

RUBTClient.java

The purpose of this class is to store information about a simple bittorrent client that parses a torrent file, connects to a tracker using the information from the torrent file, and then connects to a single peer from the tracker's peer list. The fields of the RUBTClient object store all the data needed to properly adhere to the bittorrent protocol as well as java socket, input and outputstreams unique to each client, and storage for incoming file pieces from the peer. One connection is made to the peer throughout the RUBTClient objects lifecycle. After the handshake sequence with the peer, the client attempts to request and receive pieces of the file from the peer in increasing order starting at the piece with index 0. Message interpretation and formation is handled by a MessageHandler object. Once the last piece of the file has been downloaded, the RUBTClient closes all streams and connections and exits. The downloaded file is stored in the bittorrent directory along with the torrent file and other source files.

Messagehandler.java

This class represents an object that receives messages in the form of byte arrays, parses the message to determine message id and length, and subsequently determines what actions to take. Such actions include creating and returning response messages that will be sent to the peer, or actions that the client must take like updating the currentPiece index, or changing the downloadComplete variable. A MessageHandler object verifies the the validity of piece message payloads from the peer by comparing the piece hash value from the torrent file with a SHA-1 has calculated for each completed piece that is temporarily stored in RUBTClient's purgatory byte array field. If the piece hash is validated then MessageHandler constructs another request message to be sent to the peer. If the last piece of the file is successfully verified, not more message are sent to the peer.

Peer.java

A simple class that stores the port number, ip address, and peer id of the peer extracted from the trackers peer list. This information is used in the RUBTClient createHandshake method.

Utilities.java

A collection of functions I found useful to me during the project that do not strictly belong to the other classes.

