NETOP: A Non-Cooperative Game Based Topology Optimization Model Towards Improving Search Performance

Jingya Zhou, Junzhou Luo, Aibo Song
School of Computer Science and Engineering, Southeast University, China
{jyz, jluo, absong}@seu.edu.cn

Abstract

Resource searching is an important function for resource sharing and cooperative work in large-scale distributed networks like grid and peer-to-peer (P2P). Many works have focused on optimizing network topology in order to improve search performance. However, these works rarely take into account the interaction of nodes' optimizing behaviors and the connection cost. We propose a Non-coopErative game based Topology OPtimization model (NETOP) to enhance search performance in unstructured P2P networks. Each participating node in NETOP is a rational player who selects optimizing strategy (its node degree) according to both its private information and the public information. We prove the existence and uniqueness of Nash Equilibrium (NE) of the game, and present the performance analysis of this model. Moreover, we also take network dynamics into account and extend our model to adapt to the node churn. Experimental results show that NETOP network converges rapidly and achieves higher performances. When compared with Power-law and Square-root topologies in a static condition, NETOP network achieves the same success rate with 33.3% and 6% lower connection cost, 18% and 13.2% lower average hop count, and 4.6% and 6.5% fewer messages, meanwhile in a dynamic condition, it achieves 28.9% and 11.5% lower connection cost, 14.3% and 7.7% lower average hop count, and 26.6% and 28.7% fewer messages.

Keywords: Peer-to-peer, Topology optimization, Search, Game theory.

1 Introduction

Grid and peer-to-peer networks provide good scenarios for multiple applications over the large-scale distributed networks like resource sharing and cooperative work in a distributed manner. All resources are generally disseminated across all participating nodes, and there is no centralized directory server for storing these resources information in such a large-scale distributed network, especially for the fully decentralized unstructured P2P network, it is impossible to make precise control over network topology and resource deployment. When we search for a particular resource without a centralized server, the queries should be propagated node by node for matching. This kind of distributed searching manner avoids the risk of single-point failure and shifts the searching loads to all participating nodes. Meanwhile, it also brings the problem of searching efficiency and scalability, which poses a great challenge to resource searching in the distributed networks. Furthermore, the dynamic nature of grid and P2P networks makes the searching problem more complicated. To address the problem, there currently are three kinds of research ideas on the following aspects:

- Search approaches or algorithms, which study the propagation of query messages among the nodes. Flooding [1] and random walk [2] are two commonly used methods for propagating queries. It is noted that the searching success rate lies on the number of nodes visited by queries from the perspective of probability. When a node cannot match the query it will propagate the query to all its neighbors according to flooding method, which results in revisits to some nodes, and generates more invalid query messages. LightFlood [3] is proposed to cut down the number of invalid messages by dividing the flooding into two stages while still visit the similar number of nodes as that of the standard flooding. Random walk requires the request node propagate query randomly to $k(k \geq 1)$ of its neighbors rather than all neighbors. If match failure the query will continue to propagate from neighbors to neighbors, and it can effectively avoids revisits to some nodes. Bin Wu et al. [4] present two analytical models to estimate the number of nodes visited. We use random walk as search approach in our model.
- Replication strategies. By proactively deploying resource information replicas [5-6] on several other nodes, the resources availability can be improved and access latency is cut down. Edith Cohen et al. [5] proposes that network achieves the optimal search performance only if the number of the replicas is proportional to the square-root of the query frequency. But due to the limitation of overall storage space excessive replication of information of hot resources will cover that of some cold resources, and consequently lead to failure of accessing these cold resources. It is unfair for cold resources and does not adapt well to the variation of hot resources. Furthermore, replicating those resources information will raise additional communication and storage overhead. BloomCast [7] proposes to spread and store Bloom

*Corresponding author:

Filters [8] of resources information instead of the raw information so as to reduce the overhead. However, the consistency maintenance of replica information should also be taken into account.

- Network topology optimization. In a fully decentralized unstructured P2P network, search is carried out by propagating and matching queries over the topology, and the network uses a combination of self-organization and cooperation with other peers to affect search performance [9]. This paper aims to provide a topology optimization model to improve search performance.

We first explore the relationship between the topological property (node degree) and the searching success rate, and design utility function accordingly. Then the problem of topology optimization is modeled as a multi-person non-cooperative game termed NETOP. By calculating the game's Nash Equilibrium (NE) we obtain the optimal distribution of node degrees. This paper mainly makes the following contributions:

(1) Analyze the relationship between node degree and searching success rate from the perspective of a resource provider, and present the contribution of an individual node to the success rate.

(2) Model topology optimization as a multi-person non-cooperative game, which is used to describe how each rational node chooses its connection degree during the optimization process. We prove the existence and uniqueness of Nash Equilibrium and solve it. To our best knowledge, it is the first time to utilize game theory for solving search problem.

(3) Present the expressions of the expected search performance according to Nash Equilibrium, and conduct experiment to verify the efficiency and effectiveness of NETOP.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents NETOP model for topology optimization. We extend the model for considering network dynamics in Section 4 and discuss how to obtain the public information in Section 5. Section 6 describes the performance evaluation of NETOP. Finally, we conclude our work in Section 7.

2 Related Work

Existing studies on topology construction and optimization in unstructured P2P networks mainly involve the following aspects:

- Topology construction based on nodes' interests. Gang Chen et al. [10] present a user (node) interest model, and suggest to construct a small-world network [11] through exploiting users' common interest patterns captured by the model, so as to enhance search performance. But this work cannot adapt well to frequent changes of users' interests. Mei Li et al. [12] design an approach for constructing a semantic small-world network in order to support efficient semantic-based search in P2P networks. However, semantics seems too complicated to achieve high efficiency.
- Topology optimization based on lifecycle. The dynamics of P2P networks usually exhibits the unpredictable join and departure of nodes, which has a great impact on search performance. Derek Leonard et al. [13] demonstrate that for a given average node degree d, the topology of d-regular graph performs the best resilience to node churn, and propose to keep high probability of network connectivity by selecting nodes from online nodes randomly as new neighbors instead of original offline neighbors. Zhong-Mei Yao et al. [14] study the characteristics of nodes' lifecycle in P2P networks and find it exhibit approximately heavy-tailed distribution. When a node updates its neighbors, it is suggested to choose long-lived nodes to reduce the probability of being isolated. Daniel Stutzbach et al. [15] show that the topology of Gnutella [20] network appears an "onion-like" structure, which implies nodes with similar lifecycle prefer to connect to nodes with the same or longer lifecycle, and then those long-lived nodes form a stable core that ensures search performance. Most of these previous works focus on enhancing search performance by reducing the influence of node churn only from the perspective of network dynamics. In this paper, we propose a topology optimization game model to guide nodes to construct the topology to improve search performance. Network dynamics is also taken into account to extend our model.
- Topology optimization based on node capability. Yatin Chawathe et al. [16] propose a self-organization topology. Each node evaluates its neighbors based on the capability of dealing with queries, and then connects with those one with high values for improving searching success rate. Hui-Rong Tian et al. [17] propose a reciprocal capability based adaptive topology protocol for P2P networks, which takes account of the node's rational belief of maintaining connections. Nodes are willing to establish connections only with nodes that are beneficial for them. This paper also takes node rationality into consideration, but we do work from the perspective of a resource provider rather than a resource requestor.
- Topology optimization based on load balancing. Matei Ripeanu et al. [18] propose that Gnutella network shows approximate power-law topology [21], which can achieve good search performance. A low-diameter unstructured P2P overlay called MPO with power-law characteristics is presented in [19]. However, high degree nodes in

power-law topology do not always have high capabilities, and are apt to overloading. Brian F. Cooper et al. [22] design a topology self-organization strategy to avoid overloading. Each node use connect() operation to form an ad hoc search or index connection to another one, and use break() operation to break a connection that produces too much load. We take connection cost into account for avoiding overloading in our work.

- Topology optimization based on node contribution. During a search, if a node satisfies the query, it makes contribution to the search from the perspective of a resource provider. Guo-Fu Feng et al. [23] investigate the relation among the distribution of the degree, resource access frequency and the success rate, and estimate node contribution. Then an optimal distribution of degrees is proposed in terms of node contribution. However, it primarily considers the case that each node only preserves one kind of resource. The calculation of optimal degree for case of multiple resources is simply the addition of degree of single resource, and it neglects superposition of multi-resources node degree. Brian F. Cooper [33] proposes an optimization scheme that suggests node degree should be proportional to the square root of the query frequency of resources possessed by the node. Square-root network is demonstrated to be optimal for random walk search algorithm. But it assumes that each type of resources has only one resource in the network and does not consider the maximum hop limit. Furthermore, these works do not take connection cost into consideration, and the pre-established network size and total connection degree do not reflect the network dynamics.

It is suggested to consider the node rationality in the process of topology optimization. Game theory [24] provides a good idea for solving strategies choice of non-cooperative rational entities. It wins a considerable amount of popularity for studying the problems of network resource allocation and task scheduling in grid and P2P networks [25-28]. Hong-Gang Zhang et al. [25] propose to model the interaction among nodes of unstructured P2P file sharing networks as unilateral and bilateral unstructured file sharing games, so as to solve the network resource (bandwidth) allocation problem. C. K. Tan et al. [26] model the problems of channel assignment and power control as a non-cooperative game, in which all wireless users jointly pick an optimal channel and power level to minimize a joint cost function. Krzysztof Rzadca et al. [27] suppose that task scheduling in distributed manner is analogous to the prisoner's dilemma game. Vasanth Kumar Ramesh [28] develops an auction based game theoretic framework for task scheduling in heterogeneous environments. However, few works combine game theory and studies of resource searching together. This paper introduces game theory into topology optimization, and presents a non-cooperative game based topology optimization model for solving search problems in unstructured P2P networks.

3 NETOP Model

The classic approaches of topology optimization attempt to design a mechanism in which each node selects the behaviors of topology construction with the aim of optimizing some search performance metrics. However, the interaction among rational nodes is absent from those approaches, for example, excessive behaviors of some nodes optimize their own performance, while it may result in the performance degradation of other nodes. Thus it may not necessarily guarantee the optimal performance on the whole. Game theory can be used to describe the interaction among rational nodes. As a rational node in the game, each one maximizes its utility by considering its contribution to success rate and the cost of maintaining connections. We establish the relation between node utility and its behavior of topology construction. In this study the behavior of topology construction refers to the choice of node degree. As a rational node, it chooses an appropriate connection degree according to the impact of other nodes on its own utility. In unstructured P2P networks each node maintains several connections to neighbors, the queries are spread out through these connections until the requested resource is found (search successfully) or the hop count reaches the upper bound (search failed). Assume that the resources are uniformly distributed in the network and have the identical density. The success rate of searching any resource is proportional to the number of nodes visited. From the requestor's point of view, a node wants to propagate the queries to more nodes within less hop count by increasing connection degree so as to enhance the searching success rate and reduce hop count and query messages. Nevertheless, the excessive increase in connection degree will result in a dense network in which we need high maintenance cost to reconnect the broken connects caused by node churn. As we know, the queries are satisfied by nodes that provide the requested resources when search finishes successfully. Node's dual roles (requestor and provider) allow us to study search problem from the perspective of resource provider. Node contribution is defined as how extent it satisfies queries, which can be used to represent the search performance factors. Moreover, we consider that node should pay the maintenance cost for providing the contribution, and define node utility as the difference between contribution and cost. In this way, if each node could maximize its utility, we can achieve higher search performance with lower cost.

3.1 Node Contribution

In a P2P network with N nodes, it is assumed that each node i is equipped with m_i ($m_i \ll N$) types of resources, the total number of resource types is R, and the topology is always connected. Each type has several resource instances, and the search process aims at finding the instances that belong to the required type. To avoid the impact of number of resource instances on search, we assume that each type of resource has the same number of resource instances, and is distributed uniformly in the network.

Node contribution refers to the contribution to the search when node acts as a resource provider, and is defined as the number of times that it can satisfy the query during a search. The contribution value lies on two factors: the number of times that the node is visited by the queries and the query hit rate on the node.

To make sure a certain node i been visited by the query it requires a path between node i and the request node s. In terms of random walk algorithm, the request node randomly propagates queries to k of its neighbors, and the maximum hop limit is set to h, so the path length should not more than h. Then we discuss the following two cases.

Case 1: There exists a direct connection between the request node s and node i shown in Figure 1(a). As we know, a connection connects two terminal nodes (Our discussion does not include the case that node connects itself) and contributes two degrees to the total node degree. Then a connection is considered to be constituted by both terminal nodes' degrees. The probability that there exists a connection between s and i through a certain degree of s is $\frac{d_i}{d_i + d_{-i} - d_s}$, where d_i is node i's degree, d_{-i} is the sum of all node degrees except i, and d_s is node s's degree. We use the average node degree d instead of d_s to represent an arbitrary node s's degree. Hence we get the probability of case 1: $\frac{d * d_i}{d_i + d_{-i} - d}$.

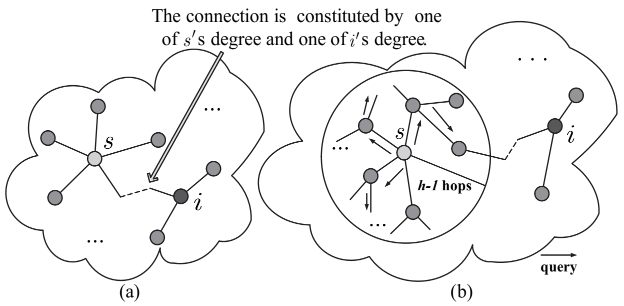


Figure 1 (a) There Exists a Direct Connection Between Node s and i; (b) There Exists a Connection Between Node i and an Arbitrary Node Visited by Queries within h – 1 Hops

Case 2: There exists a connection between node i and an arbitrary node visited by queries within $h - 1$ hops shown in Figure 1(b). We call these nodes intermediate nodes. Consider the probability that the node be visited repeatedly during a search is low for random walk algorithm, so it can be ignored here for consideration of simplicity. The number of intermediate nodes is $k(h - 1)$. In accordance with the analytical method in case 1, we get the probability of case 2: $k(h-1)\frac{(d-1) * d_i}{d_i + d_{-i} - d}$.

Combine the two cases together, the number of times that node i is visited during a search $p_i(h)$ can be calculated by:

$$\begin{aligned} p_i(h) &= \frac{d * d_i}{d_i + d_{-i} - d} * \frac{k}{d} + k(h-1)\frac{(d-1) * d_i}{d_i + d_{-i} - d} * \frac{1}{d-1} \\ &= \frac{d_i * kh}{d_i + d_{-i} - d} \end{aligned} \tag{1}$$

$$d_{-i} = \sum_{j \neq i}^{N} d_j, \; d = \frac{1}{N}\sum_{j=1}^{N} d_j$$

where $\frac{k}{d}$, $\frac{1}{d-1}$ are respectively used to represent the probability that s's connection to i is chosen to propagate queries and the probability that a intermediate node's connection to i is chosen to propagate queries during a search.

We test out our derivation by experiment. The experiment is carried out upon a random network with 10000 nodes, the node degree uniformly distributed from 1 to 50, and the average node degree is 5. As Figure 2 shown, high degree nodes receive more queries during the experiment and the pairs with higher product of k and h have the higher number of being visited during a search, which basically accords with Equation 1. The differences among pairs are more obvious for the higher degree nodes. In general, the number of being visited in the experiment is slightly lower than the theoretical value. That is because the

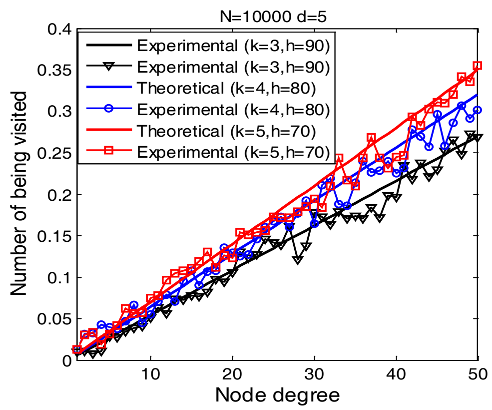


Figure 2 The Relation between the Node Degree (d_i) and Its Number of Being Visited By Queries ($p_i(h)$)

existence of a small number of revisits reduces the size of area that query arrives and makes it less than $k * h$.

It is observed in [30] that there exists a small part of resource types which are required with very high frequency in P2P networks, and these kinds of resource are known as hot resources. Suppose node i has R_i ($0 \le R_i \le m_i$) types of hot resources, then the hit rate on node i is:

$$Q_i = q_{hot} * \frac{R_i}{R_H} + (1 - q_{hot}) * \frac{m_i - R_i}{R - R_H} \tag{2}$$

where q_{hot} is the sum of access frequency of all hot resources, and it also represents the probability that the query is looking for hot resources. We use R_H to represent the number of hot resource types. Combine the above equations together, we get node i's contribution $F_i(h)$:

$$F_i(h) = p_i(h) * Q_i \tag{3}$$

In addition, assume that m is the average value of m_i ($i \in \{1,..., N\}$), we calculate the average hit rate by m/R.

3.2 Topology Optimization Game

We have discussed node contribution to the search performance in the first part, and through analyzing Equation 3 it is easy to find that the contribution value is tightly related to node degree and the hot resources possessed by node. In this study we assume that node utility is proportional to its contribution value. We also take the cost of maintaining connections into account, and assume it as $C_i = c_2 * d_i$ for simplicity, which implies it is proportional to the node degree. Equation 4 gives the definition of node utility u_i that is the difference between node contribution and its cost, c_1, c_2 ($c_1, c_2 > 0$) are two constants for making the twin values comparable.

$$u_i(d_i, d_{-i}) = F_i(h) * c_1 - C_i = \frac{d_i * khc_1}{d_i + d_{-i} - d} * Q_i - c_2 * d_i \tag{4}$$

Each node attempts to maximize its utility according to the above utility function, then the network can provide the best search performance while consume the least connection cost. Node can change its connection degree to maximize utility (other parameters k, h, R_i remain unchanged). Equation 4 implies that node i's utility lies not only on its degree but on degrees of other nodes. The change of degree of any other nodes will impact node i's utility, and vice versa. We model the problem as a multi-person non-cooperative game.

Definition 1. NETOP: the network is composed of N nodes, and each one is regarded as a player. The strategies space of N players are $D_1,..., D_N$, the utility function of them are $u_1,..., u_N$, and the set of all players' $Q_i (i \in \{1,..., N\})$ is the public information. We use $G = \{D_1,..., D_N; u_1,..., u_N\}$ to represent the game.

In this game, the utility of each node is the function of N-tuple of all nodes' strategies. If a node does not take others' strategies into account, it can boost utility by changing its strategy unilaterally. For example, a node may increase its connection degree appropriately to improve utility, but the node does not have the knowledge about others' strategies when it chooses its own strategies. In the same way, the other nodes may attempt to boost utilities by increasing their connection degrees, which results in the utility of each node may not necessarily improved, in contrast, the connection cost increases. As a rational node, it should take strategies of both its own and others together into account. Though the node does not have the knowledge about others' strategies, it can make predictions based on the public information, and adjusts its own strategy accordingly. The adjusted strategy is regarded as the optimal response to others. If all nodes choose strategies in this way, the game will reach a state of "strategy stability," namely Nash Equilibrium (NE). We give the definition of NE of this game and prove the existence and uniqueness of NE as follows.

Definition 2. NE: In NETOP game $G = \{D_1,..., D_N; u_1,..., u_N\}$, when each player $i \in \{1,..., N\}$ chooses strategy d_i resulting in N-tuple strategy profile $(d_1,..., d_N)$ then player i obtains utility $u_i(d_1,..., d_N)$. A strategy profile $(d_1^*,... d_N^*)$ is a Nash Equilibrium (NE) if no unilateral deviation in strategy by any single player is profitable for that player, that is

$$u_i(d_1^*, \dots, d_{i-1}^*, d_i^*, d_{i+1}^*, \dots, d_N^*) \ge u_i(d_1^*, \dots, d_{i-1}^*, d_i, d_{i+1}^*, \dots, d_N^*)$$
$$\forall i \in \{1, \dots, N\}, d_i \in D_i, d_i \ne d_i^*$$

Theorem. NETOP game has a unique NE.

Proof: In this game, the node's problem of strategy choice can be transformed into an optimization problem as follows:

$$\max_{d_i \in D_i} u_i(d_i, d_{-i}^*), d_i > 0, \forall i \in \{1 \dots N\} \tag{5}$$

Calculate the second derivative of utility function we get

$$u_i'' = -\frac{2khc_1 * (d_{-i} - d)}{(d_i + d_{-i} - d)^3} < 0$$

so u_i is continuous and concave in d_i. Each node's D_i is a closed convex set. In terms of theorem 1 and 2 in [31], there exists equilibrium point, and the point of optimization problem is unique. Hence the game has a unique NE. □

To solve the NE of this game, let $u_i' = 0$, we obtain:

$$\begin{cases} \dfrac{(d_{-1}-d)*khc_1}{(d_1+d_{-1}-d)^2}*Q_1-c_2=0 \\ \dots \\ \dfrac{(d_{-i}-d)*khc_1}{(d_i+d_{-i}-d)^2}*Q_i-c_2=0 \\ \dots \end{cases}, i\in\{1,\dots,N\}, d_i>0 \quad (6)$$

Substitute $d_i + d_{-i} = Nd$ into Equation 6 and make additions to all equations, we have:

$$(N-2)Nd=(N-1)^2d^2\sum_{j=1}^{N}\frac{c_2}{khc_1*Q_j}$$

$$d=\frac{c_1}{c_2}*\frac{N(N-2)*kh}{(N-1)^2*\sum_{j=1}^{N}\frac{1}{Q_j}}$$

Then we obtain the NE solution by substituting the above equation into Equation 6.

$$\begin{aligned} d_i^* &= Nd-d_{-i} \\ &=(N-1)d-\frac{c_2}{khc_1*Q_i}(N-1)^2d^2 \\ &=\frac{c_1}{c_2}\frac{N(N-2)*kh}{(N-1)*\sum_{j=1}^{N}\frac{1}{Q_j}}\left(1-\frac{N^3(N-2)}{(N-1)^3*Q_i\sum_{j=1}^{N}\frac{1}{Q_j}}\right) \end{aligned} \quad (7)$$

3.3 Performance Analysis

In terms of the game results, we can derive analytically the expected search performance of NE. The main performance metrics concerned are the number of resource instances that are found $P(h)$, the average hop count taken to find the resource $T(h)$ and the average number of messages $M(h)$.

When the game reaches NE, $P(h)$ can be represented by the sum of all nodes' contributions.

$$P(h)=\sum_{i=1}^{N}F_i(h)=\sum_{i=1}^{N}\frac{d_i^**kh*Q_i}{d_i^*+d_{-i}^*-d^*}$$
$$|P(h)|=\begin{cases}P(h), P(h)<1\\ 1 \quad, P(h)\ge 1\end{cases} \quad (8)$$

Equation 8 shows the relation between the number of resource instances found and node degree. We use $|P(h)|$ to represent the searching success rate, and q_i to denote the probability of finding the first requested resource exactly at the ith hop, then the searching success rate can also be represented by $|P(h)|=\sum_{i=1}^{h}q_i$. Now assuming that the resource is found successfully within h hop counts, the probability of finding the first requested resource instance exactly at the ith hop should be $\frac{q_i}{|P(h)|}$, so the average hop count can be calculated by:

$$\begin{aligned} T(h) &= \sum_{i=1}^{h} i*\frac{q_i}{|P(h)|} \\ &= \frac{1}{|P(h)|}\left(\sum_{i=1}^{h-1} iq_i+hq_h\right) \\ &= \frac{1}{|P(h)|}\left(T(h-1)|P(h-1)|+hq_h\right) \end{aligned}$$

where $q_h = |P(h)| - |P(h-1)|$, substitute to above equation, we get:

$$\begin{aligned} T(h)|P(h)| &= T(h-1)|P(h-1)|+h|P(h)|-h|P(h-1)| \\ &= \sum_{i=1}^{h} i|P(i)|-\sum_{i=1}^{h} i|P(i-1)| \\ &= h|P(h)|-\sum_{i=1}^{h-1} i|P(i)| \end{aligned}$$

$$T(h)=h-\frac{1}{|P(h)|}\sum_{i=1}^{h-1}|P(i)| \quad (9)$$

By analyzing Equation 9, we conclude that $T(h)$ is related to both h and $|P(h)|$, increasing h exclusively may not always lead to continuous increase of $T(h)$. In terms of random walk algorithm, the request node sends queries to k of its neighbors randomly generating k query messages, and each query message forms a query route and spreads along this route. We assume that the hit node returns a message directly to the request node when the resource is found. Then the average number of messages is given by:

$$M(h)=k+kM'(h-1), (h\ge 1) \quad (10)$$

where $M'(h-1)$ denotes the number of messages generated by every route which has a length of $h-1$. If the resource is found on one node on the path (the average hit rate is m/R), it will generate a success message, otherwise, propagate message to the next node on the path until reaching the end of the path. The recursive expression can be denoted by:

$$M'(h-1)=m/R+(1-m/R)(1+M'(h-2))$$

Since $M'(0) = 1$, we derive the following equation

$$M'(h-1)=R(1-(1-m/R)^h)/m$$

Substitute the above equation to Equation 10, we obtain:

$$M(h) = k + kR(1-(1-m/R)^h)/m, (h \geq 1) \quad (11)$$

Equation 11 implies that the number of average messages is tightly related to both the average hit rate and the parameters of search algorithm, and as far as random walk concerned it is independent of topology.

4 Model Extension

It is assumed that the nodes (players) of game are predefined without change when we define NETOP in Section 3, which is inconsistent with the fact of dynamics of P2P networks. Node churn will have influence on the search performance, so we need to extend NETOP to adapt to the description of node churn.

In P2P networks, node churn can be described by using node arrival rate and lifecycle. It is noted by [32] that the distribution of node lifecycles in P2P networks is often heavy-tailed. In this study, we assume that node arrival rate obeys Poisson distribution $Po(\lambda)$, and node lifecycle obeys Pareto distribution that is one of heavy-tailed distributions. To allow arbitrarily small lifecycles, Pareto distribution is modified in [29] by:

$$F(x) = 1 - \left(\frac{\beta}{\beta + x}\right)^{\alpha}, \alpha > 1, \beta > 0$$

Let the node arrival time is $\gamma(0 \leq \gamma \leq t)$, according to the properties of Pareto distribution, the probability that a node is still alive is $\left(\frac{\beta}{\beta+t-\gamma}\right)^{\alpha}$. Now we use $Pa_{live}(t)$ to denote the probability that a node arrives in the network at anytime of $[0, t]$, and is still alive at time t.

$$\begin{aligned} Pa_{live}(t) &= \frac{1}{t}\int_0^t \left(\frac{\beta}{\beta+t-\gamma}\right)^{\alpha} d\gamma \\ &= \frac{\beta}{t} * \frac{1-(1+t/\beta)^{1-\alpha}}{\alpha - 1} \end{aligned}$$

As we have mentioned, the distribution of node arrival rate is Poisson with λ as its expectation, the network size $N(t)$ at time t should be:

$$N(t) = \lambda t * Pa_{live}(t) = \frac{\lambda\beta - \lambda\beta(1+t/\beta)^{1-\alpha}}{\alpha - 1} \quad (12)$$

When $t \to +\infty$, the network size tends to be $\lambda\beta/(\alpha - 1)$, then it reaches a dynamic equilibrium. Therefore, NETOP can be extended to be a time-related dynamic game defined as follows.

Definition 3. Dynamic NETOP: There exists $N(t)$ nodes at time t in the network, and each node is regarded as a player. The strategies space of all players are $D_1,..., D_N(t)$, the utility function of them are $u_1,... u_N(t)$, and the set of all players' Q_i ($i \in \{1,..., N(t)\}$) is the public information. We use $G(t) = \{D_1,..., D_N(t); u_1,... u_N(t)\}$ to represent the extended game.

According to the derivation of section 3.3, the NE solution of the extended game at time t is:

$$d_i^*(t) = \frac{c_1}{c_2}\frac{N(t)(N(t)-2)*kh}{(N(t)-1)*\sum_{j=1}^{N(t)}\frac{1}{Q_j}}\left(1 - \frac{N^3(t)(N(t)-2)}{(N(t)-1)^3 * Q_i \sum_{j=1}^{N(t)}\frac{1}{Q_j}}\right) \quad (13)$$

The number of resource instances found $P(h, t)$ and average hop count $T(h, t)$ at time t are respectively:

$$P(h,t) = \sum_{i=1}^{N(t)} p_i(h,t) * Q_i = \sum_{i=1}^{N(t)} \frac{d_i^*(t) * kh * Q_i}{d_i^*(t) + d_{-i}^*(t) - d^*(t)} \quad (14)$$

$$T(h,t) = h - \frac{1}{|P(h,t)|}\sum_{i=1}^{h-1} |P(i,t)| \quad (15)$$

The average number of messages propagated during a search $M(h, t)$ is calculated by:

$$M(h,t) = k + kPa_{live}(t)M'(h-1,t), (h \geq 1)$$

where

$$M'(h-1,t) = m/R + (1-m/R)(1+M'(h-2,t))Pa_{live}(t)$$

Let $f = 1 - m/R$, since $M'(0, t) = 1$, when $h \geq 1$,we get:

$$M(h,t) = k + kPa_{live}(t)\frac{(f - f(1-Pa_{live}(t))^{d^*(t)})^h - 1}{f - f(1-Pa_{live}(t))^{d^*(t)} - 1} \quad (16)$$

where $d^*(t)$ is the average node degree at time t. Different from the static game discussed in section 3, from Equation 16 we find $M(h, t)$ is related to $d^*(t)$, which implies the average number of messages has close relations with the optimized topology at time t in dynamic NETOP.

5 Discussion

In the previous sections, we propose a topology optimization model NETOP and analyze search performance accordingly. In terms of definitions of NETOP and NE, each participating node needs the public information Q_i ($i \in \{1,..., N(t)\}$) to calculate its optimal strategy $d_i^*(t)$ at NE. In this section, we discuss approaches to achieve the public information.

The public information as the global information is the set of all nodes' hit rate $\{Q_i\}$. The super node is a choice for collecting and offering the information from a centralized point of view, but it may become a bottleneck of the whole network. More importantly, the centralized approach is contrary to the principle of P2P. We propose a distributed approach to send and receive the information. According to our approach, each node has two tables: table online and table offline shown in Figure 3. Table online is used to record all online nodes' hit rate $\{Q_i\}$ and node i's table online is initially filled out by its own hit rate, while table offline is used under the condition of dynamics to record the departed nodes. Assume that each node maintains connections to its neighbors through sending and receiving heartbeat packets periodically. For any neighbor of node i, say node j, if node i has not received any packets from it for a given period, node i will think node j has departed and add it into table offline. In order to collect more information to update the tables, the tables are sent out accompanied by queries. We use Bloom Filters of tables (BF_{on}, BF_{off}) instead of raw tables to reduce the traffic overhead. The process of tables updating is described below:

Online

Node ID	Q_{ID}
N00001	*0.0132*
N00345	*0.0041*
N01053	*0.1281*
N10790	*0.0038*
...	...

Offline

Node ID
N03081
N10432
N30512
N10018
...

Figure 3 An Example of Twain Tables

(1) Each node uses BF_{off} to update its BF_{on} by eliminating the items appeared in BF_{off}, denoted by $BF_{on} - BF_{off}$.

(2) When node j receives node i's BFi_{on}, BFi_{off}, then merge both nodes' Bloom Filters,

$$BFi_{on} \cup BFj_{on},\ BFi_{off} = BFj_{off} = BFj_{off} \cup BFj_{off}$$

and eliminate the offline items

$$BFi_{on} = BFj_{on} = BFi_{on} \cup BFj_{on} - BFi_{off}$$

In P2P networks, each node is aware of local information achieved from its neighbors. Through repeating the above updating process our approach utilizes the propagation and merge of local information to obtain the global information approximately. Consider the dynamic situations, node churn including node's joining and departure can be perceived by its neighbors, then they will update their own Bloom Filters of tables and send the information out accompanied by queries. Our approach uses two ways to update tables, one way is active mode, a node send out Bloom Filters of tables with queries for updating; another one is passive mode, which implies a node passively receives Bloom Filters of tables accompanied by queries from other nodes and merges together. Hence the refresh rate is fast, which also guarantee the rapidity of convergence and consistency of public information.

6 Performance Evaluation

In this section, we perform an experimental study to evaluate the performance of NETOP, and compare it with that of Square-root [33] and Power-law [21] topologies.

Table 1 Experimental Parameters and Their Values

Parameters	values
Network size (N)	10000
Node capacity (m_i)	Uniform distribution [5] [25]
Contribution constant (c_1)	1
Connection cost constant (c_2)	2.721*10-4
Number of resource types (R)	2000
Node arrival rate	Poisson distribution $Po(100)$
Node lifecycle	Pareto distribution (2, 100)
Search algorithm	Random walk

6.1 Experimental Methodology

We implement an unstructured P2P simulator to generate different P2P topologies and run searches. There are 2000 resource types in total, each type has 50 resource instances, and all resources are distributed uniformly in the network. The search process proceeds in discrete time steps, 100 nodes are chosen randomly to issue queries at each step. It is assumed that the resource query frequency follows Zipf's law distribution ($\tau = 2.05$) [35]. Hot resource types are 5 percent of total number of resource types, and queries for hot resources are 60 percent of total query frequency ($q_{hot} = 60\%$). Nodes' arrival process follows Poisson distribution, and the distribution of lifecycle is the shifted Pareto defined in section 4. Here the performance metrics concerned are $P(h)/C$, $T(h)$, $M(h)$ (or $P(h, t)/C(t)$, $T(h, t)$, $M(h, t)$). $T(h)$, $M(h)$ are defined as the average hop count taken to find the first resource instance and the average number of messages propagated which is discussed in section 3, respectively. The connection cost is also taken into account, let C denotes the total connection cost $C = \sum_{i=1}^{N} C_i$, and $P(h)/C$ is defined as the number of resource instances found divided by the connection cost, which is used to measure the search efficiency. According to

Equation 7, the node degree at NE is related to the ratio c_1/c_2, and we set the ratio as 1 : 2.721 * 10^{-4} so as to make the average node degree at NE less than 5. The experimental parameters and their values are listed in table 1.

6.2 Experimental Results

6.2.1 Performance of Approach of Achieving Public Information

Before optimizing network topology, we firstly evaluate the efficiency of the approach discussed in Section 5. In the case of a static network, the initial network is a random network with size of 10000 nodes and the average node degree is 5. The search process is executed 50 time steps, and 50 * 100 searches are issued in all. Each node collects the public information through visiting or being visited so as to calculate the sum of $1/Q_j$ ($j \in \{1,..., N\}$) (namely $\sum_{j=1}^{N}\frac{1}{Q_j}$, which is used in Equation 7). Everyone has its own result value, we average all these values and use it to measure the integrity of public information. As Figure 4 described, the average value is low at the first period, and then boosts rapidly until the real value. That is because a node is only aware of the information within a small area, and this area is expanded greatly with the information merge among nodes. This also interprets why the differences among all nodes' values (measured by standard deviation) are low at first, increases as the area being expanded and drops down as the merging degree being enhanced. The standard deviation tends to be zero after 20 time steps, which guarantees the rapidity of convergence and consistency of public information.

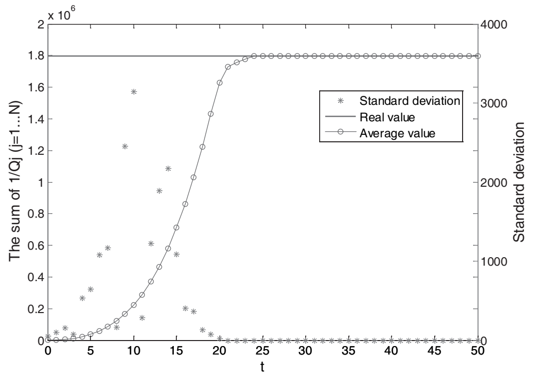


Figure 4 The Efficiency of Approach in a Static Network

In the case of a dynamic network, we focus on the efficiency of approach under the condition of node churn. Figure 9(a) indicates that the network size boosts rapidly at first, and stabilizes after 2000 time steps. So we mainly evaluate the efficiency within the first 2000 steps. As Figure 5 indicated, both the average value and real value have the similar trend with network size. Because there are new nodes joining and old nodes departing the network at the last few steps of each statistical point, the information of those nodes' hit rates have no time to spread out and update the public information. The difference between the average value and real value always exists, but it is very small and the low standard deviation guarantees the consistency of public information in a dynamic environment to a large extent.

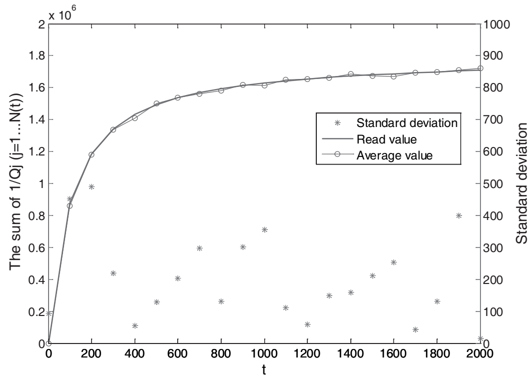


Figure 5 The Efficiency of Approach in a Dynamic Network

6.2.2 Performance Comparison in a Static Network

After collecting the public information, each node optimizes the network topology by adjusting its connection degree according to the number of hot resources possessed by itself and the public information. Figure 6 describes the NE of game, and the experimental results are basically in conformity with Equation 7. There exist some derivations between the experimental and theoretical results, because in theoretical analysis we use the number of hot resources instead of resource query frequency to calculate the hit rate, and the frequency differences among hot resources are ignored. Figure 7 shows the distribution of node degrees in NETOP network, it is observed that most nodes' degrees concentrate on the interval of [1][15], and the average node degree of NETOP is about 4.5.

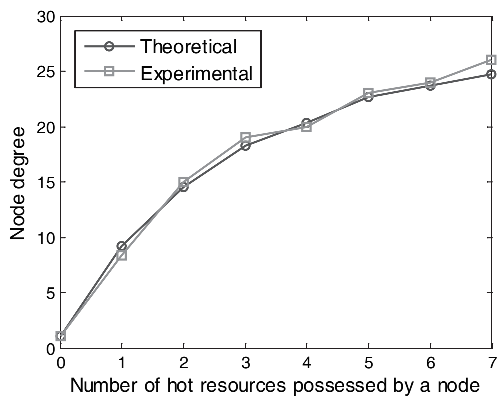


Figure 6 The Relation between the Number of Hot Resources Possessed by a Node (R_i) and Its Connection Degree (d_i)

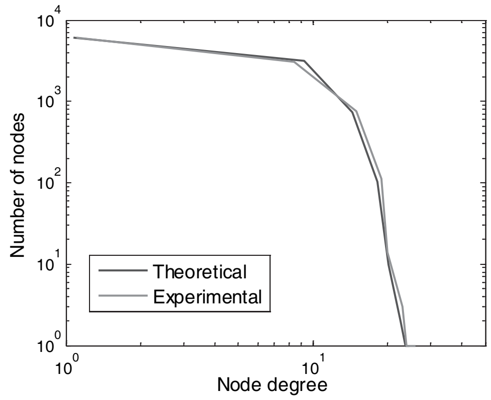
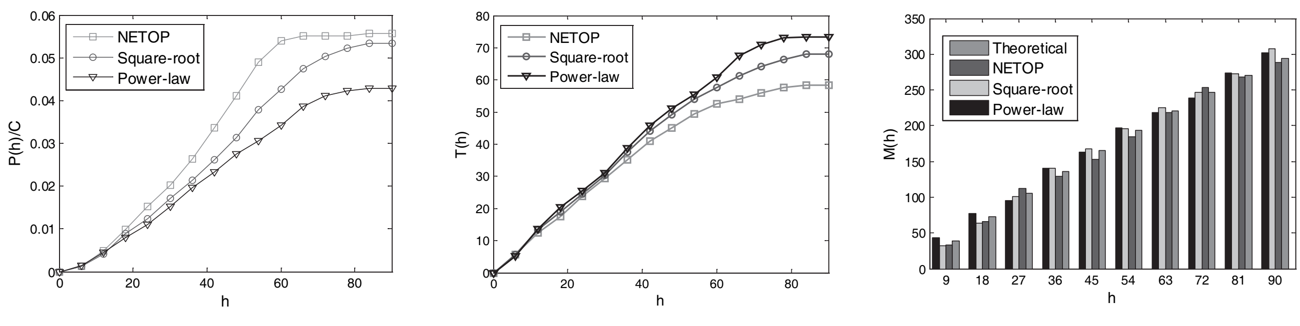


Figure 7 The Topology Optimized by NETOP

Power-law topology is a common used network model, and we construct a Power-law network according to [34]. Square-root network is proposed to reduce the average hot count compared to Power-law network. We construct a Square-root network according to square-root-construct algorithm described in [33]. Different from the other two networks which exclusively focus on the improvement of success rate and average hot count, we also take network connection cost into account. The experimental result is shown in Figure 8(a), compared with Power-law and Square-root networks, the value of NETOP is 33.3% and 6% higher respectively, which illustrates that NETOP network can achieve higher success rate with lower connection cost. It is demonstrated by [33] that the Square-root network is optimal for random walk search algorithm. But it is based on the assumption that each type of resources has only one resource instance in the network. It does not always hold for the case of multiple instances of each type, since as for an arbitrary type of resource, it neglects the superposition of contributions of multi-instances. We evaluate the average hop count $T(h)$ and messages $M(h)$ among the three types of network, and the results are depicted by Figure 8(b) and (c). Before $h < 40$, all types of network almost have the identical $T(h)$. As h continues increasing, $T(h)$ of NETOP network stabilizes faster, and is lower than that of the others, 18% and 13.2% respectively. When $h = 90$, NETOP network consumes 4.6% and 6.5% fewer messages than the other two networks. The experimental result of $M(h)$ is nearly consistent with Equation 11, and demonstrate that it is independent of topology for random walk without considering network dynamics.

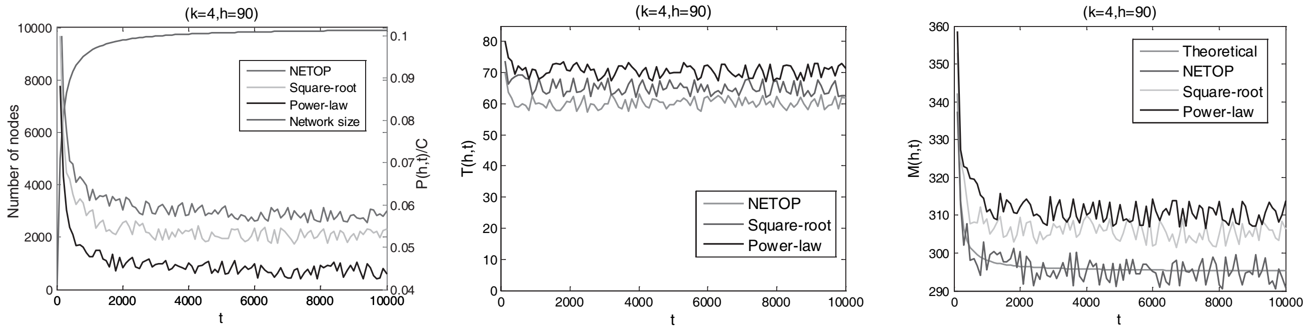
6.2.3 Performance Comparison Under the Condition of Node Churn

In this part of experiment, we evaluate the performance of NETOP under the dynamic condition of node churn, and also compare it with Square-root and Power-law topologies. We adjust network topology every 100 time steps, as for NETOP model it is sufficient for nearly all online nodes to collect the public information except the new nodes coming at the last step before each adjustment. For these new nodes, we let them choose a neighbor randomly and fetch the public information from their neighbors. Figure 9 illustrates search performance comparison under the condition of node churn. From Figure 9(a) we find the network size boost rapidly at first, and stabilize after 2000 time steps. The network connection cost C increases with the boost of network size, which results in the degradation of search efficiency. However, NETOP network still maintains 28.9% and 11.5% higher value than the others. Before stabilization, the small network size may lead to the scarcity of some types of resources. The requests for these resources call for the higher hop count and more query messages. So the average hop count $T(h, t)$ and number of messages $M(h, t)$ are higher at first shown in Figure 9(b) and (c). Moreover, as time increases the nodes' alive probability $Pa_{live}(t)$ decreases rapidly till the network comes to a dynamic equilibrium, which also interprets why $M(h, t)$ drops quickly before stabilization. In terms of Equation 16, NETOP network with lower average node degree $d^*(t)$ achieves fewer $M(h, t)$. The experimental results show that when compared with the other two networks, NETOP network requires 14.3% and 7.7% lower average hop count and 26.6% and 28.7% fewer messages.



(a) (b) (c)

Figure 8 Search Performance Comparison in Static Networks. (a) The Comparison of Search Efficiency ($P(h)/C$) under Increasing h with Others; (b) The Comparison of Average Hop Count ($T(h)$) under Increasing h with Others; (c) The Comparison Of Average Number of Messages ($M(h)$) under Increasing h with Others



(a) (b) (c)

Figure 9 Search Performance Comparison in Dynamic Networks; (a) The Comparison of Search Efficiency ($P(h, t)/C(t)$) under Increasing t with Others; (b) The Comparison of Average Hop Count ($T(h, t)$) under Increasing t with Others; (c) The Comparison of Average Number of Messages ($M(h, t)$) under Increasing t with Others

7 Conclusion

In this paper, we propose NETOP, a non-cooperative game based topology optimization model, with the aim of enhancing resource searching performance. By exploiting the relations among node degree, node contribution to searching success rate and connection cost, we present the node utility function and model the optimization problem as a multi-person non-cooperative game. In the game, each node is viewed as a rational player and attempts to maximize its utility so as to achieve higher search performance with lower connection cost. By calculating the NE of game we analyze the expected search performance theoretically. Furthermore, we take node churn into account, extend NETOP to be a time-related dynamic game accordingly and put forward a distributed approach for nodes to obtain the public information. We carry out the experiments in both static and dynamic conditions, to confirm the efficiency and effectiveness of our model. The results show that NETOP network converges rapidly and achieves higher search performances. When compared with Power-law and Square-root topologies in a static condition, NETOP network achieves the same success rate with 33.3% and 6% lower connection cost, 18% and 13.2% lower average hop count, and 4.6% and 6.5% fewer messages, meanwhile in a dynamic condition, it achieves 28.9% and 11.5% lower connection cost, 14.3% and 7.7% lower average hop count, and 26.6% and 28.7% fewer messages.

This work serves as our elementary work towards understanding the relation between topology and search performance. As future work, we want to study how to update node's neighbors more effectively based on its optimal degree. In addition, replication strategies and resource indexing techniques can also be combined to appropriately reduce the average hop count and messages.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grants No. 60903161, No. 60903162, No. 60773103 and No. 90912002, National Key Basic Research program of China under Grants No. 2010CB328104, China National Key Technology R&D Program under Grants No. 2010BAI88B03, China Specialized Research Fund for the Doctoral Program of Higher Education under Grants No. 200802860031, Jiangsu Provincial Natural Science Foundation of China under Grants No. BK2008030, China National S&T Major Project under Grants No. 2009ZX03004-004-04, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

References

[1] Vassilios V. Dimakopoulos and Evaggelia Pitoura, *On the Performance of Flooding-Based Resource Discovery*, *IEEE Transactions on Parallel and Distributed Systems*, Vol.17, No.11, 2006, pp.1242-1252.

[2] Christos Gkantsidis, Milena Mihail and A. Saberi, *Random Walks In Peer-To-Peer Networks*, *Proc. IEEE INFOCOM*, Atlanta, GA, March, 2004, pp. 120-130.

[3] Jiang Song, Lei Guo, Xiao-Dong Zhang and Hao-Dong Wang, *LightFlood: Minimizing Redundant Messages and Maximizing Scope of Peer-to-Peer Search*, *IEEE Transactions on Parallel and Distributed Systems*, Vol.19, No.5, 2008, pp.601-614.

[4] Bin Wu and Ajay D. Kshemkalyani, *Analysis Models for Unguided Search in Unstructured P2P Networks*, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol.3, No.4, 2008, pp. 255-263.

[5] Edith Cohen and Scott Shnker, *Replication Strategies in Unstructured Peer-To-Peer Networks*, *Proc. ACM SIGCOMM*, Pittsburgh, PA, August, 2002, pp.177-190.

[6] Guo-Fu Feng, Yu-Quan Jiang, Gui-Hai Chen, Sang Lu and Dao-Xu Chen, *Replication Strategy in Unstructured Peer-to-Peer Systems*, *Proc. IEEE IPDPS*, Long Beach, CA, March, 2007.

[7] Han-Hua Chen, Hai Jin and Xu-Cheng Luo, *BloomCast: Efficient Full-Text Retrieval over Unstructured P2Ps with Guaranteed Recall*, *Proc. IEEE/ACM CCGrid*, Shanghai, China, May, 2009, pp.52-59.

[8] Burton H. Bloom, *Space/Time Trade-offs in Hash Coding with Allowable Errors*, *Communication of the ACM*, Vol.13, No.7, 1970, pp.422-426.

[9] John F. Buford, *Management of Peer-To-Peer Overlays*, *International Journal of Internet Protocol Technology*, Vol.3, No.1, 2008, pp.2-12.

[10] Gang Chen, Chor-Ping Low and Zhong-Hua Yang, *Enhancing Search Performance in Unstructured P2P Networks Based on Users' Common Interest*, *IEEE Transactions on Parallel and Distributed Systems*, Vol.19, No.6, 2008, pp.821-836.

[11] Duncan J. Watts and Steven H. Strogatz, *Collective Dynamics of 'Small-World' Networks*, *Nature*, Vol.393, 1998, pp.440-442.

[12] Mei Li, Wang-Chien Lee, Anand Sivasubramaniam and Jing Zhao, *SSW: A Small-World-Based Overlay for Peer-to-Peer Search*, *IEEE Transactions on Parallel and Distributed Systems*, Vol.19, No.6, 2008, pp.735-749.

[13] Derek Leonard, Zhong-Mei Yao, Vivek Rai and Dmitri Loguinov, *On Lifetime-Based Node Failure And Stochastic Resilience of Decentralized Peer-To-Peer Networks*, *IEEE/ACM Transactions on Networking*, Vol.15, No.3, 2007, pp.644-656.

[14] Zhong-Mei Yao, Xiao-Ming Wang, Derek Leonard and Dmitri Loguinov, *On Node Isolation under Churn in Unstructured P2P Networks with Heavy-Tailed Lifetimes*, *Proc. IEEE INFOCOM*, Anchorage, AK, May, 2007, pp.2126-2134.

[15] Daniel Stutzbach, Reza Rejaie and Subhabrata Sen, *Characterizing Unstructred Overlay Topologies in Modern P2P File-Sharing Systems*, *IEEE/ACM Transactions on Networking*, Vol.16, No.2, 2008, pp.267-280.

[16] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham and Scott Shenker, *Making Gnutella-like P2P Systems Scalable*, *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August, 2003, pp.407-418.

[17] Hui-Rong Tian, Shi-Hong Zou, Wen-Dong Wang and Shi-Duan Cheng, *Incentive Compatible Adaptive Topologies For P2p Networks*, *Journal of Software*, Vol.17, No.4, 2006, pp.845-853.

[18] Matei Ripeanu, Ian Foster and Adriana Iamnitchi, *Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design*, *IEEE Internet Computing Journal*, Vol.6, No.1, 2002.

[19] Deng Li, Zhi-Gang Chen, Jia-Qi Liu and Athanasios V. Vasilakos, *A Low-Diameter Unstructured P2P Overlay and Corresponding Search Algorithm*, *Journal of Internet Technology*, Vol.10, No.3, 2009, pp.195-205.

[20] Guntella protocol specification version 0.4, http://www9.limewire.com/developer/ gnutella_protocol_0.4.pdf

[21] Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos, *On Power-Law Relationship of the Internet Topology*, *Proc. ACM SIGCOMM*, Cambridge, MA, August, 1999, pp.251-262.

[22] Brian F. Cooper and Hector Garacia Molina, *Ad Hoc, Self-Supervising Peer-to-Peer Search Networks*, *ACM Transactions on Information Systems*, Vol.23, No.2, 2005, pp.169-200.

[23] Guo-Fu Feng, Jin-Cheng Zhang, Yu-Quan Jiang, Qing Gu, Sang-Lu Lu and Dao-Xu Chen, *Optimization of Overlay Topology for Unstructured Peer-to-Peer*, *Journal of Software*, Vol.18, No.11, 2007, pp.2819-2829.

[24] Robert Gibbons, *A Primer in Game Theory*, Prentice Hall, New York, 1992.

[25] Hong-Gang Zhang, Giovanni Neglia, Don Towsley and Giuseppe Lo Presti, *Stability and Efficiency of Unstructured File Sharing Networks*, *IEEE Journal of Selected Areas In Communications*, Vol.26, No.7, 2008, pp.1284-1294.

[26] C. K. Tan, M. L. Sim and T. C. Chuah, *Game theoretic approach for channel assignment and power control with no-internal-regret learning in wireless ad hoc networks*, *IET Communications*, Vol.2, No.9, 2008, pp.1159-1169.

[27] Krzysztof Rzadca, Denis Trystram and Adam Erzbicki, *Fair Game-Theoretic Resource Management In Dedicated Grids*, *Proc. IEEE CCGrid*, Rio de Janeiro, Brazil, 2007, pp.343-350.

[28] Vasanth Kumar Ramesh, *A Game Theoretic Framework for Dynamic Task Scheduling In Distributed Heterogeneous Computing Systems*, *Theses and Dissertations Paper 828*, University of South Florida, FL, 2005.

[29] Mette Rytgaard, *Estimation in the Pareto Distribution*, *ASTIN BULLETIN*, Vol.20, No.2, 1990, pp. 201-216.

[30] Kunwadee Sripanidkulchai, *The Popularity of Gnutella Queries and Its Implications on Scalability*, Carnegie Mellon University, Pittsburgh, PA, 2001.

[31] J. B. Rosen, *Existence and Uniqueness of Equilibrium Points for Concave N-Person Games*, *Econometrica*, Vol.33, No.3, 1965, pp.520-534.

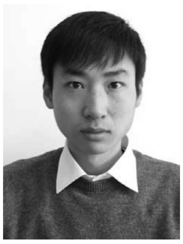
[32] Zhong-Mei Yao, Derek Leonard, Xiao-Ming Wang and Dmitri Loguinov, *Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks*, *Proc. IEEE ICNP*, Santa Barbara, CA, 2006, November, pp.32-41.

[33] Brian F. Cooper, *An Optimal Overlay Topology for Routing Peer-To-Peer Searches*, *Proc. ACM/IFIP/USENIX Middleware*, Grenoble, France, November, 2005, pp.82-101.

[34] Christopher R. Palmer and J. Gregory Steffan, *Generating Network Topologies that Obey Power Laws*, *Proc. IEEE GLOBECOM*, San Francisco, CA, November, 2000, pp.434-438.

[35] Lada A. Adamic and Bernardo A. Huberman, *Zipf's Law and The Internet*, *Glottometrics*, Vol.3, 2002, pp.143-150.

Biographies



Jingya Zhou is currently a PhD candidate in School of Computer Science and Engineering, Southeast University, China. He received his BS degree in Computer Science from Anhui Normal University, China in 2005. His current research interests include p2p computing, grid computing and service computing.

Junzhou Luo is currently a professor and the dean in School of Computer Science and Engineering, Southeast University, China. He received his MS and PhD degrees in Computer Science from the Southeast University, China in 1992 and 2000, respectively. His current research interests include grid computing, network security, service computing and protocol engineering.

Aibo Song is currently an associate professor in School of Computer Science and Engineering, Southeast University, China. He received his MS and PhD degrees in Computer Science from the Southeast University, Nanjing, China in 1996 and 2003, respectively. His current research interests include p2p computing and grid computing.