IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 1- winter 2014 (23-31)

A Survey of Advanced Search Techniques in Unstructured P2P Networks

Fatemeh Sharifkhani
Department of Electrical Engineering
Science & Research Branch, Islamic Azad University
Tehran, Iran
Leila_sharif@yahoo.com

Mohammad Reza Pakravan
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
Pakravan@sharif.edu

Received: May 9, 2013- Accepted: August 20, 2013

Abstract— **In unstructured peer to peer networks, any peer might share any file with other nodes. This uncertainty of where a specific file is located, makes the search problem in unstructured networks complicated. So far, many search algorithms have been proposed which try to maximize the success rate of an initiated query and minimize the imposed cost of search. In this paper, we survey newly introduced approaches to overcome search process problems. By reviewing these strategies and comparing them with previous search methods, we propose a new classification of informed search algorithms and we conclude that regarding this classification, informed search algorithms should be applied in less dynamic networks while blind search algorithms can be used in small networks. We believe that this taxonomy and the new classification can be useful as a guide for future search algorithm design.**

Keywords- unstructured p2p network; search algorithm

I. INTRODUCTION

Peer to peer networks are growing in popularity as an eligible substitute for traditional client-server models. This revolution in file sharing applications has put the duty of file providing on the shoulders of all participating peers in a p2p network, while left them free to ask for service from any other node in the grid.

However, discovering a desired file among all files that different peers share in different points of the network with considering high levels of content and node dynamism is a big challenge [1]. So far, various search algorithms have been proposed to address this problem. However, the properties of these search algorithms highly depend on the structure of the applied p2p network. Structured p2p networks are tightly controlled and files are placed in specific nodes of the network and therefore, discovering content is completed in a few steps. In unstructured p2p networks, however, peers are free to share any file without being imposed to share unwanted files. In these networks, any file can be shared by any node and in any location of the network, and this makes search problem more complicated. This has drawn the attention of academia to find better ways to make searching in unstructured p2p networks more efficient. The aim of these algorithms is to increase overall success rate while decreasing the generated load as much as possible. Other metrics might be taken into account in designing a search algorithm such as storage cost, processing cost, searching time and peers' available bandwidth [2].

There are surveys on the search techniques in p2p networks which thoroughly have introduced search algorithms to date of their publication and can be found in [3, 4, 5, 6, 7, and 1]

The main focus of this work is on reviewing recent approaches to enhance file discovery in completely unstructured networks. In section II, traditional search approaches in structured and practical unstructured networks are briefly described. In section III, new advances in search algorithms in unstructured networks are introduced. Section IV states a general comparison of the introduced algorithms. In section V, we draw some conclusions from our discussion and will give guidelines for future research in this field.

II. Related Works

In this section we briefly review basic search algorithms in structured networks and discuss how practical p2p file sharing systems work.

A. Structured networks

Structured overlays use a distributed hash table to assign unique key values to data objects and unique identifiers to peers and map the generated keys to live peers in the network. Content Addressable Network (CAN), Chord and Pastry are some examples of structured networks. In CAN [22], the ID space is divided among peers and each peer knows its neighbors' IDs and IP addresses. While receiving a query, each node determines the corresponding ID value of the source node and sends the message to a neighbor which is closest to coordinates of the destination. When a new peer joins the network, one of the ID zones splits in half and the responsibility of the keys stored in that zone is assigned to the new comer node. On the contrary, when a peer leaves the network, one of the failed peer's neighbors takes the responsibilities of the disconnected node. By performing a consistent hash function on the IP address of nodes and identifying data of files, Chord [23] assigns peers and files an m-bit identifier. The length of the identifier should be long enough in order to eliminate the probability hashing two different files to a single to a single identifier. Then, peers are placed in an identifier circle. To distribute keys among peers, each key is assigned to the first NodeID with equal or greater value than the key. This peer is now the *successor* of the key and it is the first peer on the ID circle from the key, *k*. When a node asks for an object, the query goes around the circle until it gets to its proper successor. In case a new peer joins the network, the keys assigned to the successor of the new comer node are reassigned between the two successive nodes. On the other hand, when a node leaves the network, all its keys are assigned to its successive peer.

In pastry [24], a 128-bit node ID is randomly assigned to each participating and joining node to network. By assigning a digit to every b-bit of the ID, the keys and NodeIDs are considered to be a sequence of digits. In this method, each node checks its routing table and sends the query to a node which shares a longer prefix with the key than the peer itself. When a peer joins the network, it gets connected to the closest topological node in the network and sends a joining message throughout the network and the closest node to the joint peer in the ID space is introduced. Then it builds up its routing tables by the help of its topological and virtual neighbors. When a peer leaves the network, its neighbors update their tables by contacting nodes on the other side of failed peer on the ID circle. Interested readers can refer to [25], [7] for more information about structured networks.

B. Practical unstructured networks

In unstructured p2p networks, any peer might share any file and this makes the search problem in these networks more challenging. So far, many unstructured p2p networks have been practically used. Napster was one of the first systems which gained a large popularity among internet users. However, the central file index of Napster caused it to be shut down by court order.

Gnutella [7, 8] is the first decentralized file sharing p2p system which became the most popular file sharing network in 2007. In this system, each node floods its request to its neighbors and if they do not find the file in their indexes, they will flood the query to their neighbors. This flooding search method, limits the scalability of Gnutella. To overcome this problem, in Gnutella 0.6 [31, 26] nodes with higher capacity are promoted to superpeers and search is performed in the superpeer layer. These supernodes perform the required search process and leaf nodes with low processing capacity are shielded from being involved in the search procedure. Bearshare, Shareaza, and Limewire are some of Gnutella clients for file sharing application.

Edonkey is another unstructured system which uses a number of servers to accelerate search process in a semi-central manner. In this system, each peer gets connected to a server and uploads its IP address and an index of its available files. The server provides the peer with a list of other servers to which the peer can send its queries. However, the discovered file is downloaded directly from a source peer and the servers are not involved in file transferring procedure.

Fasttrack is another file sharing system which has a hierarchical structure. In this network, stronger nodes form a superpeer overlay and search is performed in this overlay.

Bittorrent is the most popular file sharing application of the recent years. When a peer joins this network, it asks the server for the torrent of its requested file. Server provides the peer with a torrent which indicates file name, file size, its hash and its tracker address. A tracker is a central point which keeps track of all peers involved in downloading or uploading a specific file and is able to provide the contact information of nodes in that swarm to the requesting node. The peer, therefore, contacts the tracker and asks for a list of peers currently downloading the file. Then it contacts these peers and asks for the file pieces and starts downloading the file piece by piece. when the download of a piece is completed, the peer is able to share that piece with other requesters [7].

Although most of practical file sharing systems have applied servers for a more efficient search service, academia has shown a great tendency toward designing search algorithms in absolutely unstructured networks. Next section describes some of recent advances which have enhanced search performance in unstructured networks.

III. Resource Search in Unstructured P2P Networks

In unstructured p2p networks, search problem is more complicated as any peer might share any file in any location of the network. Search algorithms in such networks are categorized in two classes: blind and informed.

A. Blind search algorithms

Blind search algorithms forward the query based on their routing strategy without using any network information. Two basic approaches in this category are flooding and random walk (RW) algorithms. In flooding query is forwarded to all neighbors of the requesting node and if the neighbors do not find the file among their shared files, they will forward the query again to their all neighbors. The drawback of this method is the large amount of load it generates and limits the scalability of the network [8]. On the other hand, random walk search algorithm which forwards the query to a random neighbor is too slow.

To overcome the shortcomings of flooding and random walk methods, different algorithms have been proposed which expanding ring [9] and normalized flooding [10] are few to mention. Dynamic search (DS) algorithm proposed by Lin et al. is another absolutely blind method which takes advantage of benefits of both flooding and random walk. In this method two different search strategies are adopted in two different phases. In the first phase, DS acts like flooding and sends the query to *d.p* percent of neighbors, regarding that d is the link degree of the query source and p is transmission probability. This phase continues until query's hopcount reaches a predefined threshold and after that the search strategy switches to RW and the query is forwarded to a random neighbor. Dynamic search can be combined with knowledge based search algorithms and intelligently forward the query to next nodes.

B. Informed search algorithms

Informed search algorithms try to improve search performance by bringing network information to query routing strategies. These algorithms fall in two main groups: algorithms which use predefined and deterministic information of the network and algorithms which gradually learn about network properties and use this probabilistic adaptive information for query forwarding.

1) Informed search with deterministic information: two traditional examples of this category are Local Index and Routing Index methods. Local index algorithm [10] is an algorithm in which each node sends the index of its available files to nodes within radious r from itself. Each node stores the received indices and processes the query on behalf of all nodes within r hops distance. If the required resource was not found, the query is flooded to the next layer of neighbor nodes.

A variation of LI is proposed in [11] to accelerate search process of LI. In this algorithm called Trace Following Content Locating (TFCL) each node stores its own index with value r and propagates the index in the network. Any node which receives this index, decreases the value by one and sends it to its neighbors. Thus, the value associated with a file name indicates the proximity of the node to the source node. When a query reaches a node which has a trace of required file, it is enough to forward the query to nodes with bigger values. In other words, the query is forwarded to neighbors that are closer to the source.

Routing Index (RI) [12] is another method which utilises deterministic information of the network in its query forwarding policy . In this method, files are categorized according to their subjects and an index is created based on an estimation of the number of accessible files in each category for each neighbor. When a node receives a query and can not answer it, it forwards the query to a subset of neighbors which have access to a larger number of files in the same category of the requested file. This method is not suitable for today's highly dynamic p2p networks.

Lightflood search [13] is another mechanism in this category which uses some topological information for its forwarding decision. This algorithm has two stages. In the first stage, query is flooded within a limited radius of hops. Therefore, the message is spread to a large horizon with minimum redundant messages. In the second stage, last hops of flooding stage become seeds and they initiate search in a tree-like suboverlay called *FloodNet* which can pass the query to every corner of the network. In this stage, peers exchange information about their second neighbors' connectivity degrees. The neighbor with the highset secondary degree can participate as a peer in *FloodNet* and the query is flooded only in this suboverlay. This method uses two facts about flooding. First, a large number of redundant messages are generated within high hops, and second the network coverage growth rate is higher in low hops. Therefore, by combining the explained two stages, lightflood can enjoy the advantages of pure flooding while eliminating a large amount of redundant messages.

To enhance flooding and with regard to the fact of growth of redundant messages in high hops, [14] proposes a search algorithm based on peer division (SOADP) to limit the scope of pure flooding while ensuring that the hit rate is not reduced. In this method, each requesting peer sends Atomic Query (AQ) messages to their direct neighbors and Iterative Query (IQ) messages to their indirect neighbors in layer 3 and 6. If a node receives AQ messages, it only checks its own file list and discards the query. In case an IQ message is received by a node, it searches its file list and forwards the query as an AQ message to its immediate neighbors.

Integrating the idea of dominating set of nodes with cloned random walker technique, [15] proposes a new approach to discover resources in a suboverlay of dominating nodes. The main idea of this method is to enlarge the searching area and increase hit rate by sending a suitable number of walkers at each cloning distance and reduce search load and search time by flooding the query to neighbors' of dominating set nodes. Dominating set is defined as a set of nodes that every node of the network whether exists in that set or has a connection with a node in that set. Once a node receives a query and it has cloning distance with the requester, it clones a number of walkers to its neighbors. If the node is a dominating node, it floods the query to all its immediate neighbors. If the query is received by a normal node, it just randomly forwards the query to one of its neighbors. This algorithm generates a load much lower than flooding while showing a better trade-off between search delay, message overhead and success rate.

A generalized probabilistic flooding algorithm is proposed in [16] which is designed to limit the number of transmitted queries by considering resource distribution and heterogeneity of the number of direct outgoing links to other peers. In this method, each peer selects a neighbor as query's next hop based on a function of its degree and a function of its neighbors' degrees. These two functions also depend on the distance from the query's originator to control the amount of generated load.

To enhance search in unstructured p2p networks [17] tries to predict users' interest model and estimate the probabibility of sharing file f_j by a peer if it already has shared file f_i. To route a query, each peer sends the request message to a neighbor which has the highest probability of sharing that file. To estimate this probability, each file is specified with a set of unique attributes such as artist, composer, genre, album year and rating. For any two files $f_j = \{\alpha'_1,...,\alpha'_k\}$ and $f_i = \{\alpha_1,...,\alpha_k\}$ feature functions are defined to show the correlation between each attribute of the files. If the attributes are related $F(\alpha_1,\alpha'_1)$ will be 1 and otherwise, it will be 0. By giving weights corresponding to significance of each attribute, the probability of a peer's interest in sharing file f_j if it already has shared file f_i can be calculated and the query will be forwarded to a neighbor with higher probability of sharing the requested file f_j.

2) Informed search with adaptive information: in this category of informed search algorithms, nodes gradually learn about their environment and build up tables based on this knowledge. These tables are adapted by the feedback of each performed search.

In intelligent BFS [18, 19] each node creates a table and stores the number of returned results for each query from each of its neighbors. The next time it receives a query for the same file, it sends the query to the neighbor with larger number of returned results. The performance of this method improves over time as nodes gain more information about their environment. However, this algorithm has a poor performance in dynamic networks as it is not designed to adapt itself to departure of nodes. Adaptive Probabilistic Search (APS) [20] is another adaptive informed search in which each node keeps a local index for each queried object and for each directly connected node. At the beginning of the search, nodes have no information about the network and therefore, the probabilities of all neighbors for a requested file are equal. To accelerate APS, the requesting node sends k random walkers to its neighbors. Receiving nodes forward the query to neighbors with the highest probability. If a query hit occurred, the query returns to nodes it has traversed and increases their probability value. If a query failed to find an object, it decreases the probability value of nodes on its path back to the requesting node. This method can achieve high success rate over time, however, it has a poor performance in dynamic environments. Shen et al. in [21] proposed an information diffusion based search algorithm as an improvement over APS in dynamic environments. In this method, an information-diffusion cycle is defined and the probability of success is calculated for the current time. This probability is calculated by adding query hits per request for each file during the information-diffusion cycle on to the historical probability terminated at the last cycle. With this approach, APS can overcome dynamics of the network and enhance its search efficiency and success rate.

In [27] the authors propose a path tracable query routing mechanism by constructing a Tracable Gain Matrix in each peer. The rows of this matrix are a subset of neighbors which have successfully routed a query to a target. The coloumns of this matrix are assigned to queries sent or forwarded through the peer. The gain value of each element of the matrix is caculated by a function of total query hit number at target's neighbors and the distance of target peer to the requesting peer. When a peer receives a query and doesn't find the requested file in its list, it calculates the gain values of its neighbors which had successfully guided the query in previous searches. The query is then forwarded to the neighbor with highest gain value.

Ant Colony Optimization (ACO) search has been the base of some search protocols in p2p networks. This mechanism sends the query to a neighbor with higher probability of answering the request. At the beginning, each ant walks randomly in the network and leaves a small amount of pheromone on its visited nodes. Supposing that pheromone amount left is $\Delta\tau$ and the evaporation coefficient is β, the amount of pheromone on a node will be:

$$\tau_{ij}(t) = (1-\beta).\tau_{ij}(t-1) + \Delta\tau_{ij}(t) \qquad (1)$$

When a query hit occurs, a query hit message takes the same path to the requester and updates routing information. This concept has been applied in [28] to isolate freeriders in the network using the fact that freeriders do not return backward ants.

AntP2PR [29] is another scheme which tries to find the shortest routes to nodes which return a lot of results. To implement this, each node keeps a pheromone table for all possible routes, and each route is represented by the peer one hop away from the node. When a query hit occurs, the amount of pheromone of the rout leading to success is increased proprtional to number of query hits from that route. To enhance ACO-based search [30] proposes sending ICMP packets to neighbors and limits sending ants to those neighbors that are not congested. The amount of pheromone left on each node by the backward ant is proportional to the number of discovered documants and link cost.

C. Overlay reconstructing and replication distribution modifying strategies

In unstructured p2p networks each peer can join and leave the network deliberately and might be connected to any node in the network and share any desired file. Recently, some attempts have been made to enhance search efficiency by reconstructing the overlay or governing file replication over participating peers of the network. Constructing a superpeer layer was one of the first taken steps which tried to shield ordinary peers from processing and searching tasks [31]. Gia is another mechanism which modifies Gnutella network topology in a way that most of network peers get connected to high capacity nodes [32].

Hsiao and Su have proposed an overlay formation in [33] to improve search results in unstructured p2p networks. To achieve this goal, a random graph is reconstructed to satisfy three properties: first, peers try to select neighbors which are most similar to themselves. Second, to rapidly propagate messages between peers, the diameter of the network should be as low as possible, and third, each peer should send the query to one of its neighbors which is most similar to destination than the node itself, hoping that the query is forwarded in an overlay path which gets more similar to destination at each hop.

ISI [35] is another mechanism which tries to reform network topology by replacing randomly selected neighbors with peers who are more cooperative in the network. This method has two main parts which improve search result when they work together. The first part is score updating strategy which gives a score to a recently joint peer, and the score is increased when it uploads contents to other peers. In the second part, which focuses on neighbor selection, each peer is allowed to extend its neighbor list according to its score. This means that more cooperative nodes will have a greater degree. In addition, when a query hit message is received by a requester, it is allowed to substitute one of its randomly selected neighbors with the destination node if the target has a higher score than the current neighbor. This method has a better search result than Gnutella as shown in the paper.

In Diffsearch [36] an overlay of ultrapeers is created in a way that each leaf peer has at least on ultrapeer neighbor. A leaf node can promote itself to an ultrapeer when the number of visited shared files reaches a threshold. Using this query answering heterogeneity, a high query success rate is achieved by searching a small portion of nodes. This technique uses the fact that in Gnutella most of files are shared by a few number of nodes, and therefore, the peers with high query answering capabilities are given higher probability to be queried. To implement this algorithm, it is proposed that ultrapeers connect to each other in a way that they create a cluster. If an ultrapeer looks for a file that exists in a node in another cluster, it sets up a new connection with that node and therefore, two disjoint clusters will be connected together. Another search scheme, Diff-Index [37] suggests using the online time heterogeneity to enhance search efficiency in unstructured p2p networks. This technique uses the observation of that nodes sharing a great amount of files tend to stay in the network for a longer time, and therefore, with querying this small portion of nodes, success rate is increased and search traffic is decreased.

Another algorithm which tries to improve search efficiency and security simultaneously is introduced in [38, 39]. This scheme is based on construction of an overlay of trusted nodes where neighbors are selected based on their trust ratings and content similarities. This strategy gradually forms a new network topology which results in implicit semantic community structures in which peers sharing similar contents form a cluster. Search efficiency is increased in this network as most of queried files are discovered within the community, and security is improved as malicious peers are identified and isolated during search process.

To reduce the number of probed hosts and consequently reduce the overall search load, it is proposed to replicate data on several hosts [34]. The location and the number of replica vary in different replication strategies.

Thampi et al mention in [40, 3] that there are three main site selection policies. *Owner replication* in which the object is replicated on the requesting node and the number of replica increases proportional to popularity of the file. *Random replication* in which replications are distributed randomly and the *path replication* in which the requested file is copied on all nodes on the path between the requesting node and source. Depending on how search algorithm is going to be optimized, file distribution varies. In uniform replication [9] all items have the same number of replicas. In proportional replication this number is proportional to the file request rate. It is discussed in [9] that the number of replicas of a file should be proportional to the size of the file to reduce network bandwidth consumption during the search process. Square root replication distributes the replicas proportional to the square root of request rate and sounds to be a better choice than uniform and proportional replications.

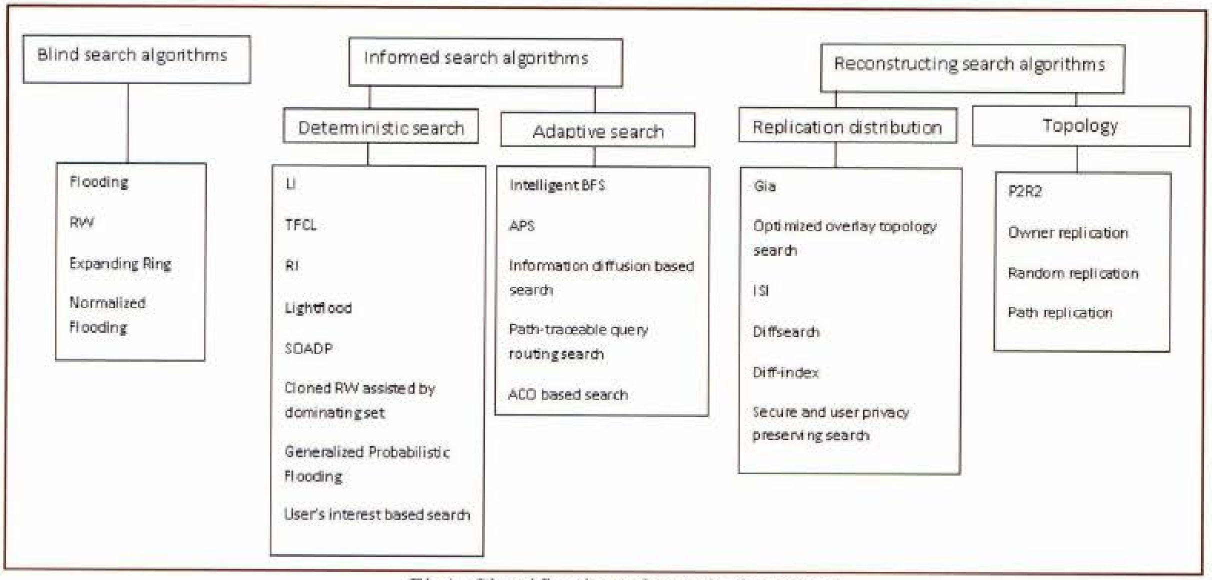

Fig1. Classification of search algorithms

Considering the heterogeneity and dynamics of p2p networks, [41] suggests a replication strategy called P2R2 in which each peer *i* calculates the probability P_{ij} which is the probability of an unsatisfied search for item *j* traversing peer *i*. Then peer *i* decides to accept a copy based on its capacity, file size and probability P_{ij}. This method gives a near optimal search efficiency as it considers heterogeneity of files and peers and dynamics of the network simultaneously.

IV. COMPARISON

In this section, we make a comparison of different search algorithms' capacities in unstructured p2p networks.

A. Load

The overall efficiency of a search algorithm is influenced by the generated search and advertisement loads. Blind search algorithms impose no advertisement load to the network since they are not designed to make decision based on any information. Despite random walk that creates a negligible load in the network, flooding generates an uncontrollable large load which mostly consists of redundant messages. DS tries to moderate these two algorithms and offer an acceptable success rate with very few redundant messages.

Informed search algorithms produce a large load to transmit network information between nodes. Adaptive informed search methods, such as APS and ACO, usually use the information brought by the feedback agents, and therefore, this information transmission load is almost equal to the generated search load. However, deterministic informed search algorithms which have to relay information tables between nodes impose a large load to the network. This load is even larger, if nodes try to update their tables in high dynamics of the network.

B. Success rate

Blind search algorithms are independent of dynamics of the network and therefore, their performances do not change over time. Broad network coverage of flooding gives a high success rate to this method. On the contrary, random walk method hardly completes the search especially if it is looking for a rare file. DS has a success rate comparable with flooding as it just tries to reduce redundancy.

In a static environment, adaptive informed search algorithms improve their success rate as time goes by since in the course of time nodes learn more about their environment and build up routing tables. However, in a dynamic network in which learning rate is less than the rate of network change, adaptive search algorithms have no chance for improving their success rate. On the other hand, routing tables of deterministic approaches lose their validity over time and degrade in performance because of network dynamics.

TABLE I. COMPARISION OF BLIND SEARCH ALGORITHMS

	Load	Success Rate	Scalability	Fault-Tolerance	Load Balancing
Flood	Large amount of redundant search load	Very good	Limited in large TTL due to redundant messages	Resilient	High capacity nodes might be discovered with a high probability
RW	Small amount	Very bad	SR is decreased for rare files	Resilient	very bad
Dynamic Search	Message production is less than flooding	Very good	SR is decreased for rare files	Resilient	High capacity nodes might be discovered with a high probability

TABLE II. COMPARISION OF INFORMED SEARCH ALGORITHMS

	Load	Success Rate	Scalability	Fault-Tolerance	Load Distribution
Local Index	High information transition load Low search load	good in static network because of routing information	Acceptable SR in larger networks	Not resilient	Not good
Routing Index	High information transition load Low search load	good in static network because of routing information	Acceptable SR in larger networks	Not resilient	Not good
Light Flood	Less than flooding	good because of vast coverage	It can adapt itself to size growth of network by growing FloodNet suboverlay	Resilient	Due to vast coverage of network, high capacity nodes might be discovered with a high probability
SOADP	Less than flooding	Less than flooding	SR is decreased for rare files	Resilient	Not good
Cloning Random Walker Assisted by Dominating Set	Less than flooding	good because of vast coverage	With increasing the dominating set, SR can remain high	Resilient	Due to vast coverage of network, high capacity nodes might be discovered with a high probability
Generalized Probabilistic Flooding	Less than flooding	good because of vast coverage	Supports scalability	Resilient	Due to vast coverage of network, high capacity nodes might be discovered with a high probability
Search Based on User's Interests	High information transition load Low search load	good because of routing information	SR is decreased for rare files	Resilient	Not good
Intelligent BFS	Less than flooding	good in static networks because of vast coverage	SR is decreased for rare files	Not resilient	Not good
APS	Comparable with RW	Improves over time Performs very bad in dynamic networks	SR is decreased for rare files	Not resilient	Not good
Information Diffusion	Comparable with RW	Improves over time Performs very bad in dynamic networks	SR is decreased for rare files	More resilient than APS	Not good
Path Traceable Query Routing	Comparable with RW		SR is decreased for rare files	Not resilient	Not good
Algorithms Based on ACO	Comparable with RW	Improves over time Performs very bad in dynamic networks	SR is decreased for rare files	Not resilient	Not good

C. Scalability

Success rate of a good search algorithm should be independent of network size. Popular files are assumed to be discovered within a few steps from a requester and therefore, a local search can lead the query to the source node. However, when a file is rare and is far away from the requesting node, blind search algorithms do not perform very well. Informed search algorithms, both deterministic and adaptive, has this ability to route the query through nodes which are far away from the source node, but have some information about its location. This property increases the success rate of informed search algorithm and therefore, gives a good scalability to the network which applies an informed method.

D. Fault-tolerance

In unstructured networks, nodes can join and leave the overlay without causing any disruption in network maintenance and normal operation of the system [1]. Blind search algorithms are hardly affected by dynamics of the network, since they investigate the existing peers in a network.

Adaptive informed search algorithms are very fragile in a dynamic environment, especially if their learning rate does not compensate the effects of dynamics of the network.

Routing tables of deterministic search algorithms lose their validity in the course of time and therefore, they are not resilient to join and departure of nodes in a grid. To update routing tables at a suitable rate, a large load is imposed to the network.

E. Load distribution

Most of attempts described in section III, part *C*, regard to the fact that consisting nodes of a network are heterogeneous in terms of update bandwidth, processing capacity, online time and rate of query answering.

With connecting more links to high capacity nodes, Gia increases the probability of these nodes to be searched. To best utilize processing capacities of participating nodes, Gnutella 0.6 performs the search in superpeer level and shields weak peers from being involved in this process.

In Diffsearch and ISI, search load is directed to nodes with a higher probability of answering the queries. Diff- Index tries to distribute the load over nodes with a higher online time.

As this study showed, the strategies which are trying to exploit network heterogeneity mostly construct an overlay or reconstruct the network graph. On the other hand, other search algorithms in blind and informed category focus on improving success rate and decreasing search load with no attention to load balancing problem.

V. Conclusions

In this paper, a few techniques to improve search efficiency in p2p networks are discussed. By studying these methods, it can be derived that blind search algorithms are well compatible with dynamics of the network. However, flooding imposes a large searching load to the network and random walk imposes a long searching time. Hybrid search algorithms which combine these two algorithms try to balance the pros and cons of flooding and random walk. Informed search algorithms, on the other hand, perform efficiently in static networks. However, their performance is degraded drastically in highly dynamic networks. To update tables in informed search algorithms with deterministic information, a large load is generated and an update protocol should be designed. The impact of network dynamics in informed search with adaptive information is even worse as in these methods peers are supposed to gradually learn about their environment and route the request based on what they have learned. However, network dynamics cause their gathered information get invalid and expired rapidly. Consequently, informed search algorithms are more suitable for networks with low dynamics and blind search algorithms are appropriate for highly dynamic but small networks. The best way to design an efficient search algorithm is to use the benefits of both blind and informed search algorithms simultaneously. We believe that by clustering peers according to their interests and what they share, we can guide the query to a corresponding cluster and blindly search within that cluster. This suggestion sounds to be appropriate as a guideline for designing search algorithms since general content dynamism in a cluster is smoother than the whole network over time. Hence, informed search algorithms can be applied to route the request message to a cluster, and the query continues its search in the cluster according to a blind protocol in order to tackle the problem of high rate of join and departure of peers.

References

[1] R. Ahmed, and R. Boutaba, "A survey of distributed search techniques in large scale distributed systems," IEEE Communications Surveys and Tutorials, vol. 13, no.2, pp. 150–167, Second Quarter 2011.

[2] F. Sharifkhani, and M. Pakravan, "A new metric for comparison of p2p search algorithms", in Proc. International Conf. p2p, parallel, grid, cloud and internet computing(3PGCIC),Victoria, Canada, pp. 191–195, November 2012

[3] S. M. Thampi, and K. C. Sekaran, "Survey of search and replication schemes in unstructured p2p networks," Network protocols and algorithms, vol. 2 no.1, pp. 93-131, 2010.

[4] C. Kang, "Survey of search and optimization of p2p networks," Peer to peer networking and application, vol. 4, no. 3, 2011.

[5] R. Ranjan, A. Harwood, and R. Buyya, "Peer-to-peer-based resource discovery in global grids: a tutorial," IEEE Communications Surveys and Tutorials, vol. 10, no.2, pp. 6–33, Second Quarter 2008

[6] E. Meshkova, J. Riihijarvi, M. Petrova, and P. Mahonen, "A survey on resource discovery mechanisms, peer to peer and service discovery frameworks," Computer networks, vol. 52, pp. 2097-2128, 2008

[7] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer to peer overlay schemes," IEEE Communications Surveys and Tutorials, vol. 7, no.2, pp. 72–93, Second Quarter 2005

[8] J. Buford, H. Yu and E. Lua, P2P Networking and Applications, Morgan Kaufmann, 2008

[9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks,"in Proc. 16th ACM Workshop Supercomputing, 2002

[10] B. Yang, and H. Garcia-Molina, "Improving search in peer-to-peer networks," in Proc. 22th IEEE Int. Conf. Distributed Computing, 2002

[11] K. Ronasi, M. H. Firooz , M. Pakravan, and A. N. Avanaki "An enhanced random walk method for content locating in p2p networks," in 27th International Conf. Distributed Computing Systems Workshops, 2007

[12] A. Crespo, and H. Garcia-Molina, "Routing indices for peer to peer systems," in Proc. 22th IEEE International Conf. Distributed Computing, 2002

[13] S. Jiang, L. Guo, E. Cohen, X. Zhang, and H. Wang "Lightflood: minimizing redundant messages and maximizing scope of peer-to-peer search," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 5, pp. 601-614, May 2008

[14] L. Jianshun, Z. Tianwu, W. Fayi, L. Jianyong and H. Daoying, "Search algorithm based on peers division in unstructured p2p networks,"in Proc. 5th Int. Conf. Intelligent Computation Technology and Automation, 2012

[15] J. S. Leu, C. W. Tsai, and W. H. Lin, "Resource searching in an unstructured p2p network based on Cloning Random Walker assisted by Dominating Set," International Journal of Computer and Telecommunications Networking, vol. 55, no. 3, pp. 722-733, February 2011

[16] R. Gaeta, and M. Sereno, "Generalized probabilistic flooding in unstructured peer-to-peer networks," ," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 12, pp. 2055-2062, December 2011

[17] G. Chen, C. P. Low, and Z. H. Yang, "Enhancing search performance in unstructured p2p networks based on users' common interest," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 6, pp. 821-836, 2008

[18] D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "Information retrieval in peer-to-peer networks," Computing in Science and Engineering, vol. 6, no. 4, pp. 20-24, 2004

[19] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in Proc. 11th Int. Conf. Information and Kowledge Management, pp. 300-307, 2002

[20] D. Tsoumakos, and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," in Proc. 3rd Int. Conf. P2P Computing, pp. 102-109, 2004

[21] X. J. Shen, Z. Q. Jiang and X. Y. Zhang, "Search algorithm based on information diffusion for unstructured peer to peer networks," Electronics Letters, vol. 47, no. 18, pp. 1026-1027, 2011

[22] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A scalable content-addressable network," in Proc. ACM SIGCOMM, 2001

[23] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan, "Chord: a scalable peer to peer lookup service for internet applications," in Proc. ACM SIGCOMM, pp. 149-160,2001

[24] A. Rowstron, and P. Druschel, "Pastry: scalable distributed object location and routing fo large-scale peer-to-peer systems," IFIP/ACM Intl. Conf. Distributed Systems Platforms, Germany, 2001

[25] Y. Zhao, and C. He, "Survey of resource-searching algorithms in structured p2p network," in 2nd Intl. Conf. Information Science and Technology, pp. 4738-4741, 2010

[26] M. Hong, J. Holliday, and C. Dong-sub, "Optimal superpeer selection for large scale p2p systems," in Int. Conf. Hybrid Information Technology, 2006

[27] M. Xu, S. Zhou, J. Guan, and X. Hu, "A path-tracable query routing mechanism for search in unstructured peer-to-peer networks," Journal of Network and Computer Applications, vol. 33, no. 2, pp. 115-127, 2010

[28] W. C. Jen, Y. K. Hsiang, and H. J. Ming, "Antsearch: an ant search algorithm in unstructured peer to peer networks," in Proc. 11th IEEE Symposium on Computers and Communications, 2006

[29] F. Loukos, H. D. Karatza, and C. X. Mavromoustakis, "AntP2PR: an ant intelligence inspired routing scheme for peer to peer networks," Simulation Modelling Practice and Theory, vol. 19, no. 2, pp. 649-661, 2011

[30] T. Dongming, L. Xianliang, and L. Yang, "ACO-based search in unstructured p2p network," in Int. Conf. Information Technology, Computer Engineering and Management Sciences, 2011

[31] T. Klingberg, and R. Manferedi, "Gnutella 0.6," in http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[32] Y. Chawate, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shanker, "Making gnutella-like p2p systems scalable," in Proc. 2003 Conf. Applications, Technologies, Architecture, and Protocols for Computer Communications, pp. 407-418, 2003

[33] H. Hung-Chang, and S. Hong-Wei, "On optimizing overlay topologies for search in unstructured peer to peer networks," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 5, pp. 924-935, May 2012

[34] E. Cohen, and S. Shenker, "Replication Streategies in unstructured peer to peer networks," in Proc. 2002 Conf. Applications, Technologies, Architecture, and Protocols for Computer Communications, pp. 177-190, 2002

[35] H. Feng, S. Tijiang, L. Zengdong, and G. Zhongwen, "ISI: integration of search and incentive strategy in p2p systems," Int. Conf. Internet Technology and Applications, 2011

[36] C. Wang, and L. Xiao, "An effective p2p search scheme to exploit file sharing heterogeneity," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 2, 2007

[37] Y. Gong, H. Deng, Z. Gu, J. Hu, and Y. Wen, "An efficient search scheme in unstructured peer-to-peer networks," in Advanced Materials Research, vol. 211-212, pp.295-299, 2011

[38] J. Sen, "An efficient, secure, and user privacy-preserving search protocol for peer to peer networks," Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence, Studies in Computational Intelligence, vol. 460, pp. 279-320, Springer, 2013

[39] J. Sen, "A secure and efficient searching scheme for trusted nodes in peer-to-peer network," in Proc. 4th Intl. Conf. Computational Intelligencein Security for Information Systems, pp. 100-108, 2011

[40] S. Thampi, and S. Sekaran, "Review of replication Schemes for unstructured p2p networks," IEEE Intl. Conf. Advance Computing, 2009

[41] M. Sozio, T. Neumann, and G. Weikum, "Near- optimal dynamic replication in unstructured p2p networks," in Proc. 27th ACM SIGMOD- SIGACT- SIGART Symposium on Principles of Database Systems, pp. 281-290, 2008

Fatemeh Sharifkhani, received her B.Sc. degree in Electrical Engineering from University of Tehran in 2003. She obtained her M.Sc. degree from Science and Research Branch of the Islamic Azad University, Tehran, Iran in 2006 and started her Ph.D. program in the same year. Her current research interests include peer to peer networking and bio-inspired networking.

Mohammad Reza Pakravan graduated as a top ranking B.Sc. student from University of Tehran in 1990. He obtained his M.Sc. and Ph.D. degrees from University of Ottawa in 1994 and 2000 respectively. He worked in Nortel Networks for more than three years as a member of scientific staff. Dr. Pakravan joined Faculty of Electrical Engineering at Sharif University of Technology in 2001. His research interests are in the area of data networking protocols and algorithms, broadband wireless and wireline communications.