International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.2, Mar-Apr 2012 pp-114-120 ISSN: 2249-6645

Pondering of Fundamental Search Methods and Protocols for Unstructured Peer To Peer NETWORKS

Shanmukha Babu.kotha[1], Anilkumar.talasila[2],Venkata vara Prasad.padyala [3],Sahitya.Kandimalla

[1]PG Scholar, CSE. K L University , Andhrapradesh, India,
[2]PG Scholar, CSE, K L University, Andhrapradesh, India,
[3]Assit.Professor, CSE, K L University, Andhrapradesh, India,
[4]Assit Professor, CSE RVR&JC college of engineering and technology, AP,India,

Abstract

Designing proficient search algorithms is a major aspect in unstructured peer-to-peer networks, because Search algorithms offer the capabilities to trace the queried resources and to route the message to the target node. Different techniques can be used to trace resources on the network. If the network is small, no intricate search techniques are needed. We can use simple broadcasting or multicasting for querying. Centralized systems with a small number of servers also do not have need of intricate query propagation techniques. However, if we want to sustain intricate queries in decentralized networks, such as unstructured P2P overlays, complicated search techniques have to be applied to query propagation to attain scalability. In this paper, we discuss the most general search algorithms and example protocols that make use of these methods and hence we illustrate the importance of search algorithms in unstructured P2P networks.

Index Terms: *Search algorithm, peer-to-peer, Unstructured networks.*

-- *** --

1 INTRODUCTION

Peer-to-peer networks are widely used for file sharing purposes. This type of usage tends to favor resilient, decentralized architectures over centralized solutions. However this comes at a penalty in ease of searching. At first, peer-to-peer systems addressed this shortcoming by incorporating a flooding mechanism for resource discovery. A node in the peer-to-peer network broadcasts a query message to its neighbours.The neighbors in turn are responsible for reporting any matches as well as forwarding the message to its neighbors, if necessary. This mechanism has been proven effective in practice for finding items which are prevalent across the peer-to-peer network, but otherwise ineffective and resource consuming. Previous works about search algorithms in unstructured P2P networks can be classified into two categories: breadth first search (BFS)-based methods, and depth first search (DFS)-based methods. These two types of search algorithms tend to be inefficient, either generating too much load on the system [1], [2], or not meeting users' requirements [3].Flooding, which belongs to BFS-based methods, is the default search algorithm for Gnutella network [4], [5]. By this method, the query source sends its query messages to all of its neighbors. When a node receives a query message, it first checks if it has the queried resource. If yes, it sends a response back to the query source to indicate a query hit. Otherwise, it sends the query messages to all of its neighbors, except for the one the query message comes from. The drawback of flooding is the search cost. It produces considerable query messages even when the resource distribution is scarce. On the other hand, random walk (RW) is a conservative search algorithm, which belongs to DFS-based methods [6], [7], 8], [9], [10]. By RW, the query source just sends one query message (walker) to one of its neighbors. If this neighbor does not own the queried resource, it keeps on sending the walker to one of its neighbors, except for the one the query message comes from, and thus, the search cost is reduced.

Different techniques can be used to discover resources on the network. If the network is small no complex search techniques are needed. One can use simple broadcasting or multicasting for querying. Centralized systems with few servers also do not require complex query propagation methods. However, if we want to support complex or free-form queries in decentralized networks, such as unstructured P2P overlays, sophisticated search techniques have to be applied to query propagation to achieve scalability and efficient operation. Below we give an overview of the most common search algorithms and example protocols that utilize these methods. Search in a graph is defined as finding a path from a start node to a destination node. In our context the destination node is the node that contains the service searched. The cost of a search can be defined in various ways, for example as the number of edges traversed in locating the destination node or the number of packets sent into the network during the search process. A simple scenario of P2P network is shown in Fig.1.

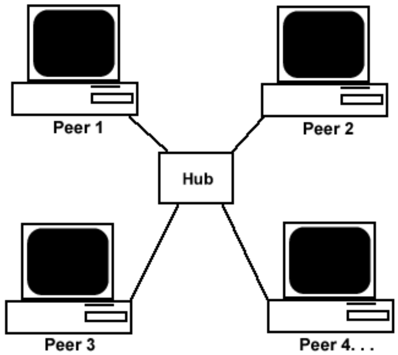

Fig. 1 P2P network

In the next section we will thrash out fundamental search methods in unstructured peer-to-peer networks.

2. BASIC SEARCH METHODS AND PROTOCOLS FOR UNSTRUCTURED NETWORKS

The search algorithms are classified into uninformed [11], where sending nodes know nothing of the surrounding networks, and informed [12] that rely on the partial network information discovered previously. Informed and probabilistic methods significantly reduce the overhead in the system, but they suffer from the partial coverage problem, in extreme cases showing very poor performance.

2.1 Uninformed search methods and related protocols

Uninformed search methods [11] can be further divided into systematic and random search algorithms. Systematic search methods typically explore the searched tree or graph according to some predefined rules. There is no place for probabilistic or random choice in such methods. Often systematic methods conduct a complete or almost complete search of the studied graph. The theoretical evaluation of the considered uninformed systematic search methods is given in Table 1, where the notation established in [13] is used: b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are used as follows: a complete if b is finite; b complete if step costs ≥ ϵ for positive ϵ; c is optimal if step costs are all identical; d is optimal if both directions use breadth-first search.

TABLE 1
Evaluation of Uninformed Systematic Search Methods [11]

Criterion / Name	Breadth-first	Depth-first	Depth-limited	Iterative-deepening	Uniform cost
Complete?	Yes[a]	No	No	Yes[a]	Yes[a,b]
Time	$O(b^{d+1})$	$O(b^m)$	$O(b^l)$	$O(b^d)$	$O(b^{1+[C^*/\epsilon]})$
Space	$O(b^{d+1})$	$O(bm)$	$O(bl)$	$O(bd)$	$O(b^{1+[C^*/\epsilon]})$
Optimal	Yes[c]	No	No	Yes[c]	Yes

A. Breadth-first search and flooding: The most well known search algorithms in this category are Breadth-first (BFS) and Depth-first (DFS) searches. BFS first explores all neighboring nodes of the sender and, if the solution is not found, proceeds to explore all the two-hop neighbors. The depth of the search is further increased until either a solution is found or all nodes of the network have been searched. Flooding [14], [15] is one of the basic search protocols. Its foundation lies on the BFS search algorithm. The query is propagated to all nodes in the network. The number of query packets typically increases exponentially further the query travels from the source node, causing huge overhead. The basic solution is to introduce a time-to-live field that limits the query propagation to a certain hop depth.

B. Depth-first search, related methods and protocols: Depth-first search explores one neighbor of the sender. If solution is not found it increases the depth of the search, i.e. explores one neighbor of the previously searched node. The search continues in depth: each new node searched is situated at the increased hop distance from the requester. If maximum depth is reached than the search path is traced back until it can branch and go again in depth. Depth-limited search is a special case of the depth-first search, where only nodes with depth less than some bound are considered. Famous time-to-live (TTL) field is means for implementing this principle. Iterative deepening depth-first repeatedly applies the depth-limited search. In each iteration the maximum search depth is increased and the search is re- run. During the iteration BFS algorithm is applied.

C. Uniform-cost search: The search starts with a root node and all the neighboring nodes are explored. The neighbor connected to the root node with the lowest path cost link is chosen. Then all possible neighbors of the nodes, that already were chosen, are searched again and the node with the lowest path cost is preferred. The process continues until the searched object is found. Uniform cost is always optimal (since at any stage the cheapest solution is chosen). Dijkstra's algorithm is an example of uniform cost search.

D. Random search methods and protocols: These methods are governed by random variables. Good performance is expected, but not guaranteed by these algorithms. The random walk is one of popular random search methods [12]. This strategy is a formalization of the intuitive idea of taking successive steps, each in a random direction. A random walk is a simple stochastic process. The mathematical properties of random walks are quite well known for a long time, one of the celebrated examples being the Brownian motion. Random Walk (RW) protocol implements the random walk search technique described above. It is one of the basic and most widely used networks protocols. Its properties are discussed in detail by Q. Lv et al. in [15]. The source node sends a query to a fixed number of neighboring nodes.

The number of query replicas does not increase with the hop distance. The method allows in many cases considerably reduce the overhead imposed on the network with a tradeoff of the reduction in hit rate, increased round-trip time, and highly variable performance.

Probabilistic forwarding [16] is another of the random search methods. For each node where the query might be forwarded a random number in range [0, 1] is generated. If this number exceeds a threshold then the node is searched, otherwise it is skipped.

Probabilistic Flooding protocol [17] is based on flooding, but the query replicas are forwarded here only to a certain percentage (p) of the node's neighbors. Normal flooding is an extreme case of the probabilistic flooding with p = 1. If p = 0 then the query is, of course, not propagated anywhere. Probabilistic flooding is one of the often proposed techniques to be used in large dense wireless networks. There the propagation via standard flooding, besides leading to huge overhead, also increases the collision rate and leads to the degradation of the network performance. Careful choice, maybe even an adaptive adjustment, of the probability of further packet propagation leads to a drastic increase in the protocol efficiency and at the same time keeps high the probability of the packet delivery to the destination. Use of this method in sparse networks is not justified, as there the sufficient probability of the packet delivery cannot be ensured.

Gossip-based protocols [18] are based on probabilistic flooding. A node forwards a message to a certain number of its neighbors if it believes that they have not already received a certain amount of the message replicas. The protocol is especially suited to the large-scale distributed systems with limited mobility and high node failure level, i.e., when the connection between nodes rarely change, but the nodes themselves are often unavailable.

2.2 Informed search methods

Informed search methods [12] make extensive use of heuristics. A heuristic is a method that does not guarantee the solution found to be optimal, but usually finds an acceptably good solution in a reasonable time.

Best-first search goes through a list of possible nodes to explore and chooses the most promising ones to be explored first. Heuristic is used to rank the neighboring nodes based on the estimated cost from the current node to the solution. There are several variations of this algorithm. Greedy search algorithm chooses the node that appears to be closest to the goal node from the current node. The algorithm makes the locally optimal choice at each stage with the hope of finding the global optimum. Beam search is similar to the best-first search, however it unfolds not one, but the first m most promising nodes at each depth.

A* search falls into the category of best-first searches. The algorithm takes into account both the cost from the root node to the current node (function g(n)) and estimates the path cost from the current node (function h(n)). Function F (n) = g(n)+ h(n) represents the path cost of the most efficient estimated path towards the goal and is continuously re-evaluated while the search runs in order to arrive at the minimal cost to the goal. A* is monotonic, it is complete and optimal on graphs that are locally finite and where the heuristics are admissible and monotonic.

Backtracking, falling in the class of constraint searches is used to find solutions to problems specified by a set of constraint variables. Backtracking in the worst case tries all possible combinations in order to obtain a solution. The method's strength is that many implementations avoid trying many partial combinations, thus speeding up the running-time. The term "backtrack" was coined by American mathematician D. H. Lehmer in the 1950s [19].

Hill climbing falls in the class of Iterative improvement methods. The method extends the search path with a node which brings the path closer to the solution than it was before attaching the node. Two major modifications of the algorithm are used. In simple hill climbing the first node that brings the user closer to the solution is chosen. In steepest ascent hill climbing all possible nodes are compared and the closest to the solution successor is chosen. Finding of only local maximum (in the case that heuristic is not good enough) is the main problem with hill climbing. Several methods exist to overcome this drawback, including iterated hill climbing, stochastic hill climbing, random walks, and simulated annealing.

Lately, simulated annealing, a generic probabilistic meta-heuristic for global optimization problems, was tried success- fully as a possible search algorithm for unstructured P2P networks [20], [21]. With this algorithm a fair approximation to the global optimum of a given function in a large search space can be achieved. Simulated annealing allows the system to move consistently toward lower energy states, yet still jump out of local minima due to the probabilistic acceptance of some upward moves during the first few iterations.

Tabu search is a local search algorithm that uses memory structures, tabu-lists, forbidding the use of certain values of attributes in the search. Tabu lists containing the prohibited values are very effective, though a very good solution that just happens to have this value might be missed. To overcome this problem aspiration criteria are introduced. They allow overriding the tabu state of a solution and including the solution in the allowed set.

Ant colony optimization [22] is a meta-heuristic inspired by the behavior of ants in finding paths from the colony to food sources. It uses many ants (or agents) to traverse the solution space and find locally productive areas. The strategy usually does not perform as well as simulated annealing and other forms of local search, but it can solve tasks where no global or up-to-date perspective can be obtained, and therefore the other, in general more effective methods cannot be applied. Ant colony optimization outperforms simulated annealing, tabu search and genetic algorithms in dynamic environments, as it can adapt continuously to the changes in real time.

Genetic algorithms [23] try to solve problems using techniques inspired by biological evolutionary mechanisms, such as yielding successive generations of possible solutions using reproduction, "survival of the fittest" and mutation methods. In genetic programming, the above approach is extended to algorithms, by regarding the algorithm itself as a "possible solution" to a problem. The genetic algorithm approach is used in Immune Search [24].

Neural search algorithms are based on the use of artificial neural networks (ANN) [25], that imitate the structure of the brain. Artificial neurons, modeling brain cells, are interconnected with each other to form a network using both forward and feedback links. The links connecting the neurons can have different weights, possibly changing during the run-time, thus producing an adaptive system. Based on these elements ANN creates a mapping between inputs and outputs, either deterministically or probabilistically. Neural search algorithms are widely used for speech and image recognition systems, pattern matching and search engines.

3. PROTOCOLS BASED ON INFORMED SEARCH ALGORITHMS

We now consider several search protocols that are based on informed algorithms. Most of them utilize the best-first approach. Many protocols are also using history-based metrics, to determine the destination or the next forwarding node.

Intelligent-BFS [26] and Directed-BFS [27] are informed versions of probabilistic flooding. The neighbors to which the query is to be forwarded are chosen judging from the success rate of the neighboring nodes for the queries of the same type. The answer packet, while traveling to the source node, updates the local indices on the bypassing nodes, increasing the probability that these nodes will forward these types of queries in the future. Hybrid flooding [28] is a further development of this approach. The algorithm uses probabilistic flooding to forward the query only to a subset of the current node's neighbors. Multiple weighted metrics are used to select these neighbors.

Degree-based random walk was first proposed by Adamic et al. in 2001 [29]. The algorithm is based on random walk and it issues a number of walkers (queries) that are forwarded to the highest-degree neighbors that have not seen the query. The neighbor connectivity is learned via the exchange of "Hello" messages. The algorithm shows a very good performance for power-law random networks when resources are concentrated in the high-degree nodes. However, if resources are not concentrated in the most-connected nodes, then the heuristic fails and can even perform worse than the basic random walk.

Distributed resource location [30] protocol makes nodes to listen listens to the bypass traffic and cache the information from the relevant search-related packets: the locations of the answer to the query as well as the description of the resource. Later, if the node receives a query searching for the resource the location of which is cached, then the query is forwarded directly to the relevant node, thus saving network resources and time. This technique is aimed for large and fairly static networks, where the initial discovery-caused overhead is compensated by saving in later requests. In a very dynamic network due to the outdating of the cached information the effectiveness of the method drastically decreases. The method exhibits increasing accuracy as the object popularity drops, as the less popular objects are less likely to be reallocated.

Adaptive probabilistic search (APS) [31] uses k random walkers and tracing messages to update supplementary information on nodes. The method employs distributed resource location mechanism. To choose the direction of packet for- warding a combination of probabilistic forwarding, historic learning and best-fist technique is used. The supplementary information on the nodes is updated using both positive and negative feedback. The nodes estimate using the history of previous requests the direction for the query to travel. However, a non-zero probability exists that a packet will be send to non-best fitting neighbor, thus enabling the protocol to explore new routes. A single entry is kept for each type of the resource for precise targeting. Compared to random walk, APS is rather a bandwidth efficient protocol and with additional adaptive techniques used [31] it achieves much higher hit rates. It is said that APS does suffer from partial coverage problem due to the use of random walk and informed propagation techniques. Routing Indices [32], Query Routing Protocol [33], [34] and Local Indices [36], [35] are examples of other protocols that extensively use metadata to exchange information between nodes.

Local Minima Search [37] is another search algorithm for unstructured networks based on greedy search and local mini- mum. It is somewhat similar to the mechanism used in DHT- based systems and suffers from the same inability to conduct complex searches. Each item is assigned a key, for example via hashing. Replicas of the (key, value) pair are propagated though the network looking for the local minimum between the key and the node ID. At the local minimum the replica is stored. The propagation method is a combination of random walk and greedy deterministic forwarding; this way the wide spread of the (key, value) pairs is ensured. The query for the item is propagated in the same fashion: first the key corresponding to the searched item is determined and then the search is started using random walk followed by the deterministic forwarding. Additional methods are employed by the protocol to improve its performance, such as dynamic adjustment of the number of replicas or use of bloom filters.

Immune search method as well as Genetic routing [38] falls into the class of protocols based on genetic algorithms. The protocol consists of two parts: query propagation through the network and the topology evolution initiated as a result of search. The originating node issues a query that is forwarded to its neighbors via random walk until the packet arrives to a node where similarity metric between the information profile and the message content exceeds the threshold. Then the message packet undergoes proliferation (more messages are issued) in order to be able to find more nodes with similar information in the neighborhood. Some of the proliferated packets are also mutated. Due to mutation the chance of message packets to meet similar items increases, which in turn helps in packet proliferation. Clustering is introduced in the system to bring similar node together. The distance a node moves towards the query originating node depends on the similarity between them, their distance and the number of times (age) the node moved before.

NeuroSearch [39] makes use of neural search algorithms and correspondingly neural networks to decide to which neighbor to forward the query. The decision whether to propagate a query to a certain neighbor is based on the output neuron of three layer perceptron neural network. Each neighbor is evaluated using seven parameters, such as was the neighbor connectivity or acknowledgment of the fact that a certain message was received before. Prior to the deployment the protocol needs to be trained on test networks to adjust neural network weights. A genetic algorithm is used during the training. With current array of input parameters the protocol performs well compared to flooding with low TTL on the network built after power-law distribution. The authors in [39] expect considerable performance improvement with introduction of history-based inputs.

3.1 Common exchange metrics

Nodes employing informed search methods exchange different types of information in order to predict the location of the searched resource. The examples of these metrics are a list of known services and their location, topology information, traffic load, power capacity, computational resources, communication channel quality, available bandwidth, historical feedback (e.g. number of successful queries forwarded) and node uptime. For the latter parameter it is generally assumed [40] that the longer the node stays without failure in the network the higher the chances are that it will be connected to the network in the future. In the next section we discuss a dynamic search algorithm [41] for unstructured P2P networks.

4. DYNAMIC SEARCH ALGORITHM [41]

Designing efficient search algorithms is a key challenge in unstructured peer-to-peer networks. Flooding and random walk (RW) are two typical search algorithms. Flooding searches aggressively and covers the most nodes. However, it generates a large amount of query messages and, thus, does not scale. On the contrary, RW searches conservatively. It only generates a fixed amount of query messages at each hop but would take longer search time. We discuss the dynamic search (DS) algorithm, which is a generalization of flooding and RW. DS takes advantage of various contexts under which each previous search algorithm performs well. It resembles flooding for short-term search and RW for long-term search. Moreover, DS could be further combined with knowledge-based search mechanisms to improve the search performance.

4.1 Operation of Dynamic Search Algorithm

```
Algorithm: The pseudo-code of dynamic search DS
Input: query source s, queried resource f, transmission probabil-
ity p
Output: the location information of f
DS(s, f, p)
/* the operation of s */
h ← 0
if (h <= n)
        h ← h + 1
        s choose p portion of its neighbors
        mi carring h visits these chosen neighbors
elseif (h > n)
        h ← h + 1
        mi carring h visits one neighbor of s
/* the operation of r */
foreach (r)
        if (r has the location information of f)
                r returns the information to s
                mi stops
        elseif (h > TTL)
                mi stops
        elseif (h <= n)
                h ← h + 1
                r choose p portion of its neighbors
                mi carring h visits these chosen neighbors
        elseif (h > n)
                h ← h + 1
                mi carring h visits one neighbor of r
```

Fig. 2 The DS Algorithm [41]

DS is designed as a generalization of flooding, MBFS, and RW. There are two phases in DS. Each phase has a different searching strategy. The choice of search strategy at each phase depends on the relationship between the hop count h of query messages and the decision threshold n of DS.

1) Phase 1. When h ≤ n

At this phase, DS acts as flooding or MBFS. The number of neighbors that a query source sends the query messages to depends on the predefined transmission probability p. If the link degree of this query source is d, it would only send the query messages to d·p neighbors. When p is equal to 1, DS resembles flooding. Otherwise, it operates as MBFS with the transmission probability p.

2) Phase 2. When h > n

At this phase, the search strategy switches to RW. Each node that receives the query message would send the query message to one of its neighbors if it does not have the queried resource. The pseudo code is shown in Figure 2.

5. CONCLUSION

In this paper we have explicated Fundamental search methods and protocols for unstructured peer-to-peer network, hence the importance of these algorithms and the need of dynamic algorithm. Next, we have discussed the DS algorithm, which is a generalization of the flooding, MBFS, and RW. DS overcomes the disadvantages of flooding and RW, and takes advantage of various contexts under which each search algorithm performs well. It resembles flooding or MBFS for the short-term search and RW for the long-term search. Finally, we conclude that this work may be useful to design a new algorithm or to modify the existing algorithms for better performance.

ACKNOWLEDGEMENTS

We like to express our gratitude to all those who gave us the possibility to carry out the paper. We would like to thank Mr.K.Satyanarayana, chancellor of K.L.University, Dr.K.Raja Sekhara Rao, Dean, and K.L.University for stimulating suggestions and encouragement. We have further more to thank Prof.S.Venkateswarlu, Dr.K.Subrahmanyam, who encouraged us to go ahead with this paper.

REFERENCES

[1] K. Sripanidkulchai, The Popularity of Gnutella Queries and Its Implications on Scalability, white paper, Carnegie Mellon Univ.,Feb. 2001.

[2] M. Jovanovic, F. Annexstein, and K. Berman, “Scalability Issues in Large Peer-to-Peer Networks: A Case Study of Gnutella,”technical report, Laboratory for Networks and Applied Graph Theory, Univ. of Cincinnati, 2001.

[3] B. Yang and H. Garcia-Molina, “Improving Search in Peer-to-PeerNetworks,” Proc. 22nd Int’l Conf. Distributed Computing Systems(ICDCS ’02), pp. 5-14, July 2002.

[4] G. Kan, “Gnutella,” Peer-to-Peer Harnessing the Power of DisruptiveTechnologies, O’Reilly, pp. 94-122, 2001.

[5] RFC-Gnutella 0.6, http://rfc-gnutella.sourceforge.net /developer/testing/index.html, 2008.

[6] C. Gkantsidis, M. Mihail, and A. Saberi, “Random Walks inPeer-to-Peer Networks,” Proc. IEEE INFOCOM ’04, pp. 120-130,2004.

[7] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman,“Search in Power-Law Networks,” Physical Rev., E, vol. 64, 046135,2001.

[8] L.A. Adamic, R.M. Lukose, and B.A. Huberman, “Local Search in Unstructured Networks,” Handbook of Graphs and Networks.pp. 295-317, Wiley-VCH, 2003.

[9] C. Gkantsidis, M. Mihail, and A. Saberi, “Hybrid Search Schemes for Unstructured Peer-to-Peer Networks,” Proc. IEEE INFOCOM ’05, pp. 1526-1537, 2005.

[10] N. Bisnik and A. Abouzeid, “Modeling and Analysis of Random Walk Search Algorithm in P2P Networks,” Proc. SecondInt’l Workshop Hot Topics in Peer-to-Peer Systems (HOT-P2P ’05),pp. 95-103, 2005.

[11] S. Russel and P. Norvig, Artificial Intelligence A Modern Approach, 2nd ed. Upper Saddle River, New Jersey 07458, USA: Pearson Education, 2003, pp. 73-81.

[12] ——, Artificial Intelligence A Modern Approach, 2nd ed. Upper Saddle river, New Jersey 07458, USA: Pearson Education, 2003, pp. 95-136.

[13] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," Request for Comments, vol. 2543, March 1999.

[14] D. Tsoumakos and N. Roussopoulos, "A Comparison of Peer-to-Peer Search Methods," in Proceedings of the WebDB’03, June 2003, pp. 61-66, san Diego, USA.

[15] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," in Proceedings of the 16th ACM International Conference on Supercomputing (ICS), June 2002, pp. 84-95, New York, USA

[16] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," ACM Transactions on Computer Systems (TOCS), vol. 17, no. 2, pp. 41-88, 1999.

[17] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in Proceedings of the 11th International Conference on Information and Knowledge Management, McLean, VA, USA, November 2002, pp. 300-307.

[18] A. Ganesh, A. Kermarrec, and L. Massoulie, "Peer-to-peer member- ship management for gossip-based protocols," IEEE Transactions on Computers, vol. 52, no. 2, pp. 139-149, February2003.

[19] S. Golomb and L. Baumert, "Backtrack programming," Journal of the ACM (JACM), vol. 12, no. 4, pp. 516-524, October 1965.

[20] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, no. 4598, p. 671, 1983.

[21] C. C. H. Wang, T. Lin and Y. Shen, "Dynamic Search in Peer-to-Peer Networks," in Proceedings of the 13th international World Wide Web Conference, May 2003, pp. 332-333, New York, USA.

[22] M. Dorigo, Ant Colony Optimization. MIT Press, 2004.

[23] J. Koza, Genetic programming: on the programming of computers by means of natural selection. Cambridge, Mass. : MIT Press, December 1992.

[24] A. D. N. Ganguly, G. Canright, "Design Of An Efficient Search Algorithm For P2P Networks Using Concepts From Natural Immune Systems," in Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, September 2004, pp. 491-500, birmingham, UK.

[25] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed., M. Horton, Ed. Prentice Hall International, Inc., 1999.

[26] I. Clarke, "Freenet home page." http://www.freenetproject.org/, [last visited on 02.07.2007].

[27] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in Proceedings of 22nd International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2002, pp. 5-14.

[28] L. X. Z. Zhuang, Y. Liu and L. Ni, "Hybrid Periodical Flooding in Unstructured Peer-to-Peer Networks," in Proceedings of the Interna- tional Conference on Parallel Processing (ICPP 03), April 2003, pp. 171-178, Kaohsiung, Taiwan.

[29] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," Physical Review, vol. 64, no. 4, pp. 46135-46 143, Sep 2001.

[30] D. Menascé and L. Kanchanapalli, "Probabilistic scalable p2p resource location services," in Proceedings of the ACM SIGMETRICS Performance Evaluation Review, vol. 30, no. 2, Marina del Rey, CA, USA, September 2002, pp. 48-58

[31] N. R. D. Tsoumakos, "Adaptive Probabilistic Search (APS) for Peer- to-Peer Networks," in Proceedings of the International Conference on Parallel Processing (ICPP 03), Linkoping, Sweden, September 2003.

[32] A. Crespo and H. Garcia-Molina, "Routing indices for peer-to-peer systems," in Proceedings of the The 22nd International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2002.

[33] M. W. G. Sakaryan and H. Unger, "Search Methods in P2P Networks: a Survey," in Proceedings of the Innovative Internet Community Systems (I2CS 2004), June 2003, Guadalajara, Mexico.

[34] C. Rohrs, "Query routing for Gnutella net- work," May 2002, http://www.limewire.com/ developer/query routing/keyword%20routing.htm [last time visited on 15.02.2008].

[35] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Networks," in Proceedings of the 22nd International Conference on Distributed Systems, Vienna, Austria, July 2002, p. 5.

[36] S. Helal, N. Desai, and V. Verma, "Konark-a service discovery and delivery protocol for ad-hoc networks," in Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), vol. 3, 2003, nOT FINISHED.

[37] R. Morselli, B. Bhattacharjee, M. Marsh, and A. Srinivaran, "Efficient Lookup on Unstructured Topologies," IEEE Journal on selected areas in communications, vol. 25, no. 1, pp. 62-72, January 2007.

[38] M. Iles and D. Deugo, "A Search for Routing Strategies in a Peer-to- Peer Network Using Genetic Programming," in Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02), Suita, Japan, October 2002, p. 341.

[39] M. Vapa, N. Kotilainen, A. Auvinen, H. Kainulainen, and J. Vuori, "Resource Discovery in P2P Networks Using Evolutionary Neural Networks," in Proceedings of the AISTA, November 2004, Luxembourg

[40] Z. Zhuang, Y. Liu, and L. Xiao, "Dynamic layer management in super-peer architectures," in Proceedings of the 2004 international Conference on Parallel Processing (ICPP 04), Montreal, Quebec, Canada, August 2004.

[41] Tsungnan Lin, Pochiang Lin, Hsinping Wang, and Chiahung Chen," Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks " IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009

Shanmukhababu.kotha, presently I am doing M tech in department computer science and engineering in K L University.

Anilkumar.talasila, presently I am doing M tech in department computer science and engineering in K L University.

P.V V Prasad, presently working as a assistant professor in department computer science and engineering in K L University.