SOBIE:A Novel Super-node P2P Overlay Based on Information Exchange

Zhigang Chen[1] ,
1. School of Information Science and Engineering, Central South University, Hunan, P. R. China
Email: czg@mail.csu.edu.cn

Jiaqi Liu[1], Deng Li [1,*],Hui Liu[2], Athanasios Vasilakos[3]
1. School of Information Science and Engineering, Central South University, Hunan, P. R. China
Email: { jiaqi1006@gmail.com, dongxuanlee@gmail.com }
2. Dept. of Computer Science, Missouri State University, Springfield, MO,USA
Email:HuiLiu@missouristate.edu
3. Dept. of Computer and Telecommunication Engineering, University of Western Macedonia,Greece
Email:vasilako@ath.forthnet.gr

***Abstract*—In order to guarantee both the efficiency and robustness in the Peer-to-Peer (P2P) network, the paper designs a novel Super-node Overlay Based on Information Exchange called SOBIE. Differing from current structured and unstructured, or meshed and tree-like P2P overlay, the SOBIE is a whole new structure to improve the efficiency of searching in the P2P network. The main contributions are 1) to select the super-nodes by considering the aggregation of not only the delay, distance, but also the information exchange frequency, exchange time and query similarity especially; 2) to set a score mechanism to identify and prevent the free-riders. Meanwhile, the SOBIE also guarantees the matching between the physical network and logical network and has small-world characteristic to improve the efficiency. Large number of experiment results show the advantages of the SOBIE including high efficiency and robustness by such different factors as the query success rate, the average query hops, the total number of query messages, the coverage rate and system connectivity.**

***Index Terms*—P2P overlay, super node, information exchange, topology matching, free-ridding**

I. INTRODUCTION

P2P systems are designed for the sharing of computer resources (content, storage, CPU cycles) by direct exchange, rather than requiring the intermediation or support of a centralized server or authority [1]. P2P systems have internal dynamic that peers may dynamically join and leave the network. Every peer is connected virtually with other peers in the application layer of the network system structure so that all peers in the system compose a virtual network called overlay [2].Although the virtual network is built on the physical network, and the former depends on the support of the latter, the building of the virtual network is independent on the physical network.

The researches on P2P systems mainly centralize on two sides [3]. One is how to build a network topology with scalability and robustness, while the other is how to present more effective and less cost searching algorithms. Although search algorithms are independent on the topology, a good topology will greatly improve the performance of the search algorithms. At present, the reseaches on the P2P topology mainly focus on applying the graph theory and complicated network theory (e.g. the small-world characteristic [4], nodes group based on interests [5] or hierarchical structures with super-node and so on) to the P2P.

In order to secure both the efficiency and robustness of P2P network, this paper proposes a Super-node Overlay Based on Information Exchange (SOBIE). The main contributions of SOBIE are following: 1) Any node in each AS has a value called score. The score integrates the parameters including distance, delay, querying similarity, information exchanging frequency and time. The system selects the super-nodes based on the score value. 2) It sets a corresponding mechanism to identify and prevent the free-riders. Meanwhile, it guarantees the matching between the physical network and logical network by partitioning the Autonomous System (AS) based on topology and geography. Moreover, the SOBIE has small-world characteristic.

The rest of the paper is organized as follows. Section II deals with the related works. We present the overlay structure with different issues in section III. In section IV we show the performance analysis and simulation results of the SOBIE topology. Finally, conclusions are made in section V.

II. RELATED WORKS

With the applications of P2P systems, how to build a reasonable topology is more and more important. There are many different P2P overlays including structured and unstructured [6].

Structured P2P overlays implement a distributed hash table (DHT) providing efficient mechanisms for storing and retrieving a data item based on an exact key. They have more scalability. However, they are unsuitable when the search key is similar, but not identical, to the key used

* Deng Li is the corresponding author. Email: dongxuanlee@gmail.com

to store the data item [7]. Moreover, in a decentralized structured network, the location of a peer in the overlay network is determined by the key space for which it is responsible. Thus It lacks adaptability and fault tolerance [8]. Most DHT-like structured P2P overlays do not consider the heterogeneity and the mismatching between physical and logical locations. Especially, if user machines are highly unreliable or join/leave the system frequently (i.e. churn), such a structured approach may introduce more management overheads, which will make the overlay be much fragile.

Unstructured P2P networks whose capability of adapting their behaviors dynamically can meet the changing specific needs of individual users in P2P system; and in addition will dramatically decrease the complexity and associated costs currently involved in the effective and reliable deployment of networks and communication services. Existing approaches to P2P applications can be divided into two general classes [9]: tree-based and mesh-based. Mesh-based P2P systems (e.g. Gnutella 0.4 [10]) typically use blind search methods like flooding that is very robust, flexible and easily supports partial-match and keyword queries. However, the large volume of query traffic generated by the flooding of messages limits the scalability and efficiency of this approach. Due to the single-parent nature, though there are many researches [11] hoping to improve the performance, tree-based overlay [12] is unbalanced and vulnerable to high "churn" rates of peers, since the departure of one peer affects all of its children.

In order to effectively solve above problems, P2P systems based on super-node are proposed by some researches [13]. The super-node based P2P systems such as KaZaA and eDonkey [14] which use a hierarchical approach to improve search efficiency. In hierarchical networks, nodes are partitioned into a set of logical groups (i.e. AS) by partitioning algorithms. Each AS selects a node which has more capability of performance such as CPU processing, memory, bandwidth and so on as the super-node. The other nodes are called ordinary-node. The super-node saves the information of the ordinary-nodes in the same AS and the super-node processes all requests instead of processing by the ordinary-nodes themselves. The searching messages are transmitted among the super-nodes which build a high speed transmitting layer. The P2P systems based on the super-node sufficiently consider the heterogeneity of networks, overcome the flooding searching problem as well as have the effectivity of centralized searching. So they improve the performance of P2P network and are broadly used in P2P networks. However there are also some problems including unreasonable super-node selection, single node failure, unmanching between physical and logical topology, and free-riding in the P2P systems based on the super-node.

The most typical applications of super-node P2P system are the Gnutella6.0 [15] and the KaZaA [14]. Gnutella6.0 presents two hierarchical structure including ultrapeer which is a super-node, and leaf-peer which is an ordinary-node. Gnutella6.0 proposes some standards for the selection of the ultrapeer including that the ulgrapeer can not be a firewall, and should be a suitable operating system with enough bandwidth and efficient update time. KaZaA classified nodes into super-nodes and ordinary-nodes, and that is similar with the Gnutella6.0. Nodes which have better performances in connection, bandwidth and CPU processing are selected as super-nodes. There are other super-node selection methods. For example, super-nodes are selected randomly, it is simple but fails to process the heterogeneous of the nodes in physical capability and content similarity; the super-node selections based on flooding and random walk [16] are not suitable to the dynamic and scalability in P2P networks, which will lead great communication costs with the growing of networks' scales. Reference [17] proposes a dispersed self-managing super-node selection method based on the gossiping protocol, of which the heuristic and self-organizational gradient topology structure is the centric. Reference [18] selects the super-node based on the capability such as the nodes which have wider bandwidth, better computing capability,longer online time and hold the more load in the system.

These current researches on the super-node selections are mainly some simple methods only considering nodes' static capabilities. But the dynamic of the node is an important characteristic in the P2P system. Moreover, most of above researches do not consider the free-riding problem which affects the efficiency and scalability of P2P systems very much when building the overlay topology structure. The free-riding nodes (i.e. freerider) are the nodes which selfishly consuming others' resources but not sharing own resources [19]. It is indicated that 70% nodes are free-ridding and 50% responses are returned by only 1% nodes in Gnutella.

Aiming at the above described problems in current researches, this paper proposes the SOBIE which greatly considers the dynamical capability such as the information exchange time, the information exchange frequency and the query similarity as well as the distance among peers in the P2P network to make the super-node selection to be more reasonable. The SOBIE also considers the accordance between the physical topology and logical topology. At the same time the freeriders are identified and forcely quitted the network when forming the SOBIE topology. The simulation results show that the SOBIE topology improves the querying success rate, decreases the query hops and the query messages, has better robustness including the more stable coverage rate and system connectivity so as to improve the performance of P2P networks comparing with the random topology [20] and the standard super-node topology [21].

III. FORMING THE SOBIE

A. Autonomous System Partition

All nodes in the P2P network based-on super-node group into different ASs by some standards. That is given n peers to build k ASs, which should satisfy the follow conditions: 1) each AS has at least one node; 2) each node must only belong to one AS.

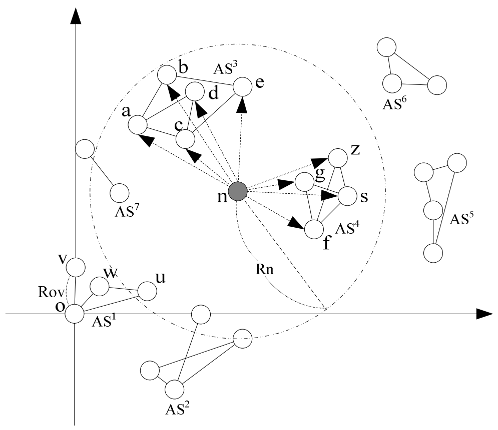


Figure 1. Autonomous system partition

Reference [22] proposes a normal standard for a good AS partition. The standard is that nodes in the same AS should be near or content correlated to each other as much as possible. But peer maybe always search satisfying resources across AS frequently. So the cost is great especially greater when the distance between the ASs is far. Some researches group an AS based on the semantic and content similarity [23]. But the topology mismatching is the key problem to reduce the searching effectiveness for the P2P network. Since the P2P network is a heterogeneous and loose couple structure without centralized controller, nodes being adjacent in logical network may be far in physical network. As a result, the information exchanging and resource locating in same AS even suffer great network delay, at the same time, searching time across ASs is more.

To solve above problems, the paper groups the AS according to the geography. The nodes being nearby to each other are grouped in the same AS, and the near ASs in geography are adjacent. Each AS selects its own super-node. This method guarantees the distance between the super-node and the ordinary-nodes to be close and avoids ordinary-nodes to select the far super-node to affect the querying delay. The distance is just a important factor to affect the Quality of Service (QoS).

We assume that the nodes are connected into a same virtual space by the Internet. The key of our partition of ASs is to find the coordinate of each node on the virtual space. It is assumed that every node in the network has a delay threshold. When the node n selects an AS to join, the AS must satisfy a requirement that the delay between the node n and all other nodes in the AS should be less than the delay threshold. The delay threshold can be mapped to distance threshold, for example, 1 minute may equal to one distance unit. So the delay measured between the nodes i and j ($Delay_{ij}$) can be mapped to distance ($Dist_{ij}$) with the same ratio. In this way the coordinate of nodes is determined by the delay among the nodes, for example, the nearer nodes have lower delay than farther nodes. Thus, each node tries to connect with the nearer nodes in order to reduce the delay. This partition reflects a general hypothesis that the delay between the nearer nodes in physical is less. Fig.1 is an example to describe the partition process of ASs in detail.

The first node in the network is always the origin (0, 0). In Fig.1, node o firstly joins the system, therefore the position of o is (0, 0). The next node is v and it chooses to be at (0, R_{ov}). In fact, node v can be at any location on the circumference of the circle whose center is (0, 0) and radius is R_{ov}. The third node w can calculate its coordinate (x, y) by mapping the $Delay_{wo}$ and $Delay_{wv}$ to $Dist_{wo}$ and $Dist_{wv}$. The new node calculates its coordinate by the distance, which is mapped from delay between the new node and the randomly selected three nodes in the virtual space. When the nodes in the virtual space are less than three, the new node calculates its coordinate by the delay between the new node and all nodes in the virtual space. The nodes group the AS by distance after calculating the coordinate. As shown in Fig.1, there are already seven ASs in the virtual space. The nodes in the same AS are connected by the solid line. The node n indicated with gray solid dot firstly calculates its coordinate （x_n,y_n） by the random selected three nodes. Then it found that there are two ASs (AS^3 and AS^4) in which all nodes are within in its radius R_n. So the node n has two candidate ASs to join. We define the $Dist_{max}$ to be the distance between the node n and the farthest node in the candidate AS. Computing all distances between the node n and all nodes in the AS^3 and AS^4.The distances between the node n and the farthest node in the AS^3 and AS^4 are respectively $Dist^3_{\max} = \max(dist_{na}, dist_{nb}, dist_{nc}, dist_{nd}, dist_{ne})$ and $Dist^4_{\max} = \max(dist_{ng}, dist_{nf}, dist_{ns}, dist_{nz})$ .The node n decides to select the AS^4 as the final candidate AS because the $Dist^4_{\max}$ is less than the $Dist^3_{\max}$. If the delay between the node n and all nodes in the AS^4 (i.e. g, f, s, z) is less than the delay threshold (i.e. $Delay^n_{threshold}$) of the node n, the node n joins the AS^4 , else it forms a new AS.

B. Information Exchange

When the information exchange begins between peer P_i and P_j, it is assumed that the query is sent from P_i to P_j. Then $V(P_i, P_j)$ in (1) denotes the P_i's evaluation to P_j about the completion of the query.

$$V(P_i,P_j) = \sum_{\forall q \text{ answered by } P_j} Qsim(q_j,q)^{\alpha} * F(P_i,P_j) / T(P_i,P_j) \qquad (1)$$

In order to find the most likely peers to answer a given query we need a function $Qsim$ to compute the similarity between different queries. $F(P_i,P_j)$ indicates the communication times between peer P_i and P_j. $T(P_i,P_j)$ denotes the time of information exchange between peer P_i and P_j. When the similarity between the different queries is the same, we can assume that the peer has more communication times in unit information exchange rather than in time joining the AS in order to decrease the percentage of the free-ridding nodes and prevent to mistake the nodes which just now join the AS and actively share resources as free-riding nodes because their whole information exchange times are less within the short time. Parameter α improves the power of similarity of queries. Since α is bigger, queries which are more satisfied are given a higher evaluation. That is, the more similar queries P_j complete, the larger evaluation P_i gives.

The cosine similarity in (2) between 2 vectors ($\vec{q}$ and $\vec{q}_i$) has been used extensively in information searches, and we use this function. In the cosine similarity model, the similarity *Qsim* of the two queries is simply the cosine of the angle between the two vectors.

$$Qsim(q,q_i)=\cos(q,q_i)=\frac{\sum(\vec{q}*\vec{q}_i)}{\sqrt{\sum(\vec{q})^2}*\sqrt{\sum(\vec{q}_i)^2}} \quad (2)$$

Thus, the score value of node P_j is:

$$Score(P_j)=\sum_{\forall i\neq j\in\{same\ AS\}}(V(P_i,P_j)*ID(P_i,P_j))\Big/\sum_{\forall i\neq j\in\{same\ AS\}}ID(P_i,P_j) \quad (3)$$

Where $ID(P_i,P_j)$ is the inverse proportion function of the distance between peer P_i and P_j. The closer peers lie in geography, the bigger the function *ID* is. It improves the power of peers adjacent geographically in order to avoid influence of network congestion so as to decrease the delay. The nodes in the AS which exchange with others have score value. The more frequency the nodes exchange with other and the higher querying similarity the nodes provide, the high evaluation the nodes are obtained. If the evaluations of the nodes are equal, the nearer nodes in physical have the higher score, that is $Score(p_j)\propto V(p_i,p_j)/D(p_i,p_j)$.

C. The description of SOBIE Building

In the super-node based P2P system, unreasonable super-node selection will reduce the performance of the system so as to make the super-node selection be meaningless. Therefore, super-node selection must base on some strict standards and processes. The dispersion, dynamic and complexity of the P2P network make the super-node selection to have more challenge.

We propose a super-node selection method based on the information exchange in the SOBIE topology. The nodes sort by their scores in one AS. The node whose score is the highest is selected as the super-node, and two nodes which have the second and the third score are the backup nodes of the super-node. The one reason to use two backup nodes is avoiding the single node failure problem. The other will be given in the next section. The query request is forwarded among the super-nodes. The source super-node always tries to connect with the highest score super-node to transmit the query message. By the score, the system selects the super-nodes and identifies the free-riders. In order to complete tasks or forward searching requests, nodes will try to connect with those higher-score nodes.

Each super-node maintains a triple index $<FileID, PeerID, Score(P_i)>$ to save the information of the ordinary-nodes in the same AS. Where the *FileID* is the ID of sharing file provided by the ordinary-node P_i, the *PeerID* is the ID of the ordinary-node P_i and *Score(P_i)* is the score value of the P_i. Some parameters are defined as follows:

- $AS^i=(\omega^i_{\min},\omega^i_{\max},\omega^i_{trigger},Score^i_{\min},t^i_{treshold})$ is the attribute set which satisfies $1\le\omega^i_{\min}\le\omega^i_{trigger}\le\omega^i_{\max}$. Where the $\omega^i_{\min}$ denotes the least number of the nodes in the AS^i; the $\omega^i_{\max}$ denotes the maximal capability (i.e. the most permitted node number) of the AS^i; the $\omega^i_{trigger}$ denotes the node number which is regulated by the AS^i to trigger the mechanism of free-riding process; the $Score^i_{\min}$ denotes the least score threshold of the nodes in the AS^i; the $t^i_{treshold}$ is the online time threshold of the node. When the node's online time has reached $t^i_{treshold}$ while its score is still less than $Score^i_{\min}$, the node can be regarded as a free-rider which should be dropped out of the AS^i.
- $Nr(i)$: The set of nodes applying to join the AS^i in the network.
- Nr_j: The node *j* applies to join the AS.
- $Nb(i)$: The set of the nodes in the AS^i.
- $Score^i_j$: The score of the node *j* in the AS^i.
- $Nb(i)_{\max}=\{j\mid j\in Nb(i),\forall k\in Nb(i),k\neq j,Score^i_j>Score^i_k\}$: The node has the most score in the AS^i.
- $Nb(i)_l=Nb(i)-Nb(i)_h$: The set of nodes whose scores are less than the $Score^i_{\min}$.
- $Nb(i)_{\min}=\{j\mid j\in Nb(i),\forall k\in Nb(i),k\neq j,Score^i_j<Score^i_k\}$: The node has the least score in the AS^i.

The key pseudocode of the SOBIE building is shown in the Fig. 2.

```
Initialization: ω^i_min, ω^i_max, ω^i_trigger, t^i_treshold,
  For ∀P_j ∈ Nb(i) in AS^i do
    Calculate Score(P_j) per t^i_treshold
  End for
SN ← ⟨ select Nb(i)_max from Nb(i) ⟩
while ( ω^i_max ≥| Nb(i) |> ω^i_trigger && Nb(i)_l ≠ ∅ )
  delete Pj( P_j ∈ Nb(i)_l , j = 1,2,...,| Nb(i)_l | ) from AS^i
If ∃Nr_j ∈ Nr(i) : Dist^i_max < Dist^j_threshold
  If | Nb(i) | +1 ≤ ω^i_max
     Nb(i) ← Nb(i) ∪ {Nr_j}
  Else if | Nb(i) | +1 > ω^i_max
     Nb(i) ← Nb(i) − {Nb(i)_min}
     Nb(i)_min becomes a Nr_k
     Nb(i) ← Nb(i) ∪ {Nr_j}
  End if
End if
Variables: | Nb(i) |, | Nb(i)_l |, Score^i_min
Fixed parameters: ω^i_min, ω^i_max, ω^i_trigger, t^i_treshold
```

Figure 2. The pseudocode of SOBIE building

D. The Structure of SOBIE

The structure of the SOBIE is shown in Fig. 3. There are three ICs respectively called IC_{local}, IC_{level} and ICl_{ayer} in each AS, which are shown in the Fig. 3 with the solid square, the blank dot and the blank square respectively, and the three nodes are connected to each other. Other nodes in the AS are called Ordinary-Nodes (ON) and their applications and resources are abstract as the information. The node with the highest value of score in an AS is as the IC_{local} in this AS. Then, the nodes having the ordinal highest values of score are named as IC_{level} and IC_{layer}. ONs do not have to keep information tables of each other, but only information about the IP and ID of the IC_{local}, IC_{level} and IC_{layer}, and only report to the IC_{local} rather than report to the others.

The ICs have different functions but contain the same backup. The ON_i^j denotes the j^{th} ON and IC_σ^i denotes the ICσ ($\sigma \in \{local, level, layer\}$) in the AS^i.

IC_{local}: IC_{local} is connected by and receives information such as ID, IP, applications, resources, activity histories (i.e. the values of score) and status from ONs. For example, ON_i directly connects to and stores their information in IC_{local}. IC_{local} also keeps the backup of information and sends its own information (i.e. information of ONs) to local IC_{level} and local IC_{layer}.

IC_{level}: One IC_{level} in the i^{th} level communicates with at most *d* IC_{level}s at the *(i–1)*th level. Each IC_{level} is connected with only one IC_{level} in the upper neighbour level. A local IC_{level} submits the application and resource information of its AS (i.e. information from local ONs aggregated in local IC_{local} are backed up to IC_{level}) to the corresponding IC_{level} in the upper neighbour level. To recover from all ICs' crash, IC_{level} also stores upper neighbour level NNs' IP addresses.

IC_{layer}: One IC_{layer} in the i^{th} layer communicates with at most *d* IC_{layer}s in the *(i–1)*th layer while the AS which lies in the highest level has one extra subordinate from the highest level AS in the lower neighbour layer. Each IC_{layer} is connected with only one IC_{layer} in the upper neighbour layer. Because of characteristics of our overlay, IC_{layer} in the upper layer only backs up the ID and IP information and score value of the nodes of its lower neighbour layer ASs.

Every layer includes many levels and ASs

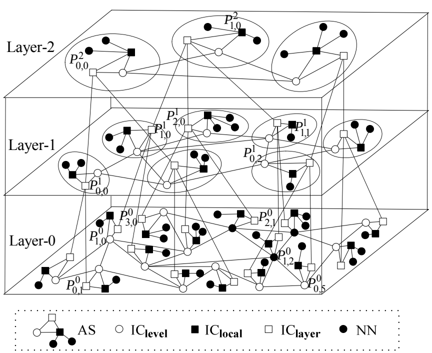


Figure 3. SOBIE topology structure

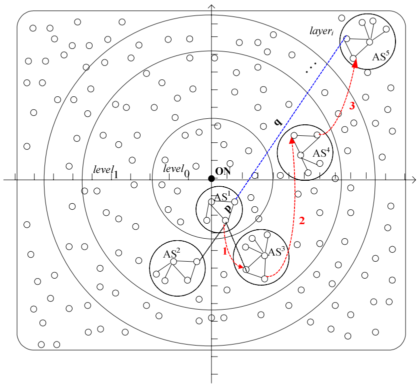


Figure 4. An example of one layer-structure

communicating to each other by ICs appearing in all ASs, levels and layers. Fig. 4 shows an example of one layer-structure, the edge *q* is one of the *long-range contacts*, and the edge *p* in AS is one of the *local contacts*. The *long-range contacts* and *local contacts* are presented in [24] to build small world network. Informally, a small world network can be viewed as a connected graph in which two randomly chosen nodes are connected by just about six degrees of separation. In other words, the average shortest distance between two randomly chosen nodes is approximately six hops. This property implies that one can locate information stored at any random node of a small world network by only a small number of link traversals. One important property of a small world network is the low average hop count between two randomly chosen nodes [25].

The average hop count is low in the SOBIE. For example, the AS^1 wants to visit the AS^5 such as the Fig .4 showing. In traditional, a peer in the AS^1 should firstly visit the node in the nearest neighbor AS such as AS^3. Then the peer in the AS^3 will visit the relative nearer AS (e.g. AS^4). Finally, the peer in the AS^4 visits the AS^5. So the AS^1 contacts the AS^5 by 3 hops. While in the SOBIE, AS^1 contacts the AS^5 by the *long-range contacts p* only one hop. So it will decrease the query hops when searching resource. The following simulations also validate this.

IV. Simulations and Evaluations

A. The Parameters of Simulation

We conducted our experiments on three P2P network topologies including the random topology, the standard super-node topology and the SOBIE topology. We apply the flooding search scheme to each topology to compare the performance of different P2P overlay topology.

Random topology: It is a random graph with different scales. The node degrees follow a power-law distribution: if one ranks all nodes from the most connected to the least connected, then the i^{th} most connected node has ω / i^{α} neighbors, where ω is a constant. Many real-life P2P networks including real Internet network have topologies that are power-law random graphs [26].

TABLE I.
DESCRIPTION AND VALUES OF PARAMETERS IN SIMULATIONS

parameter	parameter details		Parameter value
Network	Total Peers Peers are free-riders		500/2000(including good peers and free-riders) 25% original
Peer	Good peer	The condition of peer responding to queries The condition of peer providing file-sharing	If there is a match If there is file
	Free-rider	Does peer respond to queries Does peer provide to file-sharing	Never or respond once in a while Never or providng few of its own
AS^i	$(\omega^i_{\min}, \omega^i_{\max}, \omega^i_{trigger}, Score^i_{\min}, t^i_{treshold})$		(1,100, 25,10% of the SN's Score,60s)
File distribution	The number q of queries for the file whose popular rate is r Number of files The categories of files The number f of file whose popular rate is r		Zipf distribution($q(r) = C * r^{-\theta}$, $\theta = 0.726$) 4162 files including replica（m=300 diferent files） 10(identify the 10 categories as 1,2,…,10.The number one is the most popular file, the number tow popular is less than the number one's ;and the replica of number one is more than others replica.) Zipf distribution ($f(r) = C * r^{-\theta}, \theta = 0.726$)
Simulation	Experiments over which results are averaged		10

Standard super-node topology [21]: It is a two-level hierarchy, consisting of a first level of interconnected peers called super-peers and a second level of so-called leaf nodes or ordinary peers, which are only connected to a single super-peer. In super-node topology, searches are flooded among super-peers. In the paper, the term "node" is used interchangeably with "peer". Each super-node has two backup nodes. The peers including super-node, its backups and normal peers form a cluster. We set the total number (c_{size}) in any cluster is $c_{size} \in [5, 15]$.

Though there are many other unstructured searching algorithms, such as Iterative deepening [27], Biased high degree [28], Most results and Fewest result hops [16] etc., the flooding search method is very robust, flexible and easily supports partial-match and keyword queries. Considering all topologies being compared in our paper, it is rational that we just use flooding search algorithms to evaluate the performance of topologies.

Flooding algorithm: When a peer receives a search message, it both processes the message and forwards it to all of its neighbors in the overlay network. Each message is given a time-to-live value *ttl*, and search messages get flooded to every node within *ttl* hops of the source.

Studies have shown that Gnutella, Media and Web queries tend to follow Zipf-like distributions [29]. Thus, in our simulations, the number of each file follows a Zipf-like distribution according to its popular rate. It is assumed that there are *m* original files. And if a file's popular rate is r, then we can get $f_r \propto 1/r^{\theta}$, where f_r is the number of the file. Each file is replicated on nodes based on its popular rate. The more popular the file is, the more the file's number is. The query probability of the file also follows Zipf-like distribution according to its popular rate. The more popular the file is, the more the query probability of the file is.

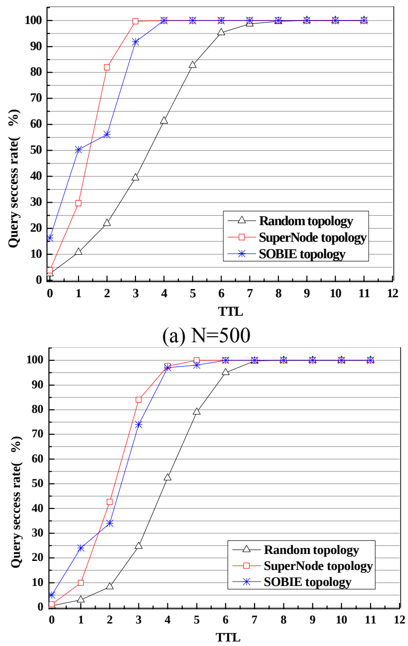
We conducted our experiments on the three topologies with different node-number (i.e. N=500 and 2000) generated by using the GT-TTM library [30]. Files including the replica are all 4162 and are distributed to all nodes in the network. The experiments randomly select the node every time as the source of one query. We set the TTL (Time To Live) to be from 0 to 11. Detailed parameters are listed in table I.

The term network validity is used to describe how the node responses to and satisfies the query requests. We evaluate the network validity based on several factors including query success rate, query hops, query message number as well as the robustness.

B. Query Success Rate

We evaluate the query success rate of three topologies with different system scales by the flooding algorithm.

Fig.5 shows the query success rate while TTL changes from 0 to 11 in different topologies. The *x*-axis is allowable the most hops to query and the *y*-axis is the query success rate in the limited hops. It is shown that the query success rate of SOBIE is always the highest in the



(a) N=500

(b) N=2000

Figure 5. Query success rate vs TTL

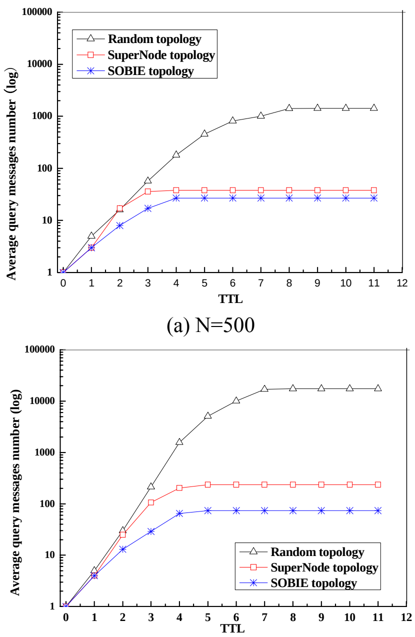
two scale systems when the TTL is less than 2. Because that the SOBIE adopts three ICs which backup information of the level and layer nodes. When an IC initiates a query request, it queries the resources in its own AS or the neighbor ASs. If the TTL is less, it is very possible that the query can be satisfied in near ASs so that the success rate is possibly high. The experiment results proof this. It is still shown in Fig. 5 that the supper-node topology and the SOBIE topology nearly get up to a steady rate (100%) when the TTL is 4, but the random topology doesn't reach the rate (100%) until the TTL being 7.

C. Query Messages and Hops

The number of query messages is one factor of the evaluation of searching efficiency. The repeated messages across one node are less, the cost should be less. We compute the number of query messages in the random topology, the standard super-node topology and the SOBIE topology when applying the flooding search scheme under two system scales.

Fig. 6 shows the number of query messages generated by a same query request in different topology. The *x*-axis is the same as that in Fig. 5. The *y*-axis is the average query messages and is the logarithmic scale.

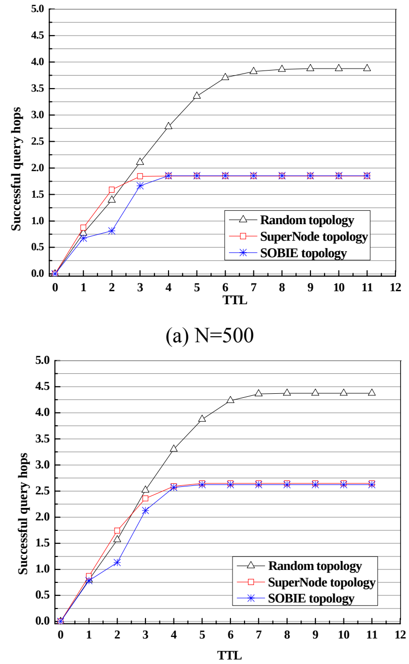
It is shown that the number of query messages in the random topology is the most, while it is the least in the SOBIE topology. In Fig. 5, the query success rate of the super-node topology is higher than the SOBIE when the TTL is between the 2 and 5. However, as shown in Fig. 6, the cost of the higher success rate is to generate more messages in the super-node topology. Therefore, we prefer to select the SOBIE which generates less query messages for searching files. In this way, SOBIE avoids to generate more messages to use more bandwidth so as to reduce the network load.



(a) N=500

(b) N=2000

Figure 6. Query message number vs TTL



(a) N=500

(b) N=2000

Figure 7. Successful query hops vs TTL

Query messages are more in the larger scale network such as Fig. 6(a) and Fig. 6(b) showing. But the trend of the change is similar. The number of query messages suddenly increases when the TTL is equal to 4 in the random topology. While that of the SOBIE gradually increases within the TTL changing from 0 to 4. The numbers of query messages in the SOBIE and super-node topology don't increase after the TTL is 4.

Fig. 7 shows the average query hops when the query is satisfied in different topologies. The larger the number of hops is, the longer the query path is. Therefore, the query delay increases and the long path will cost more network resource too.

Comparing Fig. 7(a) and Fig. 7(b), it is clear that the number of query hops increases along with the increasing of network scale. The random topology's number of query hops gets to a steady number when the TTL is 7. In the SOBIE, the number of query hops doesn't increase when the TTL is 4. Fig. 7 also shows that the number of query hops in the SOBIE is the least, while they are the most in the random topology. Therefore, we conclude that the query delay in the SOBIE is the least too. The simulation result accords with the theoretical analysis in the section III.D.

D. Robustness

In this section, we concentrate on discussing the performance of every topology when there are peers to leave system randomly. The original scale of the system is 2000.

Fig. 8 shows the change of the query success rate in different topologies, where the *x*-axis is the ratio of the number of randomly leaving nodes to all nodes' number in the system.

It can be seen from the Fig. 8 that the query success rate in SOBIE is still very high when there are peers leaving. The query success rate is still more than 90%

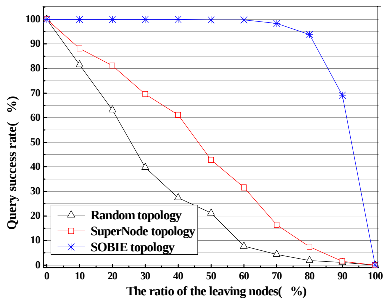


Figure 8. Query success rate

when there are 70% nodes leaving the system. While the query success rate in the random topology and super-node topology fast decreases along with the nodes leaving. The query success rate decreases to 50% below when there are less than 50% nodes leaving in the random topology and the standard super-node topology.

For efficiently evaluating the robustness of SOBIE, we define two performance parameters. They are the most coverage rate of nodes and the connectivity rate of the system.

It is defined that i is the query initiated by the i^{th} node ($i \in (N - N * rate_{lose})$), $rate_{lose}$ is the failure rate of the node, N is the system scale, n_i is the number of the nodes visited by the i^{th} query. In the system which allows adequate query hops (i.e. the query can visit all nodes connected with the source node directly or not), we define the most coverage rate of nodes is (4).

$$CoverRate = Max(n_i)/(N - N \cdot rate_{lose}) \qquad (4)$$

While the connectivity rate of the system is:

$$\text{ConnRate}=\sum_{i=1}^{N-N\cdot rate_{lose}} n_i \Big/ (N - N \cdot rate_{lose}) \qquad (5)$$

Fig. 9 and Fig. 10 show the changing of the most coverage rate and the connectivity rate of the system along with the leaving node rate increasing respectively.

The coverage rate and the system connectivity rate in SOBIE topology are still 100% even when there are 80% nodes leaving the system. The most coverage rate and the system connectivity rate have remarkable decrease after 90% nodes leave the system.

It is clear that the SOBIE topology has more robust than the other two. Because the system still has high query success rate, coverage rate and connectivity rate when there are some nodes leaving. These are the most important characteristics of the SOBIE. The experimental

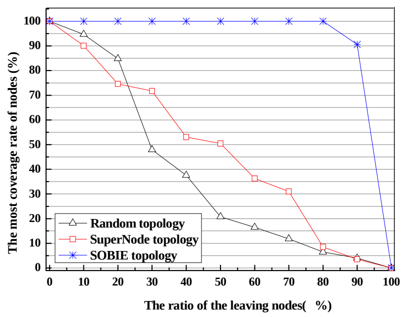


Figure 9. The most coverage rate

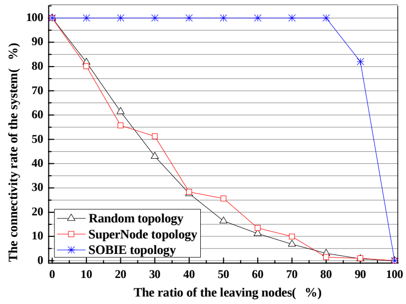


Figure 10. Connectivity rate

results are consistent with the conclusion in theory. The ICs back up and communicate to each other by the *local contacts* ASs and the *long-range contacts* ASs. The different function of ICs and the un-randomly chosen contacts make the SOBIE to be more robust and scalable.

V. CONCLUSIONS

The loose topology and the distribution of P2P systems result in the lower communication efficiency, limited throughout and scalability. In the unstructured P2P network, a reasonable topology can improve the searching efficiency, reduce the cost and utilize each peer effectively. This paper proposes a robust super-node based P2P overlay considering the information exchange, autonomous system partition and the free-ridding. The SOBIE improves the performance of the P2P network because of its scalability suitable for the heterogeneous network. The simulation results indicate that the SOBIE improved the query success rate, reduced the query hops and the number of query messages, and obtained the better robustness including the more stable coverage rate and system connectivity comparing with the random topology and the standard super-node topology.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China under Grant No. 60873012 and the Pre-Research Foundation of General Armament Department under Grant No. 9140A15030308QT4801.

REFERENCES

[1] S. Androutsellis-Theotokis and D. Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies”, *ACM Computing Surveys,* vol. 36(4), pp. 66-69,2004.
[2] D.A. Menasce and L. Kanchanapalli, “Probabilistic Scalable P2P Resource Location Services”,*ACM SIGMETRICS Performance Evaluation Rev*, vol. 30(6), pp. 48-58, 2002.
[3] X.B. Zhou, J. Zhou, H.C. Hu, “A layered interest based topology organizing model for unstructured P2P”, *Journal of Software,* vol. 18(12), pp. 3131-3138,2007.
[4] J. Zhou, H.M. Lu, Y.D. Li, “Using small-world to devise routing algorithm for unstructured peer-to-peer system”, *Journal of Software,* vol. 15(6), pp. 915-923,2004.
[5] H. Kobayashi, H. Takizawa, T. Inaba, et.al, “A self-organizing overlay network to exploit the locality of interests for effective resource discovery in P2P systems”, *Proceedings of the 2005 Symposium on Applications and*

the Internet Workshops(SAINT2005), Washington, USA, pp. 246-255, 2005.

[6] H.H. Wang, Y.W. Zhu and Y.M. Hu, "To Unify Structured and Unstructured P2P Systems", *Proc. Of the 19th IEEE Inter. Parallel and Distributed Processing Symposium,* Denver, USA, pp. 104-108, 2005.

[7] B. Wong, Y. Vigfusson, E.G. Sirer, "Hyperspaces for object clustering and approximate matching in peer-to-peer overlays", *Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, San Diego, Canada, pp. 1-6, May 2007.

[8] A. Singh, M. Haahr, "Decentralized Clustering In Pure P2P Overlay Networks Using Schelling's Model", *in ICC'2007,* pp. 1860-1866, 2007.

[9] N. Magharei, R. Rejaie, G. Yang, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches", *in INFOCOM'2007,* pp. 1424-1432, 2007.

[10] Gnutella Protocol Specification, version 0.4[EB/OL]. http://www.clip2.com/GnutellaProtocol04.pdf, 2001.

[11] D.A. Tran, K.A. Hua, and T.T. Do, "A peer-to-peer architecture for media streaming", *Journal on Selected Areas in Communications*, vol. 22(1), pp. 1-14, January 2004.

[12] T. Small, B. Li, B. Liang, "Outreach: peer-to-peer topology construction towards minimized server bandwidth costs", *IEEE Journal on Selected Areas in Communications,* Vol. 25, pp. 35-45, 2007.

[13] B.Yang, H.Garcia-Molina, "Designing a super-peer network", *International Conf on Data Engineering (ICDE' 03),* Bangalore, India, pp. 49-60, 2003.

[14] G.H. Chen, Z.H. Li, "Peer-to-Peer Network: Structure, Application and Design", Bei jing, QingHua university publishing company, pp. 53-58, September 2007.

[15] T. Klingberg. Gnutella Protocol Development[EB/OL]. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html, 2002.

[16] B.Yang, H.Garcia-Molina, "Improving search in peer-to-peer networks", *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pp. 5-14, July 2002.

[17] S.Jan, D.Jim, C.Raymond, "Using aggregation for adaptive super-peer discovery on the gradient topology", *In SelfMan, ser. Lecture Notes in Computer Science*, A. K. and Jean-Philippe Martin-Flatin, Ed. vol. 3996, Springer 2006.

[18] Z. Xu and Y. Hu, "SBARC: A Supernode Based Peer-to-Peer File Sharing System",*in ISCC*, Kemer-Antalya, Turkey, pp. 1053-1058, June 2003.

[19] M.Feldman,C.Papadimitriou, J.Chuang, "Free-Riding and Whitewashing in Peer-to-Peer Systems", *IEEE journal on selected areas in communications*, vol. 24(5), pp. 1010-1019,May 2006.

[20] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker., "Search and Replication in Unstructured Peer-to-Peer Networks", *Proc. of 16th ACM International Conference on Supercomputing (ICS'02),* New York, USA, pp. 84-95, 2002.

[21] Brian F. Cooper, "An optimal overlay topology for routing peer-to-peer searches", *In: Proc. of the 6th Int'l Middleware Conf.,* Berlin, Heidelberg: Springer-Verlag, pp. 82-101, 2005.

[22] K.Mouna, Y.Kokou, Y.Ma, "HON-P2P: A Cluster-based Hybrid Overlay Network for Multimedia Object Management", *11th International Conference on Parallel and Distributed Systems*, pp. 578-584, July 2005.

[23] H.Chen, H.Jin, X.Ning, "SemreX:A P2P Overlay Based on Semantic Similarity", *Journal of Software*, vol. 17(5), pp. 1170-1181, 2006.

[24] J. Kleinberg,"The small-world phenomenon: an algorithmic perspective", *Cornell computer science technical report*, pp. 99-1776, 2000.

[25] Y.K. Hui, C.S. Lui and K.Y. Yau, "Small world overlay P2P networks", *Twelfth IEEE International Workshop on Quality of Service*, Delta Centre-Ville (downtown Montreal) Montreal, Canada, pp.201-210, June 2004.

[26] M. Jovanovic,"Modeling large-scale peer-to-peer networks and a case study of Gnutella", University of Cincinnati, Cincinnati, Ohio, USA, 2001.

[27] B. Yang and H. Garcia-Molina, "Efficient Search in Peer-to-Peer Networks", *in ICDCS*, Vienna, Austria, pp. 5-14, July 2002.

[28] L.A. Adamic, R.M. Lukose, A.R. Puniyani and B.A. Huberman, "Search in Power-law Networks", *Physical Review*, E 64, pp.46135-46143, 2001.

[29] K.A. Hua, C. Lee, C.M. Hua, "Dynamic load balancing in Multicomputer database systems using partition tuning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 7(6), pp. 968-983, 1995.

[30] E.W. Zegura, K.L. Calvert, S. Bhattacharjee, "How to model an Internetwork", *Proc. of the IEEE INFOCOM'96*, San Francisco, CA, USA, pp. 594-602, March 1996.

Zhigang Chen was born in 1964. He received the PhD degree in Computer Science from Central South University in 1998.

He is a Professor in School of Information Science and Engineering, Central South University, P.R.China. His current research focuses on Parallel and Distributed Computing, Trusted Computing.

Prof. Chen is an IEEE member.

Jiaqi Liu was born in 1983. She is a PHD candidate in School of Information Science and Engineering at Central South University, P.R.China. Her current research interest is Parallel and Distributed Computing.

Deng Li was born in 1978. He received his PhD degree in Computer Science from Central South University in 2008.

He is an Associate Professor in School of Information Science and Engineering at the Central South University, P.R.China. His current research interests include Autonomic Communications, P2P and Grid Computing.

Dr. Li is an IEEE member.

Hui Liu is an Associated Professor in the Missouri State University. Her research interests include Wireless Networks, Mobile Computing, Parallel Algorithms and Optical Networks.

Athanasios Vasilakos is a Professor at the Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece. His research interests include Telecommunications Networks, Game Theory, Networking, Mobile Computing and Wireless Networking, Internet Arts.

Prof. Vasilakos is an IEEE member.