Introduction:

This document will prove the reader with real time thought process and also provide additonal methods for code authentisity. It also servers as a reviewing platform past idea's and implentation. Each section is labeledwith a problem that was attempting to be solved. This also gives me a good opertunity to agrate some of the random thoughts in the java docs to this document.

Problem – Seeding clients never respond to valid requests.

So we've noticed when downloading torrent from the wide many seeding clients will get valid requests and respond for the first few but then after that they respond to none. At first I assumed that perhaps we had too many outstanding request at a time. But after limiting the number of active requests we still found this to be the case. Now it was always my intention to put a timer on how long a request can go unanswered, and perhaps thats all that is needed. But it seems that there may infact be something else occuring....

Problem – Reducing complexity of peering logic.

While i think I've made good progress on seperating connection logic (writing/reading/socket maintenance) from peer logic i think we either need to add more to the management layer or create a new in between layer. Currently our "basic" Peer logic is 300 lines of code utilizing over 8 data structures. Further more it has been see that we are clearly lacking utility like knowing how much time has passed on a particular request. The primary problem is that dissemination of and maintence of requests is complex.

To solve this (@author wiselion) I added the following basic utilities from the managed connection layer:

- Time Stamped All Requests (recieved and sent)
- Added abillty to add call backs (<333) to connection
- Moved all the dissemination of pieces into a disseminator class.