CMSC417 Final Project Decomposition Document

Brian Sherwood
Patrick Logue
Hayley Lee

0. Components:

1. Client (main)
2. Tracker (HTTP)
3. Peer
4. Message
5. Pieces

1. Inter-Component Interfaces:

- The client, and the protocol driver, will be responsible for calling all other components.
- Client calls Tracker first in order to connect and receive the list of peers. Client also periodically calls Tracker to send/recv updates.
- Client will maintain the state of each peer connection as well as the state of the data fragments.
- Client calls Peer to connect to received peers, establish the handshake, send/recv messages, and download data.
- Peer will construct messages defined in the Message component
- Client will facilitate the peer protocol by deciding which Peers to communicate with based off of Pieces, parse the incoming Messages and feed the received data back to Pieces.

2. Details:

The Client is the component that drives our program. It lives on the peer's computer and implements the protocol. It will execute via command line with arguments that pass in the torrent file. The Tracker component is how our client (peer) will communicate with a central tracker server. It will use HTTP to send GET requests and receive responses. It will construct the formatted request as well as parse the bencoded dictionary into addresses for connection to the peers. The next component, Peer, will define a peer (and its state) and keep track of all peer connections. It will also handle connecting (including the handshake) then sending and receiving messages. The Message file will define all possible messages including KeepAlive and the handshake. Pieces downloads and tracks the fragmented data sent between peers using the bitfield. It additionally handles writing the data to a file, which is done after receiving a full piece. It is possible that we may split some of our components up if we feel they become bloated or they have sections that deserve their own component.

3. Testing:

- The **Messages** component can be tested by comparing captured packets from our client and the reference client in wireshark.
- **Tracker** can be tested by interacting with the poole sample and other reference trackers. It should receive a bencoded dictionary and be able to correctly parse it back to the client.
- **Peer** will be tested by either manually specifying peers or connecting to the tracker. A handshake will be sent and confirmed as valid before then testing each individual message.
- **Pieces** will be tested by capturing fake 'pieces', running SHA1, and writing to a file.
- Inter-component testing (aside from Client) will include building correct messages in Peer and sending the correctly parsed peers from Tracker to Peer.
- **Client** will be tested with each component.
- The final test will be for multiple instances of our client to successfully download the sample torrent on poole.

4. Who does what:

Messages component will be contributed to by all.
The following will be our main focus of work before Dec 5, but these sections do not have to be done exclusively by the assigned group member:

Patrick - Tracker
Brian - Peer
Hayley - Pieces

After Dec 5 we will all contribute to the overall structure of the main Client and integrating our own sections
After Dec 9 we will work together on the final document as well as debugging and any possible extra credit implementations.

5. Deadlines:

As of the 30th, we have no implementation, but by the 5th we will complete the individual components Tracker, Peer, Message, and Pieces.

On the 9th, we plan on having the Client interaction and integration with the other components finished. This will give us time to debug our working client and make it fully-functional. We will also devise experiments to evaluate performance in relation to the reference client which will help us to debug, update our client, and reach the performance of the reference client.

By the final deadline (13th), the project report and any extra-credit implementations will be implemented.

Who contributed to this document:

Brian and Patrick wrote all sections together
Hayley edited the document and contributed to the deadlines