

# AN APPLICATION OF INFORMATION RETRIEVAL IN P2P NETWORKS USING SOCKETS AND METADATA

**Ms. M. Kiruthika**

Asst. Professor, Fr.C.R.I.T, Vashi, Navi Mumbai.  
venkatr20032002@gmail.com

**Ms. Smita Dange**

Lecturer, Fr.C.R.I.T,Vashi, Navi Mumbai.  
smm\_31@rediffmail.com

## ABSTRACT

P2P networks are typically used for sharing content files containing audio, images, data etc. Our project is on similar lines. We have tried to establish a P2P network using sockets. A node [peer] acts as client as well as a server. A participating node has to be on the network. The participating node then exports the metadata of the shared files to all the peers on the network. The peers can now view the details of these files in order to choose a file that he wishes to access. Information like file-size, category, and path name are provided to the peer. Then the peer selects a particular file that can be viewed on import. Taking into consideration the tradeoff between bandwidth utilization and memory requirements we have designed our system in such a way that the user has the option to maintain or delete the shared copy of a particular viewed file. A history table, which logs all the information pertaining to the retrieved files, is also maintained. This table holds the name of the file owner, date on which the file was accessed, retrieval time and the path name.

## 1 INTRODUCTION

### 1.1 Overview of Peer to Peer Systems

Peer-to-peer systems have recently gained a lot of attention in the social, academic, and commercial communities. One of the early driving forces behind the peer-to-peer concept is that there are many PCs in homes and offices that lie idle for large chunks of time. These idle resources can be useful for sharing computation or sharing content in fact, peer-to-peer systems have become synonymous with file-sharing systems like Napster, Gnutella, Kazaa and BitTorrent.

P2P networks are typically used for connecting nodes via largely ad hoc connections. Such networks are useful for many purposes such as sharing content files containing audio, video, data or anything in digital format. Peer-to-peer (P2P) systems are application layer networks, which enable networked hosts to share resources in a distributed manner.

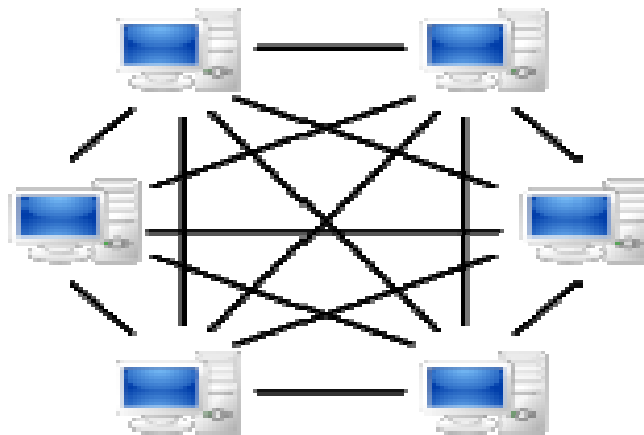
A P2P system, in which two PC/Laptop acts as independent peer nodes has been implemented. A peer can retrieve information from other peers and vice-versa. Our system can successfully perform information retrieval of song and image files

between peers. In addition if a peer wishes to view later a song file or image file he has the option to save it. If he wants to view it frequently he can access it from his own computer because there is a possibility of the peer to no longer be on the network. Hence saving a copy of the file helps to save the bandwidth. A peer can delete a viewed file thereby saving disk space. The details of the song and image file that a peer has viewed are logged into a database for future reference.

## 2 ANALYSIS

### 2.1 P2P System

A peer-to-peer (or P2P) network diverse connectivity between participants and provide cumulative bandwidth than centralized resources where a relatively low number of servers provide value to a service. P2P networks are used to connect nodes through largely ad hoc connections. A pure P2P network has equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model is actually different from client-server model where communication is usually to and from a central server.



*Fig 1:A peer-to-peer based network*

## 2.2 Socket Programming

In computer networking, an Internet socket which is commonly known as a network socket or just a socket is the endpoint of bidirectional communication flow across the Internet. Sockets are nothing but an application programming interface (API) used for inter-process communication. Based on a combination of local and remote IP addresses and port numbers sockets deliver incoming data packets to the appropriate application process or thread. Each socket is mapped by the operating system to a communicating application process or thread and the operating system forwards incoming IP packets to the corresponding application by extracting socket address (a combination of IP address and a port number) information from the IP and transport protocol headers.

Within the operating system and the application that created a socket, the socket is referred to by a unique integer number called socket identifier.

## 2.3 Use of Socket Programming with Respect to P2P Systems.

We need to use a client/server application that does the following.

1. The concurrent PEER-A[server] program runs waiting for a connection from a PEER-B [client].
2. The PEER-A and the PEER-B should run on different machines.

3. The PEER-B connects with the PEER-A.
4. After the connection is established the PEER-A sends the database which has the list of files which he is willing to share with the PEER-B.
5. The PEER-B can select the file which he wishes to view of PEER-A.
6. An entry is made in the History table about which file has been imported and from which peer is the file obtained and the total time taken for file transfer is provided.
7. Each peer can enter a new file in the database along with its attributes.
8. Since the database is updated the PEER-A has to establish a new connection with the PEER-B to view the updated database.
9. After file transfer ,the socket connections need to be closed.
10. In a P2P system, the peer acts as a server as well as a client hence we will be having both the server as well as the client running on a single computer.

## 2.4 Setting up a LAN Connection b/w Two PCs/laptops in Order to Setup a P2P System.

### Objective:

- Create a simple peer-to-peer network between two PCs
- Identify the proper cable to connect the two PCs
- Configure workstation IP address information
- Test connectivity using the ping command.

**Steps to be followed:**

**Step 1:** Identify the proper Ethernet cable and connect the two PCs.

**Step 2:** Verify the physical connection. Plug in and turn on the computers. To verify the computer connections, insure that the link lights on both NICs are lit.

**Step 3:** Access the IP settings window. Windows XP users should do the following:

- Click on Start > Settings > Control Panel and then click the Network Connection icon.
- Select the Local Area Network Connection and click on Change settings of this connection.
- Select the TCP/IP protocol icon that is associated with the NIC in this PC.
- Click on Properties and click on Use the following IP address.

**Step 4:** Configure TCP/IP settings for the two PCs. Set the IP address information for each PC

accordingly.

- Note that the default gateway IP address is not required, since these computers are directly connected. The default gateway is only required on local area networks that are connected to a router.

**Step 5:** Access the Command or MS-DOS prompt

**Step 6:** Verify that the PCs can communicate

- Test connectivity from one PC to the other by pinging the IP address of the opposite computer.

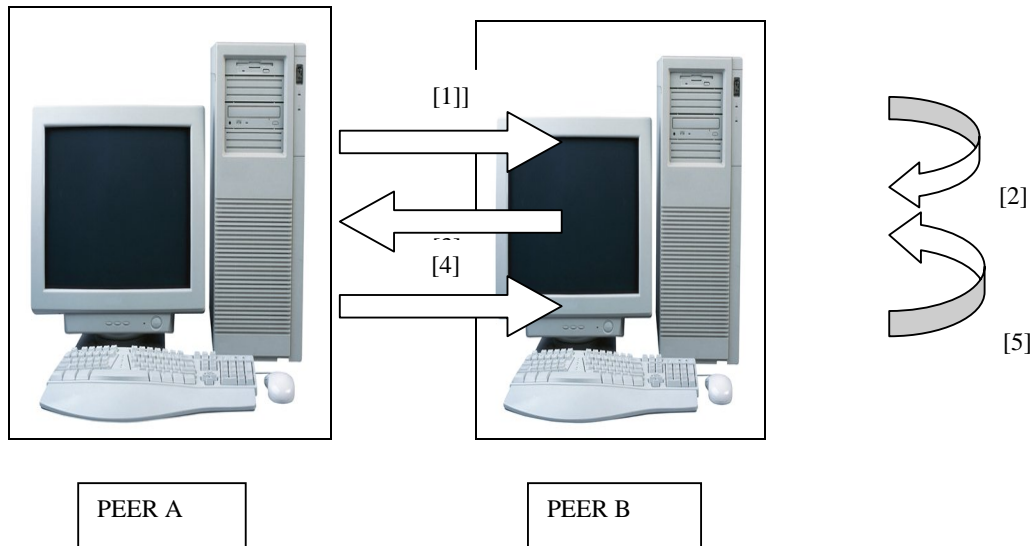
Use the following command at the command prompt.

**C:>ping 192.168.1.1 (or 192.168.1.2)**

- Look for the expected results. If not, check the PC connections and TCP/IP settings for both PCs.

**Step 7:** Restore the PCs to their original IP settings, disconnect the equipment, and store the cables.

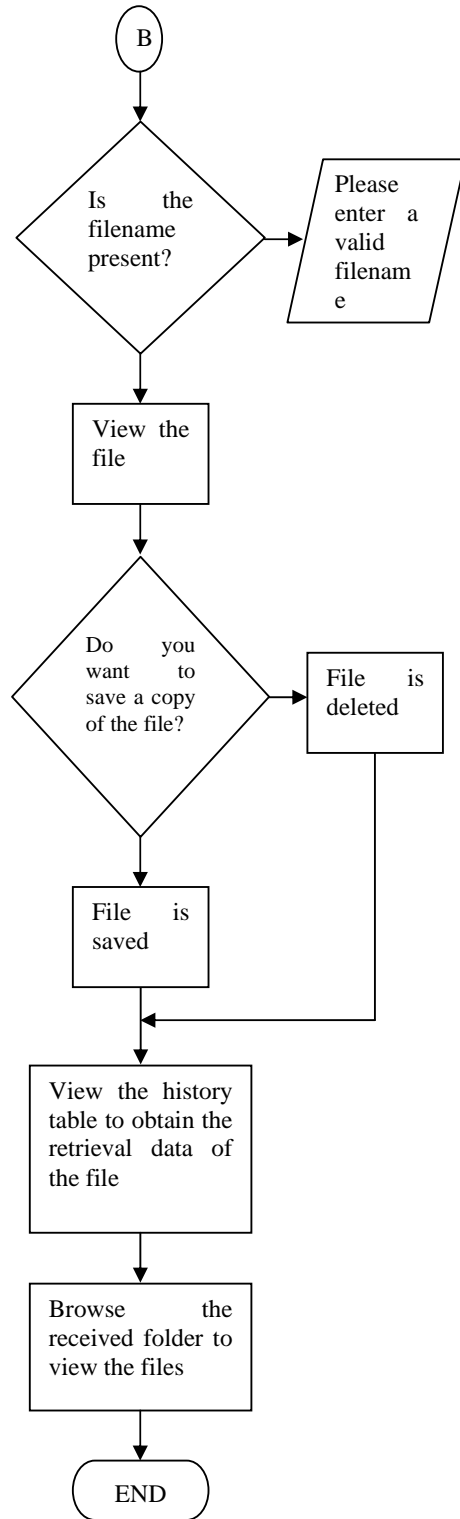
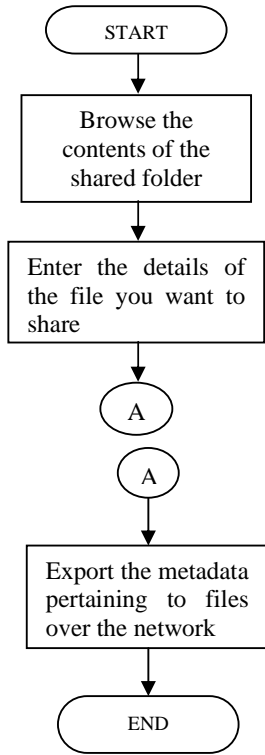
### 3 DESIGN



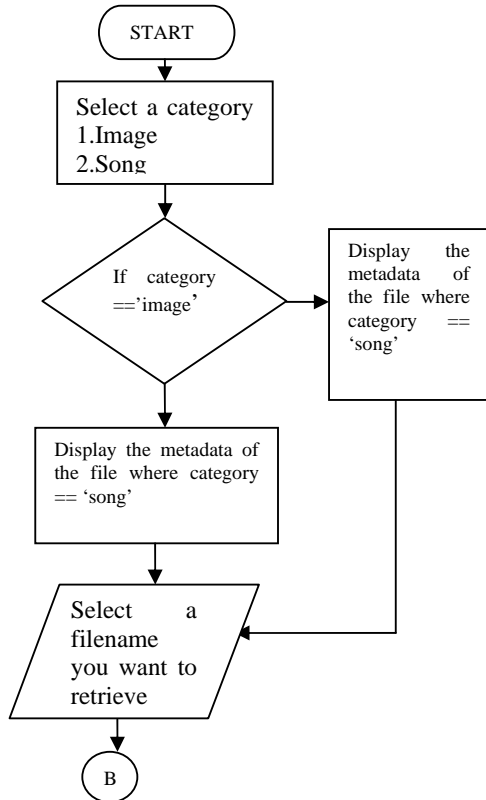
**Fig 2:**Peer A is considered as the shared side and Peer B as the requesting side in the following explanation.

1. Export the database containing metadata of the files in the shared mode over the network.
2. Choose the file you wish to view.
3. Place a request for the file with the sharing peer.
4. Send the file to requesting peer.
5. Click on the VIEW button to open/access the file.

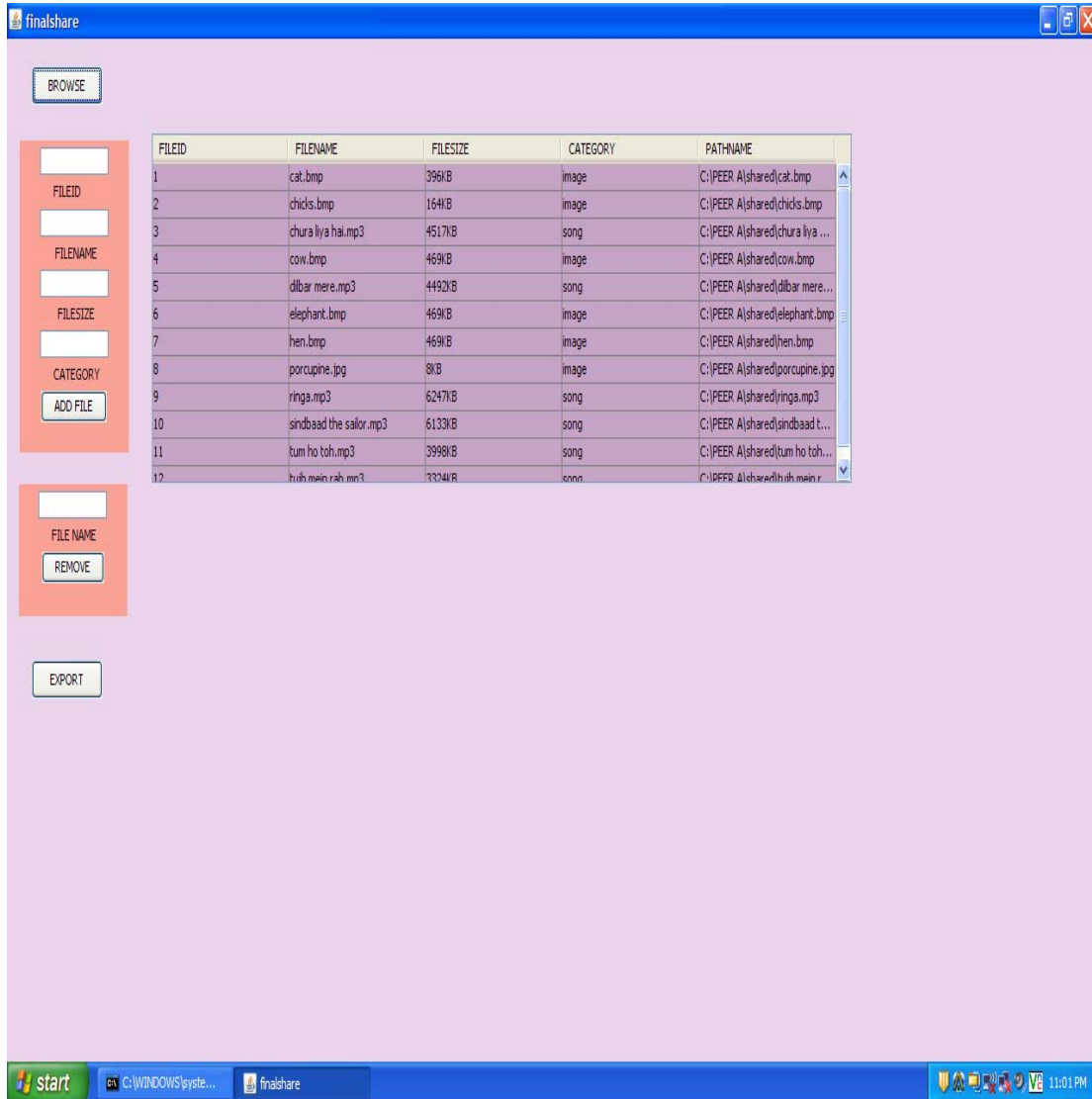
### 3.1 Module Share



### 3.2 Module Request



#### 4 INTERFACE AND I/O SCREENS



request

IMAGE

FILEID	FILENAME	FILESIZE	CATEGORY	PATH-NAME
1	cat.bmp	396KB	image	C:\PEER A\shared\cat.bmp
2	chicks.bmp	164KB	image	C:\PEER A\shared\chicks.bmp
4	cow.bmp	469KB	image	C:\PEER A\shared\cow.bmp
6	elephant.bmp	469KB	image	C:\PEER A\shared\elephant.bmp
7	hen.bmp	469KB	image	C:\PEER A\shared\hen.bmp
8	porcupine.jpg	8KB	image	C:\PEER A\shared\porcupine.jpg

VIEW

FILENAME	OWNER	ACCESS DATE	RETRIEVAL TIME	PATH-NAME
cat.bmp	10.1.1.2	03/31/2009	2213millicsec	C:\Peer B\shared\cat.bmp
chicks.bmp	10.1.1.2	03/31/2009	641millicsec	C:\Peer B\shared\chicks.bmp
ringa.mp3	10.1.1.2	03/31/2009	2203millicsec	C:\Peer B\shared\ringa.mp3
ringa.mp3	10.1.1.2	03/31/2009	861millicsec	C:\Peer B\shared\ringa.mp3
ringa.mp3	10.1.1.2	03/31/2009	981millicsec	C:\Peer B\shared\ringa.mp3
cat.bmp	10.1.1.2	04/01/2009	4657millicsec	C:\Peer B\shared\cat.bmp
ringa.mp3	10.1.1.2	04/03/2009	3646millicsec	C:\Peer B\shared\ringa.mp3
ringa.mp3	10.1.1.2	04/03/2009	861millicsec	C:\Peer B\shared\ringa.mp3
cow.bmp	10.1.1.2	04/03/2009	941millicsec	C:\Peer B\shared\cow.bmp
chicks.bmp	10.1.1.2	04/03/2009	581millicsec	C:\Peer B\shared\chicks.bmp
dilbar mere.mp3	10.1.1.2	04/03/2009	921millicsec	C:\Peer B\shared\dilbar mere....

VIEW

BROWSE

start | C:\WINDOWS\sys... | C:\WINDOWS\sys... | finalshare | request | Document1 - Microsof... | 11:02 PM

## 5 IMPLEMENTATION

The above application has been successfully implemented. The back end is implemented using MS ACCESS. The front end is implemented in JAVA using javax.swing class.

### Module: Share

Peer A will enter the metadata like fileid ,filename ,filesize ,category , pathname pertaining to the files. These details will get stored in the metadata table at the backend. The details will be visible to the peer on the front end in a jTable implemented using the class javax.swing. Details are made available to the network by exporting the details. A socket is created using java.net class for communicating with other peers. Peer A will then

wait for incoming requests.

### Module: Request

Peer B goes onto the network. The details of all the files are extracted from peer A and stored in shared-information table at the backend of peer B. The details are viewed on the frontend in a jTable which is implemented using javax.swing class. He then selects the file that he wants to view. The file transfer is done using sockets implemented using java.net class. After the file transfer is successful he has the option to save the file or delete it. All the requests that the peer B makes are logged into the history table at the back end. The user has an option for viewing the history details on the front end in a jTable implemented using javax.swing class.

## 6 TESTING

Module Name: Share

Sr. No.	Test Case	Test Data	Expected Output	Actual Output
1.	A valid file name should be entered.	No file name was entered.	The user should be prompted to enter the file name.	A prompt box was displayed asking the user to enter the file name.
2.	A file size should be entered.	The file size field was left blank.	The user should be prompted to enter the file size.	A prompt box was displayed asking the user to enter the file size.
3.	A category for the file should be entered. It could be song or image.	The category field was left blank.	The user should be prompted to enter the category of a file.	A prompt box was displayed asking the user to enter the category.

Module Name : Request

Sr. No	Test Case	Test Data	Expected Output	Actual Output
1.	In order to retrieve a file a file name should be entered.	The file name field was left blank.	The user should be prompted to enter the file name.	A prompt box was displayed asking the user to enter the file name.
2.	A valid file name should be entered.	A file name that was not shared by a peer is entered.	The user should be prompted to enter a valid file name.	A prompt box was displayed indicating that the file is not hared.
3.	After retrieving a file a user should have the option of saving its shared copy or deleting it	File retrieval was complete.	The user should be asked if he wants a shared copy of this file.	The user was given the option of having a shared copy of the file on his machine.
4.	If the user wishes to maintain a shared copy of the file he should be indicated the pathname.	The user clicked on yes in order to maintain a shared copy of the file.	Message indicating the pathname should be displayed.	The user was provided with the pathname of the shared file.

## 7 CONCLUSION AND FURTHER WORK

Our system provides many advantages over other content sharing system. The most important feature is the history table that avoids redundant transfer of files.

However, the following developments can be made in order to make the system more efficient and user-friendly.

- **Advanced querying**  
This feature enables the user to execute queries of the form 'Files with FILESIZE<=400 KB'. When such a query is fired the system will return a list of all the files that satisfy that condition along with the details regarding where the particular files are present.
- **Search engine mechanism**  
Through this feature the user will be able to search a particular file using the file name and locate it on any machine connected in the network. This will be an extremely useful add-on because it gives the user complete flexibility for file sharing.

## 8 REFERENCES

- [1] Introduction to Information Retrieval, Cambridge University Press. 2008 by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze
- [2] Information Retrieval A Book By C. J. Van Rijsbergen B.Sc., Dip. Naac, Ph.D., M.B.C.S., F.I.E.E., C.Eng., F.R.S.E.
- [3] "A Tool for Information Retrieval Research in Peer-to-Peer File Sharing Systems" published at Illinois Institute of Technology, Chicago, IL 60616, USA.
- [4] ASAP: An Advertisement-based Search Algorithm for Unstructured Peer-to-peer Systems, Peng Gu, Jun Wang, Hailong Cai, *Proc. International Conference on Parallel Processing (ICPP)*, 2007, pp. 8.
- [5] The Past, Present and Future of Web Information Retrieval, Monika Rauch Henzinger, *PODS*, 2004, pp. 46.