Joshua Owen
005004114
Mark Vosinkel
082004310

Our main method connects to the tracker and unbencodes the tracker's response. This is put into a vector of peers. Next we create a Downloader thread (or multiple) to connect to a peer and download the availible chunks of the file. This Downloader thread has three threads that it spawns off, there is a Listener thread, a Sender thread, and a KeepAlive thread. The Listener thread listens on the Downloaders InputStream, parses it, and puts it into the inoutQueue as a Message. The Sender thread waits until there is a message in the outputQueue and writes the message to the OutputStream. The KeepAlive thread puts a keepalive message into the outputQueue, sleeps for 30 seconds, and repeats. The Downloader thread waits for messages on the inputQueue and processes them based on the Message ID. The Message class contains useful fields for messages, so the Downloader thread doesn't have to parse them out.

In other words, we programmed this portion of the project with as much expandability as possible. By making the Downloader class a thread, it allows us to create multiple threads to address each peer individually. In turn, each of these threads will have the ability to keep itself alive, listen for incoming messages, and to send outgoing messages due to the subclasses within Downloader. By separating the peers into threads and handling the messages sent between them individually, there is no confusion as to which peer sent what message, and there is less unnecessary waiting involved with the communication.

Our biggest issue was in communicating with the peer. We were able to send and receive messages, for quite some time, but we had difficulty in being able to receive an actual piece of the file. I think most of this problem could have been cleared up simply by having better documentation for the BitTorrent protocol. Granted, that is not within the control of anyone dealing with this project, so it is a moot point. Other than that, the project went fairly smoothly. It consisted of a lot of trial and error in order to find what was actually needed to contact the tracker/peer, but in the end was fairly simple.