



## **Providing File Surveillance to Actively Search for Files by Querying Structured Peer to Peer Overlay Systems**

**S. VENU GOPAL<sup>1</sup> and N. SAMBASIVA RAO<sup>2</sup>**

<sup>1</sup>Department of Computer Science and Engineering, Hyderabad 501 218, Telangana State, India.

<sup>2</sup>Sumathi Reddy Womens Engineering College, Warangal, Telangana State, India.

(Received: February 16, 2015; Accepted: April 15, 2015)

### **ABSTRACT**

In a peer to peer system searching a file in each and every node it is very difficult. These days everybody using file sharing, and video streaming. So here while sharing or downloading a file there may be loss of data in the file. To avoid all these problems I proposed a concept providing file surveillance to actively search for files by using query performance in structured peer to peer overlay systems. using querying performance in structured overlay peer to peer systems if any node is leaving dynamically data should not loss.

**Key words:** Peer to peer, Surveillance, Searching.

### **INTRODUCTION**

In a peer to peer system the "peers" are computer systems which are connected to each other via the Internet. Files can be shared directly between systems on the network without the need of a central server. Here each and every computer on a P2P network acts as a file server as well as a client.

The requirements of a computer is to join a peer to peer network are an internet connection and peer to peer software. Software which gives a environment to create a network.

P2P software programs include Kazaa, BearShare, Limewire, and Acquisition. These programs connect to a Peer to Peer network, such as Torrent which allows the computer to access thousands of other systems on the network.

Once connected to the network, P2P software allows you to search for files on the other's peer or computer.

So other users on the network can search for files on your computer. If the peer having files like audio, video, and text files, if the peer want to share in peer to peer network it is easy and convenient, is

also have a lot of software piracy and illegal music downloads.

Peer-to-Peer (P2P) networks form a kind of open, decentralized overlay network on top of the Internet, on which distributed users communicate directly to find and share resources, often music and movie files. Peer social networks appear to be mainly composed of pure resource providers that guarantee high resource availability and reliability of P2P systems. The major peers that both provide and request resources are only a small fraction. The connectivity between peers, including undirected, directed and weighted connections, is scale-free and the social networks of all peers and major peers are small world networks. Content distribution is an important peer-to-peer application on the Internet that has received considerable research attention. Content distribution applications typically allow personal computers to function in a coordinated manner as a distributed storage medium by contributing, searching, and obtaining digital content. In this survey, we propose a framework for analyzing peer-to-peer content distribution technologies. Our approach focuses on nonfunctional characteristics such as security, scalability, performance, and resource management potential, and examines the way in which these characteristics are reflected in and affected by the architectural design decisions adopted by current peer-to-peer systems

### Routing characteristics of Searching for file in structured Overlays

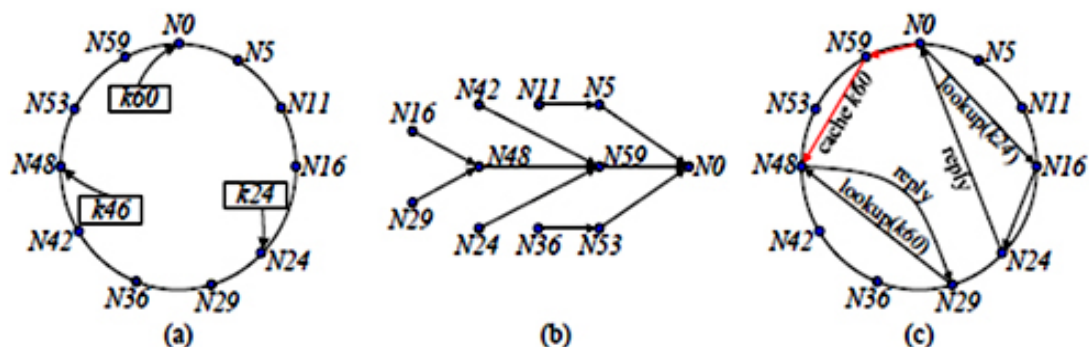
In peer to peer structured overlays each file is treated as key and peer is arranged as a graph topology that maps each key to a responsible peer. This is the successor peer of the key denoted by successor (key). The search file is stored at the peer responsible for the it's key. If any peer sends a request query for the key, the request query will pass by some peers and arrive at successor (key).

The paths of the request stored by peers are files. When a peer copies a file to other peers, this operation will improve query performance

### Applying basic Hashing for Partitioning

consider problem of data partitioning given document X, choose one of k servers to use, suppose we use modulo hashing number servers 1..k, place X on server  $i = (X \bmod k)$  data may not be uniformly distributed. Place X on server  $i = \text{hash}(X) \bmod k$ . What happened if a server fails or joins ( $k \rightarrow k+1$ )

Consistent hashing partitioning key-space among nodes and contact appropriate node to lookup/store key. First node determines the node is responsible for key 1 and sends lookup or insert to first node.



	Routing table size	Lookup/ Routing	Join/ Leave: Routing updates	Join/ eave: key Movement
Consistent Hashing	$O(n)$	$O(1)$	$O(n)$	$O(1)$

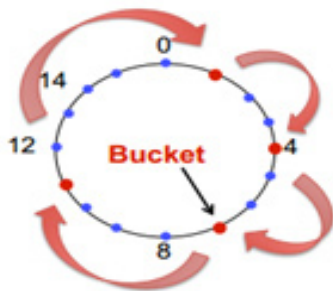


Fig. 1.

### Partitioning key-space among nodes

nodes chose random identifies e.g. hash (IP) keys randomly distributed in ID-space e.g. hash (URI) keys assigned to node "nearest" in ID-space Spreads ownership of keys evenly across nodes

### Consistent Hashing

#### Construction

Assign  $n$  hash bucket to random points on  $\text{mod } 2^k$  circle; hash key size= $k$   
Map object to random position on circle, Hash of object=closest clockwise bucket successor (key)  $\rightarrow$  bucket.

### Consistent Hashing

Using Building routing table we can get the search file without any loss of data Routing table: nodes joining and leaving the ring here first we must update local information about the change, in order for each node in the system to determine precisely which interval in the id space it is responsible for. Second issue is conveying information about the change to all the nodes in the ring so that these nodes will maintain correct information about the system change and consequently manage to route in a single hop. Maintain correct local information about each node's successor and predecessor node. Every node  $n$  runs a stabilization routine periodically, wherein it sends keep-alive messages to its successor  $S$  and predecessor  $P$ . joining node

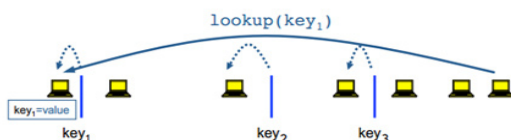


Fig. 1.2

contacts another system node to get its view of the current membership; this protocol is similar to the Chord protocol. The membership information enables it to get in touch with its predecessor and successor, thus informing them of its presence. Maintaining correct routing table, notifications of membership change events, i.e joins and leaves must reach every node in the system within a specified amount of time.

In a peer to peer network file updates in each and every node. The simple algorithm here I discuss:

4. File found in a peer to peer system Algorithm:

Step: 1. setup peer network

Step: 2. find number of files in a node

Step: 3. assign file Id to all the files

Step: 4. set source (key)

Step: 5. finds nearest node check

Step: 5.1 if file availability then

file found update cache in that particular node

else

Step: 5.2 file not found search nearest node

Repeat step 5 Exit if

Step: 6. Exit.

5. Future work

We believe there are certain aspects of query routing and a file sharing techniques available, so, we would like to investigate other techniques for file sharing. Another area that we would like to look into is bringing the network back to scale-free topology after in attack is over. The system finds nodes storing the desired file and returns their IP addresses to the user in order to select a node for downloading the file.

### CONCLUSION

This paper explains the sharing of files in structured peer to peer overlay systems without loss of information. Once connected to the network, P2P software allows you to search for files on the other's peer or computer. If any peer leaving or dynamically added to the network the requested file will going to move from source node to destination node using the concept called query performance in structured peer to peer overlay systems.

## REFERENCES

1. Domenico Talia, and Paolo Trunfio, DynamicQuerying in StructuredPeer-to-Peer Networks.19th IFIP/IEEE International Workshop on Distributed Systems:Operations and Management (DSOM 2008), LNCS 5273, pp.28-41, 2008
2. Mike Freedman,. P2P Systems and Distributed Hash Tables.Sec7on 9.4.2. COS 461: Computer Networks. Spring 2011
3. Anjali Gupta, Barbara Liskov, and Rodrigo Rodrigues, Efficient Routing for Peer-to-Peer Overlays.
4. I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse. Kelips: Building an e\_icient and stable P2P DHT through increased memory and background overhead. In Proceedings of the 2<sup>nd</sup> International Workshop on Peer-to-Peer Systems (IPTPS '03), Feb. 2003.
5. D. Liben-Nowell, H. Balakrishnan, and D. Karger. Analysis of the evolution of peer-to-peer systems. In Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing (PODC 2002), 2002.