Dynamic Search Algorithm In Unstructured Peer-To-Peer Networks

[1,] S.Vengalakshmi M.E, [2,]R.DhivyaB.E(M.E,)
[1,2,]Assistant Professor,Assistant Professor,Deptof Cse,Dept Of Cse, Syed Ammal Engineering College,Syedammal Engineering College,

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 DS provides 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.*

I. INTRODUCTION

IN unstructured peer-to-peer (P2P) networks, each nodedoes 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 [1], [2]. 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 [3]. 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 [4], [5], or not meeting users' requirements [6]. Flooding, which belongs to BFS-based methods, is the default search algorithm for Gnutella network [7], [8]. 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

- T. Lin is with the Department of Electrical Engineering and Graduate Institute of Communication Engineering, National Taiwan University, BL626, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan, ROC.

- .P. Lin and H. Wang are with the Graduate Institute of Communication Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan, ROC.

- C. Chen is with the Department of Electrical Engineering, National Taiwan University, No. 5, Lane 455, Sec. 6, Roosevelt Rd., Wunshan District, Taipei 116, Taiwan, ROC.
  Manuscript received 11 Apr. 2007; revised 17 Apr. 2008; accepted 16 July 2008; published online 21 July 2008.
  Recommended for acceptance by U. Ramachandran.
  For information on obtaining reprints of this article, please send, and reference IEEECS Log Number TPDS-2007-04-0109. Digital Object Identifier no. 10.1109/TPDS.2008.134.

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. 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 4, 12, 36, . . . , respectively. If the queried resource locates at one of the third neighbors, it takes 4þ12þ36¼52 query messages to get just one query hit.

On the other hand, random walk (RW) is a conservative search algorithm, which belongs to DFS-based methods [9], [10], [11], [12], [13]. 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. 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. As the example shown in Fig. 1, RW can only visit 12 vertices of second neighbors even when the number of walkers is set as 32. Certainly, the search is inefficient because 32 walkers only visit 12 vertices at the second hop.

Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

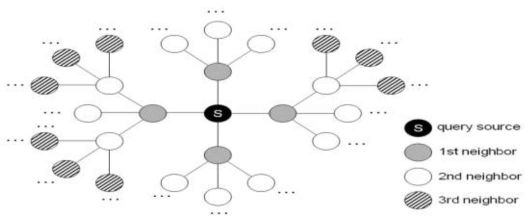


Fig. 1. A simple scenario of P2P network to demonstrate the operation of flooding and RW.

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 [14]. 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 [9], [15], [16]. 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 [10], [17], [18], [19]. Our dynamic network model is constructed based on these factors that strongly reflect the real measurement studies [17], [20], [21], [22]. 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.

II. 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 (MBFS) [23], directed BFS [6], expanding ring [17], and random periodical flooding (RPF) [24]. 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 LightFlood algorithm, which is a combination of the initial pure flooding and subsequent tree-based flooding [25], [26]. 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 knowl-edge. Thus, each node could relay query messages more intelligently. Some examples are adaptive probabilistic search (APS) [27], [28], biased RW [29], routing index (RI) [30], local indices [31], and intelligent search [32]. 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 [33], [34].

III. 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 thresh-old n of DS.

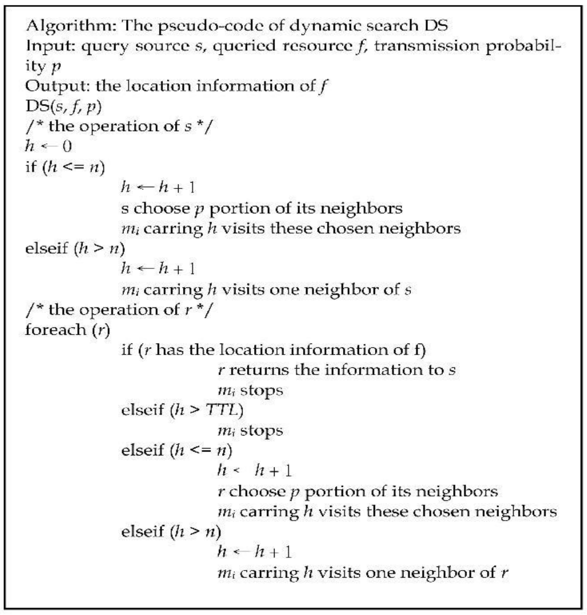
3.1.1 Phase 1. When h _ n

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 h¼n is the coverage c_n, and then the operation of DS at that time can be regarded as RW with c_n 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 c_2¼12 walkers. Let us consider an RW search with K¼12 walkers. At the first hop, the walkers only visit four nodes, but the cost is 12 messages.

656IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009



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

Fig. 2.The pseudocode of DS algorithm.

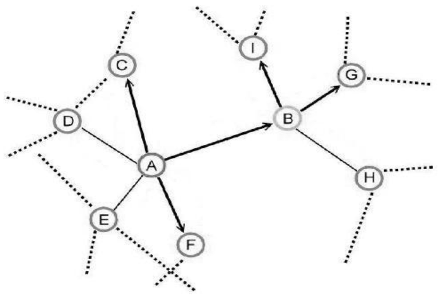


Fig. 3.Illustration for the operation of owledge-based DS algorithm.

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, m_i 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 DS andshow the effects of parameters n 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 algo-rithms 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 mes-sages to send and to which peers these messages should be forwarded. Take APS as an example. The peer applying APS search builds a probability table for each neighbor and each object. It consistently refines its probability table by the search experiences. If a search query for some object delivers to a certain neighbor successfully, the probability entry corre-sponding to that neighbor and object is increased. If the search fails finally, it will decrease the probability entry. In accordance with APS, when a certain node receives a hit from peer i, it adds 10 points for the entry of peer i; if peer i fails to respond the hit to that node, the node subtracts 10 points for the entry of peer i.

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.

IV. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of DS. We apply Newman’s random graph as the network topology, adopt the generation functions to model the link degree distribution [14], 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 generat-ing functions has appeared in many places by physicists, e.g., [10], it maybe not strict enough in the computer science context.Mihail et al. provide a strict analysis for RWs in power-law random graphs [35].

4.1 Network Model

First, we summarize Newman's work about the random graph. Let G_0ðxÞ be the generating function for the distribu-tion of the vertex degree k. G_0ðxÞ can be represented as

m

G_0ðxÞ ¼ $p_k x^k$; ð1Þ

k¼1

X

wherep_k 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)

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

(b)

(a) Probability table for node A. (b) Probability table for node B.

Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

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

m

z_1¼ hki ¼kp_k¼ $G_0{}^0$ð1Þ: ð2Þ

k¼1

X

The average number of second neighbors is

–

d

G_0ðG_1ðxÞ

z_2¼ dx Þ ¼ $G_0{}^0$ð1Þ$G_1{}^0$ð1Þ; ð3Þ

x¼1

where G_1ðxÞ is given by

$G_0{}^0$ðx

G x Þ : 4

ð1

1ð Þ ¼$G_0{}^0$Þ ð Þ

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

4.1.1 Power-Law Graphs

For the power-law random graph with the degree exponent _, p_k is proportional to k— [36]. That is,

p_k/ k— : ð5Þ

According to [11], the following approximations for the power-law distribution are obtained:

1

$G_0{}^0$ð1Þ ﬃ __ 2 ð1_ m^2—Þ ð6Þ

and

1 $_m$3__

$G_1{}^0$ð1Þ ﬃ $G_0{}^0$ð1Þ _ 3 __ ; ð7Þ

assuming2<_<3.

4.1.2 Bimodal Topologies

For the bimodal network topology [12], [29], few ultra-peers are connected to a large number of nodes, and the rest have few neighbors. This assumption is regarded as realistic and followed by most papers such as [37] and [38]. The probability that a randomly chosen peer belongs to the

ultra-peers is denoted as p_{ultra}, and the probability that this peer belongs to the other part with few neighbors is thus

p_{few}¼ 1 _p_{ultra}. The degrees of the ultra-peers and thepeers with few neighbors are denoted as k_{ultra} and k_{few}. Applying these parameters to (1), (2), (3), and (4), the

average number of neighbors at each hop for the bimodal topologies could be obtained.

4.2 Performance Analysis

4.2.1 Success Rate ðSRÞ

The success rate ðSRÞ is the probability that a query is successful, i.e., there is at least one query hit. Assume that the queried resources are uniformly distributed in the network with a replication ratio R. SR can be calculated as

SR ¼ 1 _ð1 _RÞ^C ; ð8Þ

whereR is the replication ratio, and C is the coverage. This formula shows that SR highly depends on the coverage of the search algorithms. We use (8) to obtain an important performance metric, the search time ðSTÞ, in the following.

4.2.2 Search Time ðST Þ

To represent the capability of one search algorithm to find the queried resource in time with a given probability, we define the search time ðSTÞ as the time it takes to guarantee the query success with success rate requirement SR_{req}. ST represents the hop count that a search is successful with a probabilistic guarantee. Using (8), ST is obtained when the coverage C is equal to $\log_{ð1_RÞ}$ð1_SR_{req}Þ. For MBFS search algorithms, this situation occurs when

p G^0 1 $_p$2 $_G$0 1 $_G$0 1 $_p$3 $_G$0 1 $_G$0 1 2

Þ ð Þ

_ 0 0 _ 1ð Þ þ _ 0 _ 1 ð Þ

ST_{MBFS}_

ð Þ þ $_p ST_{MBFS}$⁻ð G^0 1 G^0 1 1⁻ _ 9

þ _ _ ð Þ ð

_þ _ 0_⁻ 1 Þ⁻ ð Þ

¼ ^{log}ð1_RÞð^1_^{SR}reqÞ:

Thus, ST for MBFS is

p_ G^0 1 1_log 1 SR

ð1_R req

STMBF ^{log}p_G$_1$^{0}ð 1 ð Þ _ Þ^ð _ Þ

S 1Þ p G 1 :

¼ _ __ $_0$^{0}ð1Þ þ !

ð10Þ

ST of flooding is analog to that of MBFS with probability p ¼ 1.

The calculation of RW depends on the number of walkersk. When k is set as 1, ST for RW is obviously log$_{ð1_RÞ}$ð1 _SR$_{req}$Þ. When k is larger than 1, assume that

G$_0$^{0}ð1Þ_⁻G$_1$^{0}ð1Þ$^{⁻t_1}$_ k _ G$_0$^{0}ð1Þ_⁻G$_1$^{0}ð1Þ$^{⁻t}$; ð11Þ

i.e., k is equal to or larger than the average number of the tthneighbors of the query source, and assume that theeffect of redundant paths can be neglected, and then the calculation for ST of RW can be expressed as

$_G$0 1 $_G$0 1 $_G$0 1 $_G$0 1 $_G$0 1 t_1

ð Þ ð Þ þ_ _ ð

0ð Þ þ 0 _ 1 _þ 0ð Þ _ 1 Þ 12

SR : ð Þ

þ k_ðST$_{RW}$_tÞ ¼ log$_{ð1_RÞ}$ð1_ $^⁻$$_{req}$Þ _

ST for RW is

t 1 i

ST$_{RW}$¼ ^{log}ð1_RÞð^1_ ^{SR}reqÞ_$_k$^{P} $_{i¼}$$^⁻$$_0$ ^{G}$_0$^{0}ð^{1}Þ_$^⁻$$_{G1}0_{ð1Þ}$$^⁻$ þt: ð13Þ

Now, we consider ST for DS. When the hop count h of the query message is smaller than or equal to the decision threshold n, ST of DS is equal to (10). When h is larger than n, the calculation can be expressed as

p G^0 1 p^2 G^0 1 G^0 1 p^3 G^0 1 G^0 1 2

ð Þ ð Þ ð

_ (ð Þ þ _ (_ 1ð Þ þ _ 0 _ 1 Þ

p^n G 1 G ^{n_1}$_p$n$_{G1}$ 14

0 (1 0 _ _

$_0$ð Þ

þ _ _ 1 C Þ ð Þ

$_{G_}$0 n_

þ _ ⁻ 1 1 STDS _ $_n$ð Þ⁻ log^þ ⁻ 1 ð ^{SR}req :

_

_ _ $_1$ð Þ⁻ ð _ Þ ¼ ð1_RÞ$^{ð\_}$ þ

658 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009

Therefore, ST for DS is

logð1_RÞð1_SRreqÞ

STDS¼ n þ pn G0 1 G0 1 n_1

1

ÞG— 1 ð

_ 0ðp 10 nÞ– 1

_ ð Þ _

_ p G 1 1 p G 1 n_1 ð15Þ

_ 10 ð–Þ_ _ __ 10ð Þ

SR re

_ log 1 RÞ1 _ _ q _

ð_ ð _ Þ

ﬃ n þ pn_G00ð1Þ_–G10ð1Þ–n_1 _ 1:

We compare ST ’s for DS and RW with one walker. The improvement ratio is

ð1

STRW_ STDS G10 Þ 1

1 : 16

ð1 _ ð1Þ–

STRW ﬃ _G00 Þ _p _G10n ð Þ

In (16), the last term on the right would significantly affect the performance improvement. ST of DS would be exponen-tially decreased with n, which can be expressed as Oð1=nÞ. Larger p would also affect the performance, but the effect is slow when compared with n. The extreme case of n is that it is set as TTL, i.e., DS performs as flooding or MBFS. In this case, ST would be the shortest, whereas it would also generate ahuge amount of query messages at the same time. The tradeoff between the search performance and the cost should be taken into consideration. In the following paragraphs, we further analyze the number of query hits and the number of query messages and further combine these metrics into the query efficiency and search efficiency.

4.2.3 Query Hits ðQHÞ

The number of query hits highly depends on the coverage, i.e., the number of total visited nodes. Assume that the queried resources are uniformly distributed with the replication ratio R in the network, and the coverage is C. The number of query hits is R_C. The coverage C can be regarded as the summation of the coverage at each hop. Therefore, we first analyze the coverage C_h at the hth hop. Let $V_h$be the event that a vertex is visited at thehthhop. Supposethe probability that the vertex i is visited at the hth hop is PiðVhÞ. When the hop count h ¼ 1,$C_h$is the expectation of thevertices that are visited at the first hop. When the hop count h is larger than 1, the calculation of C_h should preclude the event that the vertex has been visited in the previous hop. Therefore, the coverage C_h at the hth hop can be written as

i PiðVhÞ

8 n1 ; for h ¼ 1;

¼

Ch n h _ 17

¼>P 1 ð Þ

> 1 Pi Vj Pi Vh ; for h 2;

i¼1

< j¼1 _ ð Þ_ ð Þ _

>P Q _ _

>

:

whereN is the total number of vertices in the network. Next, we analyze the visiting probability PiðVhÞ for

flooding, MBFS, RW, and DS, respectively. First, we consider the flooding and MBFS cases. The visiting probability PiðVhÞ of flooding or MBFS is

PiðVhÞ ¼– p _ pi G00ð1Þ; G10 1 Ch_1 ; for h ¼ 1;

1 1⁻ p pᵢ for h 2; ð18Þ

_ _ _ _ _ ð Þ⁻ _

wherep_i is the probability that vertex i is to be reached by certain edge. Aiello et al. [39] shows that p_i can be written as

m=i¹⁻

pᵢ¼P n¼m=i1_ ;

i 1

ð19Þ

where_ is the power-law exponent, and m is the maximum degree.

When considering RW, we first calculate the probability that a vertex i is the candidate of RW, i.e.,

G

pᵢ 0⁰ 1 ; ᶜh for h 1;

PᵢðRₕÞ ¼ _1 _ 1 ð Þ pᵢ G₁⁰ 1 _1 ; for h ¼ 2: ð20Þ

Þ

_ _ _ _ ð _ _

Then, the average number of candidates of RW at hop h is

n

rₕ¼PᵢðRₕÞ: ð21Þ

i¼1

X

Hence, the probability that vertex i is visited at hop h for RW is

h i

PᵢðVₕÞ ¼PᵢðRₕÞ_ 1 _ð1 _ 1=rₕÞᵏ ; ð22Þ

wherek is the number of walkers.

The calculation of visiting probability PᵢðVₕÞ for DS depends on the relation between h and n. When h_n, PᵢðVₕÞis given by (18). When h > n, (20), (21), and (22) areused to get PᵢðVₕÞ, where k in (22) is set as Cₙ, i.e., the coverage at the nth hop. Therefore, the visiting probability PᵢðVₕÞof DS is given by

for h ¼

p pᵢ G₀⁰ð1Þ; ᶜh_1 1;

PᵢVₕ 8 1_ 1_ p pᵢ G₁⁰ 1 ; for 2 h n;

ð Þ ¼> _ _ _ _ ð Þ Cₙ _ _

R⁻ 1=r

< Pᵢ ₕ 1 1 ⁻ₕ ; for h > n:

> ð Þ_ʰ _ð _ Þ i

:

ð23Þ

4.2.4 Query Messages ðQMÞ

When considering the flooding and MBFS cases, the query message e_h generated at hop h is given by

eₕ p _G₀⁰ð1Þ; for h ¼ 1; 24

¼ _p _G₁⁰ð1Þ _Cₕ_₁; for h _ 2: ð Þ

When considering the RW case, the number of query messages for each hop keeps fixed as k, i.e., the number of walkers. Therefore, the total number of query messages for RW is k_TTL.

The calculation of query messages for DS depends on h and n. The query messages e_h generated at hop h for DS can be written as

p G⁰ 1 ; for h 1;
0 ¼
eₕ ⁸p_⁻G1 0 ð₁Þ ᶜh_1; for 2 _ h _ n; ð25Þ
ð Þ
¼ ˂Cₙ; _ for h > n:

:
4.2.5 Query Efficiency ðQEÞ

The number of query hits ðQHÞ and the number of query messages ðQMÞ are the well-known performance metrics for the evaluations of search algorithms. Generally speak-ing, the objective of search algorithms is to get the most query hits with the fewest query messages, but these two metrics often conflict with each other. Therefore, it requires a more objective metric to evaluate the search performance.

Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

We adopt the performance metrics proposed in [15], query efficiency ðQEÞ and search efficiency ðSEÞ, which consider both the search performance and the cost. The similar criterion can also be found in [9]. First, we calculate QE. In [15], QE is defined as

TTL QH h 1
h 1

QE ¼ P ¼QM ð Þ_R ; ð26Þ

whereQHðhÞ is the query hits at the hth hop, QM is the total number of query messages generated during the query, and R is the replication ratio of the queried object. Since a search getting hits in a faster fashion delivers better users' experiences and should be gauged as the higher reputation, we modify (26) and show two types of QE's. $QE_1$is calculated as (26) shows, and QE_2penalizes searchresults coming from far away, i.e.,

TTL QH h =h 1

QE_2 ¼ P h¼1 QM ð Þ _R : ð27Þ

4.2.6 Search Efficiency ðSEÞ

The search efficiency ðSEÞ is proposed as a unified performance metric for search algorithms [15]. A similar criterion can be found in [9]. While the query efficiency QE does not consider the success rate SR, SE is defined as

TTL QH h =h SR

SE ¼ Ph¼1 QM ð Þ _R ; ð28Þ

where QHðhÞ=h is the query hits in the hth hop weighted by the hop count, QM is the total number of query messages generated during the query, SR is the probability that the query is successful, i.e., there is at least one query hit, and R is the replication ratio of the queried object. Thus, the success rate SR is taken into consideration. Assume that the object is uniformly distributed in the network. Then, the query hit at the hth hop is equal to the multiplication of the coverage at the hth hop and the replication rate R. Therefore, (28) can be written as

T
T
TTL L
Ch _R=h 1 _ ð1 _RÞᴾ h Cₕ

1
¼
h¼ TT
SE ¼ 1 L _ R ; ð29Þ
P h 1 h
P ¼

whereC$_h$ is the coverage at the hth hop, e$_h$ is the query messages generated at the hth hop, and R is the replication ratio. We consider two types of SEs. SE$_1$ does not penalize search results coming from far away, i.e.,

TT
L TTL $_{Ch}$
h 1C$_h$ R 1 1 R h¼1
^{¼}TTL _ ð _
SE$_1$ ¼ $_e$ _ _ RÞ^P ; ð30Þ
P h 1 h
P $_{¼}$

andSE$_2$ is calculated as (29) shows.

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 measurements in [17]

and [20] have suggested that the topology of Gnutella network has the property of two-segment power-law link distribution. Thus, we construct a P2P network of 100,000 peers in our simulator, in which the link distribution follows the reported two-segment power law. We set the first power-law slope as 0.2316 and the second as 1.1373, which are similar to the ones used in [17]. 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).

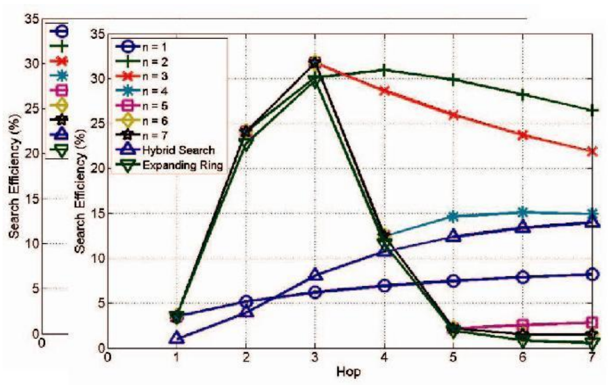
For the object distribution of the network, we assume there are 100 distinct objects with replication ratio of R ¼ 1 percent; totally, there are 100,000 objects in thenetwork. The distribution of the 100,000 objects over the network follows the measurement characteristics reported in [21]. In addition, 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.

Our dynamic peer behavior modeling largely follows the proposed idea of the peer cycle [18], which includes joining, querying, idling, leaving, and joining again to form a cycle. The joining and leaving operations of peers (include idling) are inferred and then modeled by the uptime and session duration distributions measured in [21] and [22]. These measurement studies show similar results in the peer uptime distribution, where half of the peers have uptime percentage less than 10 percent and the best 20 percent of peers have 45 percent uptime or more. We use the log-quadratic distribution suggested in [22] to rebuild the uptime distribution, which is plotted in Fig. 4. However, for the session duration distribution, those two studies lead to different results. The median of session time in [22] is about 15 minutes, while it is 60 minutes in [21]. In our modeling, we choose the median session duration time to be

20 minutes.
By these two rebuilt distributions, we can generate a
probability model to decide when a peer should join or leave the network and how long it should continually be online. The basic rule to assign peers' attributes is that peers with higher link degrees are assigned to higher uptime percentages and longer session durations, and vice versa. With these conditions, we map a 2-hour-long dynamic join/ leave pattern for peers. On average, there are 10 peers joining or leaving simultaneously. Since the mean value of uptime distribution is about 18 percent, the resulting average number of online peers is 18,152. Moreover, the maximum number of online nodes is 24,218, while the minimum number is 4,886.
We model the dynamic querying model as Poisson distribution with the idle time _¼50 minutes; that is, each peer will initiate a search every 50 minutes on average. Since there is no direct measurement about the idle time, we just use an experiential value. The choice of this parameter is insensitive to our search performance

evaluation. With

660 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009



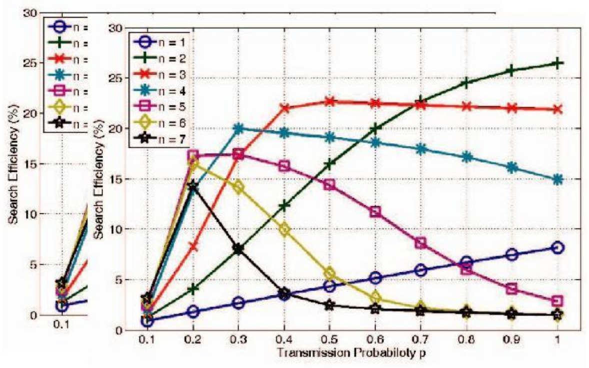


Fig. 4. SE versus hop count when p is set as 1 and n is changed from 1 to 7. Power-law topology with N ¼ 10;000. When n is set as 2, DS gets the best performance for almost all hop counts.

Fig. 5. The effects of the parameters ðn; pÞ on the SE. Power-law topology with N ¼ 10;000. TTL ¼ 7. The best SE is obtained when ðn; pÞ is set as (2, 1).

the idle time of 50 minutes, there are thus about six queries or searches processing concurrently in the network on average. Totally, in this 2-hour simulation, we generate 43,632 search queries. Furthermore, for the query distribu-tion of search objects, we model it as zipf distribution with parameter a¼0:82, similar to the ones used in [17] and [27]. 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.

V. 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

5.1.1 Effects of Parameters n and p of DS

First, N is set as 10,000. Power-law topology is adopted and the exponent _ is set as 2.1, which is analog to the real-world situation [10]. Replication ratio R is set as 0.01 in this case. Fig. 4 illustrates how the decision threshold n of DS would affect the system performance. Due to space constraints, we only show the result when p is set as 1. The case n¼1 is analog to RW with K equal to the number of first neighbors, which is roughly 3.55 in this case. The case n¼7 is equal to the flooding. As this figure shows, DS with n¼7 sends the query messages aggressively in the first three hops and gets good SE. However, the perfor-mance degrades rapidly as the hop increases. This is because the cost grows exponentially with the path length between the query source and the target. On the contrary, SE of RW is better than that of the flooding when the hop is5 to 7. When n is set as 2, DS gets the best SE for almost all hop counts. This figure shows that a good choice of parameter n can help DS to take advantage of different contexts under which each search algorithm performs well.In order to obtain the best ðn; pÞ combination, we illustrate the ðn; p; SEÞ results in Fig. 5. Here, N is set as 10,000, R is set as 0.01, and TTL is set as 7. Under this

context, when p is large (0.7-1), setting n¼2 would get the best SE. Moreover, the best n value increases as p decreases, as Fig. 6 shows. For example, when p is set as 0.2, the best n would be 6 or 7. This is because when p is small, n should be increased to expand the coverage. On the contrary, n should be decreased to limit the growth of query messages when p is large. Therefore, the parameters n and p provide the tradeoff between the search performance and the cost. It shows the best SE is obtained when ðn; pÞ is set as (2, 1). Due to space constraints, the best parameters for other contexts are skipped in this paper, which can be found through similar operation.

5.1.2 Search Time

We show the numerical results of ST in Fig. 7. In this case, N is set as 10,000, R is set as 0.01, and TTL is set as 7.Similar results can be obtained when the parameters are set as other values. The walkers K for RW are set as 1 and 32. The decision thresholds n are set as 2, 3, and 7, and p is set as 1. TTL is set as 7 in this case, thus DS with n¼7 is equal to flooding. From Fig. 7, DS with large n always gets the short ST because it always covers more vertices. On the contrary, RW with K¼1 always gets the longest ST since its coverage is only incremental by one at each hop. When K is set as 32, its coverage is enlarged and ST can beimproved. However, DS still performs better than RW with 32 walkers even when n is set as only 2. Note that when n is set as 3, DS performs as well as that with n¼7, i.e., the

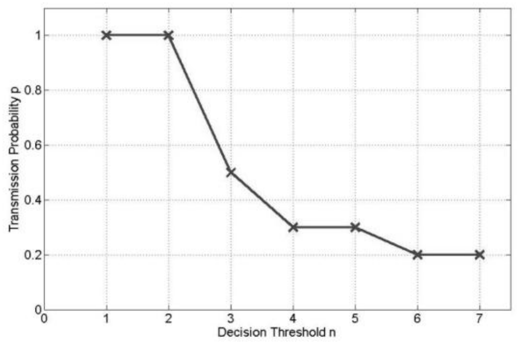
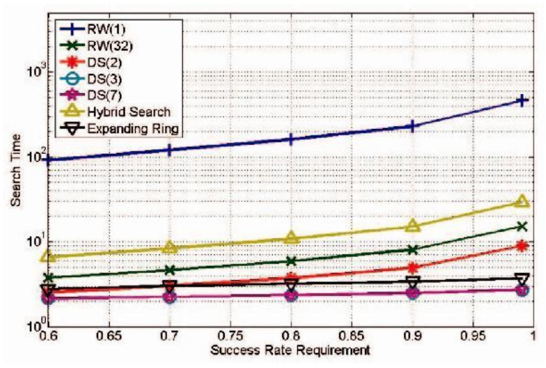


Fig. 6. The best ðn; pÞ combination when N is set as 10,000, R is set as 0.01, and TTL is set as 7.



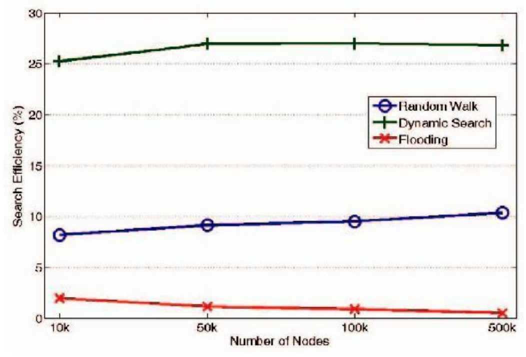


Fig. 7. ST versus SR requirement. R is set as 0.01 in this case. The walkers K for RW are set as 1 and 32, respectively. The n of DS are set as 2, 3, and 7, and p is set as 1. TTL is set as 7 in this case, thus the DS with n ¼ 7 is equal to flooding.

Fig. 8. Search efficiency for different number of nodes N in the network. This figure shows the scalability of the DS algorithm.

flooding, while does not generate as many query messages. In summary, DS with n¼2 and p¼1 would get the best SE and significantly improve ST in this case. Whileincreasing n to 3, although SE is a little degraded, the shortest ST is obtained.

5.1.3 Comparison with Other Advanced Search Algorithms

We also compare the performance of DS with that of other advanced search algorithms including Hybrid Search [12] and Expanding Ring [17]. The number of nodes N is set as 10,000. Power-law exponent _ is set as 2.1. Replication ratio R is set as 0.01 in this case. Fig. 4 shows SE's of these searchalgorithms. SE of Hybrid Search is analog to that of RW. They both increase slowly with hop counts. SE of Expanding Ring is analog to but a little worse than that of the flooding. This is because Expanding Ring would revisit the nodes it has already visited before. It would thus generate redundant messages. SE of DS is better than that of Hybrid Search and Expanding Ring for all hop counts.

Fig. 7 shows ST 's of these search algorithms. The operation of Hybrid Search is analog to that of RW with K ¼ $G^0{}_1$ð1Þ. Based on our simulation parameters, $G^0{}_1$ð1Þ isroughly 16. Thus, ST of Hybrid Search is better than that of RW(1) but worse than that of RW(32). ST of Expanding Ring is almost one hop worse than that of the flooding. When the flooding reaches the second neighbors at the second hop, Expanding Ring just revisits the first neighbors and there is no increment in coverage. For SR requirement smaller than 0.7, ST of DS(2) is shorter than that of Expanding Ring, while ST of DS(2) would be longer than that of Expanding Ring for SR requirement larger than 0.7.

5.1.4 Scalability

In order to validate the scalability of our DS algorithm, we show the search efficiency for different number of nodes in Fig. 8. Nodes N are set as 10,000, 50,000, 100,000, and 500,000, respectively. The replication ratio R is set as 0.01, and TTL is set as 7. This figure shows that our DS algorithm always performs better than flooding and RW in spite of the number of nodes.

5.1.5 Performance under Various Network Topologies and Replication Ratios

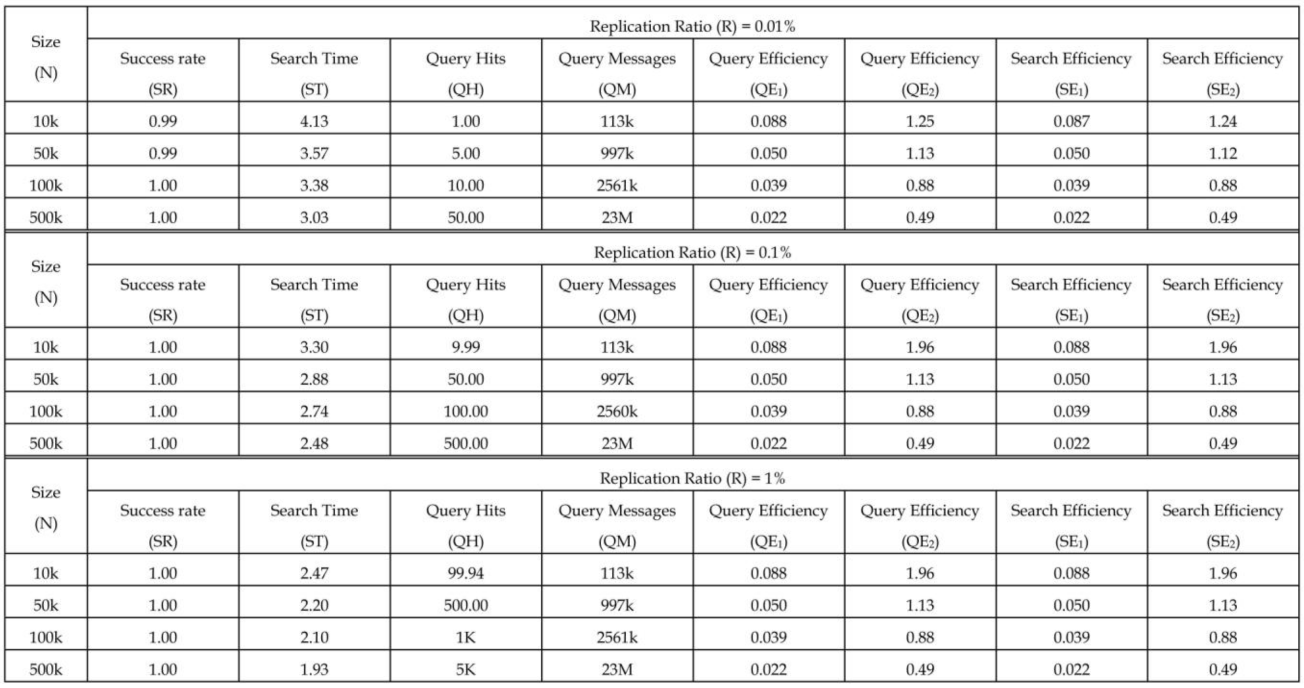
Tables 2 and 3 show the search performance under power-law random graphs and bimodal topologies, respectively. The replication ratio R is set as 0.01 percent, 0.1 percent, and 1 percent, respectively. The performance metrics including the success rate ðSRÞ, search time ðSTÞ, number of query hits ðQHÞ, number of query messages ðQMÞ, query efficiency ðQEÞ, and search efficiency ðSEÞ are listed in these tables. Two types of QE's and SE's are shown. Ones without the penalty that the search results come from far away (QE_1 and SE_1), and others with the penalty (QE_2 and SE_2), as mentioned in Section 4.2. When considering $QE_1$and QE_2, RW performs the best because it covers the fewest redundant nodes. Although RW generates the fewest query messages, its SR, ST ,QH, and the resulting SE do not perform well. In most cases, DS can perform closely to the flooding search when considering SR and ST without generating as many query messages as flooding does. In summary, DS obtains satisfactory performances in spite of the number of nodes, the replication ratio, and the network topologies. On average, it 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.

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 knowl-edge with respect to the target through some learning mechanisms. Any forwarding mechanism can improve the search performance by leveraging over the knowledge. For example, APS [27] uses the adaptive probability learning mechanism and adopts RW as the forwarding mechanism. Besides, other forwarding mechanisms, e.g., MBFS or our dynamic forwarding, are also applicable to this learning mechanism. In order to evaluate the search performance, we adopt APS learning mechanism to build the knowledge. APS learning builds a probability table for each neighbor and each object. When a query for certain object forwarding to a certain neighbor succeeds, the relative probability (or weight) of the entry for that neighbor and that object is Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply. Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

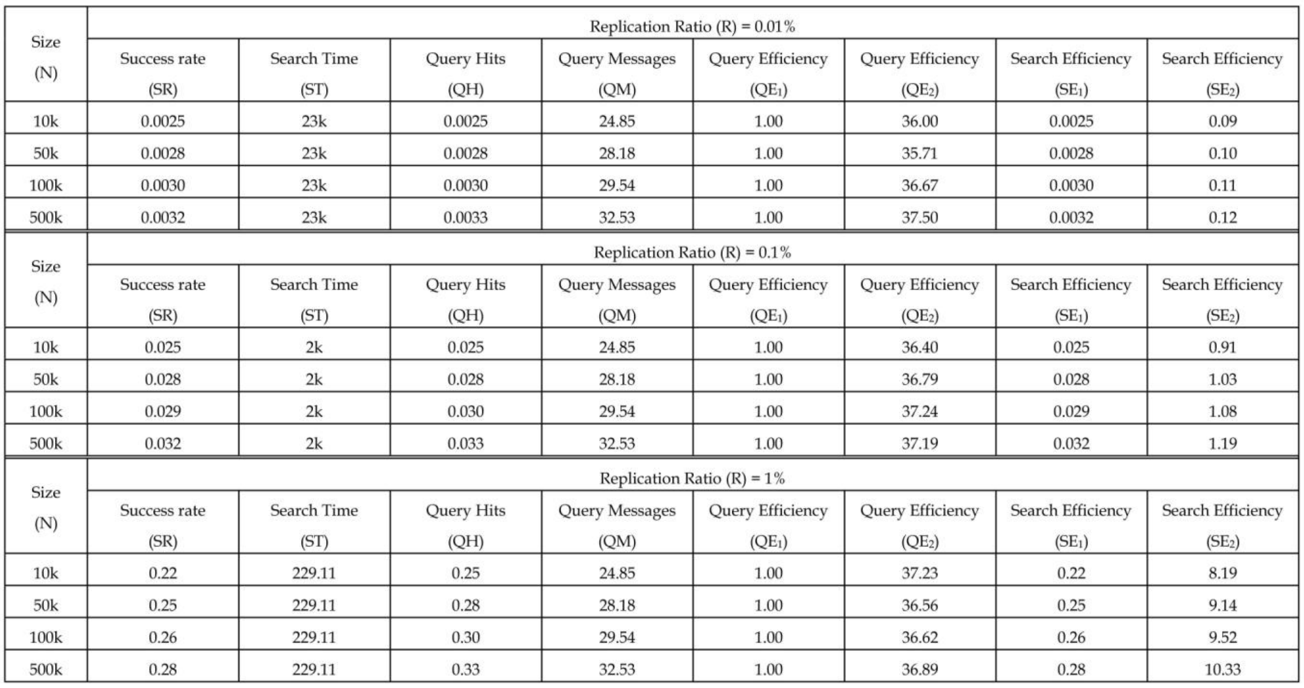
662IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009

TABLE 2a
Performance of Flooding in Power-Law Graphs



Size (N)	Replication Ratio (R) = 0.01%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.99	4.13	1.00	113k	0.088	1.25	0.087	1.24
50k	0.99	3.57	5.00	997k	0.050	1.13	0.050	1.12
100k	1.00	3.38	10.00	2561k	0.039	0.88	0.039	0.88
500k	1.00	3.03	50.00	23M	0.022	0.49	0.022	0.49
Size (N)	Replication Ratio (R) = 0.1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	1.00	3.30	9.99	113k	0.088	1.96	0.088	1.96
50k	1.00	2.88	50.00	997k	0.050	1.13	0.050	1.13
100k	1.00	2.74	100.00	2560k	0.039	0.88	0.039	0.88
500k	1.00	2.48	500.00	23M	0.022	0.49	0.022	0.49
Size (N)	Replication Ratio (R) = 1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	1.00	2.47	99.94	113k	0.088	1.96	0.088	1.96
50k	1.00	2.20	500.00	997k	0.050	1.13	0.050	1.13
100k	1.00	2.10	1K	2561k	0.039	0.88	0.039	0.88
500k	1.00	1.93	5K	23M	0.022	0.49	0.022	0.49

TABLE 2b
Performance of RW in Power-Law Graphs



Size (N)	Replication Ratio (R) = 0.01%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.0025	23k	0.0025	24.85	1.00	36.00	0.0025	0.09
50k	0.0028	23k	0.0028	28.18	1.00	35.71	0.0028	0.10
100k	0.0030	23k	0.0030	29.54	1.00	36.67	0.0030	0.11
500k	0.0032	23k	0.0033	32.53	1.00	37.50	0.0032	0.12
Size (N)	Replication Ratio (R) = 0.1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.025	2k	0.025	24.85	1.00	36.40	0.025	0.91
50k	0.028	2k	0.028	28.18	1.00	36.79	0.028	1.03
100k	0.029	2k	0.030	29.54	1.00	37.24	0.029	1.08
500k	0.032	2k	0.033	32.53	1.00	37.19	0.032	1.19
Size (N)	Replication Ratio (R) = 1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.22	229.11	0.25	24.85	1.00	37.23	0.22	8.19
50k	0.25	229.11	0.28	28.18	1.00	36.56	0.25	9.14
100k	0.26	229.11	0.30	29.54	1.00	36.62	0.26	9.52
500k	0.28	229.11	0.33	32.53	1.00	36.89	0.28	10.33

increased. Otherwise, it is decreased. Since the flooding forwards messages to all of the neighbors, the learning mechanism is useless for it, and so we do not evaluate flooding here. For the MBFS with APS learning, the transmission probability p is set as 0.2, which is chosen to keep the same amount of query messages as the other search algorithms. The initial walker for APS is 10, the same as [27].The experimental results for different search algorithms with the knowledge building mechanism are shown in Fig. 9. With APS knowledge building mechanism, all search algorithms perform much better than they do without Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

LIN ET AL.: DYNAMIC SEARCH ALGORITHM IN UNSTRUCTURED PEER-TO-PEER NETWORKS663

TABLE 2c
Performance of DS in Power-Law Graphs

Size (N)	Replication Ratio (R) = 0.01%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.75	4.46	1.00	15k	0.67	17.39	0.50	13.04
50k	0.99	3.57	5.00	91k	0.55	17.16	0.50	16.99
100k	1.00	3.38	10.00	200k	0.5	16.77	0.50	16.77
500k	1.00	3.03	50.00	1240k	0.40	16.16	0.40	16.16
Size (N)	Replication Ratio (R) = 0.1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.89	4.40	2.21	2k	1.00	22.26	0.89	19.81
50k	1.00	2.88	7.79	8k	1.00	22.01	1.00	22.01
100k	1.00	2.74	13.49	14k	1.00	21.96	1.00	21.96
500k	1.00	2.48	48.93	49k	1.00	21.88	1.00	21.88
Size (N)	Replication Ratio (R) = 1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.92	4.92	2.44	244.48	1.00	27.42	0.92	25.23
50k	0.99	2.96	4.85	485.15	1.00	27.21	0.99	26.94
100k	1.00	2.46	6.52	652.03	1.00	26.98	1.00	26.98
500k	1.00	1.93	12.96	1296.59	1.00	26.81	1.00	26.81

BLE 3a
Performance of Flooding in Bimodal Topologies

Size (N)	Replication Ratio (R) = 0.01%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	0.63	2.69	1.00	2497k	0.0040	0.14	0.0025	0.087
50k	0.99	2.69	5.00	12m	0.0042	0.13	0.0042	0.13
100k	1.00	2.69	10.00	24m	0.0042	0.12	0.0042	0.12
500k	1.00	2.69	50.00	125m	0.0040	0.11	0.0040	0.11
Size (N)	Replication Ratio (R) = 0.1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	1.00	2.28	10.00	2497k	0.0040	0.14	0.0040	0.14
50k	1.00	2.28	50.00	12m	0.0042	0.13	0.0042	0.13
100k	1.00	2.28	100.00	24m	0.0042	0.12	0.0042	0.12
500k	1.00	2.28	500.00	125m	0.0040	0.11	0.0040	0.11
Size (N)	Replication Ratio (R) = 1%							
	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
10k	1.00	1.86	100.00	2497k	0.0040	0.14	0.0040	0.14
50k	1.00	1.86	500.00	12m	0.0042	0.13	0.0042	0.13
100k	1.00	1.86	1000.00	24m	0.0042	0.12	0.0042	0.12
500k	1.00	1.86	5000.00	125m	0.0040	0.11	0.0040	0.11

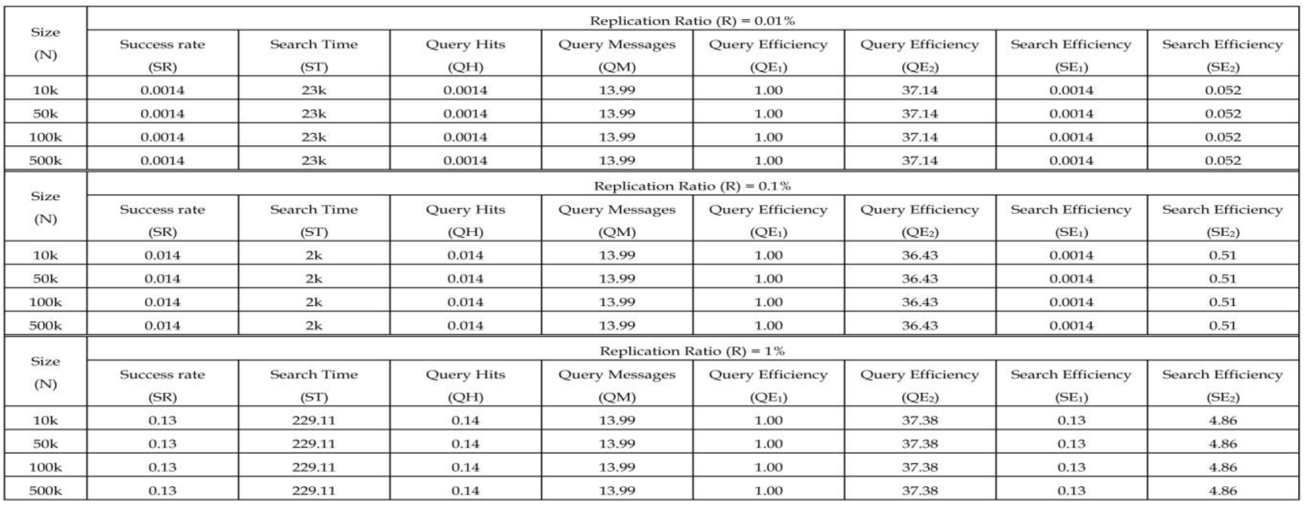
knowledge. Comparing these three search algorithms, for the case at h¼7, SE of DS is 24 percent better than that of RW, and 31 times better than that of MBFS. The outstanding performance results from the good tradeoff between the search performance and the cost.

VI. 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 Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

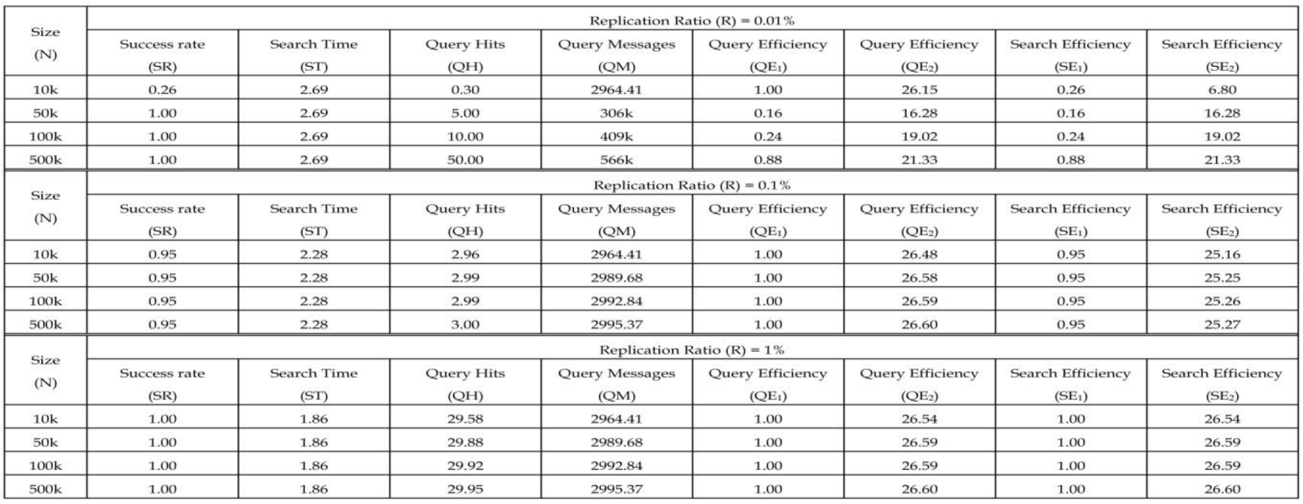
664 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009

TABLE 3b
Performance of RW in Bimodal Topologies



Size (N)	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
	Replication Ratio (R) = 0.01%							
10k	0.0014	23k	0.0014	13.99	1.00	37.14	0.0014	0.052
50k	0.0014	23k	0.0014	13.99	1.00	37.14	0.0014	0.052
100k	0.0014	23k	0.0014	13.99	1.00	37.14	0.0014	0.052
500k	0.0014	23k	0.0014	13.99	1.00	37.14	0.0014	0.052
	Replication Ratio (R) = 0.1%							
10k	0.014	2k	0.014	13.99	1.00	36.43	0.0014	0.51
50k	0.014	2k	0.014	13.99	1.00	36.43	0.0014	0.51
100k	0.014	2k	0.014	13.99	1.00	36.43	0.0014	0.51
500k	0.014	2k	0.014	13.99	1.00	36.43	0.0014	0.51
	Replication Ratio (R) = 1%							
10k	0.13	229.11	0.14	13.99	1.00	37.38	0.13	4.86
50k	0.13	229.11	0.14	13.99	1.00	37.38	0.13	4.86
100k	0.13	229.11	0.14	13.99	1.00	37.38	0.13	4.86
500k	0.13	229.11	0.14	13.99	1.00	37.38	0.13	4.86

TABLE 3c
Performance of DS in Bimodal Topologies



Size (N)	Success rate (SR)	Search Time (ST)	Query Hits (QH)	Query Messages (QM)	Query Efficiency (QE_1)	Query Efficiency (QE_2)	Search Efficiency (SE_1)	Search Efficiency (SE_2)
	Replication Ratio (R) = 0.01%							
10k	0.26	2.69	0.30	2964.41	1.00	26.15	0.26	6.80
50k	1.00	2.69	5.00	306k	0.16	16.28	0.16	16.28
100k	1.00	2.69	10.00	409k	0.24	19.02	0.24	19.02
500k	1.00	2.69	50.00	566k	0.88	21.33	0.88	21.33
	Replication Ratio (R) = 0.1%							
10k	0.95	2.28	2.96	2964.41	1.00	26.48	0.95	25.16
50k	0.95	2.28	2.99	2989.68	1.00	26.58	0.95	25.25
100k	0.95	2.28	2.99	2992.84	1.00	26.59	0.95	25.26
500k	0.95	2.28	3.00	2995.37	1.00	26.60	0.95	25.27
	Replication Ratio (R) = 1%							
10k	1.00	1.86	29.58	2964.41	1.00	26.54	1.00	26.54
50k	1.00	1.86	29.88	2989.68	1.00	26.59	1.00	26.59
100k	1.00	1.86	29.92	2992.84	1.00	26.59	1.00	26.59
500k	1.00	1.86	29.95	2995.37	1.00	26.60	1.00	26.60

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. Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.

LIN ET AL.: DYNAMIC SEARCH ALGORITHM IN UNSTRUCTURED PEER-TO-PEER NETWORKS 665

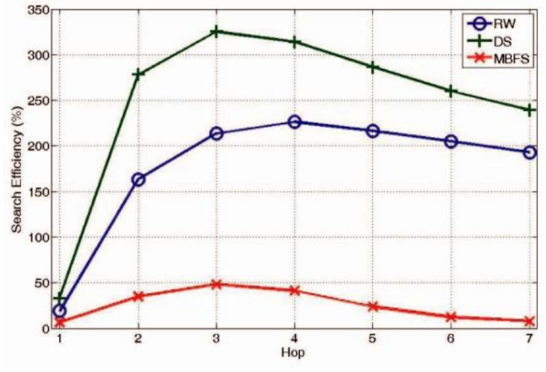


Fig. 9. Performance comparison when combined with the knowledge-based search mechanisms. DS always performs the best.

ACKNOWLEDGMENTS

This work was supported in part by Taiwan National Science Council under Grants 97-2219-E-002-024, 96-2221-E-002-091, and 96-2219-E-002-006.

REFERENCES

[1] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," Proc. 16thAnn. Int'l Conf. Supercomputing (ICS '02), pp. 84-95, June 2002.

[2] Z. Ge, D.R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-Peer File Sharing Systems," Proc. IEEEINFOCOM '03, pp. 2188-2198, 2003.

[3] K. Sripanidkulchai, The Popularity of Gnutella Queries and ItsImplications on Scalability, O'Reilly, www.openp2p.com, Feb. 2001.

[4] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the Gnutella Network," IEEE Internet Computing, vol. 6, no. 1, pp. 50-56, Jan./Feb. 2002.

[5] S. Saroiu, P.K. Gummadi, and S.D. Gribble, A Measurement Study ofPeer-to-Peer File Sharing Systems. MMCN, Jan. 2002.

[6] J. Chu, K. Labonte, and B. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems," ITCom: Scalabilityand Traffic Control in IP Networks, July 2002.

[7] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A Local Search Mechanism for Peer-to-Peer Networks," Proc.ACM CIKM Int'l Conf. Information and Knowledge Management (CIKM '02), pp. 300-307, Nov. 2002.

[8] Z. Zhuang, Y. Liu, L. Xiao, and L.M. Ni, "Hybrid Periodical Flooding in Unstructured Peer-to-Peer Networks," Proc. 32nd Int'lConf. Parallel Processing (ICPP '03), pp. 171-178, Oct. 2003.

[9] S. Jiang, L. Guo, and X. Zhang, "LightFlood: An Efficient Flooding Scheme for File Search in Unstructured Peer-to-Peer Systems," Proc. 32nd Int'l Conf. Parallel Processing (ICPP '03), pp. 627-635, Oct. 2003.

[10] S. Jiang, L. Guo, 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.

[11] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, [27] D. Tsoumakos and N. Roussopoulos, "Adaptive Probabilistic

[12] "Sampling Techniques for Large, Dynamic Graphs," Proc. Ninth Search for Peer-to-Peer Networks," Proc. Third Int'l Conf. Peer-to-

[13] IEEE Global Internet Symp. (Global Internet '06), Apr. 2006. Peer Computing (P2P '03), pp. 102-109, Sept. 2003.

[14] A.H. Rasti, D. Stutzbach, and R. Rejaie, "On the Long-Term [28] D. Tsoumakos and N. Roussopoulos, "Analysis and Comparison

[15] Evolution of the Two-Tier Gnutella Overlay," Proc. Ninth IEEEGlobal Internet Symp. (Global Internet '06), Apr. 2006.

[16] 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.

[17] of P2P Search Methods," Technical Report CS-TR-4539, UMIACS-TR-2003-107, Dept. of Computer Science, Univ. of Maryland, 2003.

[18] [29]Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P Systems Scalable," Proc.ACM SIGCOMM '03, pp. 407-418, Aug. 2003.

[19] K. Sripanidkulchai, The Popularity of Gnutella Queries and Its[30] A. Crespo and H. Garcia-Molina, "Routing Indices for Peer-to-

[20] Implications on Scalability, white paper, Carnegie Mellon Univ.,Feb. 2001.

[21] 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.

[22] 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.

[23] Peer Systems,” Proc. 22nd Int’l Conf. Distributed Computing Systems(ICDCS ’02), pp. 23-32, July 2002.

[24] 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.

[25] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, “A Local Search Mechanism for Peer-to-Peer Networks,” Proc. 11thInt’l Conf. Information and Knowledge Management (CIKM ’02), pp. 300-307, Nov. 2002.

[26] G. Kan, “Gnutella,” Peer-to-Peer Harnessing the Power of Disruptive[33] R.A. Ferreira, M.K. Ramanathan, A. Awan, A. Grama, and Technologies, O’Reilly, pp. 94-122, 2001.

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

[28] S. Jagannathan, “Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks,” Proc. Fifth IEEE Int’lConf. Peer-to-Peer Computing (P2P ’05), pp. 165-172, Aug. 2005.

[29] C. Gkantsidis, M. Mihail, and A. Saberi, “Random Walks in [34] N. Sarshar, P.O. Boykin, and V.P. Roychowdhury, “Percolation

[30] Peer-to-Peer Networks,” Proc. IEEE INFOCOM ’04, pp. 120-130, 2004.

[31] 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.

[32] L.A. Adamic, R.M. Lukose, and B.A. Huberman, “Local Search in Unstructured Networks,” Handbook of Graphs and Networks.

[33] 295-317, Wiley-VCH, 2003.

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

[35] 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),

[36] 95-103, 2005.

[37] M.E.J. Newman, S.H. Strogatz, and D.J. Watts, “Random Graphs with Arbitrary Degree Distribution and Their Applications,” Physical Rev., E, vol. 64, 026118, 2001.

[38] H. Wang and T. Lin, “On Efficiency in Searching Networks,” Proc.IEEE INFOCOM ’05, pp. 1490-1501, 2005.

[39] P. Lin, T. Lin, and H. Wang, “Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks,” Proc. Global Telecomm. Conf.(GLOBECOM ’06), Nov. 2006.

[40] Search in Power Law Networks: Making Unstructured Peer-to-Peer Networks Scalable,” Proc. Fourth IEEE Int’l Conf. Peer-to-PeerComputing (P2P ’04), pp. 2-9, Aug. 2004. M. Mihail, A. Saberi, and P. Tetali, “Random Walks with Lookahead in Power Law Random Graphs,” Internet Mathematics, 2006.

[41] L.A. Adamic, “The Small World Web,” Proc. Third European Conf.Digital Libraries (ECDL ’99), pp. 443-452, 1999.

[42] S. Behnel and A. Buchmann, “Models and Languages for Overlay Networks,” Proc. VLDB Workshop Databases, Information Systemsand Peer-to -Peer Computing (DBISP2P ’05), Aug. 2005.

[43] S. Behnel and A. Buchmann, “Overlay Networks—Implementation by Specification,” Proc. ACM/IFIP/USENIX Sixth Int’l MiddlewareConf. (Middleware), 2005.

[44] W. Aiello, F. Chung, and L. Lu, “A Random Graph Model for Massive Graphs,” Proc. 32nd Ann. ACM Symp. Theory of Computing(STOC ’00), pp. 171-180, 2000.

[45] Authorized licensed use limited to: Shree MotilalKanhaiyalalFomra Institute. Downloaded on August 12,2010 at 08:33:11 UTC from IEEE Xplore. Restrictions apply.