

Cost effective Retrieval in unformed P2P Networks

B.Sundarraaj¹, K.P.Kaliyamurthie²¹Assistant Professor, Department of CSE, Bharath University, Chennai, India²Professor & Head, Department of CSE, Bharath University, Chennai, India

ABSTRACT: Peer-to-peer (P2P) networks establish loosely coupled application-level overlays on top of the Internet to facilitate efficient sharing of resources. They can be roughly classified as either structured or unstructured networks. Without stringent constraints over the network topology, unstructured P2P networks can be constructed very efficiently and are therefore considered suitable to the Internet environment. However, the random search strategies adopted by these networks usually perform poorly with a large network size.

In this paper, we seek to enhance the search performance in unstructured P2P networks through exploiting users' common interest patterns captured within a probability-theoretic framework termed the user interest model (UIM). A search protocol and a routing table updating protocol are further proposed in order to expedite the search process through self organizing the P2P network into a small world. Both theoretical and experimental analyses are conducted and demonstrated the effectiveness and efficiency of our approach.

KEYWORDS: Unstructured peer-to-peer network, self organization, user interest model.

I.INTRODUCTION

PEER-TO-PEER (P2P) networks establish loosely coupled application-level overlays on top of the Internet to facilitate efficient sharing of vast amount of resources [1]. One fundamental challenge of P2P networks is to achieve efficient resources discovery. In the literature, many P2P networks have been proposed in an attempt to overcome this challenge [2]. Those networks can be largely classified into two categories, namely, structured P2P networks based on a distributed hash table (DHT) and unstructured P2P networks based on diverse random search strategies (e.g., flooding) [3].

Without imposing any stringent constraints over the network topology, unstructured P2P networks can be constructed very efficiently and have therefore attracted far more practical use in the Internet [4], [5] than the structured networks. Peers in unstructured networks are often termed blind, since they are usually incapable of determining the possibility that their neighbor peers can satisfy any resource queries. An undesirable consequence of this is that the efficiency of distributed resource discovery techniques will have to be compromised.

Three essential research issues have been identified and studied in this paper in order to save peers from their blindness. The first research issue questions the practicality of modeling users' diverse interests. To solve this problem, we have introduced the user interest model (UIM) based on a general probabilistic modeling tool termed Condition Random Fields (CRFs) [1]. With UIM, we are able to estimate the probability of any peer sharing a certain resource (file) f_j upon given the fact that it shares another resource (file) f_i . This estimation further gives rise to an interest distance between any two peers.

The second research issue considers the actual exploration of users' interests as embodied by UIM. To address this concern, a greedy file search protocol is presented in this paper for fast resource discovery. Whenever a peer receives a query for a certain file that is not available locally, it will forward the query to one of its neighbors that has the highest probability of actually sharing that file.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

The third research issue has been highlighted with the insight that the search protocol alone is not sufficient to achieve high resource discovery performance. This paper proposes a routing table updating protocol to support our search protocol through self organizing the whole P2P network into a small world. Different from closely related research works that are also inspired by the small-world model [7], in order to reduce the overall communication and processing cost, in this paper, the updating of routing tables are driven by the queries received by each peer.

To support our discussion in this paper, theoretical analysis has been performed with two major outcomes: 1) using deterministic limiting models (DLMs), we analytically compared four routing table updating strategies, with the strong indication that our protocol is most suitable for managing neighbor peers, and 2) it is shown in this paper that the expected number of search steps (NOP) for any query is $O(\log n)$, provided that each peer can maintain at least $\log n$ routing entries. n stands for the total number of files shared by the P2P network. Although several assumptions have been utilized to make our analysis applicable, these two results clearly justify the effectiveness of our protocols in unstructured P2P networks.

Finally, simulation studies have been performed in order to evaluate all the proposed protocols and algorithms. We have specifically compared four protocols for updating routing tables via experiments. The experiment results suggest that our protocol outperforms the rest three protocols, including least recently used (LRU) and enhanced clustering cache replacement (ECCR) proposed in [8]. Experiments under varied network conditions are reported, including different network sizes, different routing table sizes, and inaccurate UIMs. The robustness of our protocols is further evaluated in a dynamic environment, where peers continuously join and leave the network. Our experiment results indicate that the protocols presented in this paper are effective.

II.RELATED WORK

A variety of techniques for locating resources (files) in P2P networks have been devised over the last few years. Initial approaches such as Napster [4] emphasize on a centralized architecture with designated indexing peers. Due to the single point of failure problem and the lack of scalability, recent research in P2P networks focuses more on distributed search technologies. Two basic categories of P2P networks have been proposed to support distributed search, namely, structured P2P networks and unstructured P2P networks [1].

Following a fruitful stream of research motivated by Plaxton et al. structured P2P networks have been widely considered as infrastructure technologies with tremendous practical values [2]. These networks feature structured overlays in the form of DHTs. New files have to be published at a specific peer of the network according to well-engineered topology rules. These rules are further utilized to drive users' queries across the network. Structured P2P networks break the important autonomy assumption broadly accepted in the Internet, as peers are required to hold data files that are not to the interest of their users. Network maintenance may also induce considerable cost in highly dynamic environments.

Peers in unstructured P2P networks enjoy more freedom to choose their neighbors and locally shared files. In purely unstructured P2P networks such as Gnutella [3], blind search through flooding mechanisms is usually explored for resource discovery. To find a file, a peer sends out a query to its neighbors on the overlay. These neighbors, in turn, will forward the query to all of their neighbors until the query has traveled a certain radius. Despite its simplicity and robustness, flooding techniques, in general, do not scale. In large networks, the probability of a successful search may decrease dramatically without significantly enlarging the flooding radius.

In order to improve search performance, guided search, instead of flooding, must be put to use [7], [8]. The key problem is what information is actually eligible to guide the search. The basic assumption is that if a peer p' has a particular file required by another peer p , then p' is likely to have other files to be requested by p in the future. According to previous successful queries, shortcuts from peer p to several peers p_0 are established in order to expedite subsequent search processes. Similar assumptions and techniques have been explored in [7] and [10].

The distance measure, usually taking numerical values, will be utilized to guide the search process. For example, in [1], the similarity of any two peers is derived from the fraction of documents that they have for every topic. The practicality of this approach is questionable in cases where each peer shares only a small amount of files and numerous topics are available in the network. No peers may turn out to be similar with each other in this situation based on the similarity

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

measure in [1]. In [6], each shared resource is supposed to be represented by a Semantic Vector (SV). Euclidean distance is used to measure the semantic closeness between any two SVs.

Different from semantic-based measurement, this paper introduces a probability-theoretic framework for capturing users' common interests. Our UIM does not rely on any predetermined semantic structure and is flexible to change in response to new evidence regarding the files shared by peers. Moreover, UIM does not restrict the type of features employed for describing a file, and it works well with features that assume symbolic values. Another prominent strength of UIM is due to its encapsulation of users' common interests statistically based on real file sharing patterns. During the process of searching for a certain file, all files is covered along the search path can be evaluated using UIM to identify their degree of matching the user's interests. Those files with a matching degree (or probability) above a certain level can be returned to the user as a useful supplement to the file being searched for.

ECCR requires every peer in the network to store some files that are not to the interest of its users. As a result, the locality of files is essentially destroyed, resulting in the same problem as in structured P2P networks. In comparison, we are able to achieve similar search performance as in [28], without restricting peers to store files that they do not want. Both theoretical and experimental analyses will be performed in this paper to further compare ECCR and our protocols. This paper will employ this more efficient strategy for neighbor management. Moreover, in this paper, we also recognize the uneven updating problem, which has been briefly explained in Section 1, and a filtering mechanism is further introduced to tackle this problem.

III.GUIDED SEARCH BASED ON USERS COMMON INTEREST

This section presents and analyzes the guided search solution that we proposed for resource discovery in unstructured P2P networks.

3.1 P2P NETWORK ARCHITECTURE

This paper considers a loosely connected P2P network. We use p to denote a single peer in the network. P is further utilized to denote the set of all peers in the network. The main type of resource, namely, a data file, is represented by f . For every peer p , F_p is used to represent the group of files shared by p . In order to conduct distributed search over the P2P network, every peer p maintains locally a list of neighbor peers. This list serves as the routing table for peer p , denoted by R_p . There is an upper bound B_r on the size of any routing table R_p , while the size is measured in terms of the number of entries in R_p . An entry E_p of R_p is a tuple of two elements: $\langle p', f' \rangle$. It represents a link from peer p to another peer p' that shares file f' .

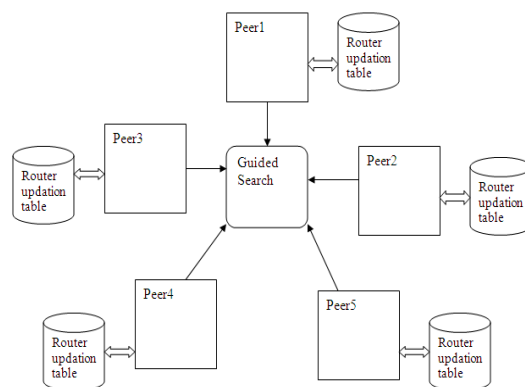


Fig 1 – Architecture

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

In order to locate (or discover) any file under request, the user of a peer p , denoted by up , sends out a query to the network. A query that originated in peer p is represented by q_p and is a tuple of six elements: $q = \langle p, f, h_q, TTL, T_s, t_e \rangle$. Here, p stands for the peer that issued the query q_p . f is the file requested by the query. h_q records the search history, which is a list of peers that have processed the query previously, including peer p itself. In order to prevent a query from incurring too much traffic in the network, time to live (TTL) in a query defines an upper bound on the allowable size of h_q . t_s refers to the time when the query is issued, while t_e is the time when the query is completed. A query is completed successfully if the requested file f has been identified. On the contrary, the query is failed if the size of h_q exceeds the TTL.

Upon receiving a query q , a peer p needs to perform several basic operations: 1) append itself to the search history h_q , 2) search the requested file f among its locally shared files (i.e., local repository), and 3) forward the query to one of its neighbor peers. Each forwarding operation is termed a hop. At the time when query q is finished, the number of hops NOP becomes an important measure of the search performance. In practice, we hope that NOP for average search tasks could be as low as possible, which essentially implies that only a small group of peers will be involved in processing any query.

To summarize, there are two widely used performance metrics for resource discovery in P2P networks: NOP and search success rate. Search success rate refers to the proportion of queries that have been successful among all the queries issued by network users.

3.2 USER INTEREST MODEL

For many Internet applications, users' diverse interests often exhibit important statistical patterns, which can be effectively exploited to improve the quality of service of these applications. This section deals with one essential problem as to how user's interests can be modeled properly.

Our UIM aims at characterizing users' common interest patterns within a probability-theoretic framework. It is adapted from a general probabilistic modeling tool termed Conditional Random Fields (CRFs) [1]. Similar with CRF, UIM defines a log-linear conditional probability distribution $\Pr(f_j / f_i)$ between any two files f_i and f_j . In this paper, $\Pr(f_j / f_i)$ refers to the probability that any user can be interested in sharing file f_j , given the fact that he/she shares another file f_i . Due to its nature for modeling conditional probability distributions, probabilistic inference in UIM is very efficient, without relying on any independence assumptions as required by other probabilistic modeling techniques such as the Hidden Markov model (HMM) [2]. It should also be noticed that to propose a generic probabilistic model suitable for a wide range of applications is not the focus of this paper. For the purpose of this paper alone, we found that UIM is expressive enough to model users' common interests and to guide the resource discovery process.

In practice, every file shared through a P2P network can be uniquely described with a list of attributes. For example, in a P2P network that shares music contents, each music file may be characterized with attributes such as title, genre, artist, and composer. In this paper, an attribute is denoted by σ and a file f is further represented by a list of attributes $f = \{\alpha_1, \dots, \alpha_n\}$.

The key structure for estimating $\Pr(f_j/f_i)$ in UIM is the feature function [4]. Each feature function stands for a certain domain-specific criterion, which is essential for evaluating $\Pr(f_j/f_i)$. If the criterion is satisfied, $F(\cdot)$ will return 1. Otherwise, 0 will become the output of $F(\cdot)$.

For any two files $f_j = \{\alpha'_1, \dots, \alpha'_n\}$ and $f_i = \{\alpha_1, \dots, \alpha_n\}$, a feature function $F(\cdot)$ will take as its input a group of attributes from both files f_j and f_i . For example, depending on specific application domains, it is possible for us to define a feature function of the form $F(\alpha_1, \alpha'_1)$. This function requires one attribute from file f_i and another attribute from file f_j . From a probabilistic perspective, $F(\alpha_1, \alpha'_1)$ indicates that the two attributes depend upon each other in the joint probability distribution $\Pr(f_j / f_i)$ involving both f_i and f_j . In particular, if the criterion defined in $F(\alpha_1, \alpha'_1)$ is satisfied, the two files are considered positively correlated. That is, the user will be interested in sharing file f_j to a certain degree if he/she shares file f_i .

No general restrictions apply to the internal representation of a feature function, and multiple features functions may be defined in order to cover different value combinations for input attributes. The definition of feature functions forms the structure core of UIM, which is domain dependent and can be constantly learned via model learning algorithms [2], [6].

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

3.3 LEARNING USER INTEREST MODEL

In the previous section, we concentrate on modeling users' common interests. To actually apply our model in practice, domain-specific UIMs have to be established and maintained based on observed file sharing patterns across the whole P2P network. The primary concern of this paper is to manage network topology and to enhance resource discovery performance with the help of UIM. However, to make our discussion complete, this section will briefly introduce the process through which UIM can be learned and updated.

In line with the basic design principles of P2P networks, UIMs are better to be learned locally by every peer. However, in order to ensure that these locally maintained UIMs will remain consistent with each other, designated servers will also be employed to fulfill certain computation-intensive learning tasks. According the establishment of UIM comprises two levels of learning tasks: 1) structure learning to determine a group of feature functions and 2) parameter learning to determine the weight associated with every feature function in the group (see (1)).

Structural learning, in general, is a challenging task that demands intensive computation resources. For this reason, dedicated servers are utilized to periodically collect file sharing information from randomly selected peers in the network. Based on collected information, a reference UIM will be created, with emphasis on its modeling structure (i.e., feature functions). This reference UIM is then broadcast through the P2P network to update the structure of those UIMs locally maintained by every peer. The use of servers will not introduce bottleneck to the network, because the distributed search protocol to be presented in the next section, which actually controls the local activity of every peer, does not rely on their existence. Meanwhile, the structure of a UIM, which captures essentially the dependence relationships among a group of file attributes, is expected to change infrequently. As a result, a UIM structure update through servers will not introduce considerable communication cost.

In comparison with structural learning, parameter learning usually happens more frequently. Instead of relying on servers, every peer locally adjusts its weight parameters upon given new observations of files shared by other peers. Those observations are obtained either from neighbor peers or from received queries.¹ This approach helps tackle users' changing interests quickly in a fully distributed manner. Moreover, because all local UIMs share the same modeling structure, the probability distributions that they define are usually similar to each other.

Distributed learning will inevitably introduce discrepancies among locally maintained UIMs, which may further affect the resource discovery performance.^[1] Meanwhile, any model learning algorithms will introduce learning errors such that the learned UIMs cannot precisely reflect users' real interests. In Section 5, experiments will be performed to study the impact of UIMs in such situations. As evidenced in Section 5, under diverse experimental settings, our file search solution has managed to achieve the same level of performance as in the case of a unique and accurate UIM.

3.4 THE SEARCH PROTOCOL

In this section, a file search protocol is presented to regulate the activities of every peer p in a P2P network upon receiving a query $q = \langle p, f, hq, TTL, ts, te \rangle$.

1. Queries are extended with information about files shared by other peers involved in the search process.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Require: (1) Query $q = \langle p_q, f_q, h_q, TTL, t_s, t_e \rangle$ received by a peer p . (2) The routing table R_p of peer p . (3) UIM.
<ol style="list-style-type: none"> 1. Add peer p to the search history h_q. 2. If file f_q is locally stored in peer p, inform peer p_q that the search is successful. 3. Else if the size of h_q is greater than TTL, inform peer p_q that the search fails. 4. Else <ol style="list-style-type: none"> a. Order all the routing entries $E_p = \langle p_e, f_e \rangle$ of R_p decreasingly based on $\Pr(f_q f_e)$. b. Iterate over every routing entry $E_p = \langle p_e, f_e \rangle$, starting from the one with the highest $\Pr(f_q f_e)$. <ol style="list-style-type: none"> 1. If p_e has never been visited before according to h_q, forward the query q on to peer p_e. 2. Else continue iteration with next routing entry. c. If no entry is chosen at step 4.b, forward query q on to p_e with highest $\Pr(f_q f_e)$.

Fig. 2. The guided search protocol.

The local decision involved in the search protocol demands three main types of information: 1) the search history h_q stored in the query q , 2) the routing table R_p of the peer p that handles the query q , and 3) the UIM. h_q and R_p , which are readily available in many P2P networks. UIM, however, is a new source of information introduced in this paper.[2] The workflow of our search protocol is shown in Fig. 2. It follows essentially a greedy search strategy. Upon receiving a query q , a peer p will try to find, among its neighbor peers, the one that is most probable of having the requested file f_q according to UIM. In case that the peer chosen to forward the query already exists in the search history h_q , another peer with a relatively high probability of satisfying the query will be chosen instead. If all neighbor peers have been visited before, then the peer that enjoys the highest probability will be selected again for query processing.

3.5 THEORETICAL ANALYSIS

Through theoretical analysis, this section seeks to justify that high resource discovery performance is achievable with the use of UIM and our search protocol. Building on top of UIM, a distance measure between any pair of peers can be derived.[2]

Require: (1) The search driven by query $q = \langle p_q, f_q, h_q, TTL, t_s, t_e \rangle$ has been successful. Peer p_i , which shares file f_q , has been identified.
<ol style="list-style-type: none"> 1. Repeat for every peer p in the search history h_q. 2. Add a new entry $E_p = \langle p_i, f_q \rangle$ to the routing table R_p of peer p. 3. While the size of R_p is greater than B_r. <ol style="list-style-type: none"> a. With respect to each entry $E_p = \langle p_e, f_e \rangle$, calculate the interest distance $d(p_e, p)$. b. Select an entry $E_p = \langle p_e, f_e \rangle$ with probability proportional to $d(p_e, p)^r$. c. Remove E_p from the routing table R_p.

Fig. 3. The protocol for updating routing tables.

IV. PROTOCOL FOR UPDATING ROUTING TABLES

In this section, a protocol for updating routing tables will be presented and analyzed.[3] An uneven updating problem will also be highlighted, and a filtering mechanism will be further introduced to tackle this problem.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

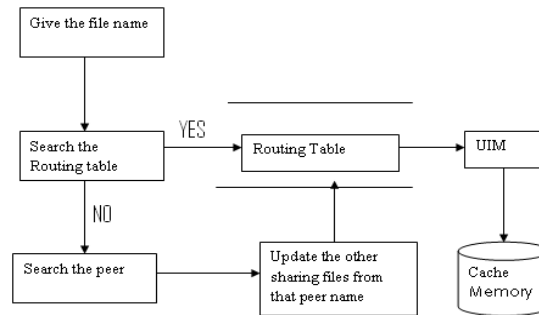


Fig 4. Flow diagram for routing table updating

4.1 ROUTING TABLE UPDATING PROTOCOL

The details of our protocol for updating routing tables are described in Fig. 3. Whenever the search process driven by any query $q = \langle p_q, f_q, h_q, TTL, t_s, t_e \rangle$ is completed successfully, a new routing entry $E_p = \langle p_i, f_q \rangle$, indicating that peer p_i shares the queried file f_q , will be temporarily added into the routing table R_p of every peer p recorded in the search history h_q . If R_p is not full, no entries of R_p will be removed. Otherwise, the size of R_p will be reduced to below Br by deleting one or more selected entries.[4]

The major difference between our protocol (see Fig. 3) and other widely used strategies for routing table update lies in the way routing entries are chosen for removal. For our approach, with respect to each routing entry $E_p = \langle p', f' \rangle$ maintained by peer p , the interest distance between p_0 and p is evaluated according to (3). The probability of removing any entry is proportional to $d(p', p')^f$. Different from this approach, three competing strategies to be analyzed in this paper for updating routing tables are summarized as follows:

- The LRU strategy. The routing entry that is least recently used to forward queries will be dropped.
- The ECCR scheme [2]. With a certain probability Pre , the least recently used routing entry will be dropped. Otherwise, the neighbor peer p_0 , which has the longest interest distance from peer p , will be removed from R_p .
- The distance-centric (DC) strategy. Either the peer p' , which has the longest interest distance from peer p , or another peer p'' , which has the second longest distance, will be removed from R_p of peer p , depending on a probability Pr_d . [5]

4.2 ANALYSIS OF ROUTING TABLE UPDATING STRATEGIES

Based on our analysis in Section 3.5, it is expected that peers' routing tables could converge asymptotically to (4) after many rounds of updating. In this section, the asymptotic properties of four routing table updating strategies will be analyzed and compared according to (4).

To make our analysis achievable, the routing table updating process will be represented through a DLM. We notice that similar techniques have been explored elsewhere with success [1]. Suppose, without loss of generality, that the distance between any two peers is restricted within the range $[0, d_{max}]$ which is further divided into h non-overlapping and equal-length segments. Starting from distance 0, these segments are numbered in increasing order from 1 to h .

V.EFFECTIVENESS OF OUR PROTOCOLS

In this section, the effectiveness of our search and routing table updating protocols (including the filtering mechanism) will be evaluated together in a network of 300 peers. Each peer shares 10 files. The files shared by different peers do not overlap. The routing table size Br is set to 10. The TTL in all queries is fixed at 50. [6] The simulation continues until 50,000 queries have been completed.

Three parameters have been introduced in our protocols, including r in Fig. 2, σ in (15), and v in (17). In order to understand the impact of varied parameter settings on the search performance, a series of experiments has been

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

performed by gradually changing the value of every parameter. Fig. 3 compares the search performance (i.e., NOP and success rate) achieved when r takes different values.

The influence of previously accepted neighbor peers in the density estimation will be enhanced through increasing σ . In this paper, success rate is considered to have a higher priority than NOP. Therefore, σ with a larger value such as $\sigma = 5$ is chosen for the succeeding experiments.

As the value of v increases, the probability of accepting any peer p' located in high-density distance regions (i.e., using p' to update routing tables) becomes smaller. [7]

Overall, the experiment results presented in this section confirm that our search and routing table updating protocols can effectively improve the resource discovery performance. Gradually, the P2P network will self organize into a small world, where NOP can be significantly reduced without sacrificing the success rate. [8]

5.1 COMPARISON OF ALTERNATIVE STRATEGIES FOR UPDATING ROUTING TABLES

In this section, experimental settings identical to Section 5.1 will be used to compare our protocol and another three strategies for updating routing tables, which have been analyzed in Section 4.2. Among all the competing strategies, LRU leads to the poorest search performance as the success rate will constantly decrease to around 0.1, which is unacceptable for most of the applications.

In order to demonstrate the usefulness of the filtering mechanism, experiments have been performed, where each peer relies solely on our protocol in Fig. 3 to update its routing table. [9]

In general, it is clear that our protocol (after using the filtering mechanism) is more effective than LRU, ECCR, and DC.

- 1) Select at random a list of u peers, denoted by $\{p_1, \dots, p_u\}$, from the network.
- 2) With respect to each of the u peers, select at random a file f locally shared by the peer. A collection of u files, denoted by $\{f_1, \dots, f_u\}$, is therefore obtained.
- 3) Based on the right hand side of Equation (3), evaluate the interest distance between each file in $\{f_1, \dots, f_u\}$ and the peer p .
- 4) Select the file $f^* \in \{f_1, \dots, f_u\}$ with the shortest distance.
- 5) Generate a query q_p to search for file f^* .

To ease our discussion, such distance measures will be referred in this paper as the hash distance. The experiment results of using the hash distance together with the two protocols that we proposed are summarized in the last two columns in. [10] As evidenced by the table, due to the exploration of users' common interests, the P2P network using UIM will perform better than when the hash distance is applied. This observation further justifies the usefulness of UIM. [11]

VI. CONCLUSION

In this paper, we have shown that the search performance in unstructured P2P networks can be effectively improved through exploiting the statistical patterns over users' common interests. Our solution toward enhancing search performance was presented in three steps:

1. A UIM has been introduced in order to capture users' diverse interests within a probability-theoretic framework. It leads us to further introduce a concept of interest distance between any two peers.
2. Guided by UIM, a greedy protocol has been proposed to drive the distributed search of queried files through peers' local interactions.
3. Finally, a routing table updating protocol has been proposed to manage peers' neighbor lists. With the help of a newly introduced filtering mechanism, the whole P2P network will gradually self organize into a small world.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Theoretical analysis has also been performed to facilitate our discussion in this paper. Specifically, the search protocol was shown to be quite efficient in small-world networks. Succeeding analysis further justifies that by using our routing table updating protocol, the P2P network will self organize into a small world that guarantees search efficiency. To conclude, it is noticed that our solution presented in this paper can be extended from several aspects. For example, UIM can include extra attributes that characterize a network user (e.g., gender, age, and occupation). Meanwhile, live data from real P2P networks are expected to be collected and applied to further test the effectiveness of our protocols. These and other efforts will be left here as our future work.

REFERENCES

1. T.M. Adami, E. Best, and J.J. Zhu, "Stability Assessment Using Lyapunov's First Method," Proc. 34th Southeastern Symp. System Theory (SSST '02), pp. 297-301, 2002.
2. Kamatchi P., Selvaraj S., Kandaswamy M., "Synthesis, magnetic and electrochemical studies of binuclear copper(II) complexes derived from unsymmetrical polydentate ligands", Polyhedron, ISSN : 0277-5387, 24(8) (2005) PP.900-908.
3. S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution technologies," ACM Computing Surveys, vol. 36, no. 4, pp. 335-371, 2004.
4. Babu T.A., Joseph N.M., Sharmila V., "Academic dishonesty among undergraduates from private medical schools in India. Are we on the right track?", Medical Teacher, ISSN : 0142-159X, 33(9) (2011) PP.759-761.
5. Suresh V., Jaikumar S., Arunachalam G., "Anti diabetic activity of ethanol extract of stem bark of *Nyctanthes arbor-tristis* linn", Research Journal of Pharmaceutical, Biological and Chemical Sciences, ISSN : 0975-8585, 1(4) (2010) PP.311-317.
6. V. Cholvi, P.A. Felber, and E.W. Biersack, "Efficient Search in Unstructured Peer-to-Peer Networks," European Trans. Telecomm., vol. 15, no. 6, 2004.
7. Singamsetty P., Panchumarthy S., "Automatic fuzzy parameter selection in dynamic fuzzy voter for safety critical systems", International Journal of Fuzzy System Applications, ISSN : 1562-2479 , 2(2) (2012) PP. 68-90..
8. E. Cohen, A. Fiat, and H. Kaplan, "Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics," Proc. IEEE INFOCOM, 2003.
9. E. Herskovits, "Computer-Based Probabilistic-Network Construction," PhD dissertation, Stanford Univ., 1991.
10. Kiran B., Kareem S.A., Illamani V., Chitrakha S., "Case of Phthiriasis palpebrarum with blepheroconjunctivitis", Indian Journal of Medical Microbiology, ISSN : 0255-0857, 30(3) (2012) PP. 354-356..
11. R.A. Horn and C.R. Johnson, Matrix Analysis. Cambridge Univ. Press, 1990.
12. Dr.A.Muthu Kumaravel, KNOWLEDGE BASED WEB SERVICE, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801,pp 5881-5888, Vol. 2, Issue 9, September 2014
13. Dr.A.Muthu Kumaravel, Data Representation in web portals, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801,pp 5693-5699, Vol. 2, Issue 9, September 2014
14. Dr.Kathir.Viswalingam, Mr.G.Ayyappan,A Victimization Optical Back Propagation Technique in Content Based Mostly Spam Filtering ,International Journal of Innovative Research in Computer and Communication Engineering ,ISSN(Online): 2320-9801 , pp 7279-7283, Vol. 2, Issue 12, December 2014
15. KannanSubramanian,FACE RECOGNITION USINGEIGENFACE AND SUPPORT VECTOR MACHINE,International Journal of Innovative Research in Computer and Communication Engineering,ISSN(Online): 2320-9801,pp 4974-4980, Vol. 2, Issue 7, July 2014.
16. .Vinothlakshmi.S,To Provide Security & Integrity for StorageServices in Cloud Computing ,International Journal of Innovative Research in Computer and Communication Engineering ,ISSN(Online): 2320-9801 , pp 2381-2385 ,Volume 1, Issue 10, December 2013