Dynamic Search Algorithm used in Unstructured Peer-to-Peer Networks

B.SRIKANTH
M.Tech Student
QIS College of Engineering & Technology,
Ongole, Prakasam Dist,
AP, India.

K.VENKATESWARA RAO
Assoc. Prof in CSE
QIS College of Engineering & Technology,
Ongole, Prakasam Dist,
AP, India.

Abstract—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 propose 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. We analyze the performance of DS based on some performance metrics including the success rate, search time, query hits, query messages, query efficiency, and search efficiency. Numerical results show that DSprovides a good tradeoff between search performance and cost. On average, DS performs about 25 times better than flooding and 58 times better than RW in power-law graphs, and about 186 times better than flooding and 120 times better than RW in bimodal topologies.

Index Terms—Peer-to-peer, performance analysis, search algorithm.

1 INTRODUCTION

IN unstructured peer-to-peer (P2P) networks, 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 global behavior 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 . 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 or not meeting users' requirements . Flooding, which belongs to BFS-based methods, is the default search algorithm for Gnutella network . 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 rawback of flooding is the search cost. It produces considerable query messages even when the resource distribution is scarce. The search is especially inefficient when the target is far from the query source because the number of query messages would grow exponentially with the hop counts. Fig. 1 illustrates the operation of flooding. The link degree of each vertex in this graph is 4. If the network grows unlimited from the query source, the number of query messages generated by flooding at each hop would be used respectively. If the queried resource locates at one of the third neighbors, it takes 4 þ 12 þ 36 52 query messages to get just one query hit.

In this paper, we propose the dynamic search (DS) algorithm, which is a generalization of flooding and RW.

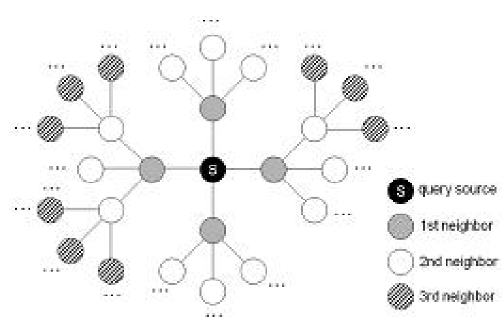

Fig. 1. A simple scenario of P2P network to demonstrate the operation of flooding and RW.

DS overcomes the disadvantages of flooding and RW and takes advantage of different contexts under which each search algorithm performs well. The operation of DS resembles flooding for the short-term search and RW for the long-term search. In order to analyze the performance of DS, we apply the random graphs as the models of network topologies and adopt the probability generating functions to model the link degree distribution . We evaluate the performance of search algorithms in accordance with some performance metrics including the success rate, search time, number of query hits, number of query messages, query efficiency, and search efficiency . Simulation experiments are performed in a dynamic P2P networking environment in order to collect convincing results for algorithm evaluations. The factors considered include the network topology, link degree distribution, peer's joining and leaving, and querying behavior as well as the activity of file sharing. Our dynamic network model is constructed based on these factors that strongly reflect the real measurement studies. Numerical results show that DS could provide a good tradeoff between search performance and cost. On average, DS performs about 25 times better than flooding and 58 times better than RW in power-law graphs, and about 186 times better than flooding and 120 times better than RW in bimodal topologies.

The rest of this paper is organized as follows: Section 2 shows the related works about the search issue in unstructured P2P networks, followed by the detailed description of the proposed DS algorithm in Section 3.

The performance analysis is given in Section 4. Numerical results and discussions are given in Section 5. Finally, the conclusion is presented in Section 6.

2 RELATED WORKS

Flooding and RW are two typical examples of blind search algorithms by which query messages are sent to neighbors without any knowledge about the possible locations of the queried resources or any preference for the directions to send. Some other blind search algorithms include modified BFS, directed BFS, expanding ring, and random periodical flooding (RPF). These algorithms try to modify the operation of flooding to improve the efficiency. However, they still generate a large amount of query messages. Jiang et al. propose a Light Flood algorithm, which is a combination of the initial pure flooding and subsequent tree-based flooding . DS and LightFlood operate analogously, but DS avoids the extra cost to construct and maintain the treelike suboverlay.

Knowledge-based search algorithms take advantage of the knowledge learned from previous search results and route query messages with different weights based on the knowledge. Thus, each node could relay query messages more intelligently. Some examples are adaptive probabilistic search (APS) biased RW , routing index (RI) local indices , and intelligent search . APS builds the knowledge with respect to each file based on the past experiences. RI classifies each document into some thematic categories and forwards query messages more intelligently based on the categories. The operation of local indices is similar to that of super-peer networks. Each node collects the file indices of peers within its predefined radius. If a search request is out of a node's knowledge, this node would perform a flooding search. The intelligent search uses a function to compute the similarity between a search query and recently answered requests. Nodes relay query messages based on the similarity. There are some other research works that focus on replicating a reference pointer to queried resources in order to improve the search time .

3 DYNAMIC SEARCH ALGORITHM

In this section, we provide the details of the proposed DS algorithm. Section 3.1 presents the operation of DS algorithm, and Section 3.2 provides the mechanism to combine DS with the knowledge-based search algorithms.

3.1 Operation of Dynamic Search Algorithm

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.

3.1.1 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.

3.1.2 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. Assume that the number of nodes visited by DS at hop h ¼ n is the coverage cn, and then the operation of DS at that time can be regarded as RW with cn walkers. However, there are some differences between DS and RW when we consider the whole operation. Consider the simple scenario shown in Fig. 1. Assume that the decision threshold n is set as 2. When h > 2, DS performs the same as RW with c2 = 12 walkers. Let us consider an RW search with K =12 walkers. At the first hop, the walkers only visit four nodes, but the cost is 12 messages.

RW would generate a large amount of redundant messages when K is set too large. Suppose that s is the query source, r is the vertex that receives the query message, f is the queried resource, mi is the ith query message, and TTL is the time-to-live limitation. Fig. 2 shows the pseudocode of DS. In short, DS is designed to perform aggressively for the short-term search and conservatively for the long-term search. Obviously, the parameters n and p would affect the performance of DS. In Section 3.2, we will analyze the performance of DSandshow the effects of parameters and p.

3.2 Knowledge-Based Dynamic Search

Some knowledge-based search algorithms, including APS, biased RW, RI, local indices, and intelligent search, are applicable to combine with our DS algorithm, and any training or caching operations are benefit from our DS algorithm as well. In this section, we present the generic scheme to incorporate these knowledge-based search algorithms with our DS algorithm. We construct the probabilistic function based on the information learned from the past experiences, with respect to each search target, search time, and local topology information. Thus, a node has more information to intelligently decide how many query messages to send and to which peers these messages should be forwarded. Take APS as an example.

Fig. 3 shows an example of knowledge-based DS algorithm. Node A initializes a search for a certain object. It makes its forwarding decision of which neighbors should be sent to in accordance with the probability table shown in Table 1. Assume the messages are sent to nodes B, C, and F. When node B receives the message, it checks its probability table shown in Table 1 and generates another two query messages to nodes I and G.

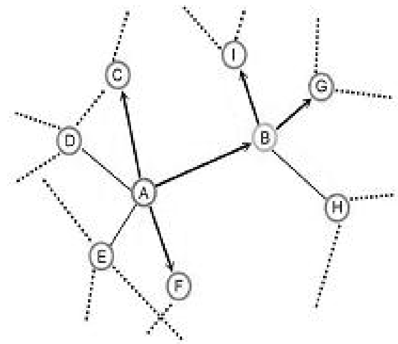

Fig. 3. Illustration for the operation of knowledge-based DS algorithm

4 PERFORMANCE EVALUATION

In this section, we present the performance evaluation of DS. Weapply Newman's random graph as the network topology, adopt the generation functions to model the link degree distribution , and analyze DS based on some performance metrics, including the success rate, search time, query hits, query messages, query efficiency, and search efficiency.

The analysis by generating functions talks about a graph all of whose parameters are exactly what they should be on an average random graph. Although the analysis using generating functions has appeared in many places by physicists, e.g.,, it maybe not strict enough in the computer science context. Mihail et al. provide a strict analysis for RWs in power-law random graphs .

4.1 Network Model

First, we summarize Newman's work about the random graph. Let G0ðxÞ be the generating function for the distribution of the vertex degree k. G0ðxÞ can be represented as

$$G_0(x) = \sum_{K=1} kpx^k \qquad (1)$$

where pk is the probability that a randomly chosen vertex in the graph has degree k, and m is the maximum degree.

TABLE 1

Node	C	D	E	B	F
Prob.	0.78	0.12	0.04	0.85	0.92

(a) Probability table for node A.

Node	G	H	H	-	-
Prob.	0.84	0.23	0.76	-	-

(b) Probability table for node B.

Based on the generating function, the average degree of a randomly chosen vertex is given by

$$Z1= (k)=\sum_{K=1} kpk=G_0(1) \qquad (2)$$

The average number of second neighbors is

$$Z2= \left[d/dx\ G_0(G1(x))\right]_{X=1} =G_0(1)G_1(1) \qquad (3)$$

where G1ðxÞ is given by

$$G_1(x) =G0(x)/G_o^1(1) \qquad (4)$$

Due to the difficulties to correctly measure and sample the operational P2P networks, there are only limited realdata about the topologies of such networks. In this paper, we will use the top two most common topologies, thepower-law graphs and the bimodal topologies, to evaluate the search performance.

- Power-Law Graphs
- Bimodal Topologies
- Search Time
- Query Hits
- Query Messages
- Query Efficiency
- Search Efficiency

4.3 Experimental Environment

We construct the experimental environment to evaluate the performance of the knowledge-based DS algorithm. For the network topology modeling, we model the P2P network as Gnutella to provide a network context in which peers can perform their intended activities.

The statistics result of the topology embedded in our simulator are that the maximum link degree is 632, mean is 11.73, and standard deviation is 17.09. Once the node (peer) degrees are chosen, we connect these peers randomly and reassure every peer is connected properly (each peer has at least one link).

For the object distribution of the network, we assume there are 100 distinct objects with replication ratio of R = 1 percent; totally, there are 100,000 objects in the network.

The distribution of the 100,000 objects over the network follows the measurement characteristics reported. In addition, due to the dynamic environment peers join and leave dynamically described in the following section, the total number of objects available in the network will fluctuate according to the network size (number of online peers), but the replication ratio will roughly remain constant.

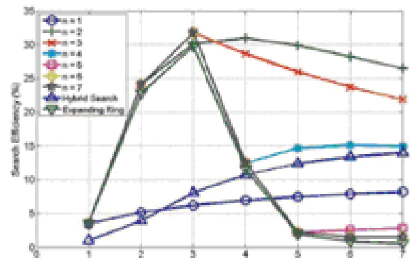

Fig. 4. SE versus hop count when p is set as 1 and n is changed from 1 to 7. Power- law topology with N =10;000. When n is set as 2, DS gets the best performance for almost all hop counts.

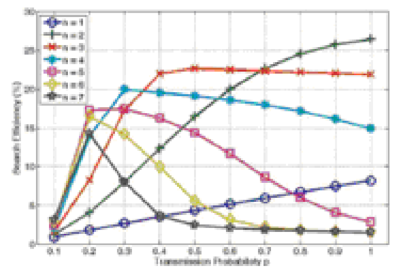

Fig. 5. The effects of the parameters ðn; pÞ on the SE. Power-law topology with N = 10;000.TTL = 7. The best SE is obtained when ðn; pÞ is set as (2, 1).

Totally, in this 2-hour simulation, we generate 43,632 search queries. Furthermore, for the query distributionof search objects, we model it as zipf distribution with parameter a = 0:82, similar to the ones used in .

Finally, our simulator's central clock is triggered per second, which measures a hop for messaging passing and serves as a basic time unit for all peer operations.

5 NUMERICAL RESULTS AND DISCUSSION

In this section, we show the numerical results of performance evaluation. We show the effectiveness of our DS algorithm and the effects of parameters n and p in Section 5.1. Then, the performance evaluation results of the knowledge-based DS algorithm are shown in Section 5.2.

5.1 Performance of Dynamic Search

- Effects of Parameters n and p of DS
- Search Time
- Comparison with Other Advanced Search Algorithms
- Scalability
- Performance under Various Network Topologies and Replication Ratios

5.2 Performance of Knowledge-Based Dynamic Search

In this section, we evaluate the search performance in a network where every node is capable of building knowledge with respect to the target through some learning mechanisms. Any forwarding mechanism can improve the search performance by leveraging over the knowledge.

6 CONCLUSION

In this paper, we have proposed 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.

We analyze the performance of DS based on some metrics including the success rate, search time, number of query hits, number of query messages, query efficiency, and search efficiency. Numerical results show that proper setting of the parameters of DS can obtain short search time and provide a good tradeoff between the search performance and cost. Under different contexts, DS always performs well. When combined with knowledge-based search algorithms, its search performances could be further improved.

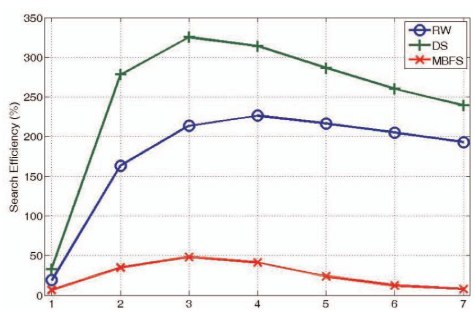

Fig. 9. Performance comparison when combined with the knowledgebased search mechanisms. DS always performs the best.

ACKNOWLEDGMENTS

This work was supported in part by Research Project work for M.Tech., on Computer Science and Engineering in QIS College of Engineering and technology, Ongole, Prakasam dist., Andhra Pradesh, India.

REFERENCES

[1] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, “Sampling Techniques for Large, Dynamic Graphs,” Proc. Ninth IEEE Global Internet Symp. (Global Internet ’06), Apr. 2006.

[2] A.H. Rasti, D. Stutzbach, and R. Rejaie, “On the Long-Term Evolution of the Two-Tier Gnutella Overlay,” Proc. Ninth IEEE Global Internet Symp. (Global Internet ’06), Apr. 2006.

[3] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, “Peer-to-Peer Computing,” Technical Report HPL-2002-57, HP, 2002.

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

[5] 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.

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

AUTHOR PROFILES

Srikath.B is Pursuing M.Tech in Qis College of Engineering & Technology ,Ongole.

Venkateswara Rao K is Currently working as Associate professor. His Areas of interest is Artificial Inelegancy .