International Journal of Multidisciplinary Research And Studies

Available online at https://ijmras.com/

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND STUDIES
ISSN: 2640 7272
Volume:03; Issue:10 (2020)

Page no.- 15/15

# STUDY ON DESIGN AND DEVELOP FILE SHARING IN P2P BASED ON NEARNESS

**Abhayanand**

*M.Phil., Roll No. :140408: Session: 2014-15*
*University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India.*
*E-mail-: abhyanandmca@gmail.com*

## ABSTRACT

specialized for the purpose of serving different computer systems, in P2P interactions the interacting friends usually have skills and duties that are similar to each other. Huh. Traditional network models that can be built on a custodian-server structure are becoming increasingly less common as newer, more advanced networking technologies continue to be developed. It is very difficult to meet customer expectations in typical client/server networks, which is one reason why P2P is becoming increasingly popular among oral exchange networks. More storage, a quicker range of laptop cycles, and increased bandwidth are the primary advantages that P2P has over the client-server version. The number of people using mobile broadband connection is increasing continuously. Additionally, these customers have high expectations for cellular broadband that can be on par with fixed network customers. The fact charges that cell networks are capable of offering at one time affect the types of offerings that can be provided through those networks


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

**KEYWORDS:** Design, P2p, advanced networking technologies, Communication.

## INTRODUCTION

Communication offering plays a tremendous role within the 0.33 era of the cell network and also in the adjacent generations. In which some computer systems are specialized for the purpose of serving different computer systems, in P2P interactions the interacting friends usually have skills and duties that are similar to each other. Huh. Traditional network models that can be built on a custodian-server structure are becoming increasingly less common as newer, more advanced networking technologies continue to be developed. It is very difficult to meet customer expectations in typical client/server networks, which is one reason why P2P is becoming increasingly popular among oral exchange networks. More storage, a quicker range of laptop cycles, and increased bandwidth are the primary advantages that P2P has over the client-server version. The number of people using mobile broadband connection is increasing continuously. Additionally, these customers have high expectations for cellular broadband that can be on par with fixed network customers. The fact charges that cell networks are capable of offering at one time affect the types of offerings that can be provided through those networks. As a result, the services and capabilities provided through each technology differ from each other. The contemporary era of networks is capable of data delivery at decent speeds and is equipped to handle multimedia offerings as a result of the latest breakthroughs in generation. Streaming video, multimedia content, and mobile TV all place more calls for a community's capacity, and this need is most effectively expected to grow in the coming years.

The proliferation of other packages that make heavy use of bandwidth has accelerated the transition to third-age mobile infrastructure. P2P multimedia apps (audio, picture and video) are not compatible with 2.2G and some .5G cell networks, because of the need for better information charges and ubiquitous conversation services of these networks. The advent of 1/3 technology (also known as 3G) networks has become necessary, which allows you to fulfill these wishes. Even a partially usable 3G cell offering can now be found in any area in the region.

## INTERCONNECTION MACHINE COMPONENTS

Distribution Layer and Gate Access Layer are the two layers that make up the interconnectivity machine. Each peer-to-peer network for document sharing that this machine is a part of


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

includes a persistent node and a backup persistent node that is a part of each distribution layer (there may be additional nodes called aggregate nodes). Is). Static nodes (FG), backup static nodes (BFG), and aggregate nodes are all called gateway nodes (AG). Because there are some FG nodes that may already be considered, a new FG node that is part of a new network needs to be tested with one or more FG nodes that may be part of different networks. Every newly created AG node has to go through an authentication process with the FG node of its own community. At some point in this authentication method both the distribution desk and access tables are created. Each desk has a list of gateway nodes. Each additional gateway node added will serve as an aid to the distribution layer.

The BFG node is an AG node, but it is also the chosen successor of the FG node. Because of this, the BFG node continues its record as the AG while the FG node also holds the data and acts as a backup. In the event that an FG node fails, the BFG node that succeeds it will act as the FG node, and a new BFG node can be elected. The FG node wins the election over the BFG node because they have a wider variety of connections available and are able to sustain the amount of load. It has been determined that a BFG node can be counted as an AG node. AG nodes are taught through their P2P network's FG nodes what types of files enter their P2P networks, as well as the types of record downloading systems and download queue systems. FG nodes are put to use inside the process of handling and maintaining the interconnection system. They contribute to the method of establishing closeness between nodes in an AG. The transmission of facts between friends on a P2P network is done using AG nodes. Access to the layer is designed with the help of friends in a P2P community. The topology that has been described can be seen in Figure 1. There are two distinct varieties of tables that can be used inside the distribution layer making and placing techniques.

To be able to communicate with each other, all gateways that are part of the same P2P community need an entry in the table. Entries to the table are composed of the FG node collectively with all AG nodes that may be part of the same network. Each FG node has a unique access right to the desk. Doing a little calculation on a public domain p2p network shows that the worst-case scenario involves about 2000 AG nodes, each with about 2000 simultaneous connections. That connection quantity will be able to accommodate all peers within the community that has the greatest number of contributors, depending on how it is evaluated. Distribution tables can be damaged in individual sections. The distribution desk of FG nodes is used to enroll AG nodes of multiple P2P networks and is used to interconnect FG


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

nodes. This data is stored through sending keepalive messages to other FG nodes and receiving them in response. It has been updated using a series of incremental improvements. AG nodes are used to connect distribution desk AG nodes that may be part of some P2P networks. When an AG node enters the distribution layer, it always sends a query to an FG node that is already found in its network. The query is sent to all FG nodes that are included in the distribution table through the FG node. The FG nodes of each separate network will reply with the IP address of the AG node in their own network, which has the best number of available connections. These items will be fetched in a database listing the distribution of AG nodes.

A metric device can be used. Downloading is the only reason it is used for. It suggests which available network is best suited for downloading a record. Most metrics are parameters that can be manually adjusted, even others can be learned from other AG nodes. The data is determined with the help of the variety of users who currently have the records, the bandwidth available from the peers, the dimensions of the records, and the queuing machines [8]. It is expected that all AG nodes in a given P2P community will be on par with different P2P networks. The metric fee can be set manually for the neighborhood AG node if it is for use handling the neighborhood weights of each partial metric; As an alternative, it can be learned by using the FG node to assign to the AG nodes.

**Structure Design and its Control**

The structure of the network is shown in parent 2. The FG nodes on each network can be connected to each of the other FG nodes on the other network (the traces formed by the black factors). AG nodes may have hyperlinks with FGs of their own P2P network (marked with the help of pink dots), and they may also have connections with selected AG nodes of alternate networks (strong black marks). There may be an opportunity that a P2P network may have AG nodes which is a good way to have connections with the same AG node within different communities.

While FG nodes broadcast the information, they have with different FG nodes, other FG nodes will give feedback on what facts are missing from their distribution table. Through this method, FG nodes are able to expand their distribution database and communicate routing data with the nodes following them. Then, a Shortest Course First (SPF) set of rules is run on the distribution database through each FG node to find the most efficient path to a certain destination. The SPF set of rules incurs a fee, which is a cost that is determined through the range of hops for a


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

destination. are kept, for individual FG nodes, and the weight of FG nodes that are included in that course is available. Likewise, it transmits the identifier of its own P2P network as well as the styles of the files that are contained in its P2P network. Through this the price value is not always affected in any way. Then, the FG node selects the route with the lowest total cost to include in its distribution desk.

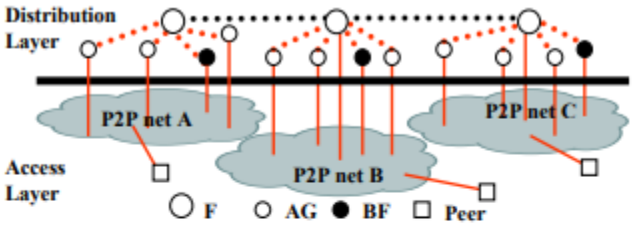


Figure 1: The access layer and the distribution

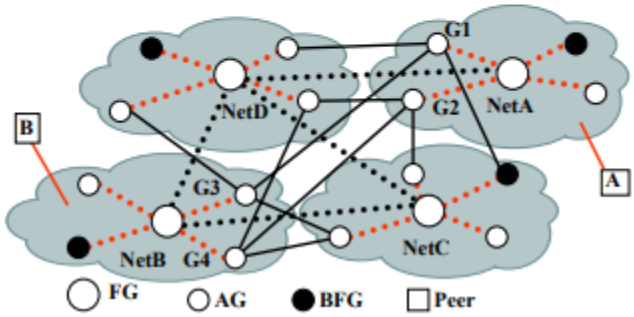


**Figure 1. 1**

## RESEARCH METHODOLOGY

### Overview of P2P File Sharing Architecture

Within the following sections, an overview of modern-day peer-to-peer record sharing architectures and structures, together with a dialogue into their key features and historical development, can be provided. There can be a wonderful communication on each system, but the emphasis can be on the approaches in which they complement each other and how they capture the shortcomings of the opposite systems. Several lessons from the system to be examined are divided into categories, which are outlined in Table 1. It is important to note that organized and loosely established systems are, by definition, decentralized.

**Table 1Classification of peer-to-peer file-sharing systems.**


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

| | unstructured network | loosely structured network | structured network |
|---|---|---|---|
| hybrid decentralized | napster | | |
| pure decentralized | Gnutella | freenet | cord, cane, tapestry |
| partially centralized | Kaza , Gutela | | |

the potential to be decentralized, hybrid decentralized, or partially centralized, while dependent and loosely established structures are usually decentralized to their maximum essential.

This survey does not cover all structures that are currently being deliberated and implemented; However, sample architectures from each class are tested, and the points stated apply to all architectures that are comparable to P2P architectures. The most up-to-date innovations and growing trends regarding these structures can be found in section 4.

## UNSTRUCTURED SYSTEM

### Hybrid Decentralized Unstructured Systems

### Napster

We certainly no longer remember hybrid decentralized systems as authentic peer-to-peer systems, as discussed earlier in this paper. To present a deeper picture , we will now quickly explain the Napster philosophy.

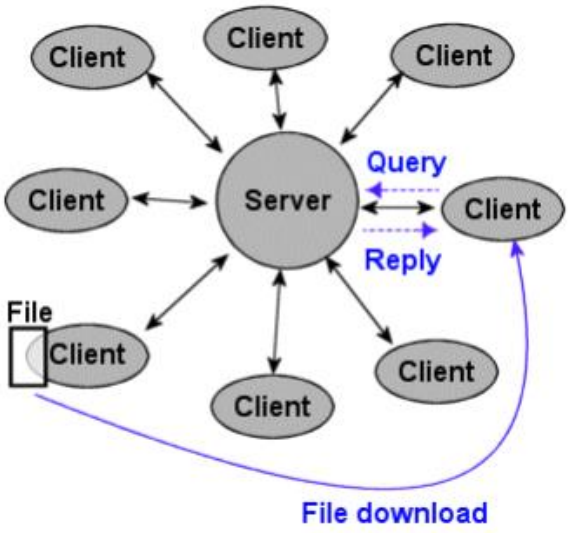


**Figure 2. Napster's Architecture.**


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

After receiving a query from a user, the server will search for the suite in its index and then provide a list of clients that own files that match the query. Then, the consumer initiates an immediate reference to the peer that has the requested document, and downloads the record from that peer.

The simplicity of Napster and the various systems worship it, in addition to its speed and accuracy in finding facts, is one of their number one selling factors. Additionally, the size of the server database and its ability to respond to queries are each potentially limiting factors in the degree to which those systems can be scaled. This is another reason why these systems are clearly not very scalable.

an extraordinary point of view, those systems do not offer a suitable peer-to-peer solution because the shared content, or at least the description of it, is also accessible to, a group, corporation, or Managed through the person who is accountable for maintaining the master server.

**Gutela (authentic structure)**

Reason being , the Gnutella community is one of the most interesting P2P networks to investigate. It was first advanced as an assignment to Null Smooth, a subsidiary of US Online . It acts as a dedicated report storage machine as its customers are given the ability to designate folders on their own computer systems that they need to make available to other peers and share those directories with them . Figure 3, copied without the author's permission from every other source and published here, depicts a "picture".

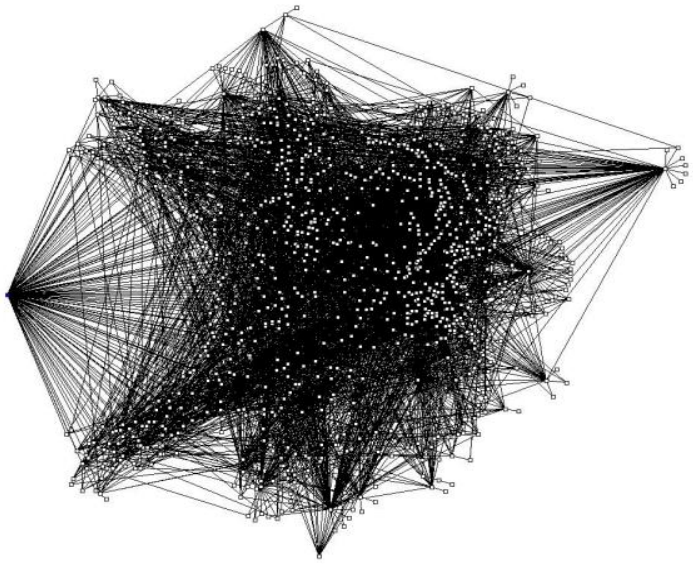

**Figure 3. A snapshot of the Gnutella**


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

## DATA ANALYSIS

This bankruptcy discusses how the layout ideas described have been put into motion and carried out in Pisum to create a hands-on sharing option. The dialogue takes place in the context of the preceding bankruptcy. With the intention of facilitating higher understanding of the framework and the reasons given on this bankruptcy, a large condensed model of the code architecture is provided. Represents a class diagram in Unified Modeling Language (UML) that shows the classes that have been implemented. The suggested code can be broken down into four addendums, each with less than a unique capability. The interface for connecting the P2P factor with the PICSMU is supplied with the help of training API packages. They are the only parts visible from the outside, and as a result, they enclose various structures that may be beneath the floor. With the help of this modular design approach, it is possible to hold or swap out P2P functionality without any changes to the relevant additives., we outline how to connect components with the API and its tutorials so that you can understand how it all works. The network bundle includes both network and peer-to-peer (P2P) functionality. IndexClientP2P is designed to behave as a peer and provides a selection of STATUS and GET methods to interact with a distributed hash desk (DHT). Those strategies are built on top of TomP2P and put into operation.

Bootstrap is a stripped-down version of IndexClientP2P, it is used to make a single peer perform tasks, it is only responsible for listening for incoming connections. If the main reason for the P2P device is to act as an access point for bootstrapping, then this is an essential factor. The index package serves as a vehicle for interaction between the API and the community. IndexClientP2P is the recipient of any processed and forwarded requests that originate from the Software's use of the API. P2PIndex is responsible for the implementation of the DHT Garage approach. This is done by building a DHT call with a selected location key, content key, and domain. Furthermore, it protects the records through encryption. Finally, the hunt package includes algorithms and record structures that make it possible to conduct DHT searches based on the content of the data.


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

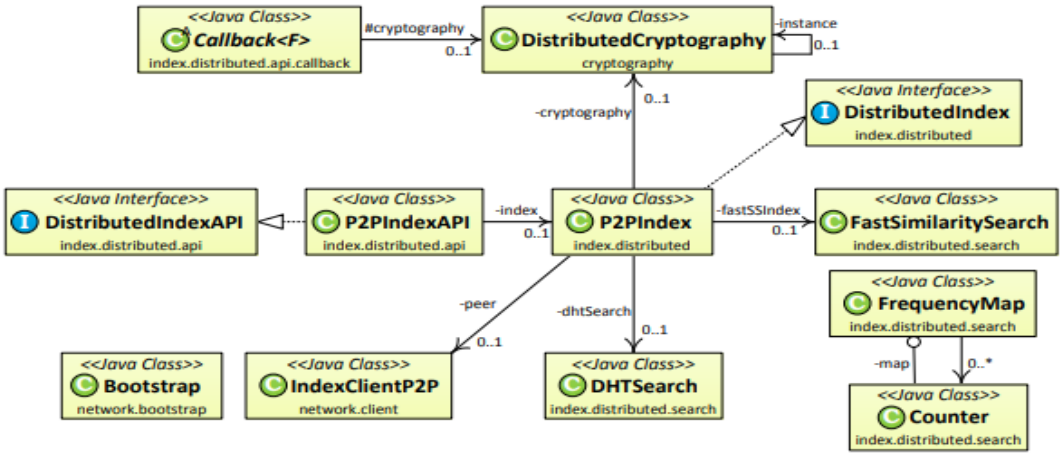


**Figure 4. Simplified UML class diagram of 1code architecture of a P2P system**

## DATA EXCHANGE USING BEANS

The Java programming language makes heavy use of the notion referred to as "beans", which refer to reusable software addons that must adhere to a certain standard. The primary purpose of these objects is to integrate several awesome things into a single entity (called a bean) which can then be shipped as a single entity. Beans, along with File Bean ,File Part Bean , and Credential Bean , are already being used in the basic implementation of PiCs Mu to facilitate the switch of data from centralized servers. This view is available throughout the entire process of incorporating the P2P gadget into PICSMU . Setting up single devices in the DHT, which can accommodate a lot of records, is better in terms of speed than suddenly putting multiple devices in the DHT. This is because single gadgets can preserve more records than some items. In case of bringing things from DHT, the same rule applies. The P2P device provides two new beans, the Distributed Cryptography Bean and the Shared Notification Bean, based primarily on this concept . (1) Distributed Cryptography Bean and (2) Shared Notification Bean First are used to combine encrypted data (including a File Bean that is encrypted) into a single item with their associated virtual signatures. This bean is used in case of individual sharing, which is specifically described in Within the same framework, the Share Notification Bean is the item that is inserted into the DHT for you to notify a specific individual that a record has been shared. It not only contains the UUID of the file but also the person of the sender of the call.

### Record-Sharing Protocol

The sequence of DHT calls that need to be completed so that it can successfully upload and download records when using public record sharing. The relationship between the software and


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

DHT is the primary emphasis, rather than taking into account the impact of cloud offerings or a centralized server. For the purpose of storing FileBean, FilePartBean, Credential Bean, and to search for phrases at some stage in public record add, you need to perform 4 basic state actions. The presence of a loop block indicates that the contained operation will be performed once (for example, for each keyword in the report) until the condition of the loop is satisfied. In evaluating GET operations, deployed operations are no longer assumed to retrieve an object from the DHT. Alternatively, the utility needs to be notified when the process is terminated so that the following method calls can be initiated. The measures that have been implemented to deal with this problem are described here .

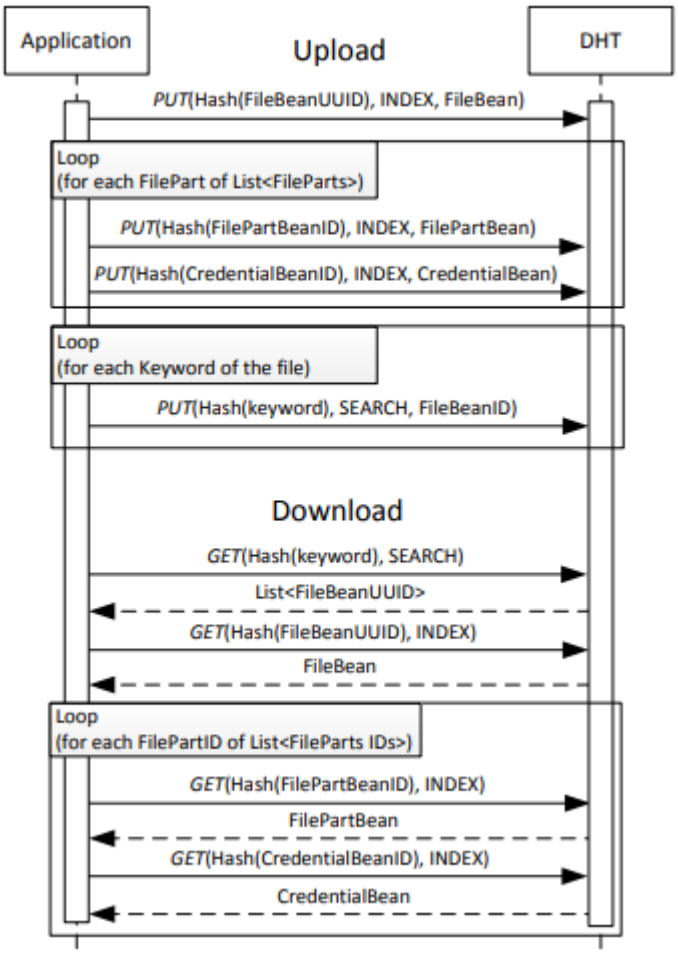


**Figure 5: Public File Exchange Protocol Sequence**

must obtain a FileBean UUID in order to be able to download a record that is freely available to the public . When you query the search domain, you will be given a list of File Bean UUIDs that are relevant to the time period you provide. File Bean is always the area to start the downloading process as it gives statistics about the total amount of File Part Beans that may be


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

required. Similarly, the identity of the Credential Bean that is used to gain access rights to the cloud service on which the File Part Bean in question is being stored is protected in each File Part Bean . Once all the necessary beans are gathered (also called an index), the encoded record sections of the actual document can be downloaded from cloud companies. Sharing files in a private setting requires the same fundamental techniques as sharing files in a public setting, but the signatures of the techniques are unique. Protocols and method calls are proven in parent 4.3. Unlike public sharing, almost all strategies use An allocated crypto graph bean as an argument instead of a selected bean. This is done in the field of a specific bean. Because non-public sharing encrypts beans to protect the confidentiality and integrity of shared records, this requires the use of a 2d wrapper object that is capable of containing encrypted statistics in addition to digital signatures. The structure of DHT region keys is another factor that differentiates the various file sharing protocols from each other. Both methods use the same region keys, but private sharing adds the identity of the recipient to each key so that one can protect you from obscurity within the DHT. Because they are unique across gadgets, the bean's identifiers are what is used to assemble the location keys. This ensures that the same DHT access will never be made again (ie, with the same region and content key). If there is already an Access in the database, a placed call that has the same location and content key as Access will either ignore it or replace it, depending on the methods being used).


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

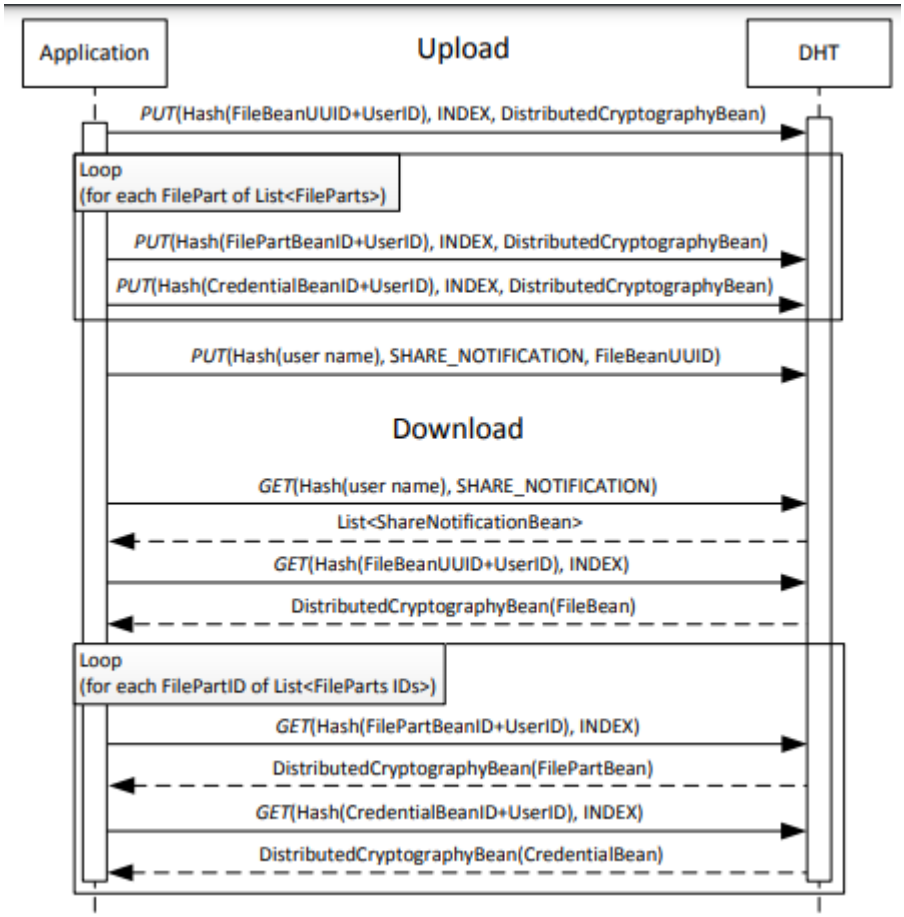


**Figure 6: Private File Exchange Protocol Sequence**

**Asynchronous and non-blocking communication**

Due to the fact that the DHT must be processed internally for some time, a utility that does not support non-blocking verbal exchange will be forced to stop processing and complete the decision before persisting. As a result, PiCs Mu Uses one of a kind technology to manage asynchronous and non-blocking interactions with its P2P element.

**Interacting with DHT Using Futures**

The concept of future gadgets serves as the foundation paper for TomP2P's non-blocking communication protocol . Because futures consider DHT calls and the results of these calls, the result of a PLACED or GET action will be returned immediately. Listeners are used to attend to unforeseen future situations during the implementation period. Listeners are interfaces that watch for the presence of activities before executing a technique in response to those events. is an example of the reasonable utility of this idea. Keeping a file of the operations being carried out is a good way, the contemporary empire of the DHT is briefly stored in future


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

items. Then, a listener interface is created in the future, and it is implemented as an anonymous internal elegance that specifies an operationCompleted (future DHT destiny) attribute. Later, a listener interface is introduced on Destiny. Because TomP2P sees all asynchronous activity on the back end, the application only wants to know when the operation is finished. The listener is the one that picks up this exact notification event and then calls the operation complete method (which takes the destination DHT destination parameter). If the system was terminated without incident, DHT data can be made available on a future object and retrieved thereafter.

**Table 2 1**

```
//get the object with the specified location key
FutureDHT future = DHT.get(0x123);
future.addListener(new BaseFutureAdapter<FutureDHT>() {

    @Override
    public void operationComplete(FutureDHT future) throws Exception {
        if(future.isSuccess()) {
            //retrieve data stored with the future object
            System.out.println(future.getData());
        } else {
            System.out.println("failure");
        }
    }
});
```

## CONCLUSION

The concept of cloud computing has given rise to a wide variety of cloud storage services, which allow personal computers to store, access, and manipulate records remotely. Use of such services requires compliance with information limits imposed by such services, which limit the types of files that can be uploaded. Machado et al. Cloud providers investigate ways to circumvent the strategy used to validate carrier data so they can save whatever they want. Encoding allows the transformation of records in any format representation that is applied to the statistical validation techniques used by cloud companies, resulting in the information becoming unreadable. PiCs Mu , a prototype C/S implementation, was built on the basis of this assumption and turned out to be a reality. The ability to store and convert files is supplied using


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

PiCs Mu as an overlay on top of current cloud offerings. The system is able to maintain a record of what and how it delivered facts because it uses index statistics.

## REFERENCE


1. Amazon Web Services website: "Products and Services". Available at: http://aws. amazon.com . Last accessed: 05.01.2013.
2. CSG website: " Tested Infrastructure for Research Activities". Available at: http: //www.csg.uzh.ch/services/testbed.html. Last accessed: 26.01.2013.
3. Dropbox website: "Simplify your life". Available at: https://www.dropbox.com. Last accessed: 06.12.2012.
4. EclEmma website: "Java Code Coverage for Eclipse". Available at: http://www. eclemma.org . Last accessed: 20.01.2013.
5. Eclipse website: "The Eclipse Foundation Open Source Community Website". Available at: http://www.eclipse.org. Last accessed: 25.01.2013.
6. Freenet website: "The Free Network". Available at: https://freenetproject.org. Last accessed: 29.11.2012.
7. Frostwire website: "Share Large Files". Available at: http://www.frostwire.com. Last accessed: 26.12.2012.
8. Google Drive website: "Keep everything. Share anything". Available at: https://drive.google.com/start. Last accessed: 06.12.2012.
9. Google Picasa website: "Organize, edit, and share your photos". Available at: http://picasa.google.com. Last accessed: 05.01.2013.
10. JUnit website: "Java Testing Framework". Available at: http://junit. sourceforge.net . Last accessed: 20.01.2013.
11. Microsoft Office 365 website: "Get big-business tools at small-business prices". Available at: http://www.microsoft.com/en-us/office365/ small-business-home.aspx. Last accessed: 05.01.2013.
12. µTorrent website: "A (very) tiny BitTorrent client". Available at: http://www. utorrent.com . Last accessed: 29.11.2012.
13. OAuth website: "An open protocol to allow secure authorization from web, mobile and desktop applications in a simple and standard method". Available at: http://oauth.net. Last accessed: 05.01.2013.



**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*

14. Otixo website: "One app for all your cloud needs". Available at: http://otixo.com. Last accessed: 06.12.2012.


**Abhayanand*,** *University Department of COMPUTER SCIENCE: B.R.A. Bihar University, Muzaffarpur India. E-mail: abhyanandmca@gmail.com.*