Group 16
Harshil Shah
Ted Nohh

Project RUBTClient – Part 1

Description:
The client starts up from the main method in RUBTClient. It creates a TorrentInfo file object which contains the meta data of the torrent file. Using that information, the client then makes a connection with the tracker. The tracker replies back with encoded data including the peer list. Then the assigned peer is extracted from the list using and assigned to a “Downloader”. The downloader's goal is to download all the pieces of the file from the assigned Peer. The downloader, first does a handshake with the peer. If successful then it proceeds with interested message and waits for unchoke. Following that it starts making sequential request for blocks of data which is later on combines then into pieces, tries to verify the hash and then sends a have message back to the peer, and finally writes the piece to the output file.

Classes:
RUBTClient.java
- Starts up the main client.
- Parses the torrent file.
- Makes connection with the tracker.
- Retrives the peer list and extracts the necessary peer.
- Spins a “Downloader” thread and assigns it the extracted peer.

Downloader.java
- On startup it performs a handshake with its peer.
- Sends an interested message.
- Waits for unchoke.
- On unchoke start sequentially requesting blocks of data
- When a piece is complete. It verifies it, sends a have message and writes to output file stream

Peer.java
- A prototype of a peer.
- It has methods to perform handshake, alive, interested, and other messages.
- Currently not using BitSet
- Has the ip and port information for the peer.

Tracker.java
- A prototype of a tracker
- Has the necessary keys to retrive data from the tracker's response dictionary.
- Has one method called connect which takes in the necessary information to make contact with the tracker.
- It has the trackers annnouce url

Miscellaneous:
- The documentation on the BitTorrent protocol is very well documented and easy to find.

- The challenges faced were to downloading the last “weird-length” block of data and to buffer the blocks for each piece and then verify.
- For Future: The downloader will not just sequentially download pieces of data but will ask the client which piece to download next, this is only if we will ever be downloading from multiple peers.

Please Note: Do not be overwhelmed by the amount of system.out.prints !!! Might seem like the matrix screen. :D