A Construction Method of Efficient SkipGraph Using The Performance of Peers in Heterogeneous Environment

Yohei Yasutomo*, Yoshitaka Nakamura** , and Osamu Takahashi**

*Graduate School of Systems Information Science, Future University Hakodate, Japan
**School of Systems Information Science, Future University Hakodate, Japan
{g2113033, y-nakamr, osamu}@fun.ac.jp

Abstract - SkipGraph is an overlay network that was applied to the data structure of SkipList in a peer to peer (P2P) network. Conventional SkipGraph does not take into account communication or the environmental performance of peers and uniformly treats all peers. However, communication and environmental performance differ for individual peers in real environments, and in some cases search efficiency deteriorates depending on the configuration of the topology of SkipGraph. We propose a method of constructing SkipGraph where the transfer delays between peers in small enough. In this method, peers are classified into three types by taking their processing speeds and communication speeds into consideration. We also evaluated the performance of the method of construction.

Keywords: P2P, Structure overlay, SkipGraph.

1 INTRODUCTION

Peer to Peer (P2P) network is network architecture. P2P networks are constructed from individual nodes (such as terminals) without servers, in contrast to the server-client model, and are superior in fault tolerance, scalability, and load distributions. All terminals of P2P network constitute an overlay network in cooperation with each other. These terminals search for contents data and transfer them on this overlay network. Many methods of constructing overlay networks have been proposed. The distributed hash table (DHT) [1] and SkipGraph[2] are typical methods. DHT is a method of effective searching data by managing the keys of data mapped to the same space with peers by hash function between multiple distributed peers. However, it is difficult to carry out range queries with DHT because the order of keys collapses due to the hash function. SkipGraph is an overlay network that applies SkipList [3] to P2P. SkipList is a forward linked list type data structure constructed by a probabilistic algorithm. It is easy to carry out range queries with SkipGraph because it does not treat hash values.

The opportunity to obtain services using P2P technology is increasing for mobile users, because mobile terminals such as smart phones, tablet terminals, and wireless communication technologies such as 3G or Wi-Fi have been developed. However, there are differences in terminal processing performance and communication environments between mobile terminals and fixed terminals. Mobile terminals generally perform worse than fixed terminals, and wireless communication causes long transmission delays. Therefore, mobile terminals have adverse effects in searches of the whole P2P network, when many mobile terminals become relay nodes.

Some researches have proposed solutions to this problem caused by differences in communication environments and peer processing performance. For example, Ref.[4] proposed asymmetry type P2P technology to perform network construction and data transmission processing in consideration of the characteristic of the terminal, and Ref.[5] proposed hierarchical P2P technology using the super node. These researches mainly use DHT as P2P construction method, but much less use SkipGraph.

There are differences in searching costs between peers in SkipGraph due to the participating positions of peers. P2P network with low search efficiency is constructed when terminals with low processing performance and poor communication environment participates in the position taking many search processing, and when terminals with high processing performance and good communication environment participates in the position taking few search processing. In this way, there are many problems to apply SkipGraph in real communication environment. We propose a method to construct efficient SkipGraph by giving priority to the terminals with small delay to take many search processing and limiting the terminals with large delay to take few search processing. We assumed a P2P network where various communication environment and terminal processing performance are mixed and classified terminals into three types by using these characteristics.

2 RELATED WORKS

2.1 SkipGraph

SkipGraph is an overlay network that applies SkipList to P2P. The structure of SkipGraph is outlined in Fig.1. SkipGraph has a number of hierarchies called “level”. “Level” is expressed as a number in the squares in Fig.1. Each peer is expressed with a square in Fig.1, and the number in the square expresses the key of data which the peer holds. This key plays a role as the node ID, and peers from a line in order of a key. The peer of SkipGraph has a bidirectional link in each level. It is determined which peers are linked by “Membership vector” which are the random binary digits. “Membership vector” are three digit numbers under the peers in Fig.1. The peers whose *n* digits prefix of Membership vector is match each other link in Level *n*. The set of all peers linking each other is called “List”. The highest level linked to peers was called “Highest level”. And all peers in Level 0 are linked to peers in ascending order.

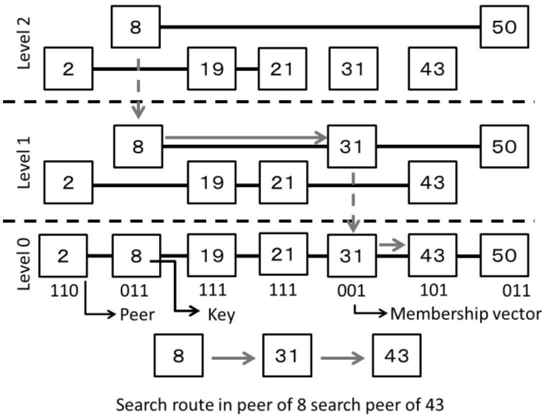

Figure 1: SkipGraph

2.1.1. Searching Process of SkipGraph

A peer (Peer S) in SkipGraph starts to search for a target peer (Peer D) from the highest level. The peer which received a search message compares its own key to the search key. If its key equates to the search key, it sends a completed search message to the Peer S. If the search key is larger than its own key, the peer starts to search other peer with lower key than the search key in the same level. If the peer is not able to find the target peer, it searches for it on the next below level. We explain the search method of SkipGraph with Fig.1. In this example, it is assumes that the peer with the key of 8 (Peer 8) searches for the peer with the key of 43 (Peer 43). First, the Peer 8 searches for the target Peer 43 on Level 2 that is the highest level. The peer searches for a peer having key between 8 and 43 because the target peer has the key of 43. Because Peer 8 cannot find the peer satisfying the search conditions on Level 2, Peer 8 lowers one hierarchy of SkipGraph and searches again on Level 1. The peer on Level 1 is able to find the peer with the key of 31 which is between 8 and 43, and forwards the search message to Peer 31. Peer 31 which received the search message starts to search the target Peer 43. Peer 31 lowers one hierarchy of SkipGraph and searches again on Level 0, because Peer 31 and 43 are on Level 1. And target Peer 43 is found on Level 0. Peer 31 sends the search message to target Peer 43 and completes the search process. The average number of hops which is necessary until search completion is log N (the number of all peers is N). SkipGraph streamlines searches because its topology is able to forward messages to distant peers on high levels.

2.1.2 Join and Leave Process of SkipGraph

A joining peer sends a message to an existing peer (agency peer) to inform its joining. The agency peer finds neighboring peers of the joining peer by using its key on Level 0, and inform the joining peer about the neighboring peers of the joining peer. After that, the joining peer sends the own membership vector to the neighboring peers on Level 0. Next the joining peer searches the neighboring peers on Level 1. Repeating this process on each hierarchy higher than Level 1, the joining peer knows the neighboring peers on each hierarchy. The average number of the messages becomes log N at the time of peer participation (N is the number of all peers). A leaving peer in SkipGraph sends messages to neighboring peers to inform its leaving from the highest level to the lowest level of SkipGraph. The neighboring peers reconstruct a topology with the messages. The average number of the messages becomes log N at the time of peer leaving.

2.1.3 Extension of SkipGraph

The research trend in SkipGraph is to extend multidimensional range searches [6] or to share multiple keys with peers [7].

Reference [8] proposed a method of construction with proximity in SkipGraph. Conventional SkipGraph may have links with too much delay to communicate because it does not assume physical localization or communication time between peers. To solve these problems, this method constructs SkipGraph with small delay by measuring the communication rate between peers.

2.2 Problems with SkipGraph

In the real communication environment, peers of P2P network are categorized into three types, such as long transfer delay, medium transfer delay, and short transfer delay according to the communication speeds and the processing speeds. Forwarding frequencies of peers in SkipGraph are changed by the position of the peer in the topology. We investigated the number of forwarding of peers at the each highest level when peers search another peers in the SkipGraph constructed by 4000 peers. Figure 2 plots relative values of the number of forwarding in every highest level with Level 12 being 100%. It is a standard at Level 12 because the average of highest level is Level 12 when 4000 peers construct SkipGraph. In Fig.2, we can see that there is 15% of difference with the rate for the number of forwarding of messages between peers under Level 10 and those over Level 17. In brief, the forwarding time increases as the highest level becomes higher. Forwarding messages on high levels in SkipGraph are more than low levels because searches in SkipGraph start from the highest level. Therefore, the forwarding efficiency in higher Level becomes important in search processing. When the peer which needs long time for message forwarding locates on higher highest level, the SkipGraph is inefficiency. However, when the peer with short transfer time is on higher highest level, the SkipGraph is efficient topology.

However, many peers in Fig.2 that have long transfer delay may be located on Level 16 or 17 because the membership vector deciding level is given at random. The position of the peers with long transfer delay influences the efficiency of the topology. Figure 3 is a specific example of construction of the inefficient topology. When Peer A and Peer B search for Peer C in this topology, the forwarding of messages goes through two low performance peers and increases delay. Therefore, this topology increases delay in searching.

Another problem is that Peer D only forwards messages to proxemics peers despite that there are other peers with lower delay.

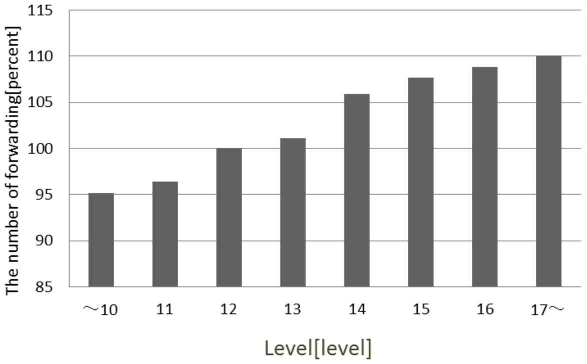

Figure 2: Number of forwarding for each Level

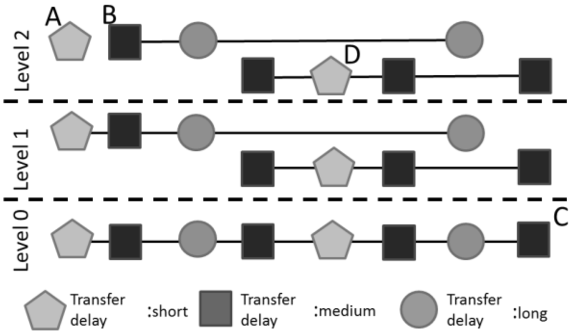

Figure 3: Example of inefficient SkipGraph

3 PROPOSED METHOD

3.1 Basic Concept

Our proposed method classifies peers joining SkipGraph in following three types from the viewpoint of transfer delay.

High Performance Peers A high performance peer is a terminal such as a server and a terminal located on a backbone network. It can forward large volumes of data with wider bandwidth than that of other peers.

Medium Performance Peers A medium performance peer is connected to a network with a general fixed line. Therefore, it is possible to communicate stably.

Low Performance Peers A low performance peer uses wireless communications such as Wi-Fi or 3G. It has slower communication speeds than medium performance peers. A peer using 3G has much more transmission delay than other peers. This type of peer is usually a mobile phone, smart phone or tablet terminal. Its IP address is frequently changing because of switching access points by the movement of the terminal. Therefore, its communication environment is unsteady.

The proposed method constructs topology such as Fig. 4. There are three types of peers in the topology. High performance and medium performance peers account for high forwarding rates at higher levels. To achieve this purpose, the proposed method applies joining and reconstruction method to each three types of peers in SkipGraph. In the joining methods of high and low performance peers, the joining peer find the highest levels that are linked to neighboring peers by the number of all peers. The average of these levels are called ``average levels'' in this paper. And the proposed method sets the membership vector so that high performance peers are located on higher than the average level and low performance peers are located on lower than the average level. Most search messages were assumed by high performance peers in this way. The proposed method decreased the number of search messages sent by low performance peers. Therefore, the proposed method can construct efficient SkipGraph.

The proposed method assumes that peers manage the number of peers participating target SkipGraph to find the average level. We explain the flow of joining and reconstruction methods with high and low performance peers in next subsection. Joining and reconstruction methods of medium peers are omitted from the explanation because a general method used in SkipGraph is applicable.

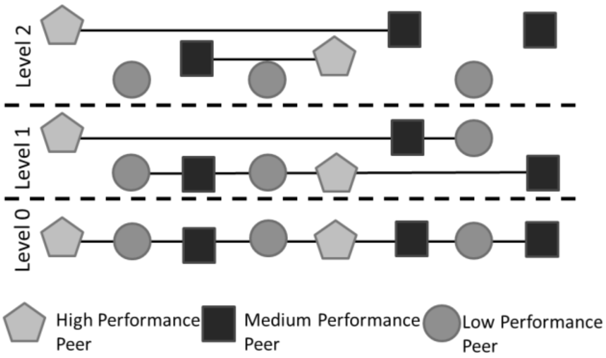

Figure 4: SkipGraph using the proposed method

3.2 Flow of the Proposed Method

Joining peers just get the key of the neighboring peer in level 0, the information about the average level of peers and key of the managing peers from agency peers when joining peers join SkipGraph. All types of joining peer joins SkipGraph with the proposed joining method. Agency peers sent joining messages about joining peers to the managing peer. The managing peer just calculates the average level with the number of peers. And the managing peer informs all peers on the recent average level when the average level is changed. High and low performance peers receive messages reconstruct the topology using the proposed method with the recent average level.

The average level can be calculated by log$N$ when the number of all peers is N. The managing peer informs all peers of the average level which is log$N$ truncated by a decimal point when the recent average level is below the last notified average level. The reason for this is that we could avoid frequent changes in the average level. For example, if the average level is a rounded value, the managing peer has to inform all peers of the average level, whenever the average level frequently keeps changing between three or four when log$N$ ranged in the neighborhood of 3.5.

3.3 Joining and Reconstruction Process of High Performance Peers

First, high performance peers use the general method of participation. When the highest levels of high performance peers is under the average level, they reconstruct the topology, and the highest level of high performance peers are over the average level. In this way, high performance peers can often forward messages at higher levels than other types of peers. We can also shorten the processing time for searches.

The highest level of high performance peers is Level i. High performance peers in Level i send messages to all peers belonging to the same list on Level i and investigate membership vectors of these peers. High performance peers compare these membership vectors with the membership vector of high performance peers that is inverted i +1 digits of prefix. And the high performance peer calculates the new highest level. When the calculated new highest level is higher than the average level, high performance peers reconstruct the topology using the inverted new membership vector. If the highest level does not attain the average level, they send messages to the all peers of the list on the next level below. They investigate the membership vectors and calculate the highest level again. In reconstructing the topology, high performance peers leave from the same level as the inverted digits. After that, high performance peer renew the membership vector, and join to the level using the new membership vector by the general method.

We explain the proposed process to achieve high performance peer (Peer 19) by using Fig.5. The Peer 19 reconstructs the topology in order to set over the highest Level 2 because the highest level of the Peer 19 is 1 while the average level is 2. The Peer 19 sends messages to the peers belonging to the list in level 1 that is highest level. The Peer 19 compares these membership vectors with the Peer 19's new membership vector that is ``110'' and calculates the highest level. Therefore, the Peer 19 understands the new highest level (Level 2) is over the average level. The Peer 19 sends the neighboring peers the renewed membership vectors ``110'' and commonly joins SkipGraph. Finally, the highest level of Peer 19 becomes 2, which is over the average level.

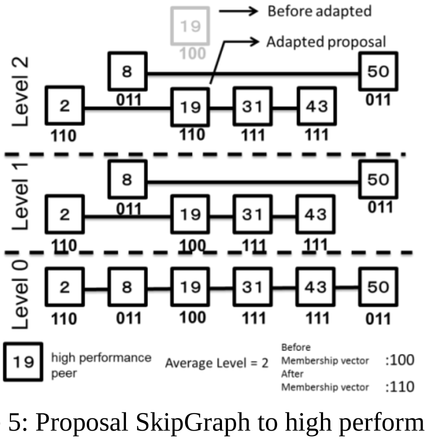

Figure 5: Proposal SkipGraph to high performance peer

3.4 Joining and Reconstruction Process of Low Performance Peers

Forwarding by low performance peers on higher levels is limited by setting the highest level of low performance peers under the average level. We can control the increase of time that occurs when searching with low performance peers on high levels.

When low performance peers join SkipGraph, they set the upper limit of the highest level (limit level) by using the average level. Limit level is calculated as follows.

$$\text{Limit Level} = \text{Average Level} - k \qquad (1)$$

k is one fixed value. We mask the higher digits of membership vector over the digit of the limit level. Low performance peers join SkipGraph with the masked membership vector. If the average level is increased after joining of low performance peers to SkipGraph, they only clear the masks of rising levels. Low performance peers participate in SkipGraph from the current highest level with the renewed membership vector again. If the average level is decreased, the low performance peers mask their membership vector and leaves to prevent the limit level from decreasing.

We explain the method of participation by low performance peers using Fig.6, where the low performance peers of Peer 19 mask the prefixed one digits of the membership vector before joining the topology. In this case, k is 2. The limit level becomes Level 1 by calculated by expression (1). Therefore, the low performance peers mask 2 digits of the membership vector. The low performance peers join with the new membership vector, which is one digit because of masking of two digits. Therefore, the low performance peers can fix the highest level at Level 1, which is lower than the average level while the highest level of low performance peers is Level 2, which is achieved with the general method.

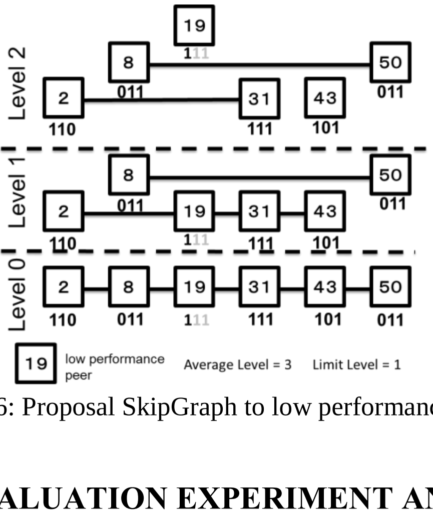

Figure 6: Proposal SkipGraph to low performance peer

4 EVALUATION EXPERIMENT AND CONSIDERATIONS

We conducted the simulation experiment to evaluate the efficiency of the proposed method using PIAX[9]. The proposed method deal with high and low performance peers. Therefore, we evaluated high performance peer, low performance peer and a combination of low and high performance peers.

4.1 Evaluation of High Performance Peers

We explain the experiment to evaluate the efficiency of the proposed method with high performance peers. The number of peers in this experiment ranged from 500 to 4000. The number of high performance peers accounted for 1/3 of the whole peers. The keys of peers are random values from zero to the number of all peers.

Peers select the keys with random values and perform range search in the range 0 to 3. We conducted 50 times of the experiment as 1 trial. And, we measured the peers' number of forwarding until search completion. The average number of forwarding of the high performance and that of other peers using the proposed method for 10 trials are outlined in Fig.7. Figure 7 indicates the number of forwarding of high performance peers are larger than that of other peers regardless of the number of all peers. That means high performance peers are able to forward messages in advance. The reason for this is that the highest level of the high performance peers located over the average level remains stable.

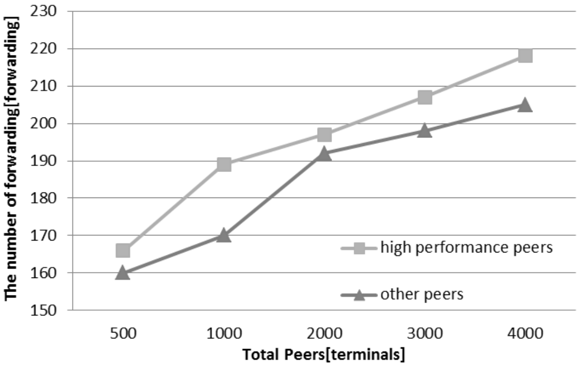

Figure 7: Number of forwarding using the proposed method for high performance peers

4.2 Evaluation and Consideration of Low Performance Peers

We explain an experiment to evaluate the performance of the proposed method with low performance peers. The experiment's setup and evaluation items are similar to those described in Subsection 4.1. The limit level in the experiment is 2 levels smaller than the average level. Figure 8 plots the average number of forwarding.

When there are more than 3000 peers, the number of forwarding of low performance peers decreases. That indicates the forwarding of messages of low performance peers is limited by the proposed method. However, when there are fewer peers than 2000 peers, the number of forwarding of the low performance peers are more than that of the other peers. The reason for this is that the number of peers on high levels is sparse because this experiment apply the proposed method to only low performance peers. Therefore, the search messages are not forwarded on high levels which can send message to peers with keys but forwarded on low levels. And the number of forwarding until search completion is increased because the forwarding on low levels is mainly sent messages to peers having near keys. When there are more than 3000 peers, the number of forwarding times of low performance peers is lower than that of the other peers. The reason for this is that the average level is difficult to shift with many peers because the average level is calculated by log*N*, and the forwarding messages on high levels are more than the case with fewer peers because the density of peers was higher on high levels.

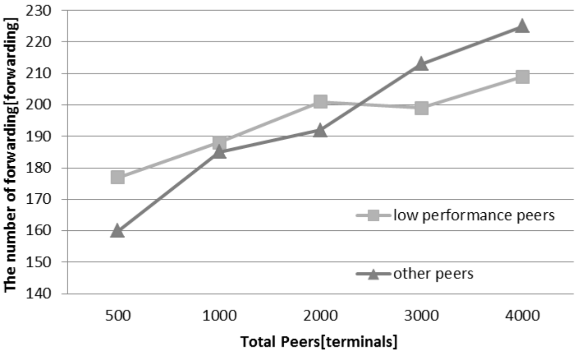

Figure 8: Number of forwarding using proposed method for low performance peers

4.3 Evaluation and Consideration of High and Low Performance Peers

We explain an experiment we carried out to evaluate the proposed method for the combination of low and high performance peers. The experimental setup and evaluation items are the same as those described in Subsection 4.1. In addition, we also evaluated searching hops. Figures 9 and 10 plot the results of the measurements.

Figure 9 indicates that when there are 2000 peers, the forwarding messages by high performance peers are more than that by the other peers, and the forwarding message by low performance peers are less than that by the other peers. Especially when there are 3000 peers, the number of forwarding of high performance peers are more than 10% of that of low performance peers. The reason for this is that the topology constructed by the proposed method raises high performance peers over the average level and low performance peers below the average level remain stable. However, we can find a problem in which forwarding by low performance peers is greater than that by medium peers when there are fewer peers than 2000. The reason for this is similar to the reason given in Subsection 4.2.

When the number of peers is from 500 to 2000 in Fig.10, there is a difference in hops between the general method and the proposed method. However, this is not a large difference and the number of hops to search by the proposed method is similar to that by the general method.

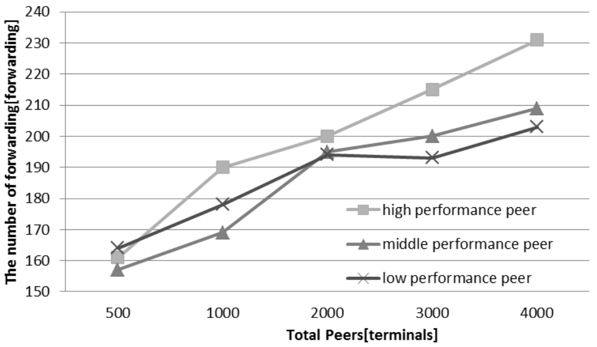

Figure 9: Number of forwarding using the proposed method for low and high performance peers

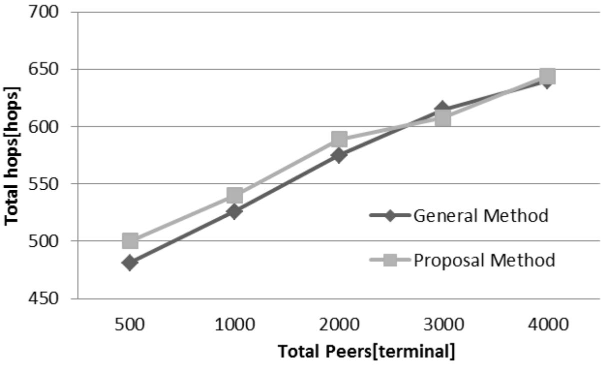

Figure 10: Number of hops using the proposed method

5 CONCLUSION

We considered communication speeds and the processing speeds of terminals and classified terminals into high, medium, and low performance peers. Our proposed method used different participation and reconstruction methods for each terminal to join SkipGraph according to this classification. And we proposed the method to let the terminals with the good communication environment such as high performance peers perform search processing more with precedence, and to limit search processing to the terminals with unstable communication such as low performance peers. We evaluated the efficiency of the proposed method by simulation experiment in varied communication environments. From the results of experiments, the proposed method is able to construct the efficient topology and to fix high performance peers over an average level and low performance peers under an average level by using limit level for all peers.

In the future work, it is necessary to examine the cases that mixture rate of terminals is an inclination to one of the terminal types and is real environment. Our current method decides the highest level of a certain peer based on the average level. For example, when there are too many low performance peers in all peers, too many terminals are located at low levels and very few peers are located at high level. The simulated experiments on other problems produced results in which the proposed approach was not efficient for a few peers.

In addition, low performance peers were assumed to be mobile terminals, which may be defective, have irregularities with SkipGraph, and cause many reconstructions because mobile terminals suffer from the unstable nature of electric waves. We also intend to reevaluate the proposed SkipGraph by addressing these problems.

REFERENCES

[1] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S., ``A scalable content-addressable network," Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01), pp.161-172 (2001).

[2] Aspnes, J., and Shah, G, ``Skip Graphs," ACM Transactions on Algorithms, Vol.3, No.4, pp.37:1-37:25 (2007).

[3] Pugh, W, ``Skip lists: a probabilistic alternative to balanced trees," Communications of the ACM, Vol.33, No.6, pp.668-676 (1990).

[4] Tagami, A., and Ano, S. ``Design and Implement of AsymmetricP2P Network with Fixed Mobile Convergence Network, "Proceedings of the 2011 Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO2011), pp.121-126 (2011).(in Japanese)

[5] Garces-Erice, L., Biersack, E.W., Ross, K.W., Felber, P.A., and Urvoy-Keller, G., ``Hierarchical P2P Systems," Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing, pp.643-657 (2003).

[6] Zhang,C., Krishnamurthy, A., and Wang, R.Y., ``SkipIndex: Towards a Scalable Peer-to-Peer Index Service for High Dimensional Data," Technical Report TR-703-04, Technical Report, Department of Computer Science, Princeton University (2004).

[7] Konishi, Y., Yoshida, M., Takeuchi, S., Teranishi, Y., Harumoto, K., and Shimojo, S., ``An Extension of Skip Graph to Store Multiple Keys on Single Node," IPSJ Journal, Vol.49, No.9, pp3223-3233 (2008).

[8] Makikawa, F.; Tsuchiya, T.; Kikuno, T., "Balance and Proximity-Aware Skip Graph Construction," Networking and Computing (ICNC), 2010 First International Conference on , vol., no., pp.268,271, 17-19 Nov. 2010

[9] PIAX: http://www.piax.org