Survey of Various Search Mechanisms in Unstructured Peer-to-Peer Networks

Harshit Kapoor
Department of CSE
Galgotias College of Engineering & Technology, Greater Noida

Kushagra Mehta
Department of CSE
Galgotias College of Engineering & Technology, Greater Noida

Diwakar Puri
Department of CSE
Galgotias College of Engineering & Technology, Greater Noida

Sandeep Saxena
Assistant Professor
Department of CSE
Galgotias College of Engineering & Technology, Greater Noida

ABSTRACT

Peer-to-Peer (P2P) [1] are widely used for file sharing purposes. This type of usage provides decentralized solutions over centralized complex architecture. Peer-to-Peer networks are gaining attention from both the scientific perspective as well as the large Internet community. Popular applications utilizing this new technology offer many attractive features to a growing number of users. P2P is an architecture which is all-together a different class of applications that use the concept of distributed resources to perform an important crucial function in a decentralized manner. The popularity and bandwidth consumption attributed to current Peer-to-Peer file-sharing applications makes the operation of these distributed systems very important for the Internet community. Efficiently discovering the queried resource is the initial and most important step in establishing an efficient peer-to-peer communication. Here, we will be describing and analyzing the performances of some existing search mechanisms deployed for the peer discovery and the content look up.

Keywords

Peer to peer networks, survey, unstructured, blind search, informed.

1. INTRODUCTION

Peer-to-peer (P2P) computing represents the notion of sharing resources available on a network. The Peer-to-Peer Working Group, a consortium lead by the industry giants such as Hewlett-Packard, Intel and IBM, defines peer computing as ”sharing of computer resources by direct exchange. ”. The resources of many users and computers can be brought together build up a large network of information which can be used by anyone and use the power of Internet up to its full potential. Seeing the current scenario, the digital revolution has resulted into the emergence of many applications, supporting file sharing which include Gnutella [12], Napster [12], Freenet and Bit Torrent etc. In a peer-to-peer system (P2P), nodes of equal roles or capabilities exchange information and services directly with each other. Every node can serve as a server and as a client therefore it is called a servent (server + client) [1][20]. Furthermore, because computers communicate directly with their peers, network bandwidth is better utilized. P2P design dictates a fully – distributed, cooperative network design, where nodes collectively form a system without any supervision. Its advantages are robustness in case of failures, extensive resource sharing, self-organization, anonymity, bandwidth utilization etc.

In unstructured peer-to-peer (P2P) networks [5], each node does not have global information about the whole topology and the location of queried resources. Because of the dynamic property of unstructured P2P networks, correctly capturing the overall behavior of a network, which also includes its topology, is also difficult.

Search algorithms provide the capabilities to locate the queried resources and to route the message to the target node. Thus, the efficiency of search algorithms is critical to the performance of unstructured P2P networks.

If we observe the present day scenario, current resource –does not use Internet and its capabilities to its full potential. There is still vast amount of untapped potential around Internet. In all cases, the very first step is to discover the resource (or the object) location inside a network.In a P2P system, each peer holds a set of documents or objects, and also has the capability of requesting any desired object from other peers in the network. These documents and objects are stored across the network at various nodes. Peers and documents are assumed to have a unique ID to differentiate, one from another.

Each peer makes requests on the basis of a query distribution concept, which controls how many requests are made for each object (e.g., popular objects [27] get many more requests than the others). Nodes that are directly linked in the network are neighbors. Peers possess knowledge only about their neighbors [23] and with such limited knowledge; it becomes very important that the search algorithm must be efficient. We also assume that with each request, the information gets erased from the memory of a peer after certain period of time and hence they are assumed to be in a soft state [3]. Each search is assigned a unique identifier, which, together with the soft state, enables peers to make the distinction between new queries and duplicate ones.

A search is successful if it discovers at least one replica of the requested object. The ratio of successful to total searches made is called the success rate (or accuracy). A search can result to multiple discoveries (or hits), which are replicas of the same object stored at distinct nodes. A Time to Live (TTL) [21] value for each query is also set, which describes the total number of hops a query can travel before it gets discarded. It also helps in establishing the termination condition for a query.

The placement of this information, used in knowledge-based mechanisms, can also vary: In centralized approaches, a central directory known to all peers exists. Ex: Local Indices mechanism. Distributed approaches can also be sub-divided into pure and hybrid. In pure approaches, all participating

peers maintain some portion of the information. Ex: Adaptive probabilistic search [6][7]. Other algorithms operate on hybrid P2P architectures [19], where certain nodes assume the role of a super-peer [18][23] and the rest become leaf- nodes. Each super peer indexes the documents and objects stored at different leaf peers. Ex: Local indices [3][7] and Routing indices [3][11].

The types of the stored indices [26] in informed approaches can be used for another categorization. Indices might relate to exact object locations, probability of discovery through a link, number of objects through a link etc.Finally, we can categorize search schemes according to the query forwarding method into flood-based (utilizing the standard flooding scheme or one of its variations) and non flood-based.

2. SEARCH TECHNIQUES

Search methods [3] can be categorized as either blind or informed, according to whether they utilize information from their previous searches to locate a resource or an object.

In blind search algorithms [10][22][24], query messages are sent to neighbors without any knowledge about the possible locations of the queried resources or any preference for the directions to forward the query messages. Nodes hold no information that relates to document location. Whereas, on the other hand knowledge-based or informed search algorithms [10][24] take advantage of the knowledge learned from previous search results and route query messages with different weight- scales based on the knowledge. Thus, each node can forward query messages more intelligently. There exists a centralized or distributed directory service that assists the peer discovery (location of resource) and content look- up.

In the following sections, we will be describing some blind as well as knowledge based search mechanisms. Searching for the location of a resource includes aspects such as the query-forwarding method, the set of nodes that receive query-related messages, the form of these messages, local processing, stored indices and their maintenance, etc.

We will also be analyzing performance metrics for some algorithms on the basis of:

- Query efficiency and quality of the query [24](Efficiency in object discovery).
- Bandwidth consumption.
- Changing topologies and relocation of objects [26].

Query efficiency [13] is nothing but the number of query messages received per query and the quality of the query is nothing but the number of documents in the query result divided by all desired documents in the system, which enhances the search efficiency [4][13]. By second metric i.e. the bandwidth consumption, Minimizing message production always represents a high-priority goal for all distributed systems. Finally, it is important that any search algorithm adapts to changing conditions, since in most P2P networks users frequently join and leave the system, as well as update their collections of objects.

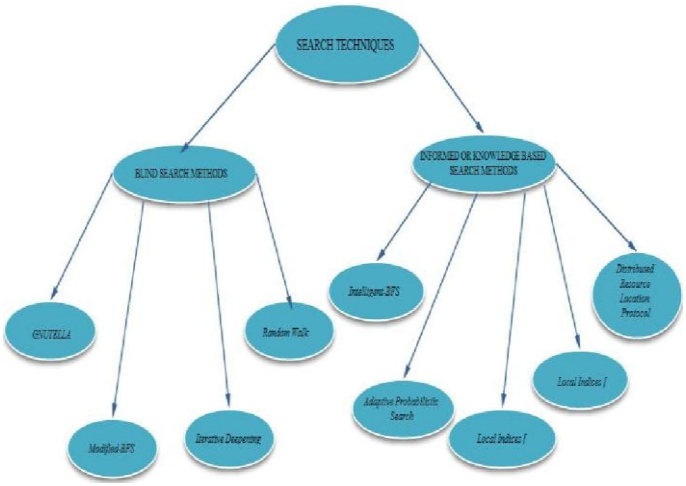

Figure 1: Hierarchy of search techniques

3. BLIND SEARCH METHODS

3.1 Gnutella [12][17][18][21] (Flooding):

The original Gnutella algorithm uses flooding (BFS traversal of the underlying graph) for object discovery, contacting all accessible nodes within TTL hops. Its basic characteristics are its simplicity and the huge overhead it produces by contacting many nodes. Flooding is an aggressive kind of a search method and covers most of the nodes, thus generates large amount of query messages. A peer broadcasts a query to its neighbors through an unstructured Peer-to-Peer (P2P) network until the Time-To-Live (TTL) decreases to zero **[14][15],** but when it comes to a large-scale network this blind search strategy usually incurs a large traffic overhead [5]. The search efficiency is too low when it comes to a large-scale network since it produces query messages even when the resource destination is scarce. The load experienced by nodes due to the flooding and duplication of query messages and expanding of networks, sometimes force a node to leave the network also.

3.2 Modified-BFS:

It is a variation of flooding which uses some ratio of the neighbors to forward the query. The ratio of neighbors to whom the message has to be forwarded is selected randomly. This type of method considerably reduces the number of messages produced but large number of peers is contacted.

3.3. Iterative Deepening [7][8][18]:

In this technique, the querying node periodically issues a sequence of BFS searches with increasing depth limits D1 < D2 < ... < Di [18]. The query is terminated when the query result is satisfied or when the maximum depth limit D has been reached. A waiting period W must be specified which signifies the time between two consecutive BFS iterations. Query satisfaction can be described by the number of hits defined by the user, which is also used as a search termination condition. In various dynamic situations, this algorithm can produce even larger loads than the standard flooding mechanism for Ex. When the user defined limit of query hits is large and is intended for such applications, which depend upon the initial number of objects returned by a query.

Iterative Deepening {a, b, c}

- Source node S initiates a BFS of depth a by sending out a query message to all its neighbors.
- Query becomes frozen at all nodes a hops away from S (Frontier nodes).
- S receives response from those nodes that have processed the query so far and waits for a time period W.
- If the query is not yet satisfied, S will start the next iteration, initiating BFS at depth b by sending a Resend message.
- A node that receives a resend message simply unfreezes the query (stored temporarily) and forwards the query to its neighbors.
- This process continues in the similar fashion till depth D is reached. At depth D, the query is dropped.

3.4 Random Walk [2][3][5][9][18][21][27]:

Random Walk (RW) is a conservative search algorithm, which belongs to DFS-based methods. By RW, the query source just sends one query message (walker) to one of its neighbors or an equal number of randomly chosen 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. The termination of a search is based on two different methods- TTL based and check method, in which the walker periodically contacts the query source that whether the termination condition has reached or not. The main drawback of RW is the long search time. Since RW only visits one node for each hop, the coverage of RW grows linearly with hop counts, which is slow compared with the exponential growth of the coverage of flooding. Moreover, the success rate of each query by RW is also low due to the same coverage issue. Increasing the number of walkers might help improve the search time and success rate, but the effect is limited due to the link degree and redundant path.

4. INFORMED OR KNOWLEDGE BASED SEARCH METHODS

4.1 Intelligent-BFS [3][7]:

This is an informed version of modified-BFS. Nodes store query-neighborID tuples for the requests that have been answered by respective peers and on this basis the nodes are ranked. When a peer receives a request query for certain object, it identifies all queries similar to the current one present in its database on the grounds of query similarity metric; it then forwards the query to a fixed number of its neighbors that have returned the most results for such queries. If a hit occurs, the query takes the reverse path to the requester and updates local indices of all the nodes in its path. At the cost of an increased message production compared to modified-BFS (because of the update process), the algorithm increases the number of hits. It achieves high accuracy, enables knowledge sharing and no overhead is produces during node arrivals/departures. Seeing the other side of the coin, the numbers of messages, which are produced, grow as the time progresses since the distribution of knowledge about the objects also grow in each and every node. It does not easily adapt to the changing nature of network that is its topology (when a peer joins or leaves), because the algorithm does not utilize negative feedback and forwarding is based on ranking.

4.2 Adaptive Probabilistic Search (APS) [6][7][18]:

In the case of APS, it builds up the knowledge on the basis of the past experiences. Every node on the network stores an index with respect to each object it has searched for, per node. This index value represents the probability that whether a specific node has the desired object and whether it will be selected for future searches. Whenever a walker is sent to the peer or on a particular path, if a query hit is encountered then the index values (probabilities) of the peers on that path are increased and if failure occur, the probabilities of those nodes get decreased. This updating process takes up the reverse path up to the requester of that resource. APS is very bandwidth efficient and zero overheads over the network during join/leave operation. The advantages of APS are mainly seen when different peers contribute with big workloads since APS gains from knowledge build-up.

4.3 Local Indices [3][7]:

In this mechanism, a node indexes all the objects that are present with the nodes inside a radius of 'r' hops. Whenever that node receives a query, first it searches for the required resource in its local database and then searches in the indexed objects of other nodes and is responsible for answering for all the nodes in a radius of 'r' hops. The method's accuracy and hits are very high, since each contacted node is responsible for answering for the whole neighborhood. Only a certain number of nodes process the query and not all the nodes engage in the processing operation. These local indices get updated whenever a node join or leaves the neighborhood.

4.4 Local Indices [3][11]:

Documents are assumed to fall into a number of thematic categories. Each node stores an approximate number of documents from every category that can be retrieved through any link attached to that particular node. The query termination condition always relates to a minimum number of hits. The forwarding process is similar to DFS: A node that cannot satisfy the query stop condition with its local repository will forward it to the neighbor with the highest "goodness" value. Goodness of a node depends upon number of related documents present with that particular node and with the nearby nodes. In RI, the destination of a packet is based upon the content of query. They give direction towards the document rather than the location of the document. This is another keyword-search approach [26], which trades index maintenance overhead for increased accuracy. RIs require flooding in order to be created and updated, so the method is not suitable for highly dynamic networks. Moreover, stored indices can be inaccurate due to thematic correlations and relocation of objects in a dynamic environment, which is the most major problems with this technique [25].

4.5 Distributed Resource Location Protocol [3][18]:

This algorithm relies on probabilistic parameters of the nodes. Nodes,which do not have any information about the location of a document, forward the query to each of their neighbors with a certain probability. Initially, random walk is used to find the location of the object. When an object is

discovered, the query backtracks up to the requester, storing the location of the found object on every node in the path of query. If in subsequent requests, again that object is requested, now that node knows where this queried object is located, hence it can directly contact that (whose location is stored) node. If that node does not currently possess that object or document, it just initiates a new search as described before. This algorithm initially spends many messages to find the locations of an object. In subsequent requests, it might take only one message to discover it. A small message production is achieved only with a large workload whose initial cost is larger than other search mechanisms. In rapidly changing networks, this approach fails and more nodes have to rely on blind search mechanism.

5. COMPARATIVE STUDY

Table 1: COMPARISION ON THE BASIS OF THEIR ADAPTING NATURE, BANDWIDTH CONSUMPTION AND QUERY HITS

Algorithms	Adapting to the changing nature of the Network	Bandwidth Consumption (Number of query messages produced)	Success Rate (Query Hits)
Gnutella (Flooding)	Efficient	Highest	Very low (large scale networks) Efficient (small scale networks)
Modified BFS	Efficient	Large	Very low (large scale networks) Efficient (small scale networks)
Iterative Deepening	Not Efficient	Large	Not effective
Random Walk	Efficient	Efficient	Efficient (large scale networks) Very low (small scale networks)
Intelligent BFS	Not Efficient	Very large	Efficient
Adaptive Probabilistic Search	Efficient	Efficient	Efficient
Local Indices	Not Efficient (every time uses flooding when a new peer joins)	Large	Efficient
Routing Indices	Not Efficient	Very efficient	Variable performance (due to object relocation and peer joining/ leaving)
DRLP	Not Efficient	High (due to initial flooding)	High (blind search) Single replica (direct query phase)

Table 2: VARIOUS ADVANTAGES AND DISADVANTAGES OF DIFFERENT ALGORITHMS

ALGORITHM/METHOD	ADVANTAGES	DISADVANTAGES
Gnutella (flooding)	Highly efficient when in small scale networks	Increases traffic overheads when comes to a large scale network
Modified BFS	Reduces average message production as compare to flooding	Results into large number of peers
Iterative deepening	Responsive and cost is low even if it visits nodes multiple times	Less efficient when it comes to long paths or large networks
Random walk	The search cost is reduce by randomly sending the query message to other nodes	The main drawback is long search time
Intelligent BFS	Accuracy, knowledge sharing and less overheads	Large no of message production, does not utilize negative feedback and me forwarding is based on ranking
Adaptive Probabilistic search (APS)	Bandwidth efficient and zero overheads	Helpful only when peer contribute to big workloads
Local indices	Accuracy and hits are very high	Uses the concept of flooding when a new pee joins so increases the messages
Routing indices	Bandwidth efficient and uses keyword search approach	Uses the concept of flooding when it needs to get created or updated so it is not suitable for dynamic network
DRLP	Small message production with large workloads so it liquidate the initial cost over many searches	This method fails when more number of nodes have to perform blind search and thus effects the number of hits

6. CONCLUSION

We present a detailed description of the various algorithms and mechanisms that can be deployed for peer discovery and content look-up in an unstructured peer-to-peer network. We have also tried to enumerate the advantages and the disadvantages of these mechanisms on the basis of various performance metrics described in the previous sections.

The specifics of any problem play a big role in choosing the right method to solve it. Each scheme that we have presented has its own positives and negatives. Important parameters that can influence our decision are that how dynamic the system is, what is its primary purpose (Ex: fast peer discovery, content look-up, query efficiency, scalability etc.), underlying topology etc.

Some of the conclusions that we present are:

1. Blind forwarding is not adequate for both high performance and low cost.
2. Various blind forwarding mechanisms does not get affected from the topology of the network i.e. peer joining/ leaving. Hence, are robust when it comes to scalability.
3. Keeping direct pointers to more number of peers in a network (Ex: DRLP) helps in increasing the accuracy but at the same time the increases the workload too.
4. Indexing of the objects and documents is greatly affected by the peer leaving/ joining.
5. Direct location information greatly increases the hits but during relocation of objects, it is ineffective. Whereas, indirect location information (Ex: APS) is much more robust in either of the cases.
6. It becomes very important for the algorithm to adapt to the changing (dynamic) nature of the network.
7. The simplicity of mechanisms behind random walks and flooding make them powerful and easy to implement since they can be easily used with other mechanisms as variation to eliminate the existing problems.

7. REFERENCES

[1] D.Raghu, CH. Raja Jacob, Gowthu, Jagadeesh Kumar, G. Monika Devi, Ramya Addanki. Dynamic search algorithm in unstructured peer-to-peer networks. International journal for computer science and technology.2011.

[2] B. Srikanth and K. Venkateswara Rao. Dynamic search algorithm used in unstructured peer-to-peer networks. Internationa journal of engineering trend and technology.2011

[3] Dimitrios Tsoumakos & Nick Roussopoulos Analysis and Comparison of P2P search methods. Infoscale '06 proceedings of the 1st International Conference on Scalable Information Systems.ACM@2006.

[4] Hsinping Wang, Tsungnan Lin, Chia Hung Chen and Yennan Shen. Dynamic Search in peer-to-peer networks. ACM 2004.

[5] Qin Lv, Pei Cao, Edith Cohen, Kai Li and Scott Shenker.

Search and Replication in unstructured peer-to-peer networks. ACM 2002.

[6] Dimitrios Tsoumakos and Nick Roussopoulos. Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks.3rd IEEE intl conference on P2P computing, 2003.

[7] Xiuqi Li and Jie Wu. Searching Techniques in Peer-to-Peer Networks.

[8] Beverly Yang, Hector Garcia-Molina. Efficient search in peer-to-peer networks. Proceedings of the ICDCS'02 conference, 2002.

[9] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Random Walks in Peer-to-Peer Networks. Performance Evaluation - P2P computing systems. ACM 2006.

[10] N.Ranjeeth Kumar, N.Deepika. An Efficient Search Algorithm in Decentralized Peer-to-Peer Networks. Int.J.Computer Techology & Applications.2012.

[11] Arturo Crespo, Hector Garcia- Molina. Routing indices for peer-to-peer networks.

[12] WU Xiao-kui. Research on Routing Method on Peer-to-Peer Network. 978-1-4244-6349-7/10 © 2010 IEEE.

[13] Tsungnan Lin, Hsinping Wang. Search Performance Analysis in Peer-to-Peer Networks. Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03). IEEE 2003.

[14] IEEE Transactions on Parallel and Distributed Systems 2009 Dynamic search algorithm in unstructured peer to peer networks By Tsungnan lin,Pochiang lin ,chiahung chin (national Taiwan university)

[15] Journal of computer science -Improving the Performance of the Peer to Peer Network by Introducing an Assortment of Methods M. Sadish Sendil and N. Nagarajan.

[16] Reza Dorrigiv, Alejandro Lopez- Ortiz, Pawel Pralat.Search Algorithms for Unstructured Peer-to-Peer Networks. LCN '07 Proceedings of the 32nd IEEE Conference on Local Computer Networks. IEEE computer society, 2007.

[17] Chao Xie and Yi Pan. Analysis of Large-Scale Hybrid Peer-to-Peer Network Topology.

[18] Sabu M. Thampi and Chandra Sekaran. K. Survey of search and replication schemes in unstructured p2p network. Network Protocols and Algorithms, ISSN 1943-3581, Vol. 2, No. 1, 2010. Cornell University Library.

[19] Christos Gkantsidis, Milena Mihail and Amin Saberi. Hybrid Search Schemes for Unstructured Peer-to-Peer Networks. 0-7803-8968-9/05 © 2005 IEEE.

[20] Stefan KraxbergerScalable Secure Routing for Heterogeneous Unstructured P2P Networks. 2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing IEEE.

[21] Hongbo Jiang and Shudong Jin. Exploiting Dynamic Querying like Flooding Techniques in Unstructured Peer-to-Peer Networks. Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP'05).

[22] Abhishek Kumar, Jun (Jim) Xu and Ellen W. Zegura. Efficient and Scalable Query Routing for Unstructured Peer-to- Peer Networks. **0-7803-8968-9/05 © 2005** IEEE.

[23] Virag Shah, Gustavo de Veciana and George Kesidis. Learning to Route Queries in Unstructured P2P Networks: Achieving Throughput Optimality Subject to Query Resolution Constraints. 2012 Proceedings IEEE INFOCOM.

[24] Ming Xu, Shuigeng Zhou and Jihong Guan. Enhancing Routing Robustness of Unstructured Peer-to-Peer Networks Using Mobile Agents. J Netw Syst Manage (2012) 20:309-352 Springer.

[25] Katja Hose, Christian Lemke and Kai-Uwe Sattler. Maintenance strategies for routing indexes. Distrib Parallel Databases (2009) 26: 231–259 Springer.

[26] John Risson, Tim Moors. Survey of research towards robust peer-to-peer networks: Search methods. Computer Networks 50 (2006) 3485–3521 ScienceDirect Elsevier.

[27] Nabhendra Bisnik and Alhussein Abouzeid. Modeling and Analysis of Random Walk Search Algorithms in P2P Networks. Proceedings of the 2005 Second International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'05) © 2005 IEEE.