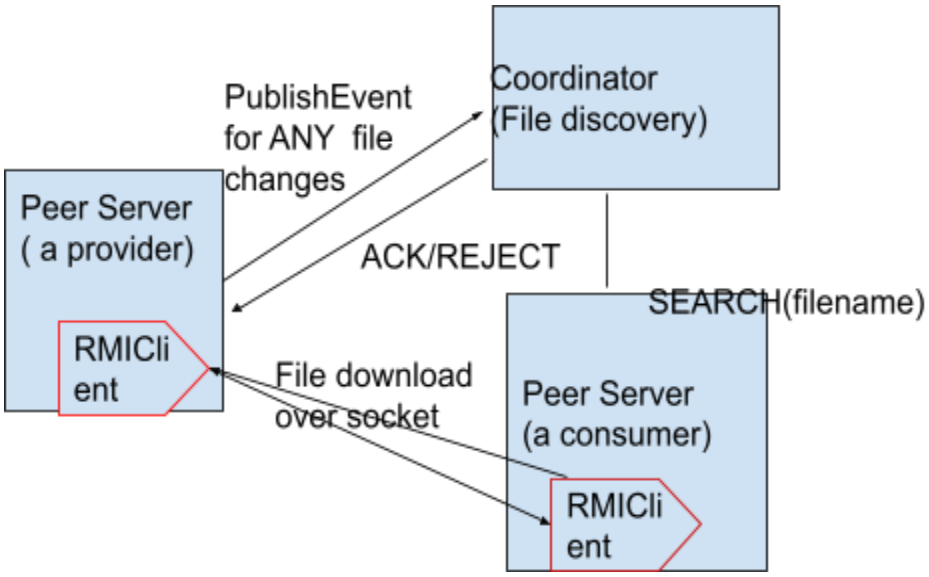
DESIGN DOCUMENT



Fig(a). P2PSystem network topology

OVERVIEW

File Download

The overall design of the project is as above. The RMIClient, contrary to its name, acts as the client part of the Peer2Peer server requests for the download of files once it is aware of the file server host and port address and this happens over TCP sockets. This was chosen design to leverage the multithreaded server and multiple client scenario for file transfers. RMI protocol was used for all interactions between Peer Servers and Coordinators due to its lightweight multithreaded ability.

Publish File Event

When there changes made to a file such as create or modify then these events are broadcasted to the Coordinator that will record these changes internaly in a thread safe hashmap that is guarded by synchronized. The hashmap has keys that designate the filename and the values are all the host and ports of peers where this file has been discovered either by creation or modification. On modification, the software is not aware of which version of the file is stale so all the peers that have last registered will contine as values for the coordinator map.

Search

The search for the filename is an exact match search on the filename.
The oldest(in time) peer to have been discovered for this filename will be returned as a candidate for other peers to download the files. Peers are not deleted from this map unless they explicitly call deregister.

FUTURE WORK

This design works well for a single coordinator but if the coordinator fails the entire system is out of reach . It has a single point of failure which shouldn't be the case in the P2P scenario. The peers should for a distributed hash table rather than a single hash table to make this fault tolerant and also scalable in the set of keys being shared amongst subgroups of peers which would reduce the number of messages that are passed for discovery.

Stateful Load balancing using network bandwidth could be performed on the set of peers to reduce latency in the scenarios where for a set of peers that have knowledge about a filename could split the load of download requests.

Also, the design could have included leader election to choose an new coordinator in terms of a failure or have had more than coordinator as a redundancy.