/download.php>
 - BitTorrent (in Python): <http://download.bittorrent.com/dl/archive/BitTorrent-5.2.0.tar.gz>
 - Azureus (in Java): http://sourceforge.net/projects/azureus/files/vuze/vuze-4.3.1.4/Vuze_4.3.1.4_source.zip/download

2 Notes

- You should submit code that:
 1. Does not crash.
 2. Does not produce debugging output by default.
 3. Has extensive **comments**. If possible documented with a tool like Javadoc or Doxygen (<http://www.doxygen.nl>).