ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2010 (1143 - 11151)
VOLUME : 10
NUMBER : 1

SEARCHING FOR RARE RESOURCES IN UNSTRUCTURED P2P NETWORKS

HaiMei XU[1,2,*] ShouQing Qi[1] XianLiang LU[2]

1.College of ChongQing Communication, ChongQing, P.R.China,400035
2.College of computer science and engineering, University of Electronics Science and Technology of China(UESTC), ChengDu, Sichuan, P.R.China,610054

Email:xuhaimei@uestc.edu.cn haimeirdz@gmail.com Email:xlu@uestc.edu.cn

Abstract

This paper has presented a novel algorithm for searching for rare resources in unstructured P2P networks. Existing protocols such as Flooding and Random Walk can effectively locate popular resources while they are limited by very low hit rate for rare resources. For users, the utility of getting rare resources is no less than that of getting popular ones. Thus high hit rate for rare resources will improve the system's efficiency. According to different capability of different nodes, this paper explores a three-grade balanced tree to distribute index replicas of rare resources uniformly across a very small portion of nodes, which is easily deployable and lightweight in overhead. Both mathematical analysis and simulation results show that it improves the hit rate for rare resources from less than 1% to more than 90%.

Keywords: *Debt Relationship, Non-cooperation Game, Pareto Efficiency, Social Utility, Individual Utility*

1. INTRODUCTION

Studies[1-5] have showed that files' query rates follow powerlaw distribution : $p(k) = ck^{-r}$.

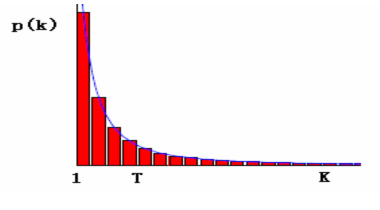

Fig.1. Probability distribution of query frequencies

In Fig.1, objects are ranked by their query frequencies, with low rankings k indicating highly accessed objects. *P(k)* represents probability distribution of k, which is a long-tail distribution. We represent rare resources in the long tail as: $R = \int_T^{\infty} p(k)dk$ and popular resources as $P = \int_1^{T} p(k)dk$. Clearly $R \ll P$, but for users, the utility of getting rare resources is not necessarily less than that of popular resources. In unstructured P2P systems such as Gnutella, KaZaA, Flooding and Random Walk can easily locate popular resources P. Boon et al. Shows 18% of the queries can't be responded even the resources exist[6]. In study[2], percolation is applied to

Received Date : 09.06.2009
Accepted Date: 15.01.2010

search for resources efficiently. Through a random walk of size $L(N,\tau)$ (N represents the size of network, τ represents the exponential of power law network), index replicas of files are distributed to the whole system, while query seeds are implanted to $\partial L(N,\tau)$ nodes. When every seed sends query at probability $q>q_c$ (q_c is the percolation threshold) to neighbors, at least one replica could be found with high probability. But the number of both replicas and query seeds approach $O(N)$ when they are distributed in low-degree nodes. In study[7], in order to guarantee the high HR for files, r_i replicas of file i are distributed in the system according to its query frequency q_i. When a large fraction of queries are insoluble, $r_1 = r_2 = ... = r_n = m$ (m is a constant); when the query rates of all types of files are comparatively high, $r_i / r_j = \sqrt{q_i} / \sqrt{q_j}$; when the query rates of all types of files are distributed uniformly, $r_i / r_j = q_i / q_j$. This way minimizes system overhead spent on insoluble queries. But for unstructured P2P networks, it is difficult to estimate query rates of all files.

THIR(Two Hop Index Replication)[3] mechanism is used to search for rare resources. In THIR, let k be the degree of a random node, then the probability $p(k) = ck^{-r}, \quad 2 \le r \le 3.475$

Where $c^{-1} = \sum_{k=2}^{k_{\max}} k^{-r} \approx \zeta(r) - 1$. $\zeta(.)$ is the Riemann zeta function, thus c can be considered constant. $k_{\max} = N^{1/r}$ where N is the network size. Superpeers have a degree k such that $N^{1/r-\delta} \le k \le N^{1/r}$ with $\delta \in (0, 1/r)$ and S be the set of superpeers that forms the network core. Also it proves that the number of superpeers is $\Omega(N^{(r-1)/r-\partial+\delta})$ with three parameters $\delta \in (0,1/r), r \in [2,3.475], \partial \in (0.0.425)$.

In THIR, replicas' storage and queries depend on only a few $\Omega(N^{(r-1)/r-\alpha+\delta})$ super peers. If a query for a rare object goes through the network core S, HR (Hit Rate) can approach nearly 100%. This way might lead to an explosion in overhead on superpeers while the majority of peers where their degrees $k \in [2, N^{1/r-\delta})$ are zero loaded. Attacks to superpeers easily incur a single point of failure. Why not amplify the superpeers' range to achieve load balance and high HR for rare resources?

In order to overcome the shortcomings of THIR, we propose a new mechanism NTIR (New Three-grade Index Replication). NTIR classifies nodes to different grades. Proportional to the capacity, quantity and *HR* of different nodes, NTIR distribute different quantity of replicas to different nodes. Simulation and mathematical analysis prove it achieves load balance and high *HR* for rare resources while incurring minimum overhead.

2. SYSTEM DESIGN

Index replicas mean that the metadata or pointers of files is distributed into the whole networks through Bloom Filters[7] or Index Tables[8-10], which is effective at improving the scalability of unstructured P2P networks while incurring much lower overhead. Our work explores the use of index replicas on a three-grade balanced tree, which enables any query to cover a large portion of the network in a few TTL (Time To Live) of Flooding, thus significantly improving.

2.1 Different *HR* for Replicas Saved in Different Nodes

It has shown that both Internet and unstructured P2P topologies hold power-law properties [3]. In this model, high capacity means high capability of CPU, large storage, great bandwidth, long online time and much low churn rates. High-capacity nodes can attract more neighbors than low-capacity ones. Therefore a node's capacity is roughly proportional to its maximum connectivity.

For reader's convenience, we summarize several notations frequently used in this paper in Table 1.

Table 1. Summary of Notation

Notation	Description
$i(i=1,2,\ldots,n)$	the grade of nodes
x_i	the number of replicas in *i-grade* nodes
h_i	the *HR* for replicas saved in *i-grade* nodes
a_i	the expense for maintaining a replica in *i-grade* nodes;
c	the size of a replica
A	the total limit of expense for maintaining all the replicas in the system
C	the total limit of the storage for all the replicas in the system
X	the limit of the number of all replicas of file *B*
N	the number of nodes in the system
n_1	the number of the first-grade nodes
n_2	the number of the second-grade nodes
n_3	the number of the third-grade nodes

Definition 1: Peers are classified as n grades by their capacity p_i. If $i<j$, then $p_i > p_j$.

Lemma 1: If $i<j$, then $h_i > h_j$. i, j=1, 2, …, n.

Proof: $i<j$ means $p_i > p_j$. The replicas saved in high-grade nodes are more reliable than low-grade nodes, and hence provide better responses to queries, therefore $h_i > h_j$.

Multiple replicas can enhance *HR* for rare resources, but the maintenance overhead of replicas can't be ignored. So the number of replicas must be controlled under limited conditions. In order to get the maximum *HR* of file *B* in the whole system, how many index replicas should be created and how to distribute the replicas in different nodes?

The *HR* for file *B* in the whole system can be computed by

$$\max \quad HR = 1 - \prod_{i=1}^{n} (1 - h_i)^{x_i} \qquad (1)$$

$$s.t \quad \sum_{i=1}^{n} a_i x_i \le A$$

$$\sum_{i=1}^{n} c x_i \le C$$

$$\sum_{i=1}^{n} x_i \le X$$

$$i = 1,2\ldots,n$$

Theorem 1: *If* $a_1 = a_2 = \ldots = a_n$, *then the optimal solution to Eq. (1) is* $HR = 1 - (1 - h_1)^X$ *where* $x_1=X$ *and* $x_2=x_3=\ldots=x_n=0$

Proof: The solution to Eq. (1) is equal to the solution to Eq. (2)

$$\begin{aligned} \min \quad & \prod_{i=1}^{n} (1 - h_i)^{x_i} \\ st \quad & \sum_{i=1}^{n} c\, x_i \le C \\ & \sum_{i=1}^{n} x_i \le X \end{aligned} \qquad (2)$$

Eq.(2) is a nonlinear programming problem and we provide the solution of dynamic programming. According to optimal theory[12-13], let $f_i(x_i)(i = 1, 2, \ldots, n)$ represents the probability of all the replicas *x* in the *i-grade* nodes being disable at the same time, and $F_k(x)$ denotes the minimum probability of failure when *x* replicas are distributed to nodes whose grades range from 1 to *k*. Eq.(2) can be rewritten as

$$\begin{cases} F_1(x) = f_1(x) = (1-h_1)^x \\ F_k(x) = \min\limits_{0 \le x_k \le x} \{ f_k(x_k) * F_{k-1}(x - x_k) \} \quad k = 2,3..n \\ \sum_{i=1}^{k} c\, x_i \le C \\ \sum_{i=1}^{k} x_i \le X \end{cases} \qquad (3)$$

By computing the value $F_1,F_2,\ldots,F_n$ recursively, in the end we can get $F_n(x)=(1-h_1)^X$, i.e. when $x_1=X$, $x_2= x_3=\ldots= x_n$, we get $1-F_n(x)=1-(1-h_1)^X$ which is the maximum *HR* for file *B*.

Theorem 1 shows that in order to achieve the highest *HR* under minimum overhead, the ideal model is to distribute all replicas in the first-grade nodes, but which might lead to information explosion when the first-grade nodes are overloaded.

2.2 The New Three-grade Index Replication Algorithm NTIR

Definition 2: According to the degree k, the nodes are classified to three grades:

The first grade: $2k_{\max}/3 \le k_1 \le k_{\max}$

The second grade: $k_{\max}/3 < k_2 < 2/3k_{\max}$

the third grade: $1 \le k_3 \le k_{\max}/3$

Theorem 2: The number of the first-grade nodes is $\Omega(N^{1/r})$.

Proof: The probability that a random node x is a first-grade one is $p(2k_{\max}/3 \le k \le k_{\max})$ For all decreasing function like k^{-r}, it is possible to bind it as following:

$$p(2K_{\max}/3 \le k \le k_{\max}) = \sum_{k=\frac{2}{3}k_{\max}}^{k_{\max}} ck^{-r} > \int_{\frac{2}{3}k_{\max}}^{k_{\max}} ck^{-r}dk$$

$$= \frac{(\frac{3}{2})^{r-1}-1}{(r-1)k_{\max}^{\ \ r-1}} = \frac{c_1}{k_{\max}^{\ \ r-1}}$$

Where $c_1 = c\dfrac{(\frac{3}{2})^{r-1}-1}{r-1}$

So the probability that x is a first-grade node is $\Omega(k_{\max}^{\ \ 1-r})$.

Let $X_1, X_2, \ldots X_N$ *be N* random variables and $X = \sum_{i=1}^{N} X_i$ such that:

$$X_i = \begin{cases} 1, \text{ if } x_i \text{ is a first-grade node;} \\ 0, \text{ otherwise}. \end{cases}$$

Let p_i be the probability that $X_i = 1$ and $k_{\max} = N^{1/r}$ [3],

$$E(X) = \sum_{i=1}^{N} p_i = \sum_{i=1}^{N} k_{\max}^{\ \ 1-r} = NK^{1-r} = N^{1/r}$$

Let $0 < \delta \le 1$ then by Chernoff bound [11, 14],

$$p(X < (1-\delta)E(X)) < e^{-E(X)\delta^2/2}$$

With the value of $E(X)$ and let $\delta = 1/2$, we get the following:

$$P(X < N^{1/r}/2) < e^{-N^{1/r}/8} < 1/N$$

For each r, there is a set of δ values that satisfy the inequality. So the number of the first-grade nodes in the network is $\Omega(N^{1/r})$

Theorem 3: The number of the second-grade nodes is $\Omega(N^{1/r})$, the number of the third-grade nodes is $\Omega(N)$.

The proof is similar to theorem 2.

Theorem 4: $n_1<n_2<n_3$ and $h_1>h_2>h_3$.

Proof: From lemma 1 and theorem 2,3, it can be proved.

For load balance, the system can be organized to a three-grade balanced tree as Fig.2. Theorem 1 and theorem 4 show that index replicas are distributed among the first-grade nodes, the highest *HR* will be achieved under the minimum overhead. This means if index replicas for rare resources are distribute uniformly, query from anywhere will hit a replica after Flooding within a few TTL in the first-grade nodes. When the first-grade nodes are overloaded, the second-grade nodes are taken into account and so on.

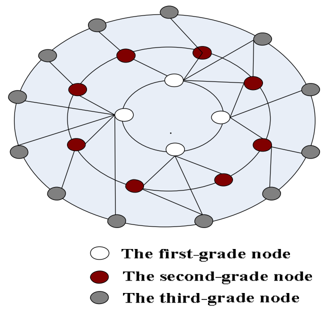

Fig.2. A three-grade balanced tree

Let r_i represents the number of index replicas in the *i-grade* nodes, q_i represents the number of the *i-grade* nodes the query has covered. The following theorem can decide how to choose r_i and q_i respectively.

Theorem 5: When $r_1 q_1 \ge N^{1/r}\left[(1+\frac{\alpha}{r}\ln N)+\sqrt{(1+\frac{\alpha}{r}\ln N)^2-1}\right]$, the probability of hit at least one replica after q_1

nodes the query has covered in the first-grade nodes is $p \geq 1 - N^{-a}$, where $\alpha > 0$ is a constant.

Proof: Let H represents the number of the first-grade nodes. For every node in the first-grade nodes, the hit probability for file replicas is $p=r_1/H$, let $X_1...X_{q1}$ be q_1 independent variables which represent the status whether the node has the replica, then

$$X_i = \begin{cases} 1 & hit\ a\ replica \\ 0 & otherwise \end{cases}$$

And

$$P[X_i = 1] = r_1 / H$$
$$P[X_i = 0] = 1 - r_1 / H$$

Let $Y = \sum_{i=1}^{q_1} X_i$, then the expectation of Y is given as below

$$E(Y) = \sum_{i=1}^{q_1} r_1 / H = r_1 q_1 / H$$

By Chernoff bound, for any $0 < \delta \leq 1$,

$$P(Y < (1-\delta)E(Y)) < e^{-E(Y)\delta^2/2}$$

Let $\begin{cases} e^{-E(Y)\delta^2/2} = N^{-a} \\ (1-\delta)E(Y) = 1 \end{cases}$ where $a>0$ is a constant.

Theorem 2 has proved that $H = \Omega(N^{1/r})$, with the value H and $E(Y)$, we get when

$$r_1 q_1 \geq N^{1/r}\left[(1+\frac{\alpha}{r}\ln N)+\sqrt{(1+\frac{\alpha}{r}\ln N)^2 - 1}\right],$$

$$p(X \geq 1) \geq 1 - N^{-\alpha}.$$

For r=3, a=1, N=10^6 when $r_1q_1 \geq 1120$, $P[Y \geq 1] \geq 99\%$ where r_1=28, q_1=40. Compared with THIR[3,15], the system's efficiency is improved significantly. The replication rate is $\lambda = r_1 / N = 0.004\%$ where r_1=28, q_1=40. Which means if we disseminates 28 replicas uniformly in the first-grade nodes, only after searching 40 first-grade nodes, query can be resolved with probability higher than 99% .

Theorem 6: *when*

$$r_2 q_2 \geq N^{1/r}[(1+\beta \mathrm{Ln}N/r)+\sqrt{(1+\beta \mathrm{Ln}N/r)^2 - 1}]$$

, the probability of hit at least one replica after q_2 nodes the query has covered is $p \geq 1 - N^{-\beta}$, where $\beta > 0$ is a constant.

The proof follows the same technique as theorem 5.

Theorem 7: *when*

$$r_3 q_3 \geq N((1+\beta LnN) + \sqrt{(1+\beta LnN)^2 - 1})$$

, the probability of hit at least one replica after q_3 nodes the query has covered is $p \geq 1 - N^{-\beta}$, where $\beta > 0$ is a constant.

The proof is similar to theorem 5.

Theorems 1, 5, 6, 7, show that the highest *HR* in the first grade nodes can be guaranteed under the minimum overhead. Generally, only if the first-grade nodes are overloaded, we take into account low-capacity nodes.

2.3 Case Study and Analyses

A single point of failure on the first-grade nodes can be avoided by the three-grade balanced tree. When the first-grade nodes are overloaded or failed, its overhead for replicas maintenance and query latency will increase. In this case, the second-grade nodes and the third-grade nodes can be taken into account as follows.

Example 1. When the first-grade nodes are overloaded while the others are lightweight. Nodes details can be found in Table 2. Assume the overhead can't exceed 20 units. In this situation, how to distribute index replicas of file *B* to the system while guaranteeing the highest *HR* of file *B*?

Table 2. The First-Grade Nodes are Overloaded

Node grade	Overhead	Hit probability
1	11 units	0.8
2	7 units	0.6
3	2 units	0.3

Table 3. The First-Grade Nodes are Under load

Node grade	Overhead	Hit probability
1	2 units	0.8
2	2 units	0.6
3	2 units	0.3

By Eq. (3), we get

$$\begin{cases} F_1(x) = f_1(x) \\ F_2(x) = \min\limits_{0 \le x_2 \le x} \{f_2(x_2) * F_1(x - x_2)\} \\ F_3(x) = \min\limits_{0 \le x_3 \le x} \{f_3(x_3) * F_2(x - x_3)\} \end{cases} \quad (4)$$

Eq.(4) is a recursive algorithm. F_1 can be computed firstly, followed by F_2, F_3 .

$$F_1(x) = f_1(x) = 1 - 0.8 = 0.2$$

$$F_2(x) = \min_{0 \le x_2 \le x}\{ f_2(0) * F_1(x), f_2(1) * F_1(x-1), f_2(2) * F_1(x-2)\}$$

$$= \min\{f_2(0) * F_1(2), f_2(1) * F_1(1), f_2(2) * F_1(0)\}$$

$$= \min\{0.2, 0.08, 0.16\} = 0.08$$

$$F_3(x) = \min_{0 \le x_3 \le x}\{ f_3(0) * F_2(x), f_3(1) * F_2(x-1), f_3(2) * F_2(x-2),$$

$$f_3(3) * F_2(x-3), f_3(4) * F_2(x-4), f_3(5) * F_2(x-5)\}$$

$$= \min\{ 0.08, 0.056, 0.078, 0.054, 0.046, 0.0672 \}$$

$$= 0.046$$

When x_1=1, x_2=0, x_3=4 the maximum *HR* is 1-0.046=0.954. Which shows when the first-grade nodes are overloaded, one replica is distributed to the first-grade nodes, four replicas to the third-grade nodes. This way can guarantee the highest *HR* under the minimum overhead.

Example 2. Assume the first-grade nodes are under loaded, nodes' detail can be found in Table 3. In this condition, in order to achieve the *HR* higher than 90%, how to distributed replicas of file *B* under minimum overhead?

By Eq. (2), we get

$$\begin{cases} \prod\limits_{i=1}^{3} (1 - h_i)^{x_i} \le 1 - 0.9 \\ \min \quad X = x_1 + x_2 + x_3 \end{cases} \quad (5)$$

From Eq.(5), we get $x_1 = 2$ □$x_2 = x_3 = 0$, which shows when the first-grade nodes are under loaded, two replicas are enough. In this case, we take into account only the first-grade nodes.

3. PERFORMANCE EVALUATION

In this section, we study the performance of our search technique and compare its performance with THIR.

3.1 Network Toplology

In order to evaluate NTIR mechanism, we use PEERSIM[14] to construct a power-law random network, which possesses two following laws:

1. Upon arrival, the new node P_{new} choose to connect with m already existing nodes p_1, p_2,…, p_m with probability

$$P(p_{new} \leftrightarrow p_i) = k_i / \sum_{k=1}^{N} k_i$$

Where k_i is the degree of node i.

2. *For load balance, when node* p_i *is overloaded, it rejects to link with* P_{new} *but recommends a neighbor with a high degree to* P_{new} .

The power-law random network consists of 10^5 *nodes. Four experiments are designed to evaluate performance of NTIR. And details about parameters of every experiment are described in section 3.2 respectively.*

3.2 Performance Metrics

HR: If there are *x* nodes query a rare file *B* and *y* nodes obtain the answer, *HR=y/x*. *HR* is the main criterion to reflect search performance for rare resources.

q: The number of nodes the query has covered before it terminates. This is measured by TTL of Flooding.

r: Replication ratio. If there are *x* replicas of file *B* distributed into the P2P system with *N* nodes, then *r=x/N*.

3.2 Simulation Results

In unstructured P2P systems, two major search algorithms are Flooding and Random Walks. For NTIR, *q* is the number of nodes the query has covered, which is same for the two algorithms, therefore we implement only Flooding algorithm in our simulations.

We first compared *HR* with and without NTIR. We simulated a system with 10^5 nodes and used NTIR to distribute 10 index replicas of 10 distinct kinds of rare files in the first-grade nodes. Then we figure out *HR* of rare files. From Fig.3, we see that more than 90% of all rare files can be found within a few TTL.

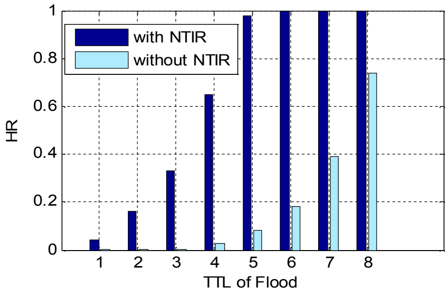

Fig.3. *HR* varied with TTL

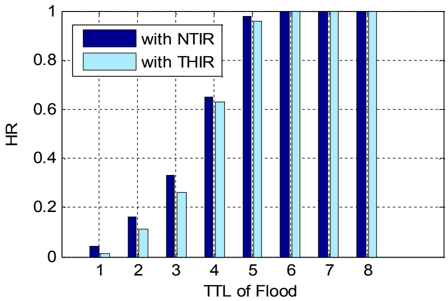

Fig.4. *HR* of two mechanisms varied with TTL

Secondly, we compared the performance of NTIR with that of THIR in the same setting: 10 index replicas of 10 distinct kinds of rare files are distributed to the system of 10^5 nodes with the two mechanisms respectively. From Fig.4, we see that *HR* of NTIR is higher than that of THIR when TTL is little. AS TTL increases, both keep an increasing trend to100%.

Thirdly, in Fig.5, we plotted *HR* changed as the size of network grew using a fixed replication ratio of 0.01% and a fixed 5 TTL of Flooding. We see evidence that the NTIR scales well as the size of the network increased.

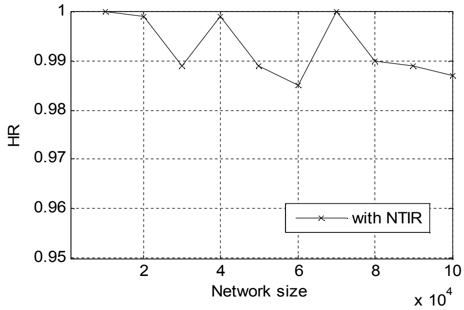

Fig.5. *HR* changed with Network Size

Fourthly, we compared *HR* of NTIR with that of THIR in the same setting as the second simulation and a fixed 5 TTL of Flooding. In Fig.6, if the first-grade nodes fail probabilistically, *HR* is maintained at a high level through the whole process. The three-grade balanced tree can avoid the single point of failure on the first-grade nodes while THIR mainly involves super nodes, the failure of super nodes leads to the decrease of *HR*.

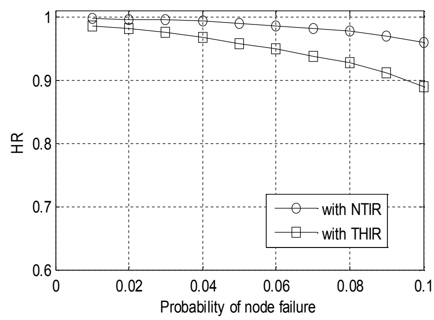

Fig.6. *HR* changed with probability of node failure

According to the above simulation results, NTIR achieves higher *HR* than THIR under low overhead.

4. CONCLUSION AND FUTURE WORK

This paper has proposed a simple distributed mechanism NTIR , which makes it easy to find rare files as well as popular files. According to peers' heterogeneity, it distributes rare files' index replications uniformly into the networks. Because NTIR don't rely on supernodes to store and deal with information , it can avoid the failure of single point caused by intentionally attacks on supernodes. For NTIR, the number of replicas is a tunable factor. When the first-grade nodes are under loaded, a few index replicas of rare resources distributed across them are enough. Otherwise the three-grade balanced tree is applied to keep a high level of *HR* under the minimum overhead. Mathematical analysis and simulation results verify that NTIR outperform THIR in terms of *HR* and overhead. Firstly NTIR has enhances the hit rate for rare resources from less than 1% to more than 90% under limited conditions, thus it is more feasible on real conditions. Secondly, the performance of NTIR scales well as the size of the network increased. Thirdly, when supernodes fail probabilistically, *HR* for rare resources can be maintained at a high level through the whole process.

Our future work includes studying NTIR's performance on real networks, implementating new techniques to reduce content delivery time and the system load[17,18], and incorpoating a P2P metric of search efficiency and preventability of polluted contents[19].

5. ACKNOWLEDGEMENTS

We are grateful for the financial support from the National Basic Research Program of China (973 Program grants 2009CB320403) and from the ChongQing Natural Science Foundation (grants CSTC, 2007ba2017).

6. REFERENCE

[1] Tathagata Das, Subrata Nandi, Niloy Ganguly. "Community based search on Power Law Networks". In *Proc. of IEEE communication Systems Software and middleware workshops*, pp. 279-282, 2008.

[2] Nima Sarshar, Oscar Boykin, VWani Roychowdhury. "Scalable percolation search on complex networks". ELSEVIER Theoretical Computer Science, Vol.355, No.1, pp. 48-64, 2006.

[3] Krishna P N Puttaswamy, Alessandra Sala and Ben Y Zhao. "Searching for rare objects using index replication". In *Proc. of the IEEE INFOCOM*, pp.1723-1731, 2008.

[4] Sarshar N, Roychowdhury V P. "Multiple power-law structures in heterogeneous complex networks". Physics Review E, Vol 72(020101), 2005.

[5] Barabasi A L, Albert R, Jeong H, et al. "Power-law distribution of the world wide web". Science, 287:2115, 2000.

[6] Boon Thau Loo, Joseph M Hellerstein, Ryan Hubsch, et al. "Enhancing P2P file-sharing with an internet-scale query processor". In *Proc. Of 30th VLDB*, Toronto, Canada, pp. 432-443, 2004.

[7] Edith Cohen, Scott Shenker. "Replication strategies in unstructured peer-to-peer Networks". In *Proc. of* SIGCOMM, ACM, Pittsburgh, Pennsylvania, pp. 697-702, 2002.

[8] Zhang Yi-Ming, Lu Xi-Cheng, Zheng Qian-Bing, et al. "An efficient search algorithm for Large-scale P2P systems". Journal of Software, Vol.19, No.2, pp.1473-1480, 2008.

[9] Lerthirunwong S, Maruyama N, Matsuoka S. "Index distribution technique for efficient search on

unstructured peer-to-peer networks". In Proc of IEEE ECTI-CON, Vol 1, pp. 97-100, 2008.

[10] Rongmei Zhang and Y.Charlie HU. "Assited peer-to-peer search with partial indexing". IEEE Transactions on Parallel and Distributed Systems, Vol.18, No 8, pp. 1146-1158, 2007.

[11] R.Motwani and P.Raghavan. *Randomized Algorithms*. Cambridge University Press,Cambridge,1995

[12] Marshall Hall, JR., *Combinatorial Theory*, John Wiley&sons Press, 1986.

[13] Sun Shi Xin, *Combinatorial Mathematics*, UESTC Press, 2002.

[14] Nima Sarshar, P Oscar Boykin, Vwani P.Roychowdhury. "Scalable percolation in power law networks". International Conference on P2P computing, pp.2-9, 2004.

[15] Xucheng Luo, Zhiguang Qin, Jinsong Han, Hanhua Chen. "DHT-assisted probability exhaustive search in unstructured P2P networks". IEEE International Symposium on Parallel and Distributed Processing, pp.1-9, 2008.

[16] PeerSim. http://peersim.sourceforge.net/, 28 October 2008.

[17] Imen Filali and Fabrice Huet. Dynamic TTL-Based Search in unstructured Peer-to-Peer Network. The 10th IEEE/ACM international Conference on Cluster, Cloud and Grid Computing, pp.438-447, 2010.

[18] Aya Takekawa, Naohiko Imaeda, Tsuyoshi Kitagawa. The study of how to reduce the content delivery time and the sysytem load in P2P system. IEEE International Conference on Complex Intelligent and Software Intensive Systems, pp.1123-1128, 2010.

[19] Hiroaki Yamanaka, Toru Fujiwara, Shingo Okamura. A metric of search efficiency and preventability of polluted contents for unstructured overlay. The 10th IEEE annual international symposium on applications and the internet, pp.181-184, 2010.

HaiMei Xu received his PH.D. degree in at College of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). She is currently assiatnt Professor of college of ChongQing Communication of China. Her research interests include distributed networks, incentive techniques and searching mechanisms in P2P networks.

ShouQing Qi is currently assistant Professor of college of ChongQing communication of China. Her research interests are in network communication and signal processing theory.

XianLiang Lu is a professor and doctoral supervisor at College of Computer Science and Engineering in UESTC. His research interests include Distributed Operating System, Computer Communications and Network Security.