

SPSI: A Hybrid Super-node Election Method Based on Information Theory

Zhiwei Gao^{1,3}, Zhimin Gu¹, Wenbao Wang²

¹Department of Computer Science,
Beijing Institute of Technology, Beijing, 100081, China

²Department of Computer Science,
Shijiazhuang Vocational Technology Institute, Shijiazhuang, 050081, China

³Department of Computer Science,
Shijiazhuang TieDiao University, Shijiazhuang, 050043, China
E-mail: gao_zhiwei@163.com, zmgu@x263.net, ww@sjzpt.edu.cn

Abstract— By exploiting heterogeneity, the super-node paradigm can lead to improved efficiency, without compromising the decentralized nature of peer-to-peer (P2P) networks. So many relevant applications such as grids, cloud computings use super-node paradigm as the lower service. However, due to inherent decentralisation, scale, dynamism, 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 (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. We are the first (to the best of our knowledge) 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— super-node, self-information, scalability, SPSI

I. INTRODUCTION

Super-nodes allow decentralized networks to run more efficiently by exploiting heterogeneity and distributing load to machines that can handle the burden. On the other hand, this architecture does not inherit the flaws of the client/server model, as it allows multiple, separate points of failure, increasing the health of the distributed network. So many distributed applications such as grids, cloud computings use super-node paradigm as the lower service. 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 decentralisation, 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 [6,7,8,9]. 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.

In order to efficiently locating data or resources among the numerous participating nodes, we explore a fast and scalable

super-node P2P network model based on RBTree protocol[5]. In our model, 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 the work doesn't resolve the super-node election problem elegantly. So this paper is the continue work of RBtree protocol.

In this paper, we present a super-node selection algorithm based on self-information theory (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.

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.

The remainder of the paper is organized as follows. Section II discusses the background theory and terminology. Section III introduces system model and algorithm. The simulation and experimental results are presented in Section IV. Section V gives related works. Section VI gives our conclusion.

II. BACKGROUND THEORY AND TERMINOLOGY

The SPSI protocol can be considered as a natural evolution of RBTree protocol. In [6,15], We propose a framework based on our own efficient and scalable Rigorous binary tree model and its theorem. 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.

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

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.

Theorem 1: Rigorous binary tree mapping theorem: For any one integer I ($I \geq 0$), 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[6,15].

Based on Rigorous binary tree extension and mapping theorem, we design RBTtree 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 2: 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 $0 \leq p \leq 1$;
- $I(p_1 \times p_2) = I(p_1) + I(p_2)$;

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

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.

Definition 3: 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:

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

Where:

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

x : Stands for values of the items and

w : Stands for weight of the item.

III.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 to network 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.

To build a topology with such characteristics, we propose a mechanism based on NEWCAST [17]. 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[17].

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 C_n 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 [18], 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 $P(C_n=x)=x^{-\alpha}$, In which $x \in [1, C_{\max}]$, α is the distribution parameters (usually the parameter $\alpha = 2$). The node size 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 structure. 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:

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

We use maximum likelihood estimation method to estimate the parameter λ , assuming that $x_1, x_2, x_3 \dots$ is a set of random sample values, representing the node's Online time, Then Likelihood function

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

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

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

As long as we estimate the average online time roughly, we can figure out the estimated value to the equation of (3). From [18] 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:

Distribution function

$$F(x_i) = 1 - \exp(-\lambda x_i) \quad (4)$$

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

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

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

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 (v_i, w_i) , where v_i denotes the one of a node's attribute value such as computational, storage capabilities, bandwidth of its network connection, online time etc. and w_i 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 C_n representing its capacity, i.e. the number of clients that can be handled by n , and the node's online time is T_n . The easiest

way is a linear combination of the two parameters, then to arrive at a parameter:

$$\delta(C_n, T_n) = \xi C_n + (1 - \xi) T_n \quad (7)$$

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

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

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

$$\text{and Online time of all nodes is} \quad T_s = \sum_{n=1}^m T_n$$

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

$$p_{C_n} = C_n / C_s \quad (8)$$

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

$$p_{T_n} = T_n / T_s \quad (9)$$

When a node has a capacity of C_n , and online time T_n , we assume that C_n, T_n is independent, then from *Theorem 2* the combined amount of information is

$$I(C_n T_n) = I(C_n) + I(T_n) = -\log P_c - \log P_t \quad (10)$$

In order to avoid the appearance of C_n, T_s in the formula, the above equation becomes

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

$$\text{Let} \quad \delta(C_n, T_n) = \log C_n + \log T_n$$

Then

$$I(C_n T_n) = -\delta(C_n, T_n) + \log(C_n * T_s) \quad (11)$$

Since $\log(C_n * T_s)$ is a constant, $I(C_n T_n)$ changes only with $\delta(C_n, T_n)$. So we can use $\delta(C_n, T_n)$ 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 $\delta(C_n, T_n)$, then $\delta(C_n, T_n)$ the bigger the better. Since $\delta(C_n, T_n)$ only connected with C_n, T_n , not with C_s, T_s , equation of (11) is more feasible than (10). So the selection of super-nodes problem becomes the selection of $\delta(C_n, T_n)$.

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 gossip-based algorithm for establishing the client-super-node relationships of the target topology is illustrated in Figure 1. The algorithm is executed only by superpeers: being more powerful, they can more easily pay the cost of their selection protocol.

The rationale behind function RANDOMPEER is the following: all superpeers try to push clients towards more powerful nodes that are willing to accept more load. To do that, the function 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.

```

RANDOMPEER()
  S ← {r | (log(cr)*0.6 + log(tr)*0.4) ≥
        (log(Cp)*0.6 + log(Tp)*0.4) ∧ r ∈ UNDERLOADED}
  q ← null
  while (S ≠ ∅ ∧ q = null)
    r ← <pick a random node from S>
    S = S - {r}
    lr ← <request load from r>
    if (lr < cr ∧ ((log(Cp)*0.6 + log(tp)*0.4)
        < (log(Cr)*0.6 + log(Tr)*0.4) ∧ 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)*0.6 + log(tr)*0.4)
        > (log(cp)*0.6 + log(tp)*0.4))
    <transfer clients of q to r>
    CLIENTS ← CLIENTS ∪ {q} - {r}
    <q become a client, r becomes a server>

```

Figure 1. Super nodes selection algorithm in SPSI

IV. EXPERIMENTAL EVALUATION

To validate our approach, we have performed numerous experiments based on simulation. We were interested in three main questions: (1) what is the behavior of the protocol with respect to its parameters; (2) what are the time and communication costs associated with its execution; and (3) how robust the protocol is.

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[17] 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.

TABLE 1. INITIAL PARAMETERS IN EXPERIMENTS

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

A. The Convergence Speed of Network Construction

Convergence speed is importance in overlay network construction. The goal of this experiment is to measure the speed of convergence. How fast the SPSI protocol compared with the other protocol? One might imagine. So we also make our protocol with SG-1, a famous super-node construction protocol based on NEWSCAST[17]. The results are showed in Figure 2. 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 has 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 3 and Figure 4 shows the impact on the number of selected super nodes with and without online time parameter. 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 3, we can see that some nodes with low remaining uptimes are selected as super-nodes, so as these

nodes leave the system, and the system has to select another node in the network to take over its work. Figure 4 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.

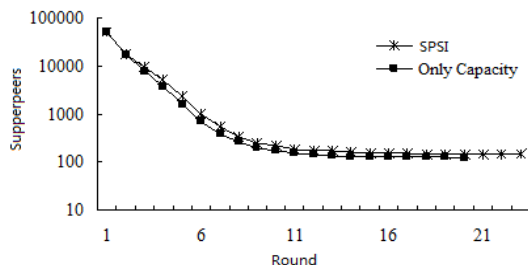


Figure 2. Convergence Test

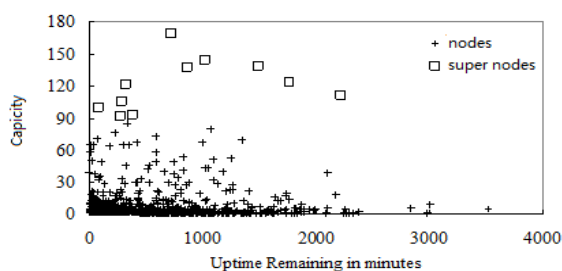


Figure 3. Super nodes selection without online time parameter

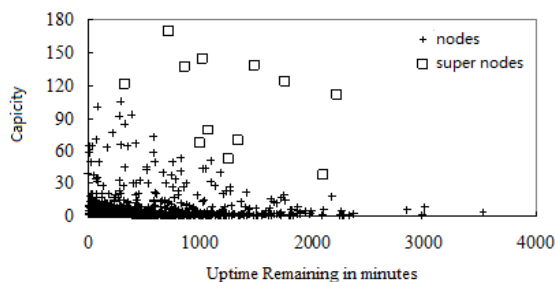


Figure 4. Super nodes selection with online time parameter

C. Communication Costs

In order to verify the effect of the protocol to lower communication costs, we conduct test 3. 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.

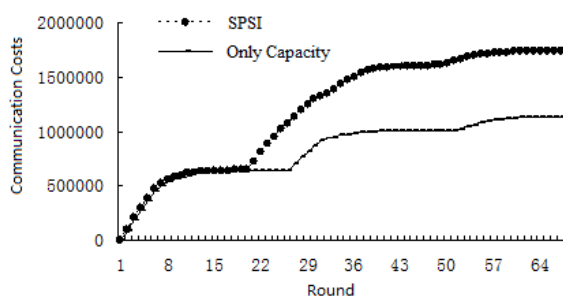


Figure 5. Communication Costs

In a super nodes based peer-to-peer network, super nodes take more responsibility. 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 5, 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 5 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.6.: 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.

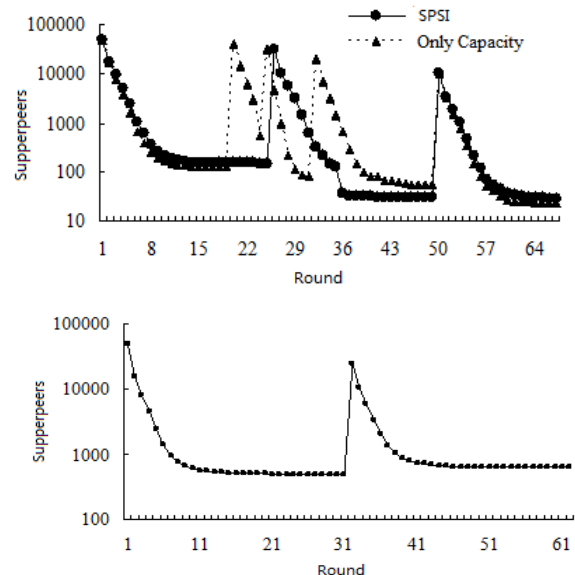


Figure 6. Roubst Test

V. 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.

Yang and Garcia Molina [1] proposed some design guidelines. A mechanism to split node clusters is proposed and evaluated analytically, but no experimental results are presented.

KaZaA [10] and Gnutella [11,12] 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.

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. They concluded that hierarchical organization could improve a 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, 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.

In [16], 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 [7], 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.

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.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Fund of China (No: 50975185).

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, "RBTtree: 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] B. Yang, H. Garcia-Molina, "Designing a superpeer network". In: *Proceedings of the 19th International Conference on Data Engineering*. (2003) 49-60.
- [9] 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.
- [10] KaZaA, [Online]. Available: <http://www.kazaa.com/>.
- [11] K. Truelove, Gnutella and the transient web, Whitepaper, 2002.
- [12] S. Q. Lv, Ratnasamy, S. Shenker, "Can heterogeneity make gnutella scalable?". in: *Proc. IPTPS*, March 2002.
- [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] 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.
- [16] 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.
- [17] 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.
- [18] 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.