End-to-end Bittorrent Publication

Andrew Wang, TJHSST Computer Systems Lab 2006-2007

Abstract

End-to-end publication through Bittorrent involves creating a .torrent metadata file, setting up a central "tracker," and an initial "seed" with a complete copy of the file. This project simplifies this process by providing an unified end-to-end publication package while also addressing the primary weakness of Bittorrent..
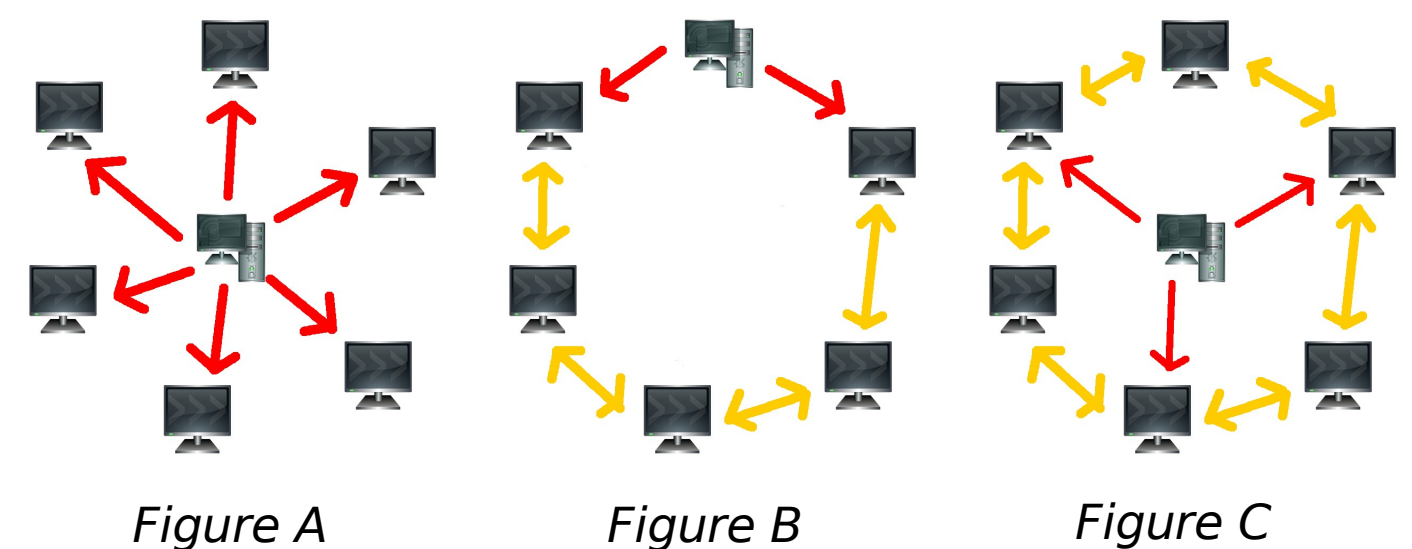
Figure A   *Figure B*   *Figure C*

Figure A describes the traditional download model, with one centralized server. Figure B describes a traditional Bittorrent download swarm. Figure C describes the Evertorrent model, combining elements of Figure A and B.

What is Evertorrent?

Evertorrent is an improvement upon the current Bittorrent publishing system. Bittorrent itself has been recognized as an efficient means of distributing large files, but it has a single weakness: if there are no clients with a complete copy of the file, the torrent "dies" because downloads can never complete.

By combining the traditional (Fig A) and Bittorrent (Fig B) models of distribution, Evertorrent (Fig C) was born. It introduces the concept of an "everseed", an initial, eternal seeder that ideally runs on the same server as the tracker. This means that the torrent will never "die" from the lack of a complete copy of the file.
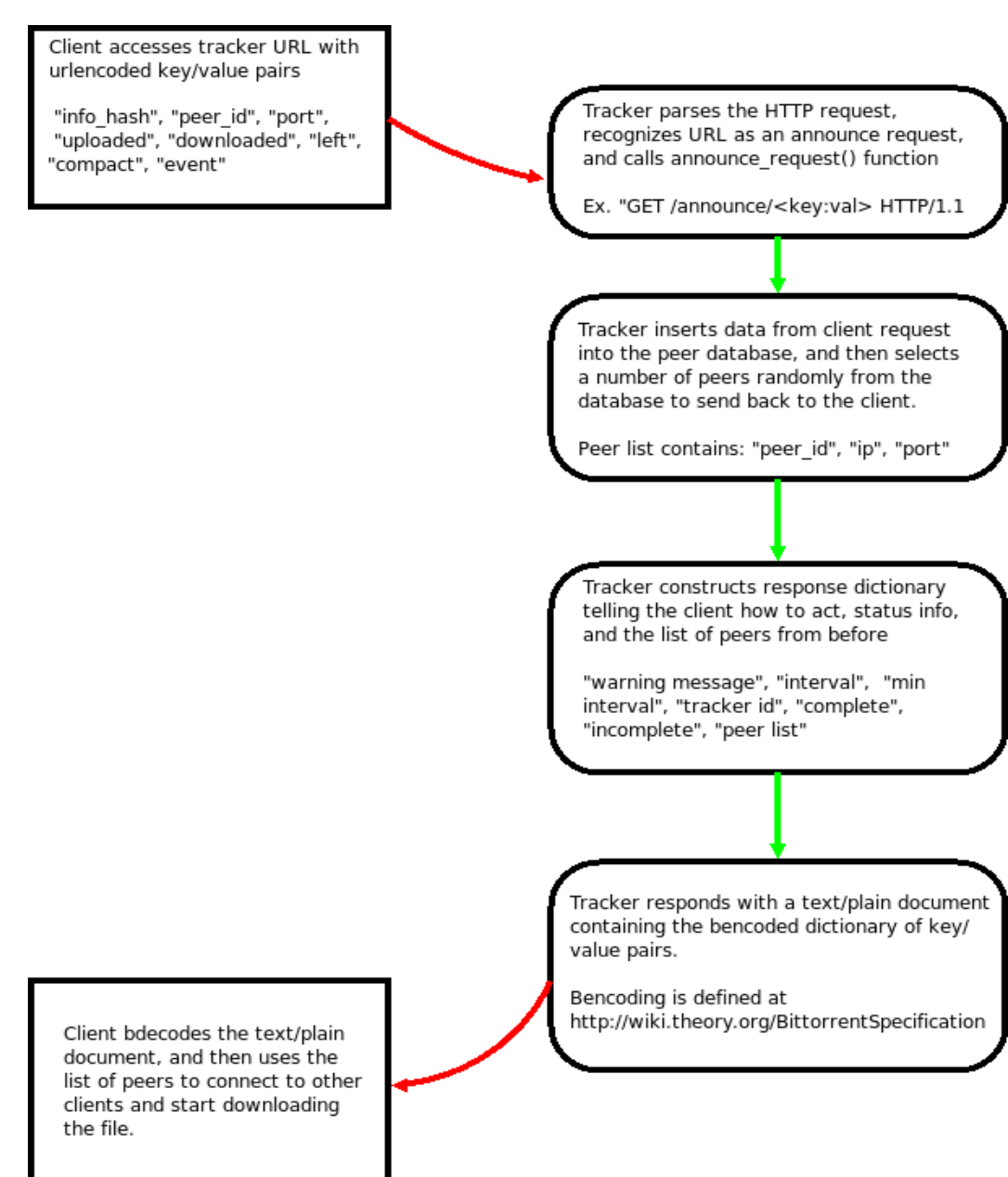
Example of a client "announce" request

Procedure

Generation of a .torrent metadata file is the first step in Bittorrent publishing. This file tells the peers how to connect to the tracker.

The tracker handles connecting the peers with each other. A peer starting a download will request the IP addresses of other peers in the "swarm," and then connect to download the file.

The innovative "everseed" is the final step, that builds on the current system by preventing torrents from ever "dying."

Results

Bittorrent metadata files can be successfully generated by my bencoding algorithm, also used in the tracker and test client. The files load correctly in a number of popular Bittorrent clients, such as the Mainline client and Azureus.

The tracker correctly handles scrape and announce requests from Bittorrent clients. Testing was done with 3rd-party clients for increased compatibility.

The planned features of a web-interface as well as a web seed have not been completed because of time constraints.