■ 703

Chord-based Resource Identifier-to-Locator Mapping and Searching for the Future Internet

Huanlin Liu, Hongyue Dai, Shuaiyong Wu, Sheng Huang
School of Communicaion and Information Engineering, Chongiqng University of Posts and Telecommunications
Chongwen Road 2#, Nan'an District, Chongqing City, 400065, China
e-mail: liuhl2@sina.com

Abstract
A great many problems, such as scalability, mapping data searching, high frequency update of mapping data, arise in the future network resource mapping system for its vast data processing need. Future Network Chord (FN Chord), an algorithm based on Chord and aims at solving the resources identity mapping and searching problem, is put forward by taking advantage of the qualities of scalability, rapid searching speed, high searching efficiency and flexible naming of chord in order to solve this problem. What's more, an extra interest node index table for FN Chord is designed to record the hotspot resource mapping location in the paper. So, the resource searching strategy, which is named as Interest Index Table Future Network Chord (IIT-FN Chord) is proposed to search the resource in the paper. The entropy weight method is used to calculate the node interest level according the interest nodes' resource item online time and visited times and to renew the interest index table. Moreover, probability replacement method is proposed to replace the outdated item on interest index table with new item. Simulation results show that the algorithm can decrease the average searching latency, average searching hops and thus increases the searching efficiency for the resource searching.

Keywords: future internet, chord, interest index table, entropy weight, average searching hops

1. Introduction

With the rapid development of Internet and the quick emergence of mobile Internet, clouding computing [2], Internet of things [3], the problem of Future Internet constantly occurs [4], such as the problems of Internet mobility, routing scalability, multi-homing and security. Many solutions are put forward to satisfy the need of future Internet network, for example, virtualization network topology [5], redesign content centric network architecture [6], and split resource identifier and locator [7], etc. What identifier and locator split means is to represent the double semantic of IP with identifier and locator respectively. At present, the identifier and locator split architecture, for which the realization of resource identifier mapping service system [8] is of great importance, is widely studied by worldwide researchers. The core issue to realize the resource identifiers mapping system is its resource mapping and searching algorithm, which should be able to satisfy scalability, mass data mapping, update and inquiry, etc.

In the P2P system, the DHT-based resource searching method includes Chord [10], CAN [11], Pastry [12] and Tapestry [13] in the P2P structured network. The Chord based method has the advantage of self-organization, scalability, robustness and rapid inquiry for its distributed hash table based identifier mapping system. So, the technology of Chord may become one of the promising candidate technologies for the resource searching of future Internet network. But in the future Internet network, the issues of scalability, quick inquiry, high frequency mapping and data update will rise in the resource mapping service system for there will be mass data and application in it. It is found that one of three or so the indexes in the Chord routing table are repetitive, which reduces the inquiry efficiency and increases the hops of inquiry [14]. Moreover, a small number of nodes with hotspot information are visited most often than the vast majority of information access nodes, which leads to long-tail effect of inquiry data [15] and increases the inquiry average hops and delay. Taking full advantage of Chord, the Future Network Chord (FN Chord), an improved algorithm of resource identifier mapping and searching in identifier and locator split future network, is put forward to decrease the average inquiry delay and the average inquiry hops in the paper. The improved algorithm based on interest index table in the future network Chord is proposed to solve the long-tail effect problem

for mass data inquiry. By considering both interest nodes online time and accessed times, we take advantage of both entropy weight algorithm and probability replacement algorithm to calculate the node's interest level and to decide which Chord entry to be replaced in the interest node index table respectively.

2. IT-FN Chord Identifier Mapping Method

To decrease the average searching hops and average searching delays in the future Internet network, an improved IIT-FN Chord resource identifier mapping and searching algorithm which is based on interest index table is proposed in the section.

An interest node index table for IIT-FN Chord is designed in table 1 firstly.

Table 1. Structure of interest node index table for one resource item

NID	Node IP and port	NID precusor node	NID successor node	VT	VC

Each node joins the IIT-FN Chord should keep information of its precusor node, its sucessor node and its finger table. And what's more, an interest node index table for one item is formed. In the interest node index table, there must be information about the servers visited by the current server in this table. It should include items like the IP address and port number of the NID (Node's hash Identification), the IP and port number of the predecessor of this NID and the IP and port number of the successor of this NID in IIT-FN Chord, experienced number of periods during the item idle times (VT) and item visited times (VC).

Four conditions must be guaranteed in the design of interest index table:
(1) Reasonable table scale;
(2) Hotter node is, more likely node should be in the table 1;
(3) Each new item should be able to join in the table;
(4) Each outdated information should be eliminated from the table as soon as possible.

Firstly, since the SHA-1 is used as the DHT hash function in IIT-FN Chord, there should be 160 items in the finger table. According to the 20/80 principle, studies show that there are only 20 items which are frequently inquired in every 100 items. So, we assume that the number of items in the interest node index table is 160*20%=32.

Secondly, calculate interesting level of each node. The importance of one resource item, namely its interesting level, is obtained by applying the entropy weight algorithm [16] which takes both idle time and accessed times into consideration. The i-th item's interesting level of node is calculated in equation (1).

$$R_i = \beta_1 \cdot VT_i + \beta_2 \cdot VC_i \tag{1}$$

Where, VT_i and VC_i represent the node's item i idle times and visited times, respectively. Parameters β_1 and β_2 are the weight coefficient of VT and VC, respectively. VT and VC are initialized when a new item is added to the node as VT0 and 1, respectively. Here, VT0 is a big value according to the network's resource access level, such as we set VT0=32, VC=1, i=1, 2, …, VT0 at the algorithm's beginning to evaluate node's interest for resource item i. If the node is not visited by other node in one time period for the item i, the item's VT0 decreases about 1. If the item's VT reduces to 0, the item i is kicked out of the interest node index table. If the node is visited by other node in one time period for the item i, the item's VC increases about 1. If we calculate all items' interest level of node, we sort all items according to each item's interest level from high to low order.

$$\beta_1 = \frac{w_1}{w_1 + w_2}, \beta_2 = 1 - \beta_1 \tag{2}$$

Where, w_1 is the entropy weight of VT, and w_2 is the entropy weight of VC.

$$w_1 = (1-e_1)/\sum_{j=1}^{2}(1-e_j)$$

(3)

The e_1 and e_2 are the entropy of VT and VC respectively.

$$e_1 = -\frac{1}{lnVT0}\sum_{i=1}^{VT0}p_{i1}\cdot \ln p_{i1}; e_2 = -\frac{1}{\ln VC}\sum_{i=1}^{VC}p_{i2}\cdot \ln p_{i2}$$

(4)

The p_{i1} and p_{i2} are i-th item's proportion of VT parameter and VC parameter, respectively.

$$p_{i1} = VT_i / \sum_{i=1}^{VT0}VT_i; p_{i2} = VC_i / \sum_{i=1}^{VC}VC_i$$

(5)

Thirdly, replace one item in the node's interest index table using the probability replacement algorithm [17] which takes all the items interesting level as probability judge criterion. The higher the interest level the item has, the less likely it will be replaced.

Set **P** is the alternative probability of each item in set R by a new item. Here, set R is the each item's interest level of node, and $\mathbf{P} = \{P_1, P_2, \ldots, P_{VT0}\}$, $\mathbf{R}=\{R_1, R_2, \ldots, R_{VT0}\}$. When a new item is added to the interesting nodes index table, the item's alternative probability P_i is initialized as 0.5. The alternative probability of i-th item in the interest index table is designed as follow.

$$P_i = 0.5 * \frac{(1/R_i)^2}{\sum_{i=1}^{VT0}(1/R_i)^2}$$

(6)

According to the equation (6), we know that the larger R_i value of a item, the smaller P_i. Lastly, we decide whether the node is outdated. An interest node is taken as an outdated node and will be eliminated from the interest node index table as long as it is not visited for a certain number of periods. In other words, if one item's VT of the node is 0, the item is kicked out of the interest node index table.

3. IIT-FN Chord resource searching

Before we describe the proposed chord-based resource searching strategy, we define some symbol as follows.

NID represents the ID of a node. VC is visited times of the target node. Usually, VC=1 when a new item is added to the interesting node index table. And VC is added 1 as target node is visited once. VT indicates the node's remaining available idle times. Usually, we assume that the initial value of VT is 32, namely VT0=32 at the beginning. If a target node is not visited during one period, VT is reduced to 1. If VT=0, then this node is eliminated from the interest index table immediately. IIT is interest node index table.

When node *n* in the IIT-FN Chord sends a resource inquiry K, the process of resource searching is designed as follow.

Step 1: Check IIT to see if there is any invalid or outdated node in it. If there is any, delete it from the IIT. Renew successor node number and precursor node number of each node on IIT. Update finger table of each node on IIT-FN Chord.

Step 2: Inquiry the resource K on node *n*. If resource K is on node *n*, the resource searching success and the searching process goes to **Step 3**. Else if resource K is not on node *n*, the resource searching goes to **Step 4**.

Step 3: Establish a new item to IIT of node *n* with VT=32, VC=1. If IIT of node *n* is full, calling the probability replacement algorithm to replace some item of IIT with this new item. Else if IIT is not full, the new item is added directly to the IIT.

Step 4: Inquiry the IIT and judge whether the resource K needs only one hop. If *n*<K< successor (*n*), then resource K is on the one hop node of successor (*n*); Else goes to **Step 5**.

Step 5: Read the IIT item by item to find out if there is any item in which precusor(*n'*)<K< *n'*. If so, we set the item with VT=32, VC=VC+1. Else, goes to **Step 6**.

Step 6: Check IIT-FN Chord's finger table of node *n* to find the node *x* which is both closest to and smaller than K. If the node *x* exist, then set *n* = *x* and resource searching success goes to **Step 2**. Else, the resource K inquiry fails and the process of resource K searching ends.
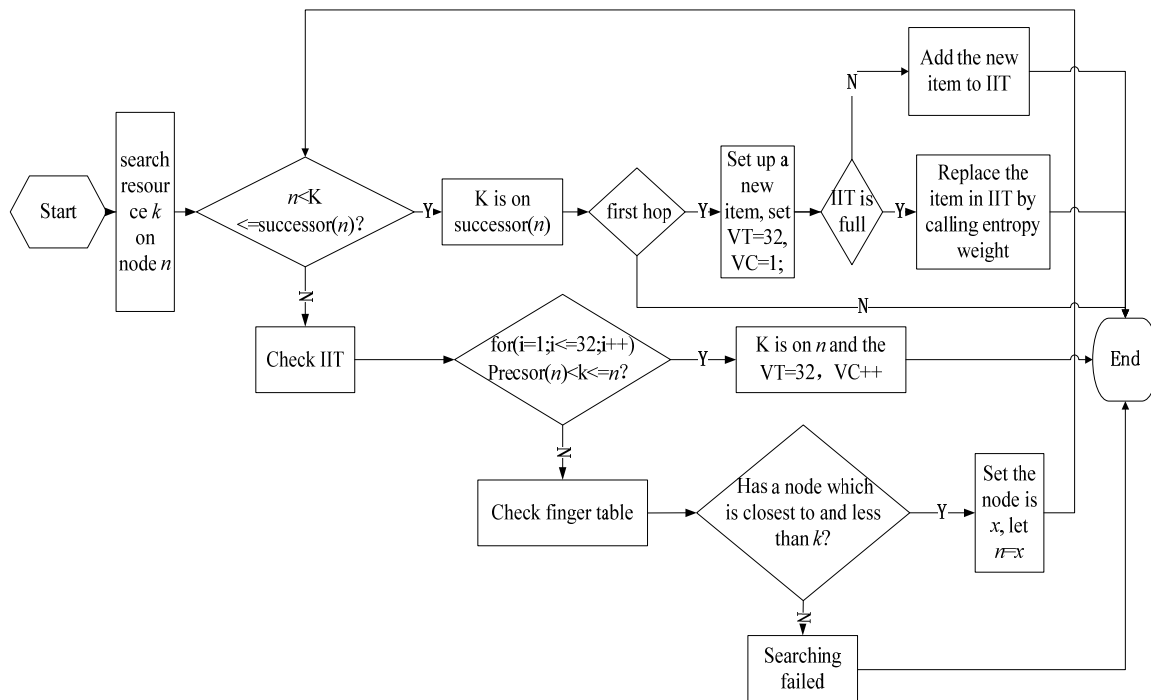

Figure 1. The flow chart of resource K searching

4. Results and Analysis

To evaluate the performance of IIT-FN Chord resource mapping and searching strategy, by comparing with the FN Chord system without the IIT, the 100 times experiments are carried out in five different nodes scalability environments, which are 128, 256, 512, 1024, 2048, respectively in P2PSim software.

In Figure 2, we compares probability density function (PDF) of query hops under different FN Chord node number N=128, 256, 512, 1024, 2048, respectively. As shown in Figure 2, under different node number in FN Chord system, the IIT-FN Chord searching strategy needs average fewer query hops comparing with FN Chord searching strategy. The reason is that we design an IIT for the IIT-FN Chord system, the resource query request gets the target node faster according to node's interest index table. For example, in Figure 2(c), the node number of FN Chord system is N=512, the number of 1, 2, 3, 4, 5, 6, 7, 8 hops success searching is 9, 17, 24, 22,12, 9,5, 2 times, respectively in IIT-FN Chord system. But, the number of 1, 2, 3, 4, 5, 6, 7, 8 hops success searching is 2, 8, 18, 27, 26, 13, 4, 2 times, respectively in FN Chord. It is quite clear that IIT-FN Chord is more likely to end up with successful searching of less hops. In Figure 3 (e), N=2048, the number of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 hops success searching is 4, 5, 12, 24, 22, 16, 9, 4, 3,1, respectively in IIT-FN Chord. But, the number of 1, 2, 3, 4, 5, 6, 7, 8, 9,10 hops success searching is 1, 4, 8, 18, 25, 24, 13, 5, 2, 0 times, respectively
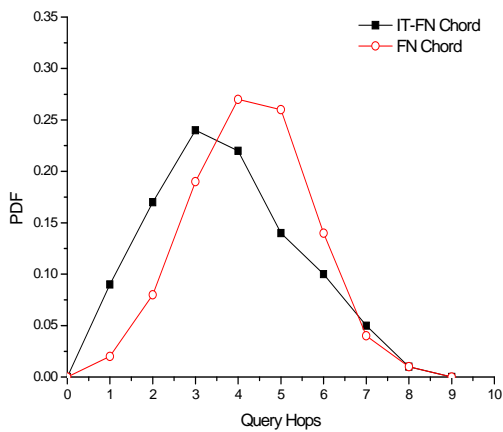
in FN Chord. It is quite clear that IIT-FN Chord is more likely to end up with successful searching of less hops.
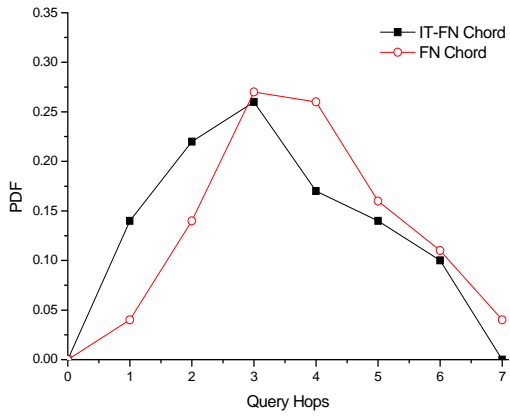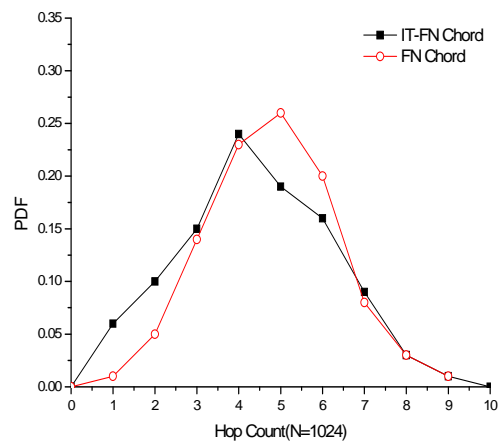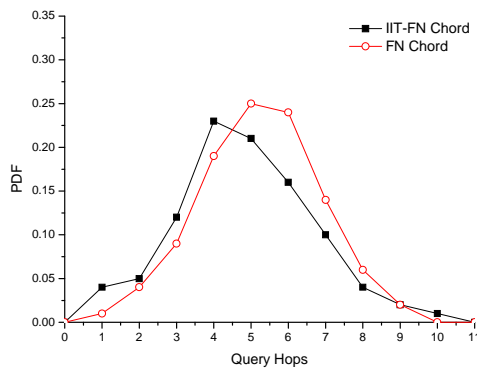

(a) node number N=128
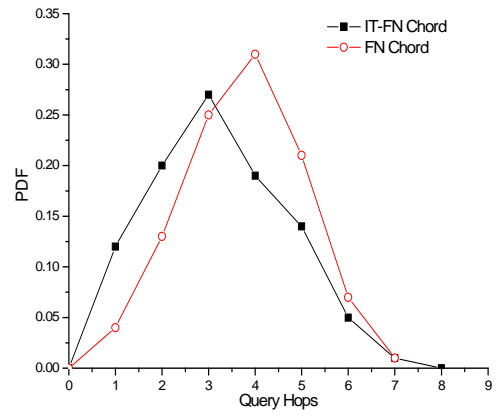
(b) node number N=256

(c) node number N=512

(d) node number N=1024

(e) node number N=1024

Figure 2. Probability density function (PDF) of query hops with different number of nodes

It is exactly the same tendency for N=128, 256, 1024. So we can conclude that:

(1) Searching process in IIT-FN Chord has a higher probability to get the needed information than in FN Chord, namely it is more likely to get the required information for IIT-FN Chord with less hops.

(2) As the number of identifier mapping servers increases, the inquiry hops of IIT-FN Chord tend more likely to be less than 0.5*logN*–0.5, here N is the node number of FN Chord system.
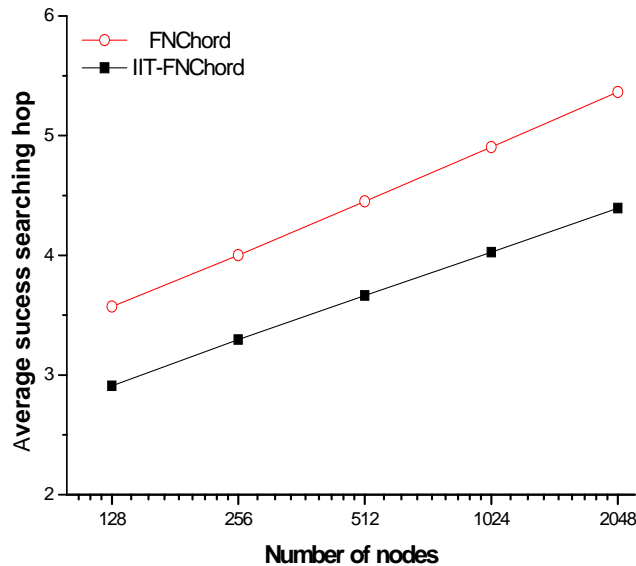

Figure 3. Average success searching hops versus number of nodes

In Figure 3, we compare the resource average success searching hops versus the number of nodes. By analyzing the FN Chord resource strategy process, we know a search can locate the target node within the limited hops, namely algorithm's complexity about 0.5*logN* in FN Chord. On the other hand, the complexity of IIT-FN Chord is about 0.5*logN*–0.5. As the number of node increases, the average success searching hops of IIT-FN Chord increases slower than FN Chord. This is simply because of the increase of success probability of one hop searching after adding the node interest node index table. With the increase of the number of nodes, the success probability of one hop searching is even higher, thus the difference value of the average success searching hops between FN Chord and IIT-FN Chord will be even larger.

As shown in Figure 4, we can find the average searching latency of FN Chord increases with the increase of the number of identifier mapping nodes in the Internet. It is the same with the IIT-FN Chord. What is more, the average searching latency of IIT-FN Chord is much less than FN Chord and grows slower as the number of the mapping nodes increases. This is simply because the constantly updated Interest node index table increases the resource search speed. In the IIT-FN Chord, the nodes on IIT keep high resource interest level by using entropy weight algorithm to calculate item's online time and visited times comprehensively. And the outdated nodes are replaced by new nodes with probability replacement method. The two measures guarantee the nodes on IIT high interest level and reduce its resource searched time.
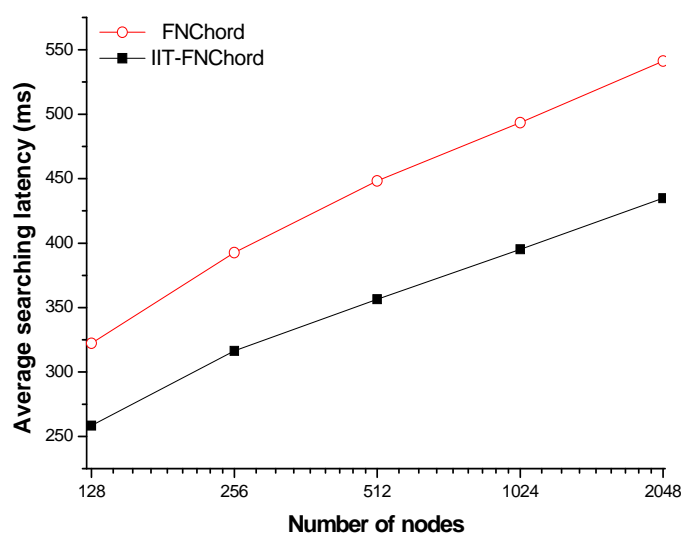
Figure 4. Average success searching latency versus the number of nodes

5. Conclusion

In this paper, an IIT-FN Chord is proposed to improve the Chord system identifier-locator mapping efficiency and speed up the resource searching for the future Internet. By designing an extra interest index table for the node, the hotspot resource can be located quickly. To improve the efficiency of interest index table, the entropy weight algorithm is introduced to calculate the resource item's interest level. The outdated resource is replaced by new item with the probability replacement method. As the applications and nodes increase in the future Internet, the IIT-FN Chord can keep Chord of P2P advantages and improves resource searching efficiency and speed. IIT-FN Chord is expected to solve the Internet inherent scalability and performance problem for the future Internet.

References

[1] Inoue A, Nagahata R, Ishii Y, et al. *Mobile Internet-access behavior analysis*. Proceedings of IEEE conference on SNPD. Kyoto. 2012: 766-770.
[2] Xu X. From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*. 2012; 28(1): 75-86.
[3] Xia F, Yang LT, Wang L, et al. Internet of Things. *International Journal of Communication Systems*. 2012; 25(9): 1101-1102.
[4] Menth M, Hartmann M, Klein D. Global locator, local locator, and identifier split (GLI-split). *Future Internet*. 2013; 5(1): 67-94.
[5] Liu Y. Development of network convergence and future internet. *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*. 2010; 22(6): 693-697.
[6] Lederer S, Mueller C, Rainer B, et al. *Adaptive streaming over content centric networks in mobile networks using multiple links*. Proceedings of IEEE Conference on Communications Workshops (ICC). Budapest. 2013: 677-681.
[7] Martinez-Julia P, Skarmeta AF. Beyond the separation of identifier and locator: building an identity-based overlay network architecture for the future Internet. *Computer Networks*. 2013; 57(10): 2280-2300.
[8] Rui T. The study of the architecture and key mechanism in the identifier and locator split network.

PhD thesis. Changsha: National University of Defense Technology. 2009.

[9] Shen H, Li Z, Li J. A DHT-Aided Chunk-Driven Overlay for Scalable and Efficient Peer-to-Peer Live Streaming. *IEEE Transactions on Parallel and Distributed Systems*. 2013; 24(11): 2125-2137.

[10] Lan M. Structured P2P algorithm with fuzzy query. *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*. 2013; 25(5): 680-685.

[11] Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network. *ACM*. 2001; 31(4): 161-172.

[12] Rowstron A, Druschel P. *Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems*. Proceedings of IFIP/ACM Conference on Distributed Systems Platforms. Heidelberg. 2001: 329-350.

[13] Ling H, String J, Rhea SC, et al. Tapestry: a resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*. 2004; 22(1): 41-53.

[14] Chen G, Wu G. G-Chord: an improved routing algorithm for Chord. *Journal of Southeast University (Natural Science Edition)*. 2007; 37(1): 9-12.

[15] Wu LL, Luesukprasert L, Lee L. *Research and the long tail: A large-scale citation analysis*. Proceedings of IEEE Conference on System Science (HICSS). Big Islang. 2009: 1-10.

[16] Fu Y, Wu X, Ye Q, et al. An approach for information systems security risk assessment on fuzzy set and entropy-weight. Acta E-l*ectronica Sinica*. 2010; 38(7): 1489-1494.

[17] Gao G. The study of Distributed Cache mechanism in P2P network. *PhD thesis*. Wuhan: Huazhong University of Science and Technology. 2011.