IEICE TRANS. INF. & SYST., VOL.E93–D, NO.6 JUNE 2010

PAPER

Constructing Overlay Networks with Short Paths and Low Communication Cost

Fuminori MAKIKAWA[†a)], ***Nonmember***, **Tatsuhiro TSUCHIYA**[†], ***Member***, ***and*** **Tohru KIKUNO**[†], ***Fellow***

SUMMARY A Peer-To-Peer (P2P) application uses an overlay network which is a virtual network constructed over the physical network. Traditional overlay construction methods do not take physical location of nodes into consideration, resulting in a large amount of redundant traffic. Some proximity-aware construction methods have been proposed to address this problem. These methods typically connect nearby nodes in the physical network. However, as the number of nodes increases, the path length of a route between two distant nodes rapidly increases. To alleviate this problem, we propose a technique which can be incorporated in existing overlay construction methods. The idea behind this technique is to employ long links to directly connect distant nodes. Through simulation experiments, we show that using our proposed technique, networks can achieve small path length and low communication cost while maintaining high resiliency to failures.

key words: *P2P application, overlay network, proximity, distributed algorithm*

1. Introduction

A Peer-To-Peer (P2P) application uses an overlay network which is a virtual network constructed over the physical network. As today's P2P applications comprise of a large number of nodes, it is increasingly important to construct communication-efficient overlay networks.

One of the most important properties that an overlay should have is short path length. By path length, we mean the number of overlay links in a path between two nodes. Clearly path length should be short to reduce the number of relay nodes that have to forward a message, and in turn to achieve efficient communication.

Geographical proximity between nodes is also an important feature to consider. The proximity between two nodes is usually expressed by the distance in the physical network between them. If no care is taken to reflect proximity in overlay construction, then a large amount of redundant traffic is produced, resulting in inefficient communication and performance degradation of the P2P application. This paper aims to address these two issues in overlay construction.

To construct a proximity-aware overlay network, existing approaches typically establish overlay links between two nodes which are nearby in the physical network [9], [17]. By shortening all links, the average communication cost between a pair of nodes can be reduced. This approach works well for structured overlay networks, because path length is usually bounded by a small value determined by the number of nodes [1], [4], [10], [12]. On the other hand, for unstructured overlay networks, this approach has the problem of increasing the path length between two physically distant nodes.

To address this problem with proximity-aware unstructured overlay networks, we propose a technique for constructing overlay networks that have short path length and reflect proximity of nodes simultaneously.

Our proposed method modifies the existing conventional approach. To reflect geographical proximity, the conventional approach uses short links between two physically nearby nodes. In addition to these short links, our method employs some long links which connect two long-distance nodes in the physical network. These long links significantly reduce the path length between distant nodes, thus solving the problem with the existing approach. Although using a long link means a large communication cost, we will show later that this does not harm communication efficiency on average, because most links are short links connecting nearby nodes.

The idea of using long links is not new. For example, the similarity with the well-known Watts-Strogatz model for small-world networks is clear. However, the proposed method has several features that distinguish it from the previous work: (1) Long links are associated with a certain distance and are established such that the difference between that value and the actual distance is minimized. In contrast, existing methods typically use random links. (2) The number of long links is controlled by a predetermined parameter. The parameter determines the ratio of long links to all links. This is in contrast to, for example, the GoCast protocol [13], in which a node has basically one random link. (3) Our proposed method can be incorporated into many existing proximity-aware overlay construction methods. The first two features allow more control over the overlay topology than existing methods. By comparing our method with GoCast through simulations, we will quantitatively show this benefit.

2. Related Work

There are several methods for constructing proximity-aware overlay networks. LOCALISER [9] is an algorithm that iteratively change link connections to reflect the proximity

Manuscript received May 25, 2009.
Manuscript revised January 22, 2010.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.
a) E-mail: f-makikawa@ist.osaka-u.ac.jp

DOI: 10.1587/transinf.E93.D.1540

of neighbor nodes. In this algorithm, physically close node pairs will have a high chance to have a mutual link. This algorithm is robust to churn, since it allows continuous topological changes. LOCALISER also has a mechanism to uniformly distribute node degree while keeping the same number of links. This mechanism provides a high resilience to node failures.

The mOverlay algorithm [17] organizes a proximity-aware overlay network in a two-level hierarchy. In this structure, nodes compose some groups and the groups compose a network. The nodes in a group have links with each other; therefore, the overlay is very resilient to the failures of nodes.

The LTM algorithm proposed in [7] also reflects proximity in constructing an overlay network. In this algorithm, each node repeatedly cuts links with high cost and creates connections with nearby nodes.

Our proposed method can be incorporated into existing proximity-aware methods. In this paper, we extend the LOCALISER and the mOverlay algorithms by adopting the proposed method.

Some methods take one step further; they consider not only proximity but also other properties. GoCast [13] is one such method. In GoCast, most nodes have exactly one random link. All other links are chosen based on proximity.

The topology aware gossip overlay [5] uses a similar approach to GoCast. In this overlay, a node maintains two lists of links: one containing links to current neighbor nodes and the other containing those to random nodes. The former link list contains some random links and some short links. These links are used for normal communications. The links in the latter list are used as a fallback when all neighbor nodes fail.

In the Foreseer architecture [2], each node uses both proximity-aware links and friend links, aimed at improving search efficiency.

These methods construct overlay networks with proximity and small path length. From these methods, we select the GoCast method to compare with our proposed method.

3. Overlay Networks

The terms and symbols used in this paper are summarized as follows:

- We call a network constructed from routers and cables a physical network and a virtual network constructed from end nodes (nodes for short) connecting to the physical network an overlay network.
- In an overlay network, a link connects two nodes. In reality, a link could be a TCP connection or could represent that the two nodes know each other's address.
- When two nodes are connected by a link, we say that these nodes are neighboring, and that one of the nodes is the neighbor node of the other. The degree of a node means the number of its neighbors. We denote the degree of node i as d_i.
- Any pair of two different nodes i, j is associated with link cost $c(i, j) > 0$ which is independent of the topology of the overlay. Intuitively $c(i, j)$ represents the communication delay required by the direct message transfer from i to j. We assume that $c(i, j) = c(j, i)$ for any nodes i, j.

We use four performance measures to evaluate overlay network topologies.

- **Path length:** The length of a path in an overlay is the number of links in the path. The path length for two nodes is defined as the path length for the shortest path between them. In this paper, we use the average path length for all nodes pairs as a performance measure for an overlay network.
- **Communication Cost:** The communication cost of a path is the sum of the cost of the links of the path. We define the communication cost between two nodes as the communication cost of the shortest path between them.
- **Clustering Coefficient:** This measure quantifies how close a given node and its neighbors are to being a clique [14]. Note that a total of $d_i * (d_i - 1)$ node pairs can be selected from the neighbors of a node i. The clustering coefficient of a node i is defined as the ratio of neighboring node pairs in those $d_i*(d_i-1)$ node pairs. The clustering coefficient of the whole network is the average of all nodes. Clustering coefficient should be small, since areas of the overlay that exhibit a high clustering coefficient are easily disconnected by node failures.
- **Reachability:** Given a set of failed nodes, we define reachability as the ratio of nodes in the largest fragment of the network to the nodes that have not crashed. For example, suppose that the total number of nodes is $10,000$ and the failure ratio is 0.2. Then the number of correct nodes is $8,000$. Now suppose that the network is partitioned into some fragments because of the node failures and that 400 correct nodes cannot be reached from the largest fragment of the network. In this case reachability is $\frac{7600}{8000} = 0.95$.

4. Existing Proximity-Aware Methods

4.1 LOCALISER

LOCALISER [9] is a fully decentralized algorithm that iteratively reshapes the topology of an overlay network. In this algorithm, each node repeatedly replaces its own links with shorter ones. As a result, the overlay network gradually becomes close to the physical network.

The following steps show how a node, say i, replaces its links. w and T are parameters. (Fig. 1 schematically shows these steps).

1. Choose two of its neighbors, node j and node k at random, and measure the link cost $c(i, j)$ and $c(i, k)$.

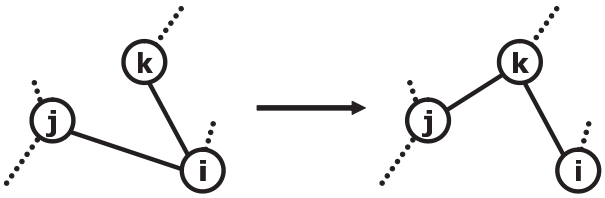


Fig. 1 LOCALISER algorithm.

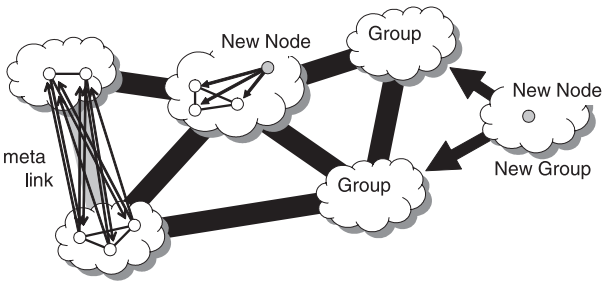


Fig. 2 mOverlay network ($M = 3$).

2. Send messages to node j and node k, which send back respectively d_j, d_k. In addition, node j sends back its estimate of $c(j,k)$.
3. Evaluate locally the cost of replacing link (i,j) with link (j,k). The cost is defined as $\Delta E = 2w(d_k - d_i + 1) + c(j,k) - c(i,j)$.
4. Perform the replacement of the links with probability $p = \min\left(1, \left(e^{-\Delta E/T}\frac{d_i(d_i-1)}{d_k(d_k+1)}\right)\right)$.

4.2 mOverlay

The mOverlay algorithm [17] constructs a two-level hierarchical network. The top level consists of groups of nodes, while the bottom level consists of nodes within each group. Figure 2 schematically represents an mOverlay network.

Each group consists of nearby nodes, which have links with each other. At the top level, a group has some "meta" links to its nearby groups. A meta link is implemented by a set of unidirectional links. If two groups are connected by a meta link, all nodes in either of the two groups have links to H nodes in the other group, where H is a design parameter. The two-level hierarchical structure can thus allow efficient communication.

When a node joins the network, it first searches for a nearby group. If a group that meets a criterion has been found, then the node joins the group. Otherwise, a new group is created and the new node joins the new group.

Creating a new group involves selecting M neighboring groups, where M is a design parameter. M meta links are added between the new group and these M groups. The selection of neighboring groups is conducted by invoking M times the following procedure for finding a nearby group.

1. Consult with a special server, called the rendezvous point, to obtain the address of an existing node called a *boot host*.
2. Measure the distance C_G to the group, say G, of the boot host by measuring the average communication cost to all nodes in the group.
3. Let $\mathcal{G}$ be the set of the neighboring groups of G. Measure the distance to each group in $\mathcal{G}$.
4. If a stop criterion is met [†], then go to Step 5. Otherwise, set G to one group in $\mathcal{G}$ such that $C_G = \min_{G' \in \mathcal{G}}\{C_{G'}\}$ and go to Step 3.
5. Let $\mathcal{G}'$ be the set of all groups whose distance has been measured. Select one group G from $\mathcal{G}'$ such that $C_G = \min_{G' \in \mathcal{G}'}\{C_{G'}\}$.

The group selected by this procedure varies depending on the boot host. If the same group has been selected more than once, then the total number of nearby groups obtained becomes less than M. In that case, new groups are selected from the neighboring groups of these selected groups such that a total of M groups are eventually selected.

4.3 Problem with the Traditional Proximity-Aware Methods

Overlay networks constructed by these proximity-aware methods have the problem that physically distant nodes tend to have only long paths. This can cause significant messaging delay, because message forwarding entails non-negligible overhead [3], [6]. Also broadcasting is affected by this problem: In P2P application, broadcast is often used for many purposes, for example, to search for a node that has a required resource. Each broadcast message is attached the maximum number of hops that the message can go through in order to prevent it from traveling in the network forever. This number is usually called *Time-To-Live* (TTL). By this limitation, the message can be discarded before reaching the target node, if the path length between the initiator node of a broadcast and the target node is greater than the TTL.

4.4 GoCast

GoCast [13] is a method for constructing an overlay network with small path length and low communication cost. This method employs two types of links: short links and random links. Most nodes have exactly one random link. All other links are chosen based on proximity.

1. When a node joins the overlay network, the node selects some physically nearby nodes. Also it selects another node in a random manner. Then, the node establishes links between these selected nodes.
2. If each node recognizes that it has too more links than it should have, then the node cuts some links with high cost such that the change in its neighbors' node degree is as minimum as possible. On the other hand, if the node has less links, then it selects some nodes as in the joining step and creates links to them.

[†]In the simulations we conducted, we stopped the search when the next G would be the group that has already been selected as G.

5. Our Proposed Technique

The aforementioned problem occurs because the previous proximity-aware methods install links only between geographically nearby nodes. Our technique alleviates this problem by introducing some *long links* in the overlay network. The technique consists of two components: long link selection and objective cost assignment.

The selection of a long link is performed whenever a new link is added to the network for the first time. The new link is selected with probability P as a long link. The link is a short link, otherwise. P is a design parameter that needs to be decided a priori.

A new link is also associated with its *ideal cost* when it is added to the network. Now suppose that a new link is added to the network and that *ideal* represents its ideal cost. If the new link is a short link, then it is associated with $ideal = 0$. If the link is a long link, then it is associated with $ideal = GOAL \geq 0$, where $GOAL$ is another design parameter. The absolute difference between the ideal cost and the communication cost works as the objective cost in installing or replacing a link. That is, a link is installed or replaced in such a way that $|c(i, j) - ideal|$ is minimized where (i, j) is a newly added link.

This technique can be naturally incorporated into the existing algorithms as follows.

p-LOCALISER: By applying our proposed technique to the LOCALISER algorithm, we have the following new algorithm, which is different from the original one in that Step 3 uses a different cost function. We call the algorithm *p-LOCALISER*. Here node i is the initiator of the algorithm. Whether each link is a long or short link is determined when the original network is built.

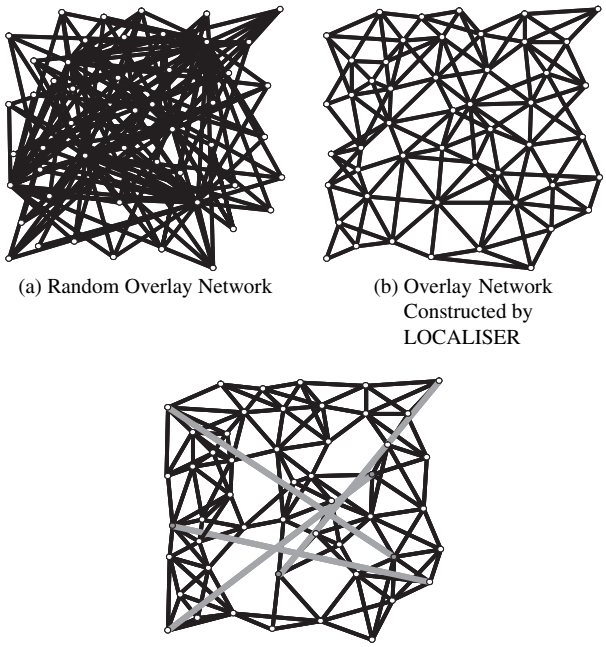
1. Choose two of its neighbors, say node j and node k, at random, and measure the link cost $c(i, j)$ and $c(i, k)$.
2. Send messages to node j and node k, which send back respectively d_j, d_k. In addition, node j sends back its estimate of $c(j, k)$.
3. Evaluate locally the cost of replacing link (i, j) with link (j, k). To reflect the ideal cost, the cost is now defined as $\Delta E = 2w(d_k - d_i + 1) + |c(j, k) - ideal| - |c(i, j) - ideal|$, where *ideal* is the ideal cost of the link currently connecting i and j.
4. Perform the link replacement with probability $p = \min\left(1, \left(e^{-\Delta E/T} \frac{d_i(d_i-1)}{d_k(d_k+1)}\right)\right)$. If the replacement happens, the new link (j, k) inherits the type (i.e., short or long) of the removed link (i, j).

Preliminary results for the case $GOAL = \infty$ can be found in our previous paper [8].

p-mOverlay: Our proposed technique can be naturally incorporated in the process of installing meta links in mOverlay. We refer to this new version of m-Overlay as *p-mOverlay*. In the original mOverlay, when a new group is created, it selects neighboring groups such that the distance to them is minimized. In contrast, the proposed technique modifies this selection process, by taking the ideal cost into account. As a result, meta links are selected as long links with probability P and are added to groups whose distance from the new group is close to the ideal cost. We let *ideal* denote the ideal cost of a meta link.

Incorporation of the proposed technique amounts to a slight change of the procedure for finding a nearby group. Specifically, Steps 4 and 5 are modified as follows:

4. If a stop criterion is met, then go to Step 5. Otherwise, set G to one group in $\mathcal{G}$ such that $C_G = \min_{G' \in \mathcal{G}}\{|C_{G'} - ideal|\}$ and go to Step 3.
5. Let $\mathcal{G}'$ be the set of all groups whose distance has been measured. Select one group G from $\mathcal{G}'$ such that $C_G = \min_{G' \in \mathcal{G}'}\{|C_{G'} - ideal|\}$.



(a) Random Overlay Network

(b) Overlay Network Constructed by LOCALISER

(c) Overlay Network Constructed by p-LOCALISER

Fig. 3 Networks constructed by LOCALISER.

5.1 An Illustrative Example

Here we describe an illustrative example. Figure 3 (a) shows a random overlay network with 50 nodes. Figures 3 (b) and 3 (c) show overlay networks obtained from the network in Fig. 3 (a) by executing the original LOCALISER algorithm and the p-LOCALISER algorithm. The black lines represent short links, while gray lines represent long links. The number of links is the same in Fig. 3 (a), Fig. 3 (b), and Fig. 3 (c).

6. Simulation Evaluation

In this section, we present the results of simulations. We compare the results of using and not using our proposed technique.

6.1 Simulation Settings

We used the George Tech transit-stub model [16] to create physical networks. Each physical network is composed of 100 transit domains, each of which has 100 stub domains. Link delays are modeled by simply assigning a propagation delay of around 10 *ms* to each physical link between two transit domains and 1 *ms* to the other physical links.

The link cost between two nodes in the overlay network is the communication cost of the shortest path in the physical network, where the communication cost in the physical network is the sum of the delays of the physical links of the path.

The transit domains are located in a 2-dimensional space. Each transit domain is connected to three other nearby transit domains on average. Each transit domain has 100 routers and each router has one stub domain. Every overlay node joins one of these stub domains. In these simulations, the average link cost between two nodes was about 50 *ms*, while the maximum delay was about 140 *ms*.

In Simulation 1 and Simulation 2, the average degree of nodes is set to 15 in random overlay networks, LOCALISER, p-LOCALISER, and GoCast. On the other hand, the average degree is lowered to 6 in Simulation 3 in order to clarify the difference between different methods, since the networks with high average degree result in near 100% reachability regardless of the construction method. We remark that in our preliminary experiments, we confirmed that different average degree does not change the qualitative properties of these methods. Unlike these algorithms, the mOverlay algorithm has a two-level hierarchical structure; thus its design parameters are set in a different way. We set its design parameters such that in the network created, each group had approximately 10 nodes on average and $M = 3$ meta links. Nodes have $H = 5$ unidirectional links in a meta link.

The behavior of the LOCALISER algorithm was simulated as follows. We first created a random overlay network. LOCALISER and p-LOCALISER were applied to the random overlay network. In each instance of the simulation, the replacements of links were executed 1000 times by each node. The parameters w and T are decided as $w = 20$, $T = 50$. With these values, the degree of almost all nodes is maintained from 12 to 18.

6.2 Simulation 1: The Parameter of our Proposed Technique

Simulation 1 was conducted to investigate the effects of the parameter values of our proposed technique. We tested LOCALISER and mOverlay equipped with our technique. In this simulation, we assumed that there were 10,000 nodes in the overlay network. No failure was considered.

We varied *GOAL*, the ideal cost for a long link, from 10 *ms* to 150 *ms*, and *P*, the probability that a new link becomes a long link, from 0% to 40%. Since the link cost between two nodes is at most 140 *ms*, when the value of *GOAL* is 150 *ms*, long links are placed in such a way that their distance is increased as much as possible. The original LOCALISER algorithm is equivalent to p-LOCALISER if *P* equals 0.

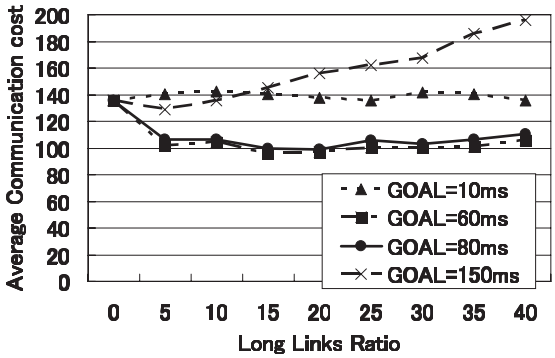


Fig. 4 Ratio of long links and average communication cost (p-LOCALISER).

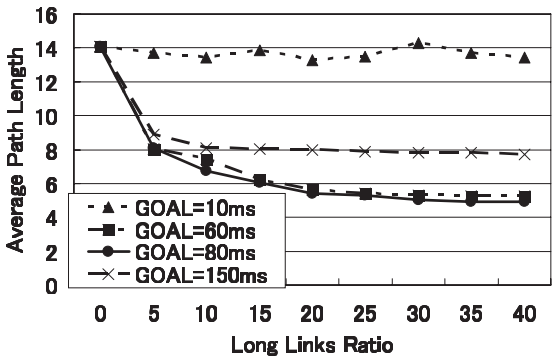


Fig. 5 Ratio of long links and average path length (p-LOCALISER).

Figure 4 shows the relations between the ratio of long links *P* and the average communication cost between any two nodes. The result for the case $GOAL = 150\,ms$ shows that the average communication cost rapidly increases as the value *P* increases. At the other extreme, when $GOAL = 10\,ms$, the average communication cost does not change clearly as the value of *P* varies. In contrast, when $GOAL = 60\,ms$ or $GOAL = 80\,ms$, the average communication cost is significantly reduced.

Figure 5 shows the relations between the value of *P* and the average path length. From the results, one can see that when $GOAL = 80\,ms$, path length is most reduced. The reduction is, however, saturated when *P* exceeds 20%.

Based on Figs. 4 and 5, we conclude that our proposed technique shows good performance with $GOAL = 80\,ms$ and $P = 20\%$ when applied to LOCALISER. This is also the case for mOverlay (Figs. 6 and 7). We therefore use these values in the following simulations.

6.3 Simulation 2: Comparison with Existing Methods

In this simulation, we evaluated the performance of LOCALISER and mOverlay with and without our technique.

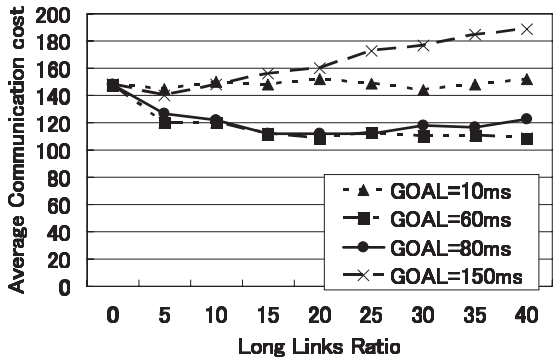


Fig. 6 Ratio of long links and average communication cost (p-mOverlay).

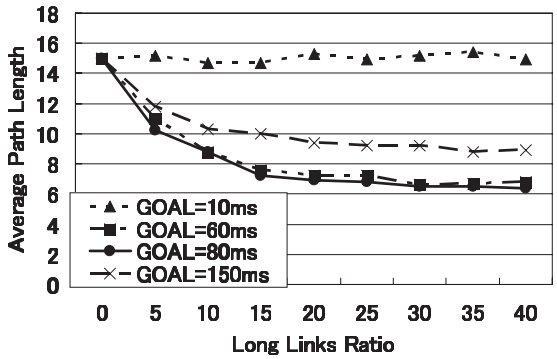


Fig. 7 Ratio of long links and average path length (p-mOverlay).

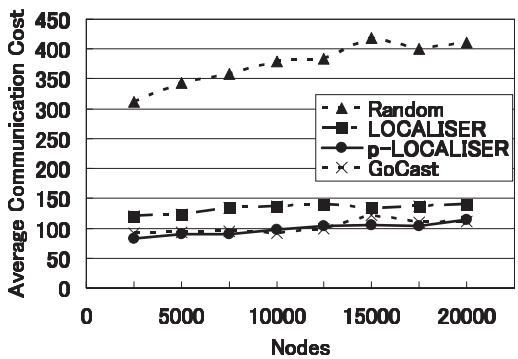


Fig. 8 Average communication cost (LOCALISER, p-LOCALISER).

We measured communication cost and path length by varying the network size: We varied the number of nodes from 2,500 to 20,000. Based on the results of Simulation 1, we set $GOAL = 80\,ms$ and $P = 20\%$.

6.3.1 Communication Cost

Figure 8 presents the results for LOCALISER with respect to the average communication cost. This figure compares GoCast, LOCALISER and p-LOCALISER, as well as the initial random network to which LOCALISER was applied. As shown in this figure, p-LOCALISER exhibits slightly lower communication cost than GoCast and LOCALISER.

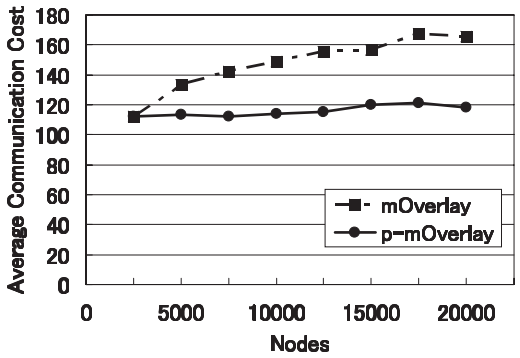


Fig. 9 Average communication cost (mOverlay, p-mOverlay).

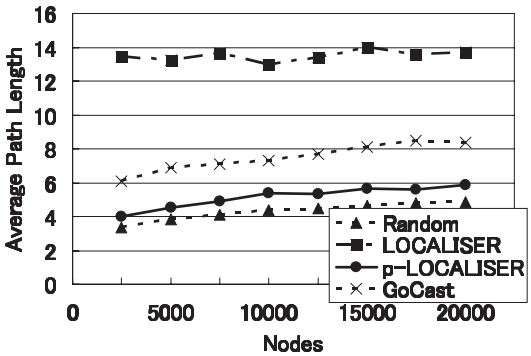


Fig. 10 Average path length (LOCALISER, p-LOCALISER).

Compared with the initial random network, on the other hand, p-LOCALISER achieves much low communication cost.

Figure 9 shows the result of using mOverlay. The benefits of using the proposed technique are much clearer in this case. In the network constructed by mOverlay, far distant nodes have to use meta links between different groups. These meta links are few in numbers and they only connects nearby groups in the original mOverlay algorithm. As a result, long meta links installed by the proposed technique have much more clear effects on the communication cost than in the case of LOCALISER which produces flat-structured overlays.

6.3.2 Path Length

Figure 10 shows the results on path length for LOCALISER. The results clearly show the benefit of using the proposed technique: With respect to average path length, p-LOCALISER achieves significantly lower values than GoCast and LOCALISER. As shown in Fig. 11, our technique also considerably reduces the path length for mOverlay. This reduction in path length greatly enhances the reachability of broadcast messages, because it prevents the messages from expiration of their TTL.

Moreover, with our proposed technique, the path length only moderately increases as the number of nodes increases. That is, our technique improves proximity-aware methods in that the constructed network can better scale to the network

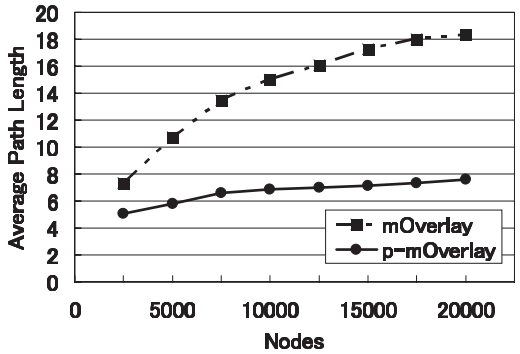


Fig. 11 Average path length (mOverlay, p-mOverlay).

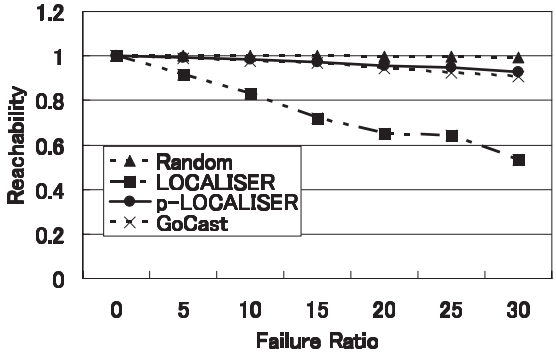


Fig. 12 Random failure rate and reachability (LOCALISER, p-LOCALISER).

Table 1 Clustering coefficient, path length and communication cost.

	Clustering	Path Length	Com Cost
Random	0.0023	4.3	379
LOCALISER	0.59	12.9	136
p-LOCALISER	0.36	5.4	98
GoCast	0.40	7.3	92

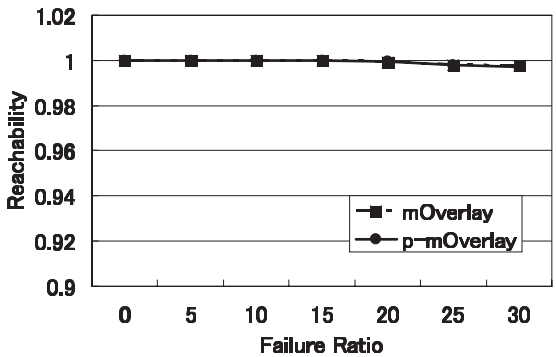


Fig. 13 Random failure rate and reachability (mOverlay, p-mOverlay).

size.

6.3.3 Clustering Coefficient

We compared the clustering coefficient between random, LOCALISER, p-LOCALISER, and GoCast when the number of nodes is 10000. Table 1 shows the clustering coefficient obtained by these four methods, as well as average path length and average communication cost.

The simulation results agree with the well-known fact that random networks have low average path and small clustering coefficient. Networks with high clustering coefficient and low average path length are called small-world networks [11]. The networks constructed by p-LOCALISER and GoCast exhibit small-worldness, while the LOCALISER network has large path length and thus does not exhibit this characteristic.

As stated in Sect. 3, a large clustering coefficient implies vulnerability to failures. Although p-LOCALISER produces networks with relatively high clustering coefficient, the value is significantly smaller than the LOCALISER networks. From this observation one may suppose that our method enhances the fault tolerance of the original LOCALISER algorithm. In the next set of simulations, we show that this is indeed the case.

6.4 Simulation 3: Resilience to Node Failures

In Simulation 3, we evaluate the resilience of the overlay networks to random node failures.

In this simulation, we assume random failures of nodes in the overlay network. In P2P applications, node failures occur not only because of node crashes but also of node joins and leaves. In such applications, users frequently join and leave from the network whenever they want. Nodes that have left from the network cannot be distinguished from those that have crashed. This means that it is usually the case that a very large fraction of nodes have failed simultaneously.

Figures 12 and 13 show the relations between the ratio of failed nodes and reachability. In Fig. 12, our proposed method exhibits much higher reachability than the LOCALISER method does. This reason is explained as follows. In the original LOCALISER, nodes in the constructed network only have links to its nearby nodes. Because of this, nearby nodes share most of their neighbors or the neighbors of their neighbors. As a result, the failure of a node affects many of its neighbors simultaneously, resulting in high probability of network partitioning. The long links added by the proposed technique decrease such probability, thus making the network more resilient to random failures. The high reachability that GoCast shows is also explained by the same argument, except that in GoCast high resiliency is resulted from using random links, instead of long links.

Figure 13 shows the results for mOverlay. mOverlay, by its design, achieves very high resilience to random node failures. All nodes in each group of mOverlay share mutual links and a meta link between different groups is shared by all pairs of nodes between the two groups. Because of this property, network partitioning does not occur unless all nodes in a group have failed.

7. Conclusion

We discussed an approach for constructing overlay networks where pairs of nodes have a small path length and low communication cost. We proposed a technique which installs long links in an overlay network. This technique can solve the problem with existing proximity-aware overlay construction methods, which only provide long paths to distant nodes. By incorporating the proposed technique into such methods, namely, LOCALISER and mOverlay, we demonstrated that the technique can be used in combination with existing overlay construction algorithms. Using simulations, we evaluated these algorithms with and without our proposed technique, as well as other overlay construction methods. The result showed that the proposed technique significantly reduces path length. In both cases of LOCALISER and m-Overlay, more than 50% reduction is usually achieved compared to the original algorithms. Even compared to GoCast, p-LOCALISER, that is, the LOCALISER algorithm incorporated with the proposed technique, achieves substantial reduction, which is around 30% for a large range of network sizes. Moreover our proposed technique makes the network more durable against a high ratio of node failures. For example, it doubles the message reachability of the LOCALISER when the ratio of failed nodes is 30%.

For evaluation we considered four performance measures: path length, communication cost, clustering coefficient and reachability. Many of them are in a trade-off relation, and thus which of these is the most important depends on the characteristic of the system that employs the overlay network. If the system exhibits high node join and leave rates, then clustering coefficient is probably the most important. Path length may be the most important if the size of the messages traversing the overlay are large, because forwarding large messages imposes high load on relaying nodes and links. In many P2P applications, the dominant traffic on overlay paths is that of data query messages, which are small in size. In that case, communication cost should be the most important metric.

In future work, we plan to conduct further simulations to evaluate the efficiency of our proposed technique in more practical settings. These simulations will take into account dynamic node joins and leaves and the bandwidth of physical links. The idea of installing random links to enhance robustness is seen not only in P2P overlays but also in P4P [15], which is a new architecture framework for providing cooperative control over P2P applications and the underlying network. We believe that our "constrained-random" path techniques can be naturally incorporated in that context.

Acknowledgments

This work was supported in part by the MEXT Global COE program (Center of Excellence for Founding Ambient Information Society Infrastructure).

References

[1] J. Aspnes and G. Shah, "Skip graphs," ACM Trans. Algorithms, vol.3, no.4, pp.37:1–37:25, Nov. 2007.
[2] H. Cai and J. Wang, "Exploiting geographical and temporal locality to boost search efficiency in peer-to-peer systems," IEEE Trans. Parallel Distrib. Syst., vol.17, no.10, pp.1189–1203, 2006.
[3] K.T. Chen and J.K. Lou, "Toward an understanding of the processing delay of peer-to-peer relay nodes," Proc. International Conference on Dependable Systems and Networks, pp.410–419, USA, 2008.
[4] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," Proc. 2003 conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp.381–394, 2003.
[5] J. Leitao, J. Pereira, and L. Rodrigues, "Topology aware gossip overlays," INESC-ID Tec. Rep. 36/2008, Jan. 2008.
[6] L.S. Liu and R. Zimmermann, "Adaptive low-latency peer-to-peer streaming and its application," Multimedia Systems, vol.11, no.6, pp.497–512, 2006.
[7] Y. Liu, L. Xiao, X. Liu, L.M. Ni, and X. Zhang, "Location awareness in unstructured peer-to-peer systems," IEEE Trans. Parallel Distrib. Syst., vol.16, no.2, pp.163–174, 2005.
[8] F. Makikawa, T. Matsuo, T. Tsuchiya, and T. Kikuno, "Constructing overlay networks with low link costs and short paths," Proc. 6th IEEE International Symposium on Network Computing and Applications (IEEE NCA07), pp.299–304, July 2007.
[9] L. Massoulié, A.M. Kermarrec, and A.J. Ganesh, "Network awareness and failure resilience in self-organising overlay networks," Proc. 22nd IEEE Symposium on Reliable Distributed Systems (SRDS '03), pp.47–55, 2003.
[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," Proc. 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01), pp.161–172, 2001.
[11] Y. Ren, C. Sha, W. Qian, A. Zhou, B. Ooi, and K. Tan, "Explore the "small world phenomena" in pure P2P information sharing systems," Proc. 3rd International Symposium on Cluster Computing and the Grid, pp.232–239, 2003.
[12] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," Proc. 2001 ACM SIGCOMM Conference, pp.149–160, 2001.
[13] C. Tang, R.N. Chang, and C. Ward, "GoCast: Gossip-enhanced overlaymulticast for fast and dependablegroup communication," Proc. International Conference on Dependable Systems and Networks, pp.140–149, Japan, 2005.
[14] D. Watts and S. Strogatz, "Collective dynamics of small-world networks," Nature, vol.393, pp.440–442, 1998.
[15] H. Xie, Y.R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider portal for applications," ACM SIGCOMM Computer Communication Review, vol.38, pp.351–362, 2008.
[16] E.W. Zegura, K.L. Calvert, and S. Bhattacharjee, "How to model an Internetwork," Proc. IEEE Infocom, vol.2, pp.594–602, 1996.
[17] X.Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A construction of locality-aware overlay network: mOverlay and its performance," IEEE J. Sel. Areas Commun., vol.22, no.1, pp.18–28, Jan. 2004.

Fuminori Makikawa received his M.E. degree from Osaka University in 2007. He is currently studying towards the Ph.D. degree in the Graduate School of Information Science and Technology at the same university. He has been engaged in the research on peer-to-peer networking.

Tatsuhiro Tsuchiya received the M.E. and Ph.D. degrees in engineering from Osaka University in 1995 and 1998, respectively. He is currently an associate professor of the Department of Information Systems Engineering at Osaka University. His research interests are in the areas of model checking and distributed fault-tolerant systems.

Tohru Kikuno received his M.S. and Ph.D. degrees from Osaka University in 1972 and 1975, respectively. He was with Hiroshima University from 1975 to 1987. Since 1990, he has been a professor at Osaka University. His research interests include the quantitative evaluation of software development and the analysis and design of fault-tolerant systems. He served as symposium chair of the 21st Symposium on Reliable Distributed Systems (SRDS 2002).