

# Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks

<sup>1</sup>D.Raghu, <sup>2</sup>CH. Raja Jacob, <sup>2</sup>Gowthu, <sup>3</sup>Jagadeesh kumar, <sup>3</sup>G. Monika Devi, <sup>4</sup>Ramyaa Addanki

<sup>1</sup>Dept. of CSE, NOVA College of Engineering&Tech, Jangareddy Gudem , AP, India

<sup>1</sup>Dept of Computer Science and Engineering, Jangareddy Gudem , AP, India

<sup>2</sup>Nova college of Engineering and Technology, Jangareddygudem,W.G.Dt, AP, India

## Abstract

Peer-to-Peer (P2P) is not a new concept but it is not just limited to file sharing. It refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner. P2P aims at avoiding the dependency on centralized points, providing direct communication between peers, enabling resource aggregation and exchange of services. With the growth of unstructured P2P networks like Napster, Gnutella etc., large scale distributed applications such as distributed data storage and cooperative backup are gaining interest.

## Keywords

P2P Networks, Dynamic Search Algorithm

## I. Introduction

The emergence of Napster, a P2P file-sharing system, in late 1998 is considered the real start of P2P systems. It attracted many people to share their music files and download music files from their peer nodes. But now P2P is not just limited to file sharing. Instead, it refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner. The resources of many users and computers can be brought together to yield large pools of information and significant computing power. In a *peer-to-peer* system (P2P), nodes of equal roles or capabilities exchange information and services directly with each other. Every node can serve as a server and as a client therefore it is called a *servent* (server + client). Furthermore, because computers communicate directly with their peers, network bandwidth is better utilized. However, there are often inherent drawbacks to P2P precisely because of their decentralized nature. For example, in Gnutella, users search for files by flooding the network with queries, and having each computer look for matches in its local disk. Clearly, this type of solution may have difficulty scaling to large numbers of sites or complex queries. In Napster, on the other hand, users cannot search for files globally; they are restricted to searching on a single server that has only indexed a fraction of the available files.

## II. P2P systems and their architectures

We aim at introducing the different types of P2P architectures. There are several different types of P2P architectures, which are divided into two categories, Pure P2P/decentralized and Server-mediated/centralized P2P. In a Pure P2P system there is no central server operating between two peers. Gnutella and Freenet are examples of pure P2P systems. Other is a server-mediated P2P where each machine has a distinct role and the central server is responsible to maintain a registry of shared information and respond to queries for the information. Napster is an example of a server-mediated P2P system.

## A. Centralized Server-mediated P2P

This kind of architecture works in a same way as P2P but it just involves a server for the peer discovery and content lookup. Every node informs the servers about the files and resources that it wants to share, the servers just stores the information in the database and on request for any file it just replies with the information of the nodes containing the file rather than the node flooding the request to every node in the group. Napster is an example of it.

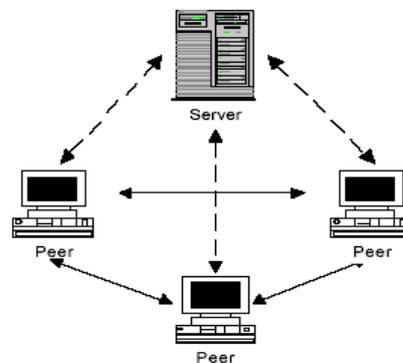


Fig. 1:

**1. Napster:** Napster was started in 1999 by Shawn Fanning at Boston, which became the start of P2P systems and it started as a way to share music files with others [1]. It contains a central server, which maintains a constantly updated directory. Nodes send query of the desired file to the central server and the server replies back with the address of the node containing the file. The client node sets up direct HTTP connection with the other node and the file transfer takes place directly. Therefore Napster started with not a pure P2P but a Peer to Server systems. The steps involved in Napster [3] are: Each user must have Napster software in order to take part in file transfers. The user runs the Napster program. Once executed, this program checks for an Internet connection.

1. If an Internet connection is detected, another connection between the user's computer and one of Napster's Central Servers will be established. This connection is made possible by the Napster file-sharing software.
2. The Napster Central Server keeps a directory of all client computers connected to it and stores information on them as described above.
3. If a user wants a certain file, they place a request to the Napster Centralized Server that it's connected to.
4. The Napster Server looks up its directory to see if it has any matches for the user's request.
5. The Server then sends the user a list of all that matches (if any) it as found including the corresponding, IP address, user name, file size, bit rate etc.
6. The user chooses the file it wishes to download from the

list of matches and tries to establish a direct connection with the computer upon which the desired file resides. It tries to make this connection by sending a message to the client computer indicating their own IP address and the file name they want to download from the client.

7. If a connection is made, the client computer where the desired file resides is now considered the host. The host now transfers the file to the user.
8. The host computer breaks the connection with the user computer when downloading is complete.

### B. Decentralized or Pure P2P

A network is termed decentralized when there is no central server that comes in between the peers. The network consists of randomly connected hosts. Decentralized P2P can be further classified into Structured and Unstructured based on the file placements policy and the query search policy.

### C. Unstructured

Unstructured P2P systems are a class of network in which the search query send by any peer does not follow a particular structure in the network. Gnutella and Kazaa are the two most popular unstructured P2P file sharing networks. In case of Gnutella, the peer sends the query to its neighbor and the neighbors forward the query to their neighbors. In case of Kazaa, the query is first send to a local server but the local server keeps on exchanging the shared list of files with other local servers.

**1. Gnutella:** Gnutella is an example of an 'unstructured' P2P file sharing network that began in March 2000 when the first client software was made public by Justin Frankel of AOL. Shortly afterwards its development was halted by AOL due to potential copyright violation as people were mainly using the client for distributing copies of music files. The Gnutella file sharing protocol was reverse engineered from the original client code and a second generation of clients was born. Due to the P2P nature of Gnutella once the new clients had once the new clients had been released it was impossible to stop the file sharing network from growing in size and popularity. Gnutella is unstructured in the sense that the network consists of randomly connected hosts and file placement on hosts is unrelated to the overlay network structure.

The steps involved in Gnutella are

1. Peer joins the network.
2. Peer sends the search query for a particular file to all its neighbors.
3. If any of the neighbors has the file, that particular peer will inform the originator about the presence of file while other peers will forward the query to their neighbors.
4. If TTL of the query is over, then the query will be rejected.

Due to complete decentralization of network there is no single point of failure. Even if one peer is disconnected, other peers will not be affected. Service is guaranteed for most of the time as it is not affected by disconnection of any peer. However it cannot ensure correct results as there may be some node that has the file but before the query reaches the remote peer, TTL is over so the peer will not gain the access to the file. The major disadvantage of Gnutella is that it floods queries because of which network congestion is increased unnecessarily. To overcome the disadvantages of Gnutella and Napster, one hybrid model started. Kazaa is an example of such a model. Kazaa:

Kazaa is a hybrid of Napster and Gnutella and it covers many of the disadvantages of both of them. Kazaa was developed in March 2001 by programmers Niklas Zennström and Janus Friis of the Dutch company Consumer Empowerment. It was designed so that users could share all types of files like movies, books, games, etc. Users could disseminate both content they had created themselves and content they had purchased or downloaded elsewhere. It works by creating a network of its own on top of the existing internet infrastructure [5]. Making up this virtual network are the various users interconnected in a web like structure. Each user acts as both a client and a server. Kazaa involves the concept of Super Nodes where each super node acts as a Napster server. Peers upload their shared list of files onto the super node and each query goes first to the super node and these super nodes exchange the shared list of files with other super nodes periodically. Whenever a peer search for a file, the query first goes to the super node and the super node reply back the peer with the information of the file. Peer selects the super node based on the capacity and the availability of super node.

### D. Structured P2P systems

Structured P2P systems are classified on the way they follow any structure to forward the query into the network and on their file placement policy. Freenet is a lightly structured P2P system, which provides anonymity to the peers.

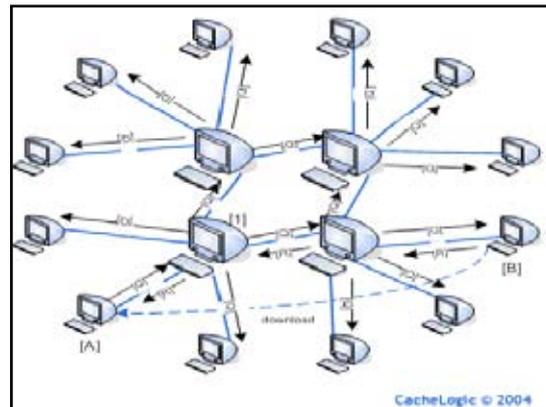


Fig. 2:

**1. Freenet:** Systems like Napster, Gnutella and Kazaa do not provide anonymity to the users; therefore Freenet was proposed which will provide anonymity to peers. With the use of Freenet it will be impossible to know is user is initiating or forwarding a query and if a user is storing the data or forwarding it. The remote peer containing the searched file will reply back to the originator with "Smart query". It is a message in which data flows in reverse path of the path from which the query came. Let us take an example to understand the meaning of smart query where "a" node searches for a file.

"a" will send query to "b"

"b" does not has the file so it will forward the query to "c".

"c" is a dead end, so it will return the response of request failed to "b".

Now "b" will forward query to other end i.e. "e" will forward it to "f".

"f" will forward it to "b".

Since "b" has already checked the query so it will send request failed response back to "f".

"f" will send the failed response to "e".

"e" will send query to "d".

“d” has the data so it will send data reply to

“e” will send data reply to “b”

“b” will send the data reply to “a”.

There are many other structured P2P systems which follow different protocols like Chord, Pastry, and CAN etc. These are based on Distributed Hash Table (DHT) which provides the information lookup service for P2P applications. In systems using DHT each node stores a (key, value) pair and map this key to node. On the basis of the hashing function for calculating the node id, the particular peer is connected and checked for the availability of the file. Key is similar to filename and value is similar to contents of file. Subsets of (key, value) pair is distributed among peers. DHT interface is defined which consists of

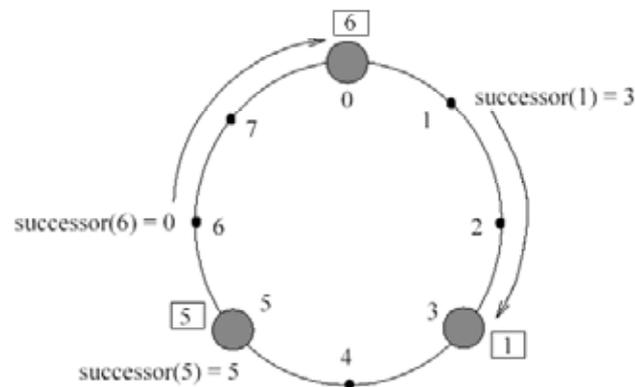


Fig. 3:

Some of the implementations of DHT Interface used as P2P systems are

- Chord
- Pastry
- Etc.

**2. Chord:** Chord builds an efficient distributed hash table using the consistent hashing [22] technique. Chord maps keys onto nodes in a way that facilitates locating these keys later when the system is queried about them. Each key and each node is assigned an  $m$ -bit identifier using a hash function. The identifier of the key is the hash of the key itself (e.g., name of the file). Identifiers are ordered in an identifier circle. Each key is assigned to the first node whose identifier equals or follows the key's identifier. This node is called the successor of the key. Nodes know their successors in the identifier circle and maintain a routing table, called the finger table [6]. When a node  $n$  receives a request for a key  $k$ , it searches its finger table for a node  $j$  whose identifier is closest to  $k$ , which by construction of the identifier circle will know more about  $k$  than node  $n$  does. Then,  $n$  asks  $j$  to resolve the query. Hash functions ensure even distribution of nodes and key among peers on the circle. However one major issue with Chord is that they do not take care for Network locality, which means two successive nodes in the circle, can be geographically distant from each other. So time taken becomes very high even when they are very close on the identifier circle.

**3. Pastry:** It is similar to Chord in a way that it also places nodes in the identifier circle but the difference is that it takes care of the network locality. So it provides faster and accurate results than Chord.

### III. Conclusion

In this paper we introduced P2P systems and then discuss

different types of P2P systems. In P2P systems, nodes are given equal importance and they are self-organized. Peers can decide what to share, whom to share with, how to share using particular system. In centralized P2P systems like Napster, there is a single point of failure. But the benefit of using centralized systems is that it provides more accurate results. P2P is not just limited to file sharing now, instead it uses different decentralized systems like Gnutella and Kazaa which can used to share any type of files. But these systems does not provide anonymity but Freenet provides anonymity i.e. users do not know about each other. Some other structured systems use Distributed Hash Tables (DHT), which distributes the file among peers. Chord, Pastry, etc. are some of the techniques which place the peers into circle and then search for the file in the circle.

### References

- [1] [Online] Available : <http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p4.html>
- [2] [Online] Available : <http://delivery.acm.org/10.1145/850000/844350/p25-lui.pdf>
- [3] [Online] Available <http://www.comp.lancs.ac.uk/computing/users/blundeIn/DeptSite/public/GnutellaOptimisationTechniques.pdf>
- [5] [Online] Available : <http://www.bandwidthcontroller.com/limit-kazaa-bandwidth.html>
- [6] [Online] Available : <http://www.cs.sfu.ca/~mhfeeda/Papers/p2pSurvey.pdf>
- [7] [Online] Available : [www.stanford.edu/class/cs347/notes/Notes11.ppt](http://www.stanford.edu/class/cs347/notes/Notes11.ppt)
- [8] [Online] Available : [http://www.cs.princeton.edu/~qlv/download/searchp2p\\_sig02.pdf](http://www.cs.princeton.edu/~qlv/download/searchp2p_sig02.pdf)
- [9] [Online] Available : <http://www.oreilly.com/catalog/peertopeer/chapter/ch01.html>
- [10] [Online] Available : [http://www.webopedia.com/DidYouKnow/Internet/2005/peer\\_to\\_peer.asp](http://www.webopedia.com/DidYouKnow/Internet/2005/peer_to_peer.asp)
- [11] [Online] Available : <http://plum.sfsu.edu/csc835/presentations/topic7/>
- [12] [Online] Available : <http://momo.lcs.mit.edu/regions/qlrn/node1.html>
- [13] [Online] Available : <http://tisu.mit.jyu.fi/embedded/TIE370/oldlectureslides/P2PIntroduction.pdf>
- [14] [Online] Available : [http://www.usenix.org/events/osdi02/tech/full\\_papers/castro/castro\\_html/](http://www.usenix.org/events/osdi02/tech/full_papers/castro/castro_html/)
- [15] [Online] Available : <http://www.cse.msu.edu/~liuyunha/GLOBECOM03.pdf>