**A"Simple"Napster"Style"Peer"to"Peer"File"Sharing"System**

**CONTENTS**

1 Indexing server

2 Peer Client

3 Peer Server

4 Trade offs

5 Other files :

6 Future Scope

7 Exception Handeling

8 Design Diagram

**Indexing server :**

Indexing server is a centralized file registry server that stores information of all the files over Peer Servers. Even though the server is centralized, in order to avoid the bottleneck, it supports Threading. Hence for each new request, regardless of its sender, a new thread is generated. Upon receiving he requests it can perform following operations :

1. Registry

It maintains one central registry containing information of file and its peer location. This registry is stored after each successful operation on server database. A file named *registry.txt* will be created when server starts for the first time. This securely stores state of registry in the *registry.txt* file and in case of any server restart, the last stored object of registry will be retreived from file *registry.txt*

2. Add:

Indexing server allows to add one or more files at a time in registry. This operation will fetch the Peer address and list of comma separated file names given by the peer. Then it will add them to registry uniquely. Which means, if file already exists, it won't allow to make duplicate entry for same file.

3. Remove :  
This functionality is provided in order to maintain correctness. For example if a file gets removed by peer server then this mechanism will allow that peer server to remove its entry from registry. However it is required that only the same peer removes that entry from registry. Taking data security in mind, any peer is not allowed to remove an entry of file from registry for some other peer.

4. Search :

This operation allows peers to request for all locations of one file over available peer servers. It returns array of Peers where the file is stored

**Peer Client :**

Peer client is a menu driven program designed in order to give user choices of operations he want to perform. Following are the operations supported by Peer Client

1. Start Peer Server

This will start Peer Server on Port given in properties file

2. Register :

This allows client to manually register files at a time or single single on the server. It also supports uploading entire shared registry over the server. This will register files from subfolders as well.

3. Remove : i

If the file is no longer allowed to be shared, the peer client can remove its entry from server

4. Search :

A client can search for one file over network and can see whereall it exists.

5. Download :

Client can download a file from Peer Clients, if they are running

**Peer Server :**

Peer serve starts running and listens for download requests on its port number.

Once a file fownload request is received, it will create a Thread for it which will then upload the file. Its automated server, hence almost no manual interferance will be required.

Operation:   
Upload:

Peer server sends the file to the requesting socket.

**Trade off :**

The peer servers will write the file of maximum size 100 mb. It was a trade off in order to avoid memory overflows but allow upto 100 mb sized file

In order to allow exact matching of filenames, subfolder locations of files are not stored. Hence, a file will not be allowed to be uploaded if it exists in sub folder of a shared folder

**Other files :**

*properties* file : Stores information of Store Server IP, Server port, Self port and selfDirectory --> Location of shared folder where all files to be allowed to upload are stored.

*serverProperties* file : Stores server port number to start listening on

*registry.txt* file : This file will be used to maintain registry contents in order to avoid data loss in case of any failure on server side. Registry status will be saved each time any proxy will make changes to registry.

Registry file simply writes the object ***Registry*** to the file files hence any changes in the serialized class will call failure to read the old registry status. Solution to this problem is not yet known. Make sure to delete registry each time you make changes to *Registry* and/or *Peer* class.

**Future Scope :**

1.Indexing Server :

a. Before allowing adding any file, we can check whether file actually exists on peer

b. Instead of making centralized indexing server, we can make distributed Indexing server

2. Peer Server

a. uploading a file from subfolder in shared folders can be added

**Exception Handeling**

This program allows to check whether :

a. inputs are entered correctly

b. Peer who is removing the file is the same who registered it.

**Design Diagram :**

Sequence diagram

