

What we Implemented:

Jiaxi Tang:

- SHA1 Hash (copied from a4)
- Data structures for the peers and files
- Data structures for storing the above structures in a list
- Functions to update the peer and file structures
- Functions to add and remove from the list
- Functions to search in the list based on the provided attribute(s)
- Functions to update data structures based on the peer message
- Urlencoder
- Function that convert bit field to byte array
- Function that convert byte array to bit field
- I have implemented what I have promised to implement. I may need to add more functions to search the list or update the peer and file data structures in the future. But for now, I think I have done what I have promised.

Evan May:

- Bencoder implemented
- Infinite loop created alongside the necessary variables in the data structures
- The logic that allows communication and file download between peers
- Function to generate a peer id
- Function to generate a handshake for the peer based on the metafile received from the tracker
- Function to send a handshake to peers
- Function to validate that the hash from the peer handshake matches the handshake generated from the metafile
- Most functions that send messages to peers
- I have implemented almost all of what I promised to implement. The only thing that I was not able to fully implement was some of the message functions. This was because I did not know what structures they would be saved in. When they are created I will finish the implementation.

Mohamed Aamir:

- Read a metafile of any given length.
- Parsing (ben-decoding) metafile/torrent file to get all the individual elements necessary to send the tracker request.
- Create and send a request to the tracker using the information from the metafile.
- Parse the response and store it in data structures for the rest of the implementation.