Journal of Computer Science 5 (3):184-190, 2009
ISSN 1549-3636

Selective Flooding Based on Relevant Nearest-Neighbor using Query Feedback and Similarity across Unstructured Peer-to-Peer Networks

[1]Iskandar Ishak and [2]Naomie Salim
[1]Department of Computer Science, Faculty of Computer Science and Information Technology,
University Putra Malaysia, 43400 UPM, Selangor, Malaysia
[2]Faculty of Computer Science and Information System,
University Technology Malaysia, 81310 Skudai, Johor, Malaysia

Abstract: Problem statement: Efficient searching is a fundamental problem for unstructured peer to peer networks. Flooding requires a lot of resources in the network and thus will increase the search cost. Searching approach that utilizes minimum network resources is required to produce efficient searching in the robust and dynamic peer-to-peer network. **Approach:** This study addressed the need for efficient flood-based searching in unstructured peer-to-peer network by considering the content of query and only selecting peers that were most related to the query given. We used minimum information to perform efficient peer selection by utilizing the past queries data and the query message. We exploited the nearest-neighbor concept on our query similarity and query hits space metrics for selecting the most relevant peers for efficient searching. **Results:** As demonstrated by extensive simulations, our searching scheme achieved better retrieval and low messages consumption. **Conclusion:** This study suggested that, in an unstructured peer-to-peer network, flooding that was based on the selection of relevant peers, can improve searching efficiency.

Key words: Unstructured peer-to-peer searching, information retrieval, nearest neighbor

INTRODUCTION

Peer-to-peer system has phenomenally become popular in recent years for its ability to search and communicate across the globe. The principal operations in any peer-to-peer networks is to efficiently search and locate data or file, which is ultimately challenging due to its dynamic and robust nature. The challenges are to develop searching efficiently able o locate the file intended. The dynamic nature of the peer-to-peer network where peers can join and leave at anytime make the searching to become efficient is even difficult. The demand for advance searching technique is always present as the peer-to-peer networks becoming larger and more complex and the network become faster and storage become cheaper.

In a peer-to-peer network, a peer acts as client and a server of the system. Peer-to-peer presents attractive solution through its scalability, fault-tolerance and autonomy. Many real-world large scale peer-to-peer networks are unstructured. However, in their basic structure; peer-to-peer suffers high cost when dealing with locating content efficiently due to use of primitive searching and routing technique that uses large overhead and long query time. Therefore it is crucial to select relevant peers to route query message to reduce the search cost and better retrieval in unstructured peer-to-peer network without the loss of the unstructured peer-to-peer identity and characteristics.

Unstructured peer-to-peer networks such as Gnutella[1,2] rely on a random process, that peers are interconnected in a random way. Typically, unstructured peer-to-peer search is based on flooding. Basic unstructured networks which apply flooding technique for propagating user queries is very expensive both in processing time and resources. Several studies have addressed the completeness and scalability issues of flooding[3,4]. Although flood-based search is generally considered inefficient, a number of efforts have been done to improve the searching in unstructured peer to peer to become more efficient

Structured peer-to-peer networks have been developed to provide strict data location management[5,6]. It uses distributed hash tables that assist the routing mechanism to maintain desirable properties for quick lookup. Structured approach offers better search performance in terms of response time and communication overhead when compared to the

Corresponding Author: Iskandar Ishak, Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia

unstructured system. However, despite its highly effective approach, the system lacks partial match lookup capability. The system also incur larger overhead than unstructured peer-to-peer systems especially when the overlay is re-arrange whenever there is failing peers or leaving peers.

This study proposes relevant nearest neighbor based search technique that exploits minimal information in each peer. The algorithm is formulated to achieve efficient search and high retrieval rate in unstructured peer-to-peer networks.

Related work: The earliest technique for peer-to-peer routing is based on the Naïve Breadth-First Search (BFS) algorithm or Flooding. This technique is used in file-sharing peer-to-peer application Gnutella[1]. In this approach, each query from a peer will be broadcasted to all the peers in the network but restricted by the TTL (time to live) value. Flooding may generate O(N) message where N is the number of node. As a result, the query consumes a great deal of processing resources and excessive network. In a worst case situation such as low bandwidth network, flooding could make the network become a bottleneck.

Although, it is a robust and simple technique for query routing but it involves a great deal of communication overhead, that is, high in number of messages. Hop count is also increased exponentially. Some of the messages might visit the same node that has been searched previously. Therefore, communication overhead and scalability are the main problems in this approach. These problems have been proven in a number of papers[2,7,8].

In the random Breadth-First-Search (BFS) approach[9,10], each peer forwards a search message to only a fraction of its peers. Each node randomly selects a subset of peers connected to it and then propagates the search message to those peers. The advantage of this technique is it does not need any global knowledge. Each peer is able to make local decision in a quick manner since it needs only small portion of connected peers to route the query. This approach may generate only a fraction of messages compared to flooding approach.

In random walks[4] approach, the requesting peer sends out a number of query messages to an equal number of random neighboring peers. Each of the query messages will follow their own path in which intermediate peers will forward the messages to randomly chosen neighbor. These query messages are known as walkers. The walkers will be terminated on both success and failure occasions and determined through the use of TTL of the messages or through the use of checking method in which the walker periodically contact the source peer whether the termination condition is met. It achieves significant message reduction when compared to the flooding approach. However, the success rate and the number of hits depend largely on the network topology and the random choices it made.

Another unstructured peer-to-peer searching approach is the Directed BFS combined with the Most Result in the Past in[11]. Each peer forwards a search message to a number of peers which returned the most results for the last most current queries. The nature of this approach is it allows peers explore larger network segments and find most stable neighbors.

A content-based searching for peer-to-peer based system is proposed in[12]. In this approach, each peer will have a special index called filters to facilitate query routing only to those that may contain relevant information. Each peer maintains one filter that summarizes all documents that exist locally in the peer, called local filters. A merged filters is the filter that summarizing the document of a set of its neighbors. When a query reaches a peer, the peer will check its local filter and uses the merged filter to route the query to the peers whose filters match the query.

Intelligent Search Mechanism[9] is proposed as a searching technique based on the similarity of the query. In this approach, each peer has its own profile table that stores the information they get from peers that answered their queries. The information stored in Table 1 is the query ID, peer ID and the query keywords that have been answered and also the query hit. Only the latest peer that answered the query will be kept into a table. Routing is based on the similarity values of the query word with the keyword from the past queries stored in the profile. Peers that have high similarity with the query will be selected for routing.

Ant Colony optimization is also used in unstructured peer-to-peer search in[13]. The approach is called SemAnt where it emulates the nature of ants cooperating between themselves to find food based on the pheromone. The peers are the ones who act like an ant and cooperated between them in creating pheromone trails. The pheromone trails is the probabilistic overlay networks and also indicates the most promising path for a given query. As a result, the more popular a query becomes, the better the trail. The experiments shown that the search algorithm is stable, robust and converges fast whole its performance is pretty much acceptable.

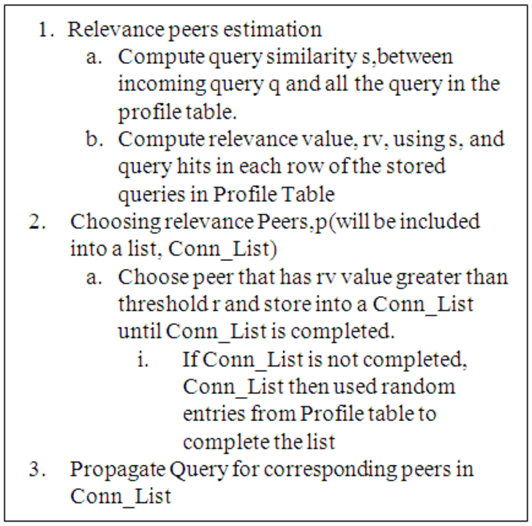
Table 1: Profile table

Query	ID	Connection and hits	Timestamps
Amazon rain forest	123112	P2(34), P34(2), P5(56)	10211
Gulf oil rigs	124451	Null	10222
Waste disposal	144512	P4(34), P8(4)	10233

MATERIALS AND METHODS

Relevance-Nearest Neighbor based search (RNN) using query feedback and similarity: Our search approach consist of 3 components; profile table, relevance peers estimation and nearest neighbor selection. Profile table is used to store past query message and query hits from other peers who have answered previous queries. The Relevance Nearest Neighbor based Search or RNN is based on the concept of giving the same weight both on the query hits and query content similarity with the incoming query and selecting only the nearest relevant peers. We also include the nearest neighbor approach to minimize the search cost but at the same time able to retrieve high query hits or recall. The search method is basically a flooding based search but is based on selective flooding. The search algorithm is shown in Fig. 1. The objective of this searching approach is to have an efficient search and high recall.

For a peer $p \in P$, we use T(p) to denote set of past query maintained by p. Each item in T(p) has two attributes; query term, q and number of hits n, so we denote each data item in T(p) as a pair of q(p), n(p). By using each q and n on T(p), we determine the relevance of a peer by using the similarity of all q(p) in T(p) to calculate the similarity between them. On the other hand, we use n(p) to calculate the stability of each peer in T(p). We define a reference point, which is the highest or optimum value of query similarity and also the highest query hits denoted as d point.



```
1. Relevance peers estimation
    a. Compute query similarity s,between
       incoming query q and all the query in the
       profile table.
    b. Compute relevance value, rv, using s, and
       query hits in each row of the stored
       queries in Profile Table
2. Choosing relevance Peers,p(will be included
   into a list, Conn_List)
    a. Choose peer that has rv value greater than
       threshold r and store into a Conn_List
       until Conn_List is completed.
        i. If Conn_List is not completed,
           Conn_List then used random
           entries from Profile table to
           complete the list
3. Propagate Query for corresponding peers in
   Conn_List
```

Fig. 1: Relevance Nearest Neighbor Search algorithm that uses the Selective flooding concept

Relevance peer estimation: The relevance-based component is based on the work in[14]. This component uses the peer similarity-hits graph model. Our peer-similarity graph model captures both the peer query hits and peer similarity with corresponding incoming query (Fig. 2). We used both information gathered in the profile table which is based on the work done in[9]. We incorporated both, query content and connection stability information to determine relevant peer to route query. Each peer stores information about past queries and the query hits in a table. There will be no global knowledge shared between all the peers but each peer will also have a list of data collected from the answered query and store it in neighbor profile table (Table 1).

The profile table contains the ID of the answering peer, connection ID, the query words that have been answered by other peers and a timestamp of the returned query. These query words are the words that matched the query sent by this peer and the words are contained in the peer are only answered query words. The list will keep the last M queries and a Least Recently Used (LRU) policy will keep the most recent queries in the table.

The relevance value will be based on two parameters, query hits and the similarity value between the query to be routed and the stored past queries. Query hits determine peer connection stability with the processing peers. The more query hits, the more stable the peer is connected and thus giving the impression of the particular peers connection reliability. Similarity value is based on cosine similarity (1). q is the incoming query while q_i is the past queries stored in the profile table in each peer. As an example, let peer A has a list of past queries; d.

Query q is an incoming query and is waiting to be routed q will be compared with all the queries, q_i, in d. The similarity between them will determine the relation between the content that the particular peer has in its storage with the query terms. The most relevance of peers is peers that have among the most query hits and it has a content related to the incoming query.

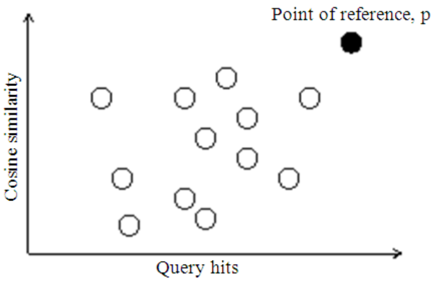


Fig. 2: Similarity and Query hits metric space

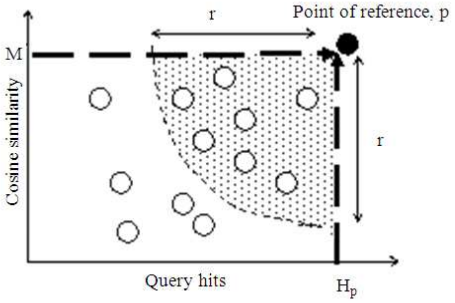


Fig. 3: Point of reference estimation

We calculate the relevance value using formula described in (3). In this formula, we are actually calculating the distance of relevance value of the peers inside the profile table with the most optimum relevance point in the similarity-query hits graph. M is the maximum cosine value, in which for the purpose of easy calculation, we decided to define M = 1. h_i is the returned hits values for a particular query, while H_p is the maximum hits retrieved from all h that have been recorded. The formula to define the maximum hits (2), involved the use of nearest-neighbor concept, which will be explained later. N_p is the total number of query hits of all peers stored in the neighbor profile table:

$$\text{sim}(q,q_i) = \frac{\Sigma(q * q_i)}{\sqrt{\Sigma(q)^2 * \Sigma(q_i)^2}} \quad (1)$$

$$H_p = 2(h_i) \quad (2)$$

$$R(q,q_i) = \sqrt{\left(\frac{H_p - h_i}{N_p}\right)^2 + \left(M - \text{sim}(q,q_i)\right)^2} \quad (3)$$

Nearest neighbor selection: We determine our group of peers within the relevance value by using the nearest neighbor principal. It is based on the fast algorithm for nearest neighbor search proposed in[15]. The purpose of the application of nearest neighbor method is to avoid comparing all the peers inside the table because table with size of N will require N times comparison and relevance calculation process. As we can see in Fig. 3, “nearest neighbors” in our context are the peers that resides within the area of radius r. However, instead of selecting a static value[14], we decided to make the value r dynamic. We determined the dynamic r value by exploiting the inequality of triangle (Fig. 4). The inequalities will determine a bound for peer selections and therefore, less relevance values will be compared to the distance r (Fig. 5).

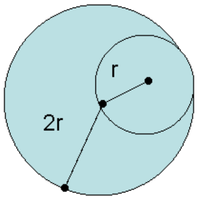


Fig. 4: Doubling the normal search radius

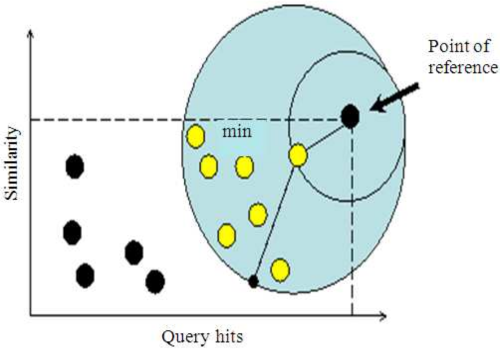


Fig. 5 Nearest-neighbor search space

RESULTS

We evaluate the performance of our searching approach by extending a peer-to-peer simulator called Peerware. We generate 1020 peers with a total of 95676 documents. Each node holds random number of documents between 5-1486 documents. The document collection used is the Reuters-21578 document collection which appeared on the Reuters newswire in 1987. Three different number of query set are used; q100 that contain 100 random query terms; q75 contains 75 queries and q50 with 50 query terms. Each query terms contain between two to five words. Each peer is country-based and each peer holds news about that particular country. One country could have more than one peer representatives in the network.

We compare our approach with the Most-Query Hits (MQH) and Intelligent Search Mechanism (ISM) approaches. We compare the search approaches based on query recalls, number of messages used and search efficiency. Recalls are the number of query matches with the content of each peer, while number of messages used is the number of messages used to answer a query. Search efficiency is the performance evaluation parameter that is calculated by dividing recalls with number of messages used:

$$\text{Search efficiency} = \frac{\text{Recall}}{\text{Messages used}} \quad (4)$$

DISCUSSION

Our experiments showed that our approach (RNN) is efficient in terms of network usage compared to the other two searching techniques. The experiments that employed q50 showed that our searching approach recorded the highest recall when compared to the ISM and MQH techniques. The recall is 4.87% higher compared to the MQH approach and slightly over than ISM approach with 1.67% more than ISM recall (Fig. 6). Figure 7 shows the message usage for all three approaches for query set q50 in which our approach recorded the highest number of message usage than MQH and ISM (4.79 and 2.1% respectively). However, we recorded highest efficiency with 38.1% higher than MQH and 28.8% efficiency higher than ISM (Fig. 8).

RNN also recorded highest recall when employing the q75 query set in the experiment, our approach still managed to record highest recall; as shown in Fig. 9. The RNN search recorded 4.31% higher recall than MQH and 0.85% higher recall than ISM. The RNN recorded 9.4% messages higher than MQH but it yield better search efficiency with 16.5%. RNN also recorded higher number of messages used than the ISM with 5.12% higher but the RNN approach registered 20.62% higher search efficiency than ISM. Messages used and search efficiency graph for experiment using the query set q75 are shown in Fig. 10 and 11 respectively.

Our experiment using q100 query set also showed that RNN approach recorded high recall than other search technique. As shown in Fig. 12, RNN recorded 1.67% higher recall than MQH approach and 0.22% higher recall than ISM. In terms of message usage,

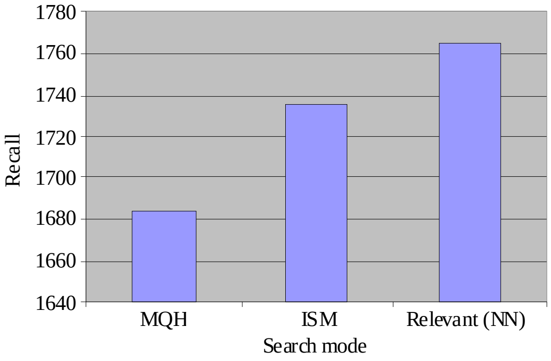


Fig. 6: Recall recorded for 50 queries searching

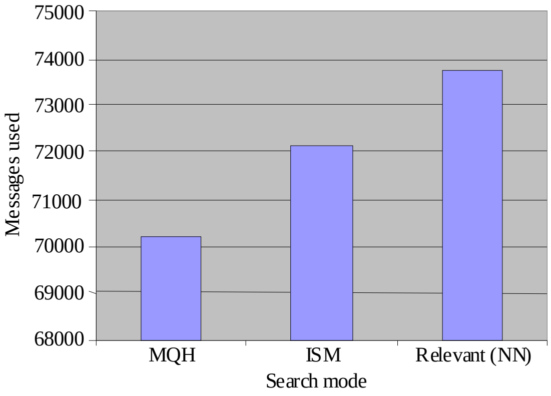


Fig. 7: Messages used for 50 queries searching

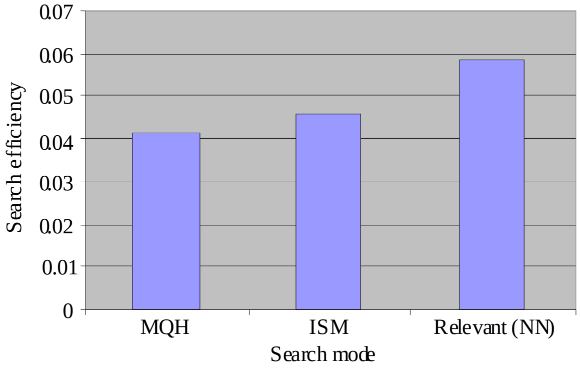


Fig. 8: Search Efficiency for 50 queries searching

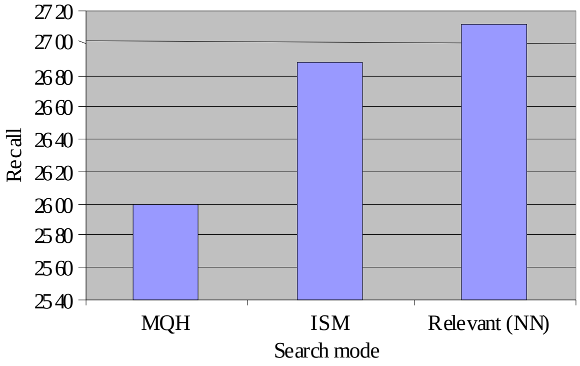


Fig. 9: Recall recorded for 75 queries searching

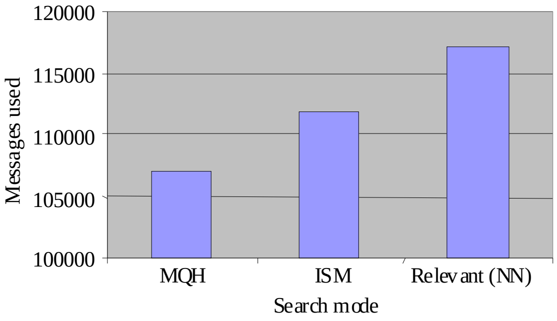


Fig. 10: Messages used for 75 queries searching

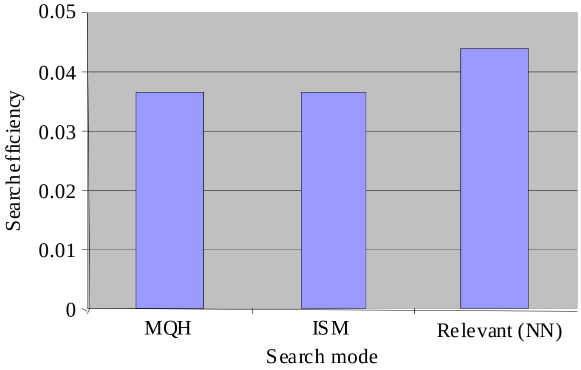


Fig. 11: Search efficiency for 75 queries searching

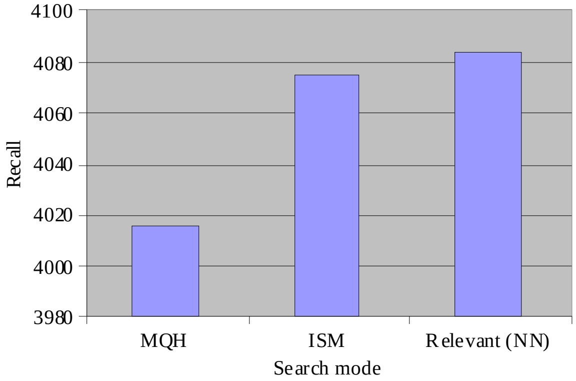


Fig. 12: Recall recorded for 100 queries searching

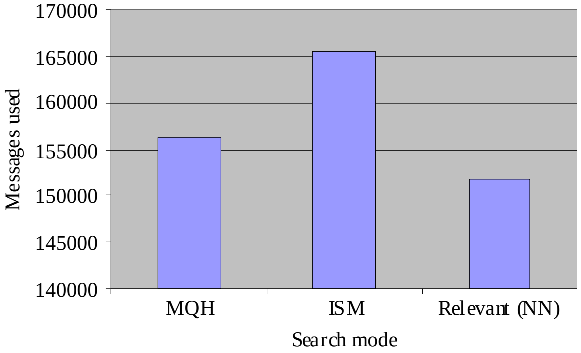


Fig. 13: Messages used for 100 queries searching

we can show in Fig. 13 that RNN approach recorded 3.01% less message than MQH and 8.5% messages less than ISM approach. In the same experiment setting, we found out that RNN search is the most efficient with 13.51% more efficient than MQH and 17.95% more efficient than ISM (Fig. 14).

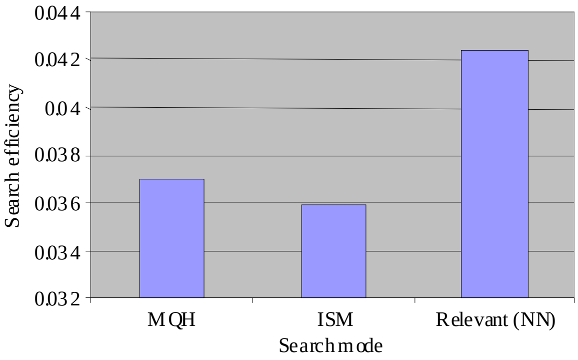


Fig. 14: Search Efficiency for 100 queries searching

CONCLUSION

This study exploits the query content and query feedback data for developing flood-based search in unstructured peer-to-peer that is minimal in cost but giving high retrieval. RNN exploits very minimal data and also nearest-neighbor concept to reduce the cost of searching in unstructured peer-to-peer networks. Our simulation tests showed that our searching approach performs better than the other two flood-based searching approaches that also use minimal data and local indices. We showed that by using minimal information of query hits and similarity, efficient search in unstructured peer-to-peer can be achieved

ACKNOWLEDGEMENT

The researchers would like to express their cordial thanks to Demetris Zeinalipour for giving the permission to use the full data set of Reuters-21578 document collection and the Peerware simulator.

REFERENCES

1. Grossel, Y. and R. Manfredi, 2009. Gtk-Gnutella. http://gtk-gnutella.sourceforge.net
2. Ripeanu, M., 2001. Peer-to-peer architecture case study: Gnutella network. Proceeding of the 1st International Conference on Peer-to-Peer Computing, Aug. 27-29, IEEE Xplore Press, USA., pp: 99-100. DOI: 10.1109/P2P.2001.990433
3. Cohen, E. and S. Shenker, 2002, Replication strategies in unstructured peer-to-peer networks. Proceedings of the 2002 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 19-23, ACM Press, New York, USA., pp: 177-190. http://portal.acm.org/citation.cfm?id=633043

4. Lv, Q., P. Cao, E.C.A.T. Labs-Research, K. Li and S. Shenker, 2002. Search and replication in unstructured peer-to-peer networks. Proceedings of the 16th International Conference on Supercomputing, June 22-26, ACM Press, New York, USA., pp: 84-95. http://portal.acm.org/citation.cfm?id=514206
5. Ratnasamy, S., P. Francis, M. Handley, R. Karp and S. Shenker, 2001. A scalable content-addressable network. Proceedings of the 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, ACM Press, San Diego, California, USA., pp: 161-172. http://portal.acm.org/citation.cfm?id=383072
6. Stoica, I., R. Morris, D. Karger, M.F. Kaashoek and H. Balakrishnan, 2001. Chord: A scalable peer-to-peer lokup service for internet applications. Comput. Commun. Rev., 31: 149-160. http://direct.bl.uk/bld/PlaceOrder.do?UIN=105697340&ETOC=RN&from=searchengine
7. Ramanathan, M.K., V. Kalogeraki and J. Pruyne, 2002. Finding good peers in peer-to-peer networks. Proceeding of the International Symposium on Parallel and Distributed Processing, Apr. 15-19, IEEE Computer Society, Washington DC., USA., pp: 24-31. DOI: 10.1109/IPDPS.2002.1015499
8. Kwok, S.H., K.Y. Chan and Y.M. Cheung, 2005. A server-mediated peer-to-peer system. ACM SIGecom Exchanges, 5: 38-47. http://portal.acm.org/citation.cfm?id=1120686
9. Zeinalipour-Yazti, D., V. Kalogeraki and D. Gunopolus, 2005. Exploiting locality for scalable information retrieval in peer-to-peer networks. Inform. Syst., 30: 277-298. http://portal.acm.org/citation.cfm?id=1187497
10. Dimakopolous, V. and E. Pitoura, 2003. A peer-to-peer approach to resource discovery in multi-agent systems. Lecture Notes Comput. Sci., 2782: 62-77. DOI: 10.1007/b12011
11. Yang, B. and H. Garcia-Molina, 2002. Efficient search in peer-to-peer networks. Proceeding of the International Conference on Distributed Computing System, (ICDCS'02), Vienna, Austria, pp: 1-32. http://www.cs.utexas.edu/~browne/CS395Tf2002/Papers/GarciaMolina-showDoc.pdf
12. Koloniari, G. and E. Pitoura, 2004. Content-based routing of path queries in peer-to-peer systems. Adv. Database Technol., 2992: 29-47. http://www.springerlink.com/index/9tc9e0ep063medy7.pdf
13. Michlmayr, E., 2006, Ant algorithms for search in unstructured peer-to-peer networks. Proceeding of the 22nd International Conference on Data Engineering Workshop, IEEE Computer Society, Washington DC., USA., pp: 142-142. DOI: 10.1109/ICDEW.2006.29
14. Ishak, I. and N. Salim, 2008. Exploiting query feedbacks for efficient query routing in unstructured peer-to-peer networks. Proceeding of the International Conference on Multimedia, Internet and Web Engineering, August, World Academy of Science, Engineering and Technology, Singapore, pp: 1-5. http://www.waset.org/pwaset/v32/v32-67.pdf
15. Orchard, M.T., 1991. A fast nearest-neighbor search algorithm. Proceeding of the International Conference on Acoustics, Speech and Signal Processing, Apr. 14-17, IEEE Xplore Press, USA., pp: 2297-2300. DOI: 10.1109/ICASSP.1991.150755