A Epidemic Style Super-node Election Method Based on Self-information Theory

Zhiwei Gao*, yingxin Hu*

*Department of Computer Science, Shijiazhuang TieDiao University, Shijiazhuang, 050043, China

gao_zhiwei@163.com, huyinxin@163.com

Abstract—**Many distributed applications such as cloud computings, grids use peer-to-peer (P2P) paradigm as the lower service. In P2P technology, the super-node paradigm can lead to improved efficiency, without compromising the decentralized nature of P2P networks. So the above applications adopt super-node paradigm to provide services. However, due to inherent dynamism, decentralisation, scale and complexity of P2P environments, self-managing super-node selection is a challenging problem. This paper present a super-node election protocol based on self-information theory and gossiping technology(SPSI). In SPSI, every node has a information vector (VI), and SPSI uses a weighted mean mechanism based on VI to promote the "best" nodes to super-node status. As we know we are the first to use self-information theory to select super-node. The paper also includes extensive simulation experiments to prove the efficiency, scalability and robustness of SPSI.**

Keywords—**self-information quantity, super-node, scalability, SPSI**

I. Introduction

Many distributed applications such as cloud computing, grids use super-node paradigm as the lower service. Super-nodes allow these applications to run more efficiently by exploiting heterogeneity and distributing load to machines that can handle the burden. On the other hand, because this architecture allows multiple, separate points of failure, increasing the health of the distributed network, it does not inherit the flaws of the client/server model. The use of P2P protocols is expected to improve the efficiency and scalability of information services in these systems [1],[4],[5].

However, due to inherent decentralisations, scale, dynamism, and complexity of P2P environments, self-managing super-node selection is a challenging problem.

A number of P2P systems address the heterogeneity of P2P environments by electing super-nodes and assigning them extra responsibilities [6],[7],[10],[11]. Solutions based on flooding, random walking or other traditional election algorithm, potentially require communication with all peers in the network and thus do not scale to large networks. Other solutions such as manual or static configuration of super-nodes are inappropriate due to a lack of global knowledge of application characteristics.

II. RELATED WORK

In this section, we briefly review some related work. We start with P2P based on super-node technology, and then present the related work on super-node selection problem.

The super-node approach to organize a P2P overlay is a trade-off solution that merges the client-server model relative simplicity and the P2P autonomy and resilience to crashes. The need for a super-node network is mainly motivated by the fact to overcome the heterogeneity of peers deployed on the Internet.

Meirong Liu[1] et al. present a super-peer-based coordinated service provision framework (SCSP) to coordinate the service groups to work collaboratively and share their service peers. The SCSP is made up of an S-labor-market model, a recruiting protocol based on a weighting mechanism, and an optimal dispatch algorithm.

KaZaA [8] and Gnutella [9], [10] have explored using heterogeneity of peers to improve search performance. These systems have efficient peers hold more neighbors and process more queries. An efficient peer (super-peer) acts as a server in a local area, builds an index of the shared files provided by those peers connected to it and offers a searching index service for those who have connected to it by flooding queries to other super-peers. These systems mainly explored improving performance by decreasing the number of transmitted messages and latency hops.

Yang and Garcia Molina [12] proposed some design guidelines. A mechanism to split node clusters is proposed and evaluated analytically, but no experimental results are presented.

Garces-Erice[13] et al. studied hierarchical DHTs, in which peers are organized into groups, and each group has its autonomous intra-group overlay network and lookup service. The groups themselves are organized in a top-level overlay network. To find a peer that is responsible for a key, the top level overlay first determines the group responsible for the key; the responsible group then uses its intra-group overlay to determine the specific peer that is responsible for the key. They concluded that hierarchical organization could improve a

Manuscript received May 21, 2012. This work was supported by the National Natural Science Fund of China (No: 50975185).

Z.W is a Associate Professor of Information Science and Technology, Shijiazhuang TieDao University. Gao's research interests center around peer-to-peer computing, PKI and certificate chain, PKI based on P2P technology. (phone: 086-0311-87939496; fax: 086-0311-8795288; e-mail: gao_zhiwei@163.com).

Y.X is a Associate Professor of Information Science and Technology, Shijiazhuang TieDao University. (Email: huyinxin@163.com)

system's scalability. A hierarchical system demonstrates better stability due to selection of peers who are more reliable as the members of the upper overlay, generates fewer messages in a wide area and can significantly improve the lookup performance by transmitting queries through the upper overlay.

Nejdl et al. proposed a design organizing super-peers with a hypercube structure in [14]. In their approach, every super-peer serves a subset of peers and all super-peers are arranged in a hypercube topology. Because the topology is vertex-symmetric, it features inherent load balancing among super-peers. When a super-peer wants to transmit a query, according to the spanning tree algorithm, it forwards the query to its neighbors instead of flooding the system with queries. Each super-peer wants to maintain at most d neighbors' information and it takes at most d logic hops for a query from any super-peer to the farthest super-peer, where d is the dimension of the hypercube.

Mizrak [15] et al. proposed a design based on the Chord ring. In their approach, there are two rings, named the inner-ring and outer-ring respectively. Each peer is placed on a circular identifier space in the "outer-ring", using a DHT algorithm such as Chord. Of all the peers, m peers who joined the system first are selected as super-peers to create a smaller core "innerring". The outer-ring is divided into m equal arcs and each arc is assigned to one super-peer. Each super-peer is responsible for maintaining two pieces of information: the addresses of the peers contained within its arc and the mapping between arcs and their responsible super-peers. Each peer registers in only one super-peer, and requests searching services from its super-node. Each super-peer offers searching services for its registered peers and the other super-peers. The lookup is performed using super-peers in constant time. When a super-peer's load approaches its capability, it may share part of its load with its neighbors if they have sufficient excess capacity or with a new super-peer selected from the volunteer peers. In either case the super-peer splits its arc appropriately and reassigns pieces of this range to the neighbors accepting the load.

In [16], the authors propose a socio-economic inspiration based on Shelling's model to create a variation of the super-node topology. Such variation allows ordinary peers to be connected with each other and to be clients of more than one super-node at the same time. This topology focuses on efficient search. As in our case, the super-nodes are connected to each other to form a network of hubs and both solutions are suited for unstructured networks. However, they do not address the problem of the super-node election.

In [18], a mechanism for the construction and the maintenance of overlay topologies based on super-nodes SG-1 was proposed. This mechanism is based on the well-known gossip paradigm, with nodes exchanging information with randomly selected peers and re-arranging the topology according to the requirements of the particular P2P application. In [19], the author presents SG-2, a protocol for building and maintaining proximity-aware super-node topologies. Like SG-1, SG-2 also uses a gossip-based protocol to spread messages to nearby nodes and a biology-inspired task allocation mechanism to promote the "best" nodes to super-node status.

Unlike all these studies, our implementation is based on information theory, and as we know we are the first to introduce information theory to super-node selection. Our contribution is as follows: (1) Propose a super-node election protocol SPSI based on self-information. (2) Give the relation between node's capacity and online time through experiments. (3) Propose a weighted mean algorithm to describe node's properties. Our model's efficiency is equal to SG-1 or SG-2, but the super-nodes we elected are more stable, so the costs of network maintenance are lower than them.

III.BACKGROUND THEORY AND TERMINOLOGY

The framework of SPSI can be considered as a natural evolution of Rigorous binary tree model. We propose a framework based on our own efficient and scalable Rigorous binary tree model and its theorem[6,17]. If the size of network is small, file lookups are resolved with only one hop. As the system's scale become larger, it can expand automatically based on the super-node's capability and suit for large scale system. In the worst case, file lookups are resolved with only three hops. But this model did not discuss super-node selection problem, and this is the main concern of this thesis.

Here, we first provide the definition of a rigorous binary tree and the other relevant theory to give readers a better understanding of our model.

A. Rigorous binary tree and its mapping theorem

Definition 1: Rigorous binary tree

For a random node of a binary tree, if it has at least one child node, its left child node and right child node must exist at the same time. If this condition is satisfied, the binary tree is defined as a rigorous binary tree.

Definition 2: Rigorous binary tree extension

After a random leaf of a rigorous binary tree produces two child nodes, the original rigorous binary tree becomes a new rigorous binary tree. This is called rigorous binary tree extension.

Definition 3: Rigorous binary tree code algorithm: The letter T represents a rigorous binary tree, "A" represents a random node in T, ha represents the depth of node A, and N_a represents its code. The code of T's root node was set as 0. The code of A's left child is equal to N_a. The code of A's right child is equal to (N_a + 2^{h_a}), The depth A's child is h_a+1.

Theorem 1: Rigorous binary tree mapping theorem: For any one integer I (I >=0), there is one and only one leaf node X whose code (N_x) and depth (h_{x}) can accord with N_x =I % 2^{h_x} , among all leaf nodes in a fixed rigorous binary tree. (Here % denotes modular arithmetic.) the proof process is included in our prior work[6], [17].

Figure.1 illustrates a rigorous binary tree. When we extend its leaf node G in Figure.1(1) by adding two children nodes to G, the rigorous binary tree becomes that described in Figure.1(2). According to the rigorous binary tree code algorithm, node A is the root node, so its code and depth are (0, 0). Node B is A's left child, so B's code is equal to A's code (0) and B’s depth is A’s depth plus one (0+1=1). Node C is A’s right child, so C’s code is equal to ($0+2^0=1$) and C’s depth is equal to B’s depth (1). In the same way, we can compute the remaining nodes’ code and depth as described in Figure. 1(3).

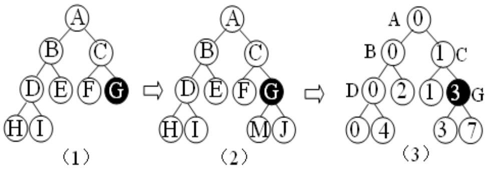


Figure 1. The extending and coding of a rigorous binary tree

Based on Rigorous binary tree extension and mapping theorem, we design RBTree model. In the model, when the number of peers registered in a super-peer reaches the quantity limit, in order to balance the load and avoid a bottleneck, the super-peer will use SPSI protocol to select a high-powered peer from its registered-peer table as a new super-peer, code it with the rigorous binary tree code algorithm and share one part of its load with the new super-peer.

B. Self-information

Definition 4: Let *E* be an event belonging to a given event space and having probability $\Pr(E) = p_E$, Let *I(E)* – called the self-information of *E* – represent the amount of information one gains when learning that *E* has occurred (or equivalently, the amount of uncertainty one had about *E* prior to learning that it has happened).

Theorem 2: The only function defined over $p \in [0, 1]$ and satisfying

- I(p) is monotonically decreasing in p;
- I(p) is a continuous function of p for $0 \leqslant p \leqslant 1$;
- I(p1 × p2) = I(p1) + I(p2);

where $\mathrm{I}(p) = -c \bullet \log_b(p)$, *c* is a positive constant and the base b of the logarithm is any number larger than one.

In SPSI, every node has a information vector (VI), and SPSI uses a weighted mean mechanism based on VI to promote the “best” nodes to super-node status.

C. Weighted Arithmetic Mean

In calculation of arithmetic mean, the importance of all the items was considered to be equal. However, there may be situations in which all the items under considerations are not equal importance. For example, we want to find average number of marks per subject who appeared in different subjects like Mathematics, Statistics, Physics and Biology. These subjects do not have equal importance. If we find arithmetic mean by giving Mean. For example, A student obtained 70, 80, 80, 70, and 65 marks in the subjects of Math, Statistics, Physics, Chemistry and Biology respectively. And we assume weights 5, 4, 2, 3, and 1 respectively for the above mentioned subjects. The solution was listed in table 1.

TABLE 1.

SOLUTION OF WEIGHTED ARITHMETIC MEAN

Subjects	**Marks Obtained**	**Weight(*w*)**	***wx***
Math	70	5	350
Statistics	80	4	320
Physics	80	2	160
Chemistry	70	3	210
Biology	65	1	65
Total		$\sum w = 15$	$\sum wx = 1105$

Defination 5: arithmetic mean computed by considering relative importance of each items is called weighted arithmetic mean. To give due importance to each item under consideration, we assign number called weight to each item in proportion to its relative importance. Weighted Arithmetic Mean is computed by using following formula:

$$\overline{X}_w = \frac{\sum wx}{\sum w}$$

Where:

$\overline{X}_w$: Stands for weighted arithmetic mean.

x : Stands for values of the items and

w : Stands for weight of the item.

IV. SYSTEM MODEL AND ALGORITHM

Generally speaking, our goal is to create a topology where the most powerful nodes (in terms of capacity) and the enough stable nodes are promoted to the role of super-nodes.

The main topology features of the SPSI protocol algorithm are that each client just connected to a super node, and super node of each other connected together by random. This protocol can find a smaller number of nodes and super nodes set which has a longer online time, and these super nodes can serve as client nodes to cover the rest nodes. Such a topology structure can be easily used to implement file sharing, also can reduce the traffic caused by the application program.

To build a topology with such characteristics, we propose a mechanism based on NEWCAST [19]. Topology information such as identifier, capacity, online time, current role and neighborhood of participating nodes are disseminated through periodic gossip messages between randomly selected nodes. Based on the received information, nodes update their neighborhoods in order to obtain a better approximation of the target topology.

In NEWSCAST, the state of a node is called partial view and it is constituted of a fixed-size set of peer descriptors. A peer descriptor contains the address of the node, along with a logical timestamp identifying the time when the descriptor was created. The size of a partial view is denoted by s. Generally, we chose the maximal view size to c = 30 and can get enough robust target topology[19].

A. *Node Capacity*

Apparently, Nodes are heterogenous: they may differ in their computational and storage capabilities, and also (and more importantly) on the bandwidth of its network connection. In order to distinguish nodes that are capable to act as super-nodes from nodes that can join just as clients, we associate each node n with a parameter Cn representing its capacity, i.e. the number of clients that can be handled by n. In other words, we use the concept of capacity to abstract in a single quantity all the characteristics listed above. In order to simplify our simulations, we assume that each node knows its capacity parameter; in a real implementation, this value could be computed on the fly, by performing on-line measurements; the result is strongly dependent on the particular application to be implemented. The techniques used to perform this computation are outside the scope of this paper.

In [20], through measurements done over existing P2P networks, the author concluded that most of the nodes have low capacity, while very few of them are able to support a large number of clients and the node's capacity obeys power-law distribution. So we have node *n* has a capacity of *x* with the probability $\mathrm{P}(C_n = x) = x^{-\alpha}$, In which $x \in [1, C_{\max}]$, α is the distribution parameters (usually the parameter α = 2). The node capacity does not necessarily follow a strict power distribution, but it provides a reasonable distribution close to that.

B. *Online time of nodes*

In any super nodes based peer-to-peer network, super nodes take charge of both data block index and overlay organization. When a super node logouts from the system or a super node decides to alleviate its load, it has to transfer the corresponding block index and child nodes to another super node. This process will bring about considerable communication cost, so it is necessary to select a highly stable peer to act as a super node.

Many research papers such as [2], [3] study the online time of nodes through measurement method. They observed that session times (in minutes) with a mean=266, standard deviation=671. Network nodes joining or leaving is considered a Poisson distribution, and online time of nodes is subject to the negative exponential distribution of λ. So its probability density function is:

$$\mathrm{f}(x) = \lambda e^{-\lambda x} \qquad x > 0 \qquad (1)$$

We use maximum likelihood estimation method to estimate the parameter λ, assuming that x1, x2, x3 ... is a set of random sample values, representing the node's Online time, Then Likelihood function

$$\mathrm{L}(\lambda) = \lambda^n \prod_{i=1}^{n} \exp(-\lambda x_i) = \lambda^n \exp(-\lambda \sum_{i=1}^{n} x_i) \qquad (2)$$

then $$\frac{\mathrm{d} \ln L(\lambda)}{\mathrm{d}\lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0$$

We obtain $$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\bar{x}} \qquad (3)$$

As long as we estimate the average online time roughly, we can figure out the estimated value to the equation of (3). From [20] we set $\bar{x} = 266$, then the parameter is $\hat{\lambda} \approx 0.004$. When the distribution parameter is determined, it will generate the random exponential distribution number as the node's online time, specific methods are:

$$\mathrm{F}(x_i) = 1 - \exp(-\lambda x_i) \qquad (4)$$

Online Time $$x_i = \frac{\ln(1 - \mathrm{F}(x_i))}{-\lambda} \qquad (5)$$

Depending on the fundamental theorem of random variable sampling, there is R = F (x), where R is a uniformly distributed random variable among [0,1]. As the 1-R and R have the same distribution, so (5) can be written

$$x = -\frac{\ln R}{\lambda} \qquad (6)$$

C. *Weighted Arithmetic Mean in SPSI*

Our goal in SPSI is to select "best" nodes as super-nodes. And we formalize the problem as follows: We are given a set S of n nodes(vi,wi), where vi denotes the one of a node's attribute value such as computational, storage capabilities, bandwidth of its network connection, online time etc. and wi denotes the weight of the value.

In peer-to-peer systems, different application emphasizes different capacity of super-nodes. Here our emphasis is the relation of node's online time and the other capacity. In order to simplify discussion, we use the concept of capacity to abstract in a single quantity all the characteristics such as computational, storage capabilities, bandwidth of its network connection etc. we associate each node n with a parameter Cn representing its capacity, i.e. the number of clients that can be handled by n, and the node's online time is Tn. The easiest way is a linear combination of the two parameters, then to arrive at a parameter:

$$\delta(C_n, T_n) = \xi C_n + \eta T_n \qquad (7)$$

where coefficient ξ is a value between 0 and 1, $\xi + \eta = 1$,denoting the importance of node's online time in relation with node's capacity.

Apparently, There are some questions we must resolve. One is C_n and T_n has different dimensions, and the other is C_n and T_n has different quantity scale. If Cn or Tn is very large, and T_n or C_n is very small, so $\delta(C_n,T_n)$ is large. This selected node is not our expected super-node. In order to resolve these questions, we introduce the information quantity theory to resolve it.

Defination 6: In joint probability space [XY,P(xy)], on the condition of event y ∈ Y, Event x ∈ X 's conditional information quantity is:

$$I(x\mid y)=-\log p(x\mid y) \qquad (8)$$

For the same reason,

$$I(y\mid x)=-\log p(y\mid x) \qquad (9)$$

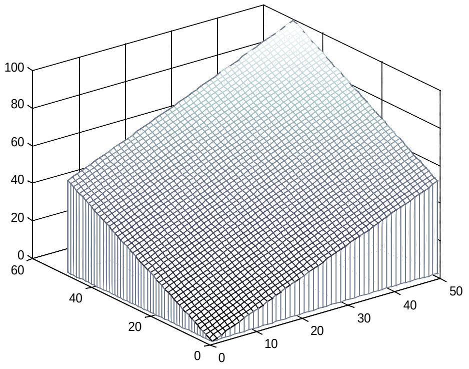


Figure 2. $\delta C_n ,Tn)=Cn+Tn$

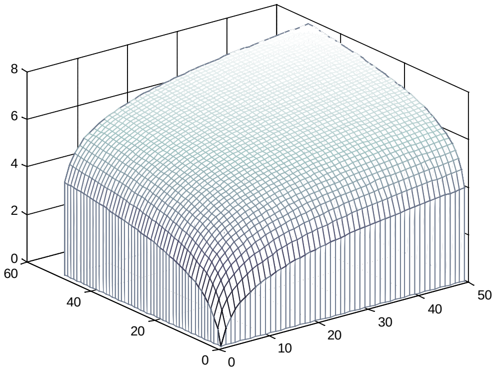


Figure 3. $\delta C_n ,Tn)=logCn+logTn$

Defination 7: In joint probability space [XY,P(xy)], any joint event xy, the joint information quantity of (x∈X，y∈Y) is:

$$I(xy)=-\log p(xy) \qquad (10)$$

Based on the definition of joint information quantity, conditional information quantity, then

$$\begin{aligned} I(xy)&=-\log p(xy)\\ &=-\log(p(x)p(y\mid x))\\ &=-\log p(x)-\log p(y\mid x)\\ &=I(x)+I(y\mid x) \end{aligned} \qquad (11)$$

For the same reason,

$$I(xy)=I(y)+I(x\mid y) \qquad (12)$$

When the event X and Y independently of each other, then

$$I(xy)=I(x)+I(y) \qquad (13)$$

Assuming that the total number of node in the network is m, and node n has a capacity of Cn and online time Tn. The total capacity of all nodes in the network is

$$C_s=\sum_{n=1}^{m}C_n$$

and Online time of all nodes is

$$T_s=\sum_{n=1}^{m}T_n$$

Apparently, the probability of node n with capacity Cn become super-node is

$$p_{Cn}=C_n/C_s \qquad (14)$$

The greater Cn is the more probability to be a super-node. Similarly, the node's probability to become super-node is

$$p_{T\,n}=T_n/T_s \qquad (15)$$

When a node has a capacity of Cn, and online time Tn, we assume that Cn, Tn is independent, then from (13) the combined amount of information is

$$I(C_nT_n)=I(C_n)+I(T_n)=-\log P_c-\log P_t \qquad (16)$$

In order to avoid the appearance of Cn , Ts in the formula, the above equation becomes

$$I(C_nT_n)=-\log\frac{C_n}{C_s}-\log\frac{T_n}{T_s}$$

Let

$$\delta(C_n,T_n)=\log C_n+\log T_n$$

Then

$$I(C_nT_n)=-\delta(C_n,T_n)+\log(C_n*T_s) \qquad (17)$$

Since $\log(C_n*T_s)$ is a constant，I (C_n,T_n) changes only with δ(C_n,T_n), So we can use δ(C_n,T_n) as conditions for selecting a super node。The difference is that if the amount of information based on self-selected super-node, then I(C_n,T_n) the smaller the better, If bases on δ(C_n,T_n), then δ(C_n,T_n) the bigger the better. Since δ(C_n,T_n) only connected with Cn, Tn, not with Cs , Ts, equation of (17) is more feasible than (16). So the selection of super-nodes problem becomes the selection of δ(C_n,T_n).

D. Super-node Selection Algorithm

Our goal is to produce a super-node topology characterized by a about minimum number of super-nodes and the stability of every super-nodes is taken into account. In order to do that, we adopt a classification criteria based on the measure introduced above: nodes with higher $\delta(C_n, T_n)$ are considered better candidates as superpeers. At each time, the target topology is the one composed by the about minimum set of nodes whose total capacity is sufficient to cover all other nodes as clients, moreover the super-nodes are stable as far as possible. Clearly, only in a static network the target topology may be obtained; in the presence of dynamism, the real topology will just approximate it.

The epidemic style algorithm for establishing the super-node and client relationships of the target topology is illustrated in Figure 4. The algorithm is executed only by super-nodes: being more powerful, they can more easily pay the cost of their selection protocol.

The rationale behind function RANDOMGET is the following: all super-nodes try to push clients towards more powerful nodes that are willing to accept more load. To do that, RANDOMGET performs a random selection among those superpeers that are underloaded and whose capacity is larger or equal than the capacity of the local node. Since UNDERLOADED may contain obsolete information, multiple selections are made until a node is found whose capacity is effectively larger than the current number of clients. Ties (nodes with the same capacity) are broken by selecting the node with the larger number of clients. The process continues until such a node is found or no other nodes can be probed.

```
RANDOMGET ()
    Define $\xi$ 0.6
    $\xi$=1- $\eta$
    S←{r|(log(cr)* $\xi$ + log(tr)* $\eta$)≥
        (log($C_p$)* $\xi$ + log($T_p$)* $\eta$)∧r∈UNDERLOADED}
    q←null
    whil(S≠Φ∧q=null)
        r←<pick a random node from S>
        S=S-{r}
        lr←<request load from r>
          if (lr<cr∧((log($C_p$)* $\xi$ +log(tp)* $\eta$)
          <(log($C_r$)* $\xi$ +log($T_r$)* $\eta$)∨lr>lp))
          q←r
    return q

UPDATE(C,p)
    CLIENTS←CLIENTS∪C
     if (lp == 0∧lp<cp)
        CLIENTS←CLIENTS∪{q}
         <q becomes a client>
     else if (   r∈CLIENTS： (log(cr)* $\xi$ + log(tr)* $\eta$)
                  > (log(cp)* $\xi$ + log(tp)* $\eta$))
         <transfer clients of q to r>
        CLIENTS←CLIENTS∪{q}-{r}
        <q become a client， r becomes a server>
```

Figure 4. Super nodes selection algorithm in SPSI

V. Experimental Evaluation

To validate our framework, we have performed numerous experiments based on simulation. Three main questions were interested in by us: first, what is the behavior of the protocol with respect to its parameters; second, what are the communication costs and time consumed associated with its execution; and third, how robust the protocol is.

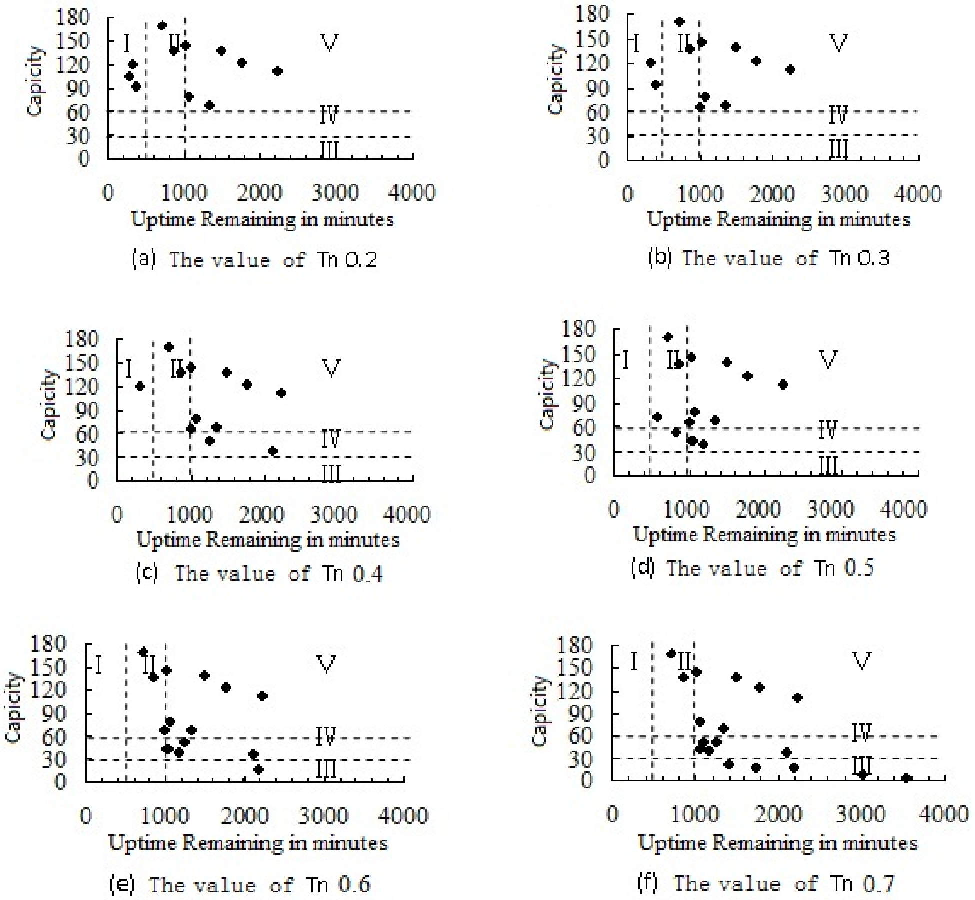
TABLE 2.

INITIAL PARAMETERS IN EXPERIMENTS

Parameters	**Values**
SIZE	40000
MAXCAPACITY	180
MAXTIME	4000
DEGREE	30
REDUCED_DEG	30
ATTEMPTS	30
RATIO	1
LIMIT	0.95
WHEN	30
CRASH	0.90
LAMD	0.02
ALPHA	1.8

All experiments are performed using Peersim and its round-driven Style. In all figures, 20 independent experiments have been performed. Unless stated otherwise, most of the parameters are fixed in all experiments: the maximum capacity of a peer is 500; and the size s of partial views used in NEWSCAST[19] is 30. All these values can be reasonably adopted or measured in realistic settings; yet, the behavior of the algorithm observed under variations of these parameters are analyzed in the following. The initial parameter settings are showed in Table 1.

Value of weighted arithmetic mean coefficient From (7), There are some questions we must resolve. One is C_n and T_n has different dimensions, and the other is C_n and T_n has different quantity scale. If C_n or T_n is very large, and T_n or C_n is very small, so $\delta(C_n, T_n)$ is large. Here the selected node is not our expected super-node. In order to resolve these questions, we introduce the information quantity theory to resolve it. The value of ξ is important. We get the value of ξ from experiments. From [20] we set $\bar{x} = 266$, then the parameter is $\hat{\lambda} \approx 0.004$. When the distribution parameter is determined, it will generate the random exponential distribution number as the node's online time, and the value of C_n is generated from [18].



(a) The value of Tn 0.2

(b) The value of Tn 0.3

(c) The value of Tn 0.4

(d) The value of Tn 0.5

(e) The value of Tn 0.6

(f) The value of Tn 0.7

Figure 5. Values of coefficient Test

For a given online time weight T_n, 0.2,0.3,……,0.8, and the corresponding node content weight C_n as 0.8,0.7,……,0.2. The experiments are carried out, and the results are shown in Figure 5.

To illustrate easily, we divide the figure into four region. $T_n \in [0,500]$ are defined region I, $T_n \in [500,1000]$ is region II, and the nodes of the region are more active than the nodes of region I. $C_n \in [0,30]$ are defined region III, the nodes belong to this region, their capacity are lower, and $C_n \in [0,30]$ are defined region IV. If $T_n > 1000$ and $C_n > 500$, that is to say that the nodes belong to this region have longer online time and higher capacity, these nodes are those we try to select as super-nodes.

From Figure 5 we can see that when the weight of T_n is 0.2, there are too many nodes in region I, that is to say the super-nodes we selected are too active, and the network have to reselect when the super-node is left. When the weight of T_n is 0.6 or 0.7, and too much nodes are located in region III, region IV, that is to say the super-nodes we selected have lower capacity. So it is better to set the weight of T_n 0.4 or 0.5 and it is better to set the weight of C_n is 0.6.

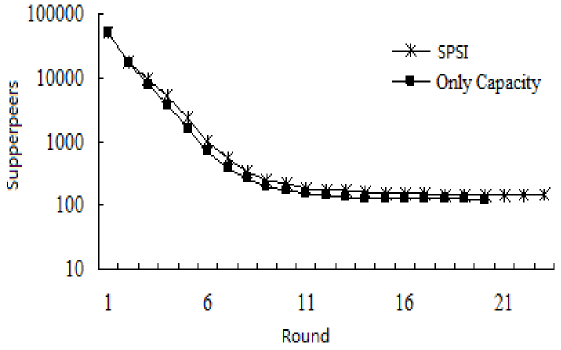


Figure 6. Network convergence Test

A. The Convergence Speed of Network Construction

The goal of convergence experiment is to measure the speed of convergence, it's important in overlay network construction. In the experiments we also make our protocol with SG-1, a famous super-node construction protocol based on NEWSCAST[19] to compare the convergence speed. The results are showed in Figure 6. In the picture, dashed line indicates the number of super nodes with SPSI protocol, while Solid line represents the number of super nodes with SG-1. It can be seen that the convergence speed of SPSI and SG-1 are basically consistent. The time needed to reach such utilization thresholds is independent from network size and around 10 and 13 rounds, respectively. As initial configuration, we selected a topology that is the farthest from the target: a random topology where all nodes behave as super-nodes, although none of them is responsible for any client. The curves represent the number of super-nodes contained in the network after the specified number of rounds, averaged over 20 experiments. Individual experiments are shown; their x-coordinates have been shuffled with a small random increment to separate similar results. The algorithm proves to be extremely fast, independently from the distribution considered: after less than 15 rounds, the resulting topologies approximate extremely well the target.

B. The Selected Super-nodes

Figure 7 and Figure 8 shows the online time parameter's impact on the number of selected super nodes. The horizontal axis is the uptime remaining (in minutes) of nodes, the vertical axis is the capacity of node. There are 1000 nodes in the network being tested. Each "fork" represents client nodes (shown using "+") and a box indicates the super node (shown using □).

From Figure 7, we can see that some nodes with low remaining uptimes are selected as super-nodes, and as these nodes leave the system, the system has to select another node in the network to take over its work. Figure 8 shows that most super nodes are located at the center of the graph, explaining that the selected super-node has a certain capacity and a longer time line. It can be seen that the SPSI protocol can effectively avoid selecting active nodes as super nodes, thus increasing the stability of the target topology.

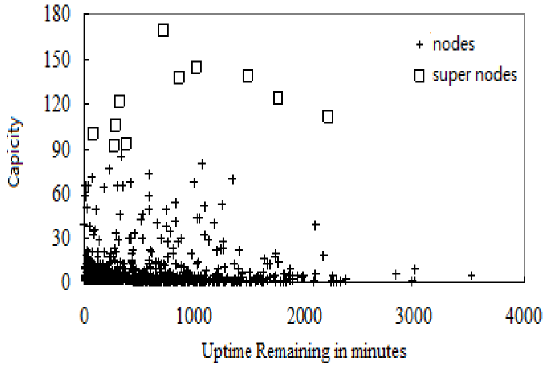


Figure 7. Super nodes selection without online time parameter

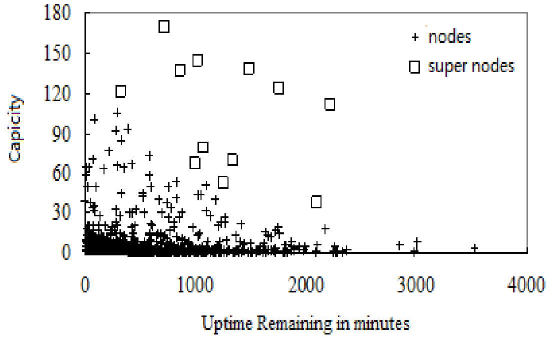


Figure 8. Super nodes selection with online time parameter

C. Communication Costs

In order to verify the effect of the protocol to lower communication costs, we conduct experiments 9. Two communication costs are to be considered: the total number of probes sent in protocol to discover the load of other super-nodes, and the total number of client transfers performed.

Super nodes take more responsibility in a super nodes based peer-to-peer network. When a super node logouts from the system or a super node decides to alleviate its load, it will bring about considerable communication cost. The main purpose of the SPSI protocol is to select a relatively stable node as super-node and save part of the network overhead. In Figure 9, the solid line represents the results of SPSI protocol, while the dotted line shows only the case of the node capacity. The SPSI protocol can select out relatively stable nodes when selects super-nodes, thus it has fewer reconfiguration. Only using the node capacity as selection method, the super nodes will have higher activity, and the network is not stable enough, thus it has more reconfiguration than the self-information algorithm.

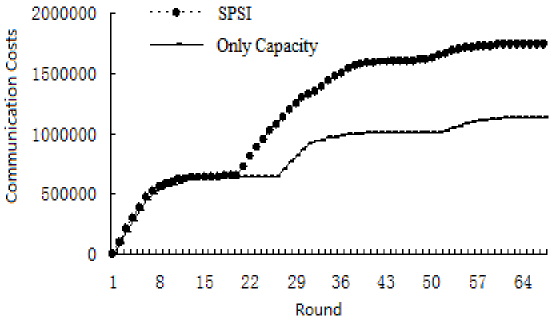


Figure 9. Communication Costs

Figure 9 shows the difference between the capacity and the self-information amount algorithm in the overhead costs. The overhead costs include exchange of messages between nodes and conversion cost between client nodes. It can be seen from the figure, the cost has not been decreased with the self-information amount algorithm in the early time of the network construction, but after a long period, when there are nodes exiting the network, the cost of the self-information amount algorithm is significantly less than that of capacity algorithm. It can be seen that the algorithm can effectively limit the number of active nodes as super nodes, thereby reducing the overhead for building the network.

D. Roubst Test

In order to demonstrate the robustness of our protocol, we hypothesis a catastrophic scenario and the test result is shown in Fig.10.: at round 30, 50% of the super-peers are removed. After the initial period when all clients whose super-peer has crashed become super-peer by themselves, the protocol behaves as usual and repair the overlay topology by selecting new super-peers among the remaining nodes.

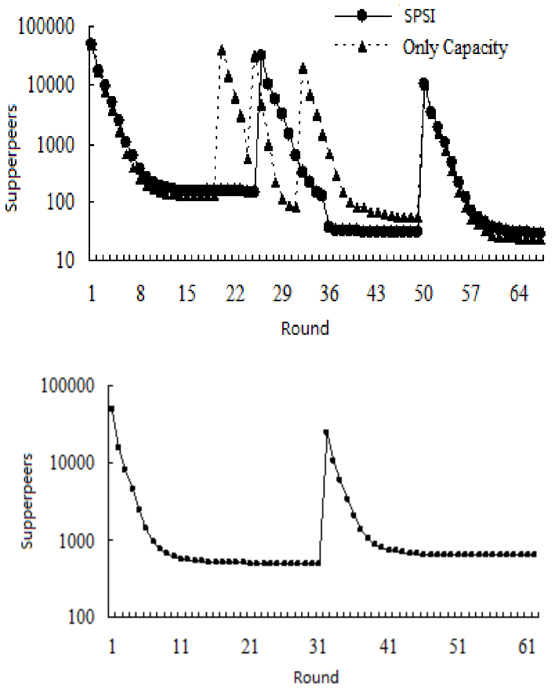


Figure 10. Roubst Test

VI. Conclusions

This paper presented SPSI, a fully decentralized, self-organizing general protocol for the construction of super-node-based overlay topologies. To our best knowledge, we are the first to introduce information theory to super-node selection. The number of generated super-nodes is a little more than SG-1 but small with respect to the network size (only 3-5%), and it's important that the more stable peers are promoted as super-nodes, so the communication costs are degraded and the target topology is more stable. The protocol shows also an acceptable robustness to churn.

References

[1] L. Meirong, K. Timo, O. Zhonghong, Z. Jiehan, R. Jukka, Y. Mika, “Superpeer-based coordinated service provision”, Journal of Network and Computer Applications,Vol 34, pp. 1210–1224, July 2011.

[2] S. Moritz, E.N. Taoufik, W.B. Ernst, “Long term study of peer behavior in the KAD DHT”. IEEE/ACM Trans. Netw. Vol. 17, pp. 1371-1384, October 2009.

[3] Z. Liu, C. Wu, B. Li, S. Zhao, “Distilling Superior Peers in Large-Scale P2P Streaming Systems”. In Proc. of INFOCOM'2009. pp.82~90.

[4] D.T. Talia, P.Trunfio, “Towards a Synergy between P2P and Grids”, IEEE Internet Computing, vol4, pp. 94-96, July.2003.

[5] A.I. Iamnitchi, I. Foster, J.Weglarz, J. Nabrzyski, “A Peer-to-Peer Approach to Resource Location in Grid Environments”, In: Proceedings of the 11th Symposium on High Performance Distributed Computing, Edinburgh, UK, August 2002.

[6] Z. W. Gao, Z. M. Gu, P. Luo, “RBTree: a new and scalable p2p model based on gossiping”. In:Proceedings of the IEEE International Symposium on Ubiquitous Multimedia Computing (UMC 2008). October, 2008.

[7] J. P. Gian, M. Alberto, and B. Ozalp, “Proximity-aware Superpeer Overlay Topologies”. IEEE Transactions on Network and Service Management (TNSM), Vol. 4, pp:74-83, September 2007.

[8] KaZaA, [Online]. Available: http://www.kazaa.com/.

[9] K. Truelove, Gnutella and the transient web, Whitepaper, 2002.

[10] S. Q. Lv, Ratnasamy, S. Shenker, “Can heterogeneity make gnutella scalable?”. in: Proc. IPTPS, March 2002.

[11] B. Yang, H. Garcia-Molina, “Designing a superpeer network”. In: Proceedings of the 19th International Conference on Data Engineering. (2003) 49–60.

[12] A.T. Mizrak, Y. Cheng, V. Kumar, S. Savage, “Structured superpeers: Leveraging heterogeneity to provide constant-time lookup”. In: Proceedings of the 3rd IEEE Workshop on Internet Applications. (2003) 104–111.

[13] L. Garces-Erice, E. Biersack, P. Felber, K. Ross, G. UrvoyKeller, “Hierarchical peer-to-peer systems”, in: Proceedings of EuroPar, Klagenfurt, Austria, 2003.

[14] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, A. L¨oser, “Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks”, in: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, 2003.

[15] A. Mizrak, Y. Cheng, V. Kumar, S. Savage, “Structured super-peers: Leveraging heterogeneity to provide constant-time lookup”, in: IEEE Workshop on Internet Applications, 2003.

[16] A. Singh and M. Haahr, “Creating an adaptive network of hubs using Schelling’s model,” Commun. ACM, vol. 49, no. 3, pp. 69–73, 2006.

[17] H.J. Liu, P. Luo et.al, “A structured hierarchical P2P model based on a rigorous binary tree code algorithm”. Future Generation Computer Systems-The International Journal of Grid Computing Theory Methods and Applications 23 (2): 201-208 Feb 2007.

[18] A. Montresor, “A robust protocol for building superpeer overlay topologies,” In Proc. of the 4th Int. Conf. on Peer-to-Peer Computing. Zurich, Switzerland: IEEE, August 2004.

[19] J. Márk, V. Spyros, G. Rachid, K. Anne-Marie, and S. Maarten, “Gossip-based peer sampling”. ACM Transactions on Computer Systems, 25(3):8, August 2007.

[20] S. Saroiu, P. K. Gummadi, and S. D. Gribble, “A measurement study of peer-to-peer file sharing systems”. In Proc of Multimedia Computing and Networking 2002 (MMCN ’02), San Jose, CA, USA, January 2002.

Zhiwei Gao is an associate professor at the Department of Computer Science, ShiJiaZhuang TieDao University. He is a Ph.D. student in the Department of Computer Science at Beijing Institute of Technology, Beijing, China. He received her Master degree from school of information and computer technology,Beijing Jiao Tong University.His current research interests are in network security, distributed computing and peer-to-peer systems.

Yingxin Hu is a lecture at the Department of Computer Science, ShiJiaZhuang Railway Institute. His current research interests are in e-commerce, distributed computing and peer-topeer systems. He received his master’s degree in computer science from ShiJiaZhuang Railway Institute, China.