

# Project Description

## End-to-end publishing using Bittorrent

Student: Andrew Wang

### 1 Background

Bittorrent is a promising peer-to-peer network that allows for fast download speeds regardless of the number of peers downloading the file. Currently, separate tools exist that make .torrent files, "track" the peers downloading the file, host .torrent files, and initially upload the file. This project aims to unify this process by making an end-to-end software suite that simplifies the process of publishing a file on the Bittorrent protocol for download.

### 2 Description

Bittorrent is a relatively simple protocol, but the process of using Bittorrent is comparatively difficult. To download a file using Bittorrent, the user must first acquire a Bittorrent metadata file (usually called a "torrent" file) that contains descriptive information of the file to be downloaded, but most importantly the URL of a "tracker." The user must then pass open the torrent file with a Bittorrent client, which then queries the tracker and finally begins downloading the file.

The tracker is the only centralized part of Bittorrent, and is how the clients in the Bittorrent "swarm" become known to each other. The tracker stores the IP addresses and Bittorrent port of each client in the swarm, as well as passes this information on to other clients at a client's request. The client-tracker communication is done through client requests to certain tracker URLs with GET argument data and tracker response with a "bencoded" dictionary containing both ASCII and binary data. The "bencode" format was invented by the creator of Bittorrent, Bram Cohen, as a easy way of passing typed data over text files.

The weakness in the Bittorrent distribution method is the number of different tools that are required to distribute a file. A web server, a metadata file generator, a tracker, and an initial uploader client are the minimum requirements, and likely more than the average user can deal with. This can be simplified by combining all of these parts into an aggregate package that handles all of these things automatically, and presents itself though a simple web interface.