Effective and Secure Content Retrieval in Unstructured P2P Overlay Networks using Bloom Cast Technique

Priya Ponnusamy[1], M. Yoha lakshmi[2]

[1]*Assistant professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology*

[2]*II M.E, Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore,*

ABSTRACT:

P2P network stands among one of the best and popular network tool. This technique made the sharing of contents through internet easier. For unstructured P2P networks Bloom Cast serves as an efficient technique for full-text retrieval scheme. In order to achieve a guaranteed recall at a communication cost of the network, Bloom Cast makes copies of the contents in the network uniformly at a random across the P2P networks. To encode the complete document the Bloom Cast utilizes Bloom Filters. To support random node sampling and network size estimation Bloom Cast utilizes its hybrid network which is a combination of a lightweight DHT and an unstructured P2P overlay. There are possibilities of malicious codes and false transactions in unstructured P2P networks. At times it generates false identities resulting in false transactions with other identities. The proposed method here uses the concept of DHT and reputation management which provides efficient file searching. For secure and timely availability of the reputation data from one peer to the other peers the self certification (RSA ALGORITHM and MD5) is used. The peers are here repeated in order to check whether a peer is a malicious peer or a good peer. The transaction is aborted at once a malicious peer is detected. The identity is attached to the peer that has reputation. The peers are provided with identity certificates which are generated using self-certification, and all of them maintain their own (and hence trusted) certificate authority which issues the identity certificate(s) and digital signature to the peer.

Keywords: *Bloom Cast, Bloom Filters, MD5 with RSA Algorithm, Self-Certification.*

I. INTRODUCTION

An overlay network is a type of the computer network which is built on the top of another existing network. Nodes that are present in the overlay network can be thought of as being connected as virtual links or logical links, each one of which corresponds to an appropriate path connected through many physical links, in the existing network. The major applications of overlay networks are distributed systems such as cloud computing, peer-to-peer systems, and client-server systems. They are known so because they run on top of the Internet. Initially the internet was built as an overlay network upon the telephone network whereas nowadays with the invention of VoIP, the telephone network is turning into an overlay network that is built on top of the Internet. The area in which the overlay networks used is telecommunication and internet applications.

Overlay networks provide us with the following advantages and opportunities to better utilize the increasingly growing Internet information and resources. (1) In overlay networks the network developers and application users can easily design and implement their own communication environment and protocols over the Internet. For example data routing and file sharing management. (2) Routing data's in an overlay network is very flexible. It can be quickly detectable and network congestions can be avoided by adaptively selecting paths based on different metrics such as probed latency. (3) Due to flexible routing the end-nodes in overlay networks are highly connected to each other. One end-node can always communicate to another end-node via overlay network as long as the physical network connections exist between them. Thus scalability and robustness in overlay networks are two attractive features. (4) The increasingly more end-nodes of high connectivity, to join overlay networks enables effective sharing of a huge amount of information and resources available in the Internet.

Typical overlay networks include multicast overlays, peer-to-peer overlays (e.g. Gnutella and Kazaa).

1.1. Searching in structured networks

Structured P2P networks employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired resource or data, even if it a rare one. But this process needs more structured pattern overlay links. The most commonly seen structured P2P network is to implement a distributed hash table (DHT), in which deviating of hashing is used to assign ownership of files to that particular peer. It is not similar to the traditional hash table assignment in which in a for a particular array slots a separate key is assigned. The term DHT is generally used to refer the structured overlay, but DHT is a data structure that is implemented on top of a structured overlay.

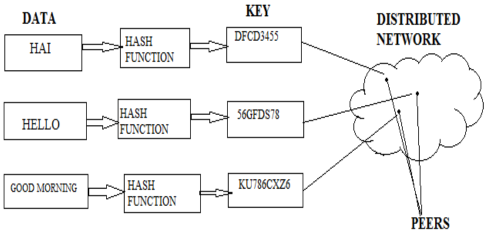


Figure.1 DHT (Distributed hash Table)

A distributed hash table (DHT) is a class of a decentralized distributed system that provides a lookup service similar to a hash table; (key, value) pairs that are stored in a DHT. Any participating node can efficiently retrieve the value associated with a given key. Maintenance of mapping from keys to values is distributed among the nodes in such a way that a change in the set of participants causes a minimal amount of disruption. This allows DHT to scale to extremely large numbers of nodes and to handle operations such as continuous arrival of nodes, departure, and failure.

DHT based P2P systems have several advantages such as it is scalable, robust and efficient. As a result, DHT has become a general infrastructure for building many P2P distributed applications. Applications such as content delivery, physical goods exchange, services or spaces, networking, science and searches adds more advantage to DHT.

Here we use bloom filters, which is a hash based data structure used to reduce the amount of communication required. It has more benefit that it compares the keyword with the entire match list and found the exact match of the keyword. Here we can easily find the locality of the document where it is actually present. We can search the content with less amount of time.

This allows to achieve higher lookup performance for a given memory bandwidth, without requiring large amounts of buffering in front of the lookup engine.

1.2. Searching in Unstructured networks

Unstructured P2P networks are formed when the overlay links are established randomly. The networks here can be easily constructed by copying existing links of another node and then form its own links over a time. In an unstructured P2P network if a peer wants to find out a desired data in the network the query is flooded through the network. This results in finding many peers that share their data. The major disadvantage here is that the queries may not be resolved frequently. If there exists popular content then the available peers and any peer searching for it is likely to find the same thing. In cases where a peer is looking for rare data shared by only a few peers, then it is highly improbable that search will be successful. Since the peer and the content management are independent of each other, there is no assurance that flooding will find a peer that has the desired data. Flooding causes a high amount of signaling traffic in the network. These networks typically have poor search efficiency. Popular P2P networks are generally unstructured.

II. CREATING AN UNSTRUCTURED P2P NETWORK

Peer-to-peer (P2P) computing or networking is a distributed application architecture that divides the tasks among the peers. Peers are active and more privileged participants in the application. They are said to form a P2P network of nodes.

These P2P applications become popular due to some files sharing systems such as Napster. This concept paved a way to new structures and philosophies in many areas of human interaction. Peer-to-peer networking has no restriction towards technology. It covers only social processes where peer-to-peer is dynamic. In such context peer-to-peer processes are currently emerging throughout society.

Peer-to-peer systems implement an abstract overlay network which is built at Application Layer on the top of the physical network topology. These overlays are independent from the physical network topology and are used for indexing and peer discovery. The contents are shared through the Internet Protocol (IP) network. Anonymous peer-to-peer systems are interruption in the network, and implement extra routing layers to obscure the identity of the source or destination of queries.

III. CONTENT SEARCHING USING BLOOM CAST

In this section we are going to have a detailed discussion about the concepts of content searching and bloom cast.

3.1. Content Searching

In unstructured peer to peer, information about one peer is unknown to the other. (i.e.) to enable communication between the peers, the peers in the network should know some information about the other peer in the network.

The proposed system uses the distributed hash table where each and every peer has the separate hash table. The information stored in the hash table is based on Reputation management (tracking users past activity).It helps to perform the file searching operation efficiently.

In a content search function, the input is a set of keywords representing a user's interests and the output is a set of resources containing these keywords. In the content search context, resources represent text documents or metadata of general resources. Some of these resources are software applications, computer platforms, or data volumes. Content search is useful when a user does not know the exact resource names of interests; this case is common in P2P-based searches as well as in web searches.

Flooding is the basic method of searching in unstructured P2P networks; however, large volume of unnecessary traffic is seen in blind flooding based search mechanism. This greatly affects the performance of P2P systems. The further study shows that a large amount of this unwanted traffic is divinable and can be avoided while searching in P2P networks.

The bloom hash table is used to store the resources which help in effective searching of resources with desired capabilities. Information about the path-name is also provided by the bloom hash table. This design enables resource discovery without knowledge of where the corresponding data items are stored.

3.2. Bloom Cast

Bloom Cast is a novel replication strategy to support efficient and effective full-text retrieval. Different from the WP scheme, random node sampling of a lightweight DHT is utilized by the Bloom Cast. Here we generate the optimal number of replicas of the content in the required workspace. The size of the networks is not depending on any factor since it is an unstructured P2P network. The size of the network is represent here as N. By further replicating the optimal number of Bloom Filters instead of the raw documents, Bloom Cast achieves guaranteed recall rate which results in reduction of communication cost for replication. Based on the Bloom Filter membership verification we can easily design a query evaluation language to support full-text multi keyword search.

Bloom Cast hybrid P2P network has three types of nodes: they are structured peers, normal peers, and bootstrap peers. A Bloom Cast peer stores a collection of documents and maintains a local storage also known as repository. A bootstrap node maintains a partial list of Bloom Cast nodes it believes are currently in the system. There are many ways to implement the bootstrap mechanism in the previous P2P designs.

3.3. Bloom Filters

Bloom Filters to encode the transferred lists while recursively intersecting the matching document set. A Bloom Filter is an efficient data structure method that is used to test whether the element belongs to that set or not. False positive retrieval results are also possible, but false negatives are not possible; i.e. a query returns either it is 'inside the set' or 'not inside the set'. Elements can only be added to the set and cannot be removed. When more elements are added to the set then the probability of false positives increases. Bloom Casting is a secure source specific multicast technique, which transfers the membership control and per group forwarding state from the multicast routers to the source. It uses in-packet Bloom filter (iBF) to encode the forwarding tree. Bloom Casting separates multicast group management and multicast forwarding.

It sends a Bloom Cast Join (BC JOIN) message towards the source AS. The message contains an initially empty collector Bloom filter. While the message travels upstream towards the source, each AS records forwarding information in the control packet by inserting the corresponding link mask into a collector. After this, it performs a bit permutation on the collector.

The figure for Bloom Filter and their memory storage is designed here to show the interconnections between source and specific multicast protocols. In Bloom Cast the transit routers do not keep any group-specific state. But in traditional IP multicast approaches the forwarding information is installed in routers on the delivery tree.

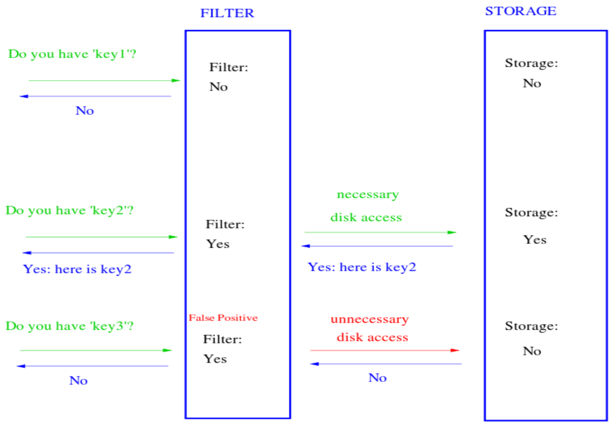


Figure.2 Bloom Filters

The above figure clearly explains the working mechanism of the bloom filters. If the user wants to search any content, initially the query is given. Then there available three possibilities of the result. The first of that is the bloom filter will initially check whether the required content is actually present in the storage area, if it finds the key in storage, then it will gives the positive result to the user. The second is, if the original content is not found in the storage, then it will give the negative result. The third is, the original content may be deleted from the storage area and due to un-updating of the content, the bloom filter may have the chance to show the false positive results.

IV. SELF CERTIFICATION AND CONTENT AUDITING

Each and every peer has the unique identity, based on this, the peer is identified and the transaction is begined. The certification is attached with identity of the peer. The certification uses the concept of RSA, where the algorithm generates the private key and public key, these identities are attached with reputation of the given peer. The sender sends the information which is associated with its private key and signature, the receiver encrypts using its public key, these in formations are updated periodically in Distributed Hash Table. DHT allows to search for specific content identified by a hash key and to eventually perform Boolean operations upon the results of searches that used different keys .It provides considerable fast search times in respect to unstructured solutions. Using the index value the files are stored and retrieved. If malicious peer performs false transaction means it can be identified easily and the transaction is aborted.

The reputation of a peer is associated with its handle. This handle is commonly termed as the "identity" of the peer even though it may not be a peer it receives a recommendation for each transaction performed by it. All of its recommendations are accumulated together for calculating the reputation of a given peer.

Self-certification obviates the centralized trusted entity needed for issuing identities in a centralized system. Peers using self-certified identities remain pseudononymous in the system as there is no way to map the identity of a peer in the system to its real-life identity.

A malicious peer can use self-certification to generate a large number of identities and thereby raising the reputation of one of its identities by performing false transactions with other identities. There is no need for the malicious peer to collude with other distinct peers to raise its reputation. It only needs to generate a set of identities for itself.

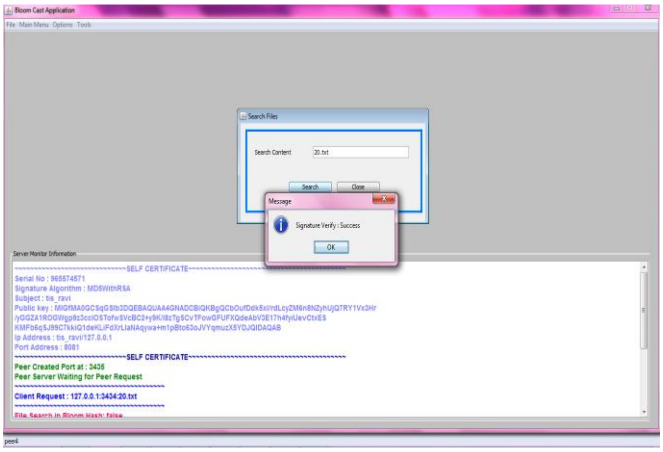
By using the content auditing technique we can easily identify whether the received content is hacked (modified) or not. Here we designed the training set of data. During content auditing the original data is compared with the training set of data. The user can set the probability level for audit the content. If the training sets are matched with the original data and it is up to the threshold level, then the user can predict that there is no hacking occurs, or else the hack is occurred and the rate of hacked content is also known.

V. EXPERIMENTAL EVALUATION

The proposed method is implemented using java. Here we have done the entire work as the simulation using the java simulation tool. The initial step is to create the unstructured P2P network. Each peer consists of the unique port number and IP address. Any number of peers can be created in the network according to the user's requirement. From peer the user can send request and receive response. In the network the query can be posted from any of the peers. It will display if the required content is found in Bloom Filter. Otherwise, it passes the query to next peer. The bloom cast stores the filename (key) in the form hash code. The actual filename is converted into binary values. Then the hash codes are generated with the corresponding binary values. Then the required files are sent to the corresponding work space created by the user.

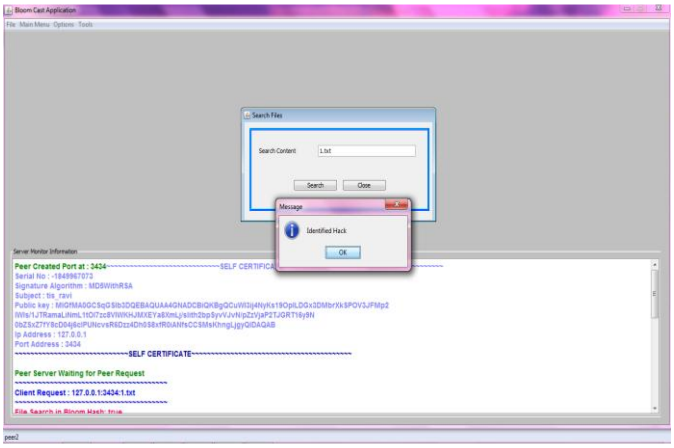
Since it is the unstructured P2P network, there is no central control. In order to provide security to the user, we preferred the self certification technique. It provides a unique identity to each and every peer in the network. That identity is known as the self certificate. It consists of information such as serial number, public key, IP address, port address. These certificates are assigned for the trust of the user.

After that by using the content auditing method, the user can able to identify whether the received content is hacked or not. If the contents are matched with the training set, then the user will receive the original content. If the contents are not matched with the training set, then the user will receive the hacked (modified) content. Then the user can also predict the rate of the content auditing.


Bloom Cast Application
File Main Menu Options Tools
Search Files
Search Content
20.txt
Search
Close
Message
Signature Verify : Success
OK
Server Monitor Information
peer4

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~SELF CERTIFICATE~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Serial No : 965574571
Signature Algorithm : MD5WithRSA
Subject : tis_ravi
Public key : MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCbOufDdk5xl/rdLcyZM6n8NZyhUjQ7RY1Vx3Hr
/yGGZA1ROGWgp8z3ccIO5Tofw5VcBC2+y9K/I8zTg5CvTFowGFUFXQdeAbV3E17h4fyiUevCtxES
KMFb6q5J99C7kkiQ1deKLiFdXrLlaNAqywa+m1pBto63oJVYqmuzX5YDJQIDAQAB
Ip Address : tis_ravi/127.0.0.1
Port Address : 8081
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~SELF CERTIFICATE~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Peer Created Port at : 3435
Peer Server Waiting for Peer Request
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Client Request : 127.0.0.1:3434:20.txt
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
File Search in Bloom Hash: false
```

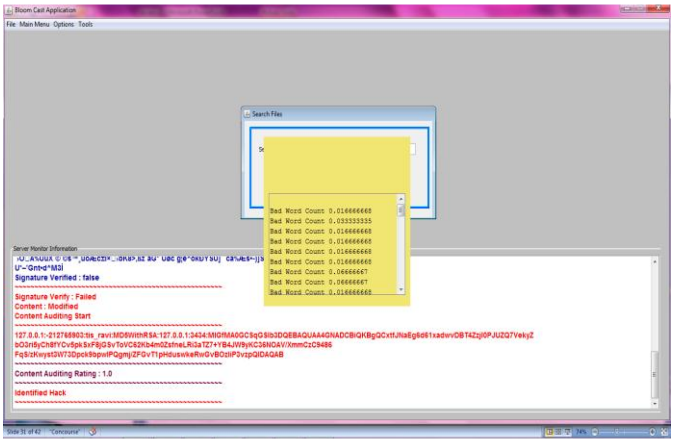
Figure.3.Signature gets verified during content searching, if there is no hacked content



```
Peer Created Port at : 3434~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~SELF CERTIFICA
Serial No : -1849967073
Signature Algorithm : MD5WithRSA
Subject : tis_ravi
Public key : MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuWI3ij4NyKs19OplLDGx3DMbrXk5POV3JFMp2
lWIs/1JTRamaLiNmL1tOI7zc8VIWKHJMXEYa8XmLj/slth2bp5yvVJvN/pZzVjaP2TJGRT16y9N
0bZ5xZ7fY8cD04j6cIPUNcvsR6Dzz4Dh0S8xfR0iANfsCCSMsKhngLjgyQIDAQAB
Ip Address : 127.0.0.1
Port Address : 3434
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~SELF CERTIFICATE~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Peer Server Waiting for Peer Request
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Client Request : 127.0.0.1:3434:1.txt
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
File Search in Bloom Hash: true
```

Figure.4.Signature doesn’t gets verified and identified hack



```
Bad Word Count 0.016666668
Bad Word Count 0.033333335
Bad Word Count 0.016666668
Bad Word Count 0.016666668
Bad Word Count 0.016666668
Bad Word Count 0.016666668
Bad Word Count 0.06666667
Bad Word Count 0.06666667
Bad Word Count 0.016666668
```

```
[illegible]
U'~'Gnt•d^M3ì
Signature Verified : false
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Signature Verify : Failed
Content : Modified
Content Auditing Start
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
127.0.0.1:-212765903:tis_ravi:MD5WithRSA:127.0.0.1:3434:MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCxtfJNaEg6d61xadwvDBT4ZzjI0PJUZQ7VekyZ
bO3ri5yCh8fYCv5pk5xF8jG5vToVC62Kb4m0ZsfneLRi3aTZ7+YB4JW9yKC36NOAV/XmmCzC9486
FqS/zKwyst3W73Dpck9bpwIPQgmj/ZFGvT1pHduswkeRwGvBOzliP3vzpQIDAQAB
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Content Auditing Rating : 1.0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Identified Hack
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure.5. Content Auditing Rating after identify the hack

VI. CONCLUSION

By examining the results of the bloom hash table, we found that this is significantly faster than a normal hash table using the same amount of memory, hence it can support better throughput for router applications that use hash tables.

We here propose an efficient and secured full-text retrieval scheme in an unstructured P2P networks using Bloom Cast method. Bloom Cast method guarantees the recall with high probability. Thus it is considered more effective. The overall communication cost of a full-text search is reduced below a formal bound. Thus it is one among the reason that Bloom Cast is efficient and effective among other schemes. Moreover the communication cost for replication is also reduced because we replicate Bloom Filters instead of the raw documents across the network.

During the transfer of files there is possible of distribution of viruses, worms and trojan horses and malicious peers to overcome this the self certification (MD5 with RSA) is used, it provides authentication and authorization. It easily finds the malicious peers and aborts the transaction. Therefore the proposed method provides the efficient and secure communication between the peers.

Further the content auditing method helps the user to find the rate of the hacked (modified) content by fixing the probability level using the training data sets that are pre-defined by the user.

REFERENCES

[1] R.A. Ferreira, M.K. Ramanathan, A. Awan, A. Grama, and S. Jagannathan, “Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks,” Proc. IEEE Fifth Int’l Conf. Peer to Peer Computing (P2P ’05), pp. 165-172, 2005.

[2] E. Cohen and S. Shenker, “Replication Strategies in Unstructured Peer-to-Peer Networks,” Proc. ACM SIGCOMM ’02. pp. 177-190, 2002.

[3] P. Reynolds and A. Vahdat, “Efficient Peer-to-Peer Keyword Searching,” Proc. ACM/IFIP/USENIX 2003 Int’l Conf. Middleware (Middleware ’03), pp. 21-40, 2003.

[4] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, “Fast Hash Table Lookup Using Extended Bloom Filter: An Aid to Network Processing,” Proc. ACM SIGCOMM, 2005.

[5] C. Tang and S. Dwarkadas, “Hybrid Global-Local Indexing for Effcient Peer-to-Peer Information Retrieval,” Proc. First Conf. Symp. Networked Systems Design and Implementation (NSDI ’04),p. 16, 2004.

[6] D. Li, J. Cao, X. Lu, and K. Chen, “Efficient Range Query Processing in Peer-to-Peer Systems,” IEEE Trans. Knowledge and Data Eng., vol. 21, no. 1, pp. 78-91, Jan. 2008.

[7] G.S. Manku, “Routing Networks for Distributed Hash Tables,” Proc. ACM 22nd Ann. Symp. Principles of Distributed Computing (PODC ’03), pp. 133-142, 2003.

[8] H. Shen, Y. Shu, and B. Yu, “Efficient Semantic-Based Content Search in P2P Network,” IEEE Trans. Knowledge and Data Eng.,vol. 16, no. 7, pp. 813-826, July 2004.

[9] A. Broder and M. Mitzenmacher, “Network Applications of Bloom Filters: A Survey,” Internet Math., vol. 1, no. 4, pp. 485-509, 2004.

[10] V. King and J. Saia, “Choosing a Random Peer,” Proc. ACM 23rd Ann. Symp. Principles of Distributed Computing (PODC ’04), pp. 125-130, 2004.

[11] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H.Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” Proc. ACM SIGCOMM ’01, pp. 149-160, 2001.

[12] C. Tang and S. Dwarkadas, “Hybrid Global-Local Indexing for Effcient Peer-to-Peer Information Retrieval,” Proc. First Conf. Symp. Networked Systems Design and Implementation (NSDI ’04), p. 16, 2004.

[13] F.M. Cuenca-Acuna, C. Peery, R.P. Martin, and T.D. Nguyen, “Planetp: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities,” Proc. 12th IEEE Int’l Symp. High Performance Distributed Computing (HPDC ’03), pp. 236-246, 2003.

[14] K. Sripanidkulchai, B. Maggs, and H. Zhang, “Efficient Content Location Using Interest-Based Locality in Peer-to- Peer Systems,”Proc. IEEE INFOCOM ’03, 2003.

[15] M. Li, W.-C. Lee, A. Sivasubramaniam, and J. Zhao, “SSW: A Small-World-Based Overlay for Peer-to-Peer Search,” IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 6, pp. 735-749, June 2008.