A Two-dimensional Structured Overlay Based on Flexible Routing Tables

Masashi Hojo, Hiroya Nagao, Takehiro Miyao and Kazuyuki Shudo

Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552 Japan
Email: hojo.m.aa@m.titech.ac.jp

***Abstract*—An overlay network enables a large number of sensor nodes to perform effectively data collection and message delivery. Geographical routing is necessary for applications such as sensor network and it requires the overlay network to reflect nodes' locations. In addition, nodes' locations are generally biased and path lengths on the overlay should be short, in other words, the number of nodes relaying a message should be small. This paper describes a new overlay routing algorithm to perform geographical routing on a two-dimensional space satisfying the above requirements. The proposed algorithms provides unique and desirable features derived from Flexible Routing Tables (FRT), that is a design framework for structured overlays, because the algorithm is designed based on FRT. The derived features are dynamic routing table size and high extensibility. The proposed algorithm adopts P2P Delaunay Network as its topology and forms shortcut links based on estimated number of hops.**

I. Introduction

A structured overlay is an application layer network composed of nodes on a computer network such as Internet. A node in an overlay has its identifier (ID) and the nodes construct a logical structural network based on their IDs. A message is relayed node by node and arrives at a responsible node for a target ID. Each node determines a next hop to which the node relays a message by referring its routing table, that holds pairs of a node ID and an IP address. In consequence such a routing mechanism, structured overlays prevent spreading messages like flooding and they are highly scalable.

Assigning a node ID to each node that reflects its location in the real space, a structured overlay can perform message delivery without wasteful long-distance communication. It supports geographical routing and desirable for applications such as wireless mesh networks. There has been routing algorithms including Z-net [1] and SONAR [2] proposed for the purpose. In these overlays, while retaining communication paths to neighboring nodes, a node manages shortcut links to distant nodes to keep path length short.

These structured overlays are designed to work in large-scale and unstable situations. Therefore, since it is difficult to maintain a list of all of nodes, the number of nodes in a routing table is limited. The routing table size is determined in advance. Since optimal routing table size changes by a frequency of nodes' arriving and leaving and stability of the network, it is to be desired to adjust the size depending on the situation. Furthermore, although these overlays keep path length short, the shortest path can be inefficient in terms of network latency, nodes' reliability and so on. This problem occurs for routing table construction focused only on node IDs.

In this paper, we propose a structured overlay on a two-dimensional ID space to solve these problems. Adopting P2P Delaunay Network [3] as its topology, the proposed overlay guarantees reachability to any nodes. In addition, to keep path length short, it constructs a routing table by selecting some distant nodes. The routing table is constructed by estimating hop length between present node and candidate nodes for ununiform ID distribution.

To solve the problems in existing overlays, the proposed overlay is designed based on Flexible Routing Table (FRT) [4]. FRT is a method for designing routing algorithms for overlay networks. Structured overlays based on FRT construct a routing table not only to keep path length short but also to provide flexibility in it. The flexibility in a routing table is, for example, availability of additional indices to improve routing performance and dynamic adjustability of routing table size. Furthermore, FRT-based overlays can behavior both single-hop routing and multi-hop routing algorithms depending on the situation of the number of nodes and routing table size.

This paper is an extended version of our previous work [5]. The differences include additional algorithms (Method I and III in Sec. III-B and detailed analyses by frequency distributions (Sec. IV-A2).

In this paper, we describe the proposed overlay, and demonstrate their routing performance by simulation. Moreover, we observe its flexibility from FRT and availability in the actual environment.

II. Related Work

In this section, first, we refer to structured overlays on a two-dimensional ID space. Next, we describe P2P Delaunay Network, that our proposed overlay adopts as its topology. Finally, we explain FRT, a method for designing routing algorithms for overlay networks.

A. Structured Overlays on a Two-Dimensional Space

1) Z-net: Z-net [1] is a structured overlay on a two- or higher dimensional ID space. This overlay uses Z-Ordering, a

Proc. 20th IEEE Symposium on Computers and Communications (IEEE ISCC 2015), pp.309-314, July 2015

method to transform a multidimensional ID into a bit string (Z-address) with a space-filling curve. Each node maintains a part of ID space including its Z-address, and using the area's Z-address as a key, nodes construct a Skip graph network [6].

The ID transformation of Z-ordering conserves proximity in ID space. Therefore, Z-net can perform range searches on a multidimensional space by specifying and searching with the segment of Z-address. However, there are problems that there are some spots where the continuity of Z-address does not hold, and to execute a range search, the segment of Z-address will often become multiple fragments and the querying will become complex as a result.

2) SONAR: SONAR [2] is a structured overlay extended from CAN [7], a structured overlay on an n-dimensional torus. In CAN network, each node maintains rectangular areas and connects other nodes whose areas adjoin them. When a new node arrive in a CAN network, an existent node divides one of its areas and imparts it to the node. In SONAR network, nodes construct a CAN network and form shortcut links to distant nodes to keep path length short. Target nodes of the shortcut links are chosen for each 2^i hops distant in each dimension in a CAN network.

Since SONAR constructs a CAN-based network, it resolves the problem in Z-net, namely the complexity of range searches. However, there are some problems, such as the complexity of a shape of nodes' management areas with repeating nodes' arriving or leaving and a high cost with maintenances of shortcut links.

B. P2P Delaunay Network

P2P Delaunay Network [3] is a structured overlay on a two-dimensional ID space. It uses a Delaunay diagram that is famous in computational geometry, and each node constructs communication paths to other nodes according to it. A Delaunay diagram is constructed by drawing edges between points in the space (called generating points) in accordance with particular conditions. In a P2P Delaunay Network, nodes are considered as generating points and communication paths as edges in a Delaunay diagram. A P2P Delaunay Network has features derived from a Delaunay diagram:

- Since a Delaunay diagram is a connected graph, there are routing paths between every two arbitrary nodes.
- A P2P Delaunay Network can adopt a Voronoi tessellation to determine nodes' management areas.
- Every node can reach an arbitrary node by a greedy routing of Euclidean distance.
- A P2P Delaunay Network can perform range searches in such a simple way that the nodes in a target area spread the query to their neighbor nodes.
- Even in a situation that nodes repeat arriving or leaving, the nodes' management areas do not become complex. Therefore, unconformity in routing tables is less likely to occur.

Owing to these features, P2P Delaunay Network is suitable for Infrastructural networks to manage a two-dimensional space. The simple way of range searches as described above

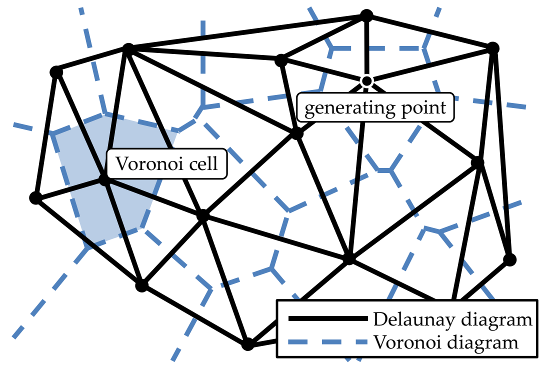


Fig. 1: Delaunay/Voronoi diagrams.

resolves the problem in Z-net of the complexity. And in SONAR, because of unconformity in routing tables under the churn condition, it's necessary to verify and update routing tables periodically. However, in P2P Delaunay Network, a Voronoi re-tessellating is invoked whenever a node arrives or leaves, and conformity of routing tables are kept. For these advantages, our proposed overlay adopts P2P Delaunay Network as its topology.

Figure 1 shows an example of a Delaunay diagram and a Voronoi tessellation. A Voronoi tessellation is executed based on Euclidean distance between generating points of a Delaunay diagram; arbitrary points on the space belong to their nearest generating point.

1) Autonomous and Distributive Generation Algorithm: The algorithm has been proposed to construct a P2P Delaunay Network by nodes on a two-dimensional space in autonomous distributed way [3]. In this algorithm, each node autonomously calculates a relation of connection between its neighbor nodes, and a P2P Delaunay Network by the whole of nodes is constructed. Since a node's arriving or leaving affects only its vicinities, there is high scalability of reconstructing the overlay and recalculating a node's management area.

2) Long Range Contact: In a P2P Delaunay Network, since each node has connections only to their neighbor nodes, path length to distant nodes are long. Therefore, a technique has been proposed to construct Long Range Contacts (LRCs) to keep path length short [8]. In this technique, each node constructs LRCs in four directions based on number of hops in a P2P Delaunay Network. The LRCs based on number of hops adapt a bias of the nodes' distribution. Therefore, this technique is available in such a case of using nodes' locations of the real space to their IDs.

C. Flexible Routing Tables

Flexible Routing Tables (FRT) is a method for designing routing algorithms for overlay networks. Whereas many conventional structured overlays strictly define a routing table should be built as a combination of node IDs and construct it by finding closest nodes to the IDs, FRT proposes an entirely different routing table constructing method. That is, an overlay based on FRT defines a total order of the routing table set $\leq_{\mathrm{RT}}$, and improves a routing table in accordance with it. In contrast to conventional structured overlays that restrict candidates for a routing table, FRT-based overlays provide various advantages explained in Sec. II-C1.

1) Advantageous Features of FRT: The algorithms designed based on FRT have the following advantages:

- It's possible to adjust routing table size L dynamically. L is adjusted in response to node's capacity or network stability.
- It's possible to design an algorithm that consistently handles both single-hop routing and multi-hop routing. If a node can hold all other nodes in its routing table, the node can forward a message in single-hop. Otherwise, a message is relayed node by node. For example, FRT-2-Chord [9] is a structured overlay based on FRT, that handles single-/multi-hop routing on one-dimensional ring topology.
- The FRT-based algorithms are extensible to consider various indices of routing performance. For example, Proximity-aware Flexible Routing Tables (PFRT) [10] considers network proximity and Grouped Flexible Routing Tables (GFRT) [4] considers node groups.

2) Routing Table Management Operations: To construct and to improve a routing table, FRT prepares two operations as follows.

- *Entry leaning*: Entry leaning is an operation to insert node's information into a routing table. Executing this operation, a node inserts a new other node's information into its routing table without screening. The node's information is available from various communication results, and in addition, it's possible to get it by active lookup.
- *Entry filtering*: Entry filtering is an operation to remove an entry from a routing table. When the number of entries in a routing table exceeds L, this operation is executed in order to retain a limitation of routing table size. The entry is selected according to total order of the routing table set described in Sec. II-C3.

In FRT-based routing table construction, at first, entry leaning is repeated until exceeding L. Then, a couple of entry leaning and entry filtering is executed continuously and a routing table is refined gradually.

3) $\leq_{\mathrm{RT}}$:Total Order of the Routing Table Set: A FRT-based overlay defines a total order of the routing table set $\leq_{\mathrm{RT}}$ to determine a better one of two routing tables. Between two routing tables E and F, if we have $E \leq_{\mathrm{RT}} F$, then we determine E is better F. FRT-based structured overlay improves routing efficiency by refining routing tables accordingly $\leq_{\mathrm{RT}}$. We can design a FRT-based overlay by defining $\leq_{\mathrm{RT}}$ considering a topology and distance of the overlay.

III. Proposed Overlay

In this section, we propose a structured overlay on a two-dimensional space that provide with flexibility not in the previous methods. The proposed overlay guarantees reachability and determines a node's management area by adopting P2P Delaunay Network as its topology. Moreover, it keeps path length short by forming shortcut links to distant nodes based on FRT.

TABLE I: Parts of a routing table.

components	size
Delaunay neighbor nodes set D	expected value of 6 or less
shortcut links list S	L (adjustable)

To construct a routing table, Euclidean distance is less useful to reduce path length in the situation that nodes distribute ununiformly. Therefore, the proposed overlay adopts the number of hops on a P2P Delaunay Network to select targets of shortcut links. Since number of hops between two nodes, unlike Euclidean distance, cannot be calculated immediately from their IDs, we devise a method to estimate it indirectly as described later. Using the estimation, each node judges its routing table quality and refines it based on FRT.

Besides keeping path length short, the proposed overlay provides features of FRT. For example, to manage location-based information in an overlay, there is often a limitation of distance that a node can directly communicate within. In such cases, considering the distance of a node to permit to insert into a routing table, the proposed overlay can refine a routing table while following the limitation.

A routing table of the proposed overlay is made up of two parts as Tab. I. We describe ways to construct them below.

A. Delaunay Neighbor Nodes Set

Since the proposed overlay adopts P2P Delaunay Network as its topology, each node manages their neighbor nodes in a Delaunay neighbor nodes set D. From a feature of a Delaunay diagram, expected value of degree of a generating point is 6 or less. Therefore, a size of D will be also 6 or less.

Operations of finding and managing Delaunay neighbor nodes is based on the autonomous and distributive generation algorithm of P2P Delaunay Network described in Sec. II-B1.

B. Shortcut Links List

Since path length to distant nodes in a P2P Delaunay Network is long, the proposed overlay constructs shortcut links to reduce it. The target nodes of shortcut links are managed in a shortcut links list S.

To select the target nodes, it is necessary to introduce some index to reduce path length effectively. If nodes distribute uniformly, an index based on Euclidean distance is useful to reduce it. However, in the situation that nodes distribute ununiformly, the index cannot work effectively. This is because that the Euclidean distance-based shortcut links can be formed in a sparse area more than necessary or be not formed in a dense area. Therefore, we adopt the number of hops in a P2P Delaunay Network that allows to form proper shortcut links even though nodes distribute ununiformly.

Since the number of hops between two nodes cannot be calculated directly, we devised three methods to estimate it. These methods are similar, but their details are different. In

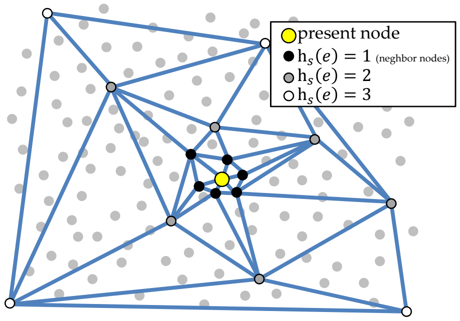


Fig. 2: Routing table entries on a two-dimensional space.

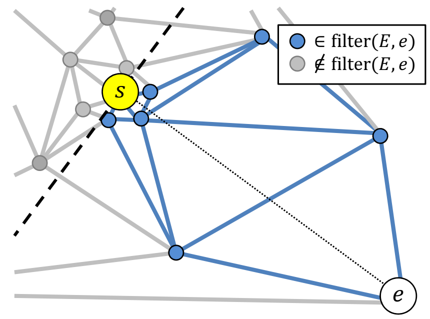


Fig. 3: Routing table entries in $\mathrm{filter}(E, e)$.

below, at first, we describe a common concept to be used in these methods. Next, we explain each method focused on the difference.

1) Similar Concept in Three Methods: In the proposed overlay, the estimation focuses how each entry is distant from a present node from the point of view of the number of hops in its routing table.

We define the *hop counts in a routing table* $\mathrm{h}(e)$ as follows.

Definition 3.1 (hop counts in a routing table $\mathrm{h}(e)$*):* For an entry e in a routing table, $\mathrm{h}(e)$ is graph distance in a Delaunay diagram constructed from the present node s and all entries in the s's routing table.

Figure 2 is an example of the Delaunay diagram of a routing table. In a routing table, since Delaunay neighbor nodes are close to a present node, we have $\mathrm{h}(e) = 1$ for $e \in D$. The target nodes of shortcut links are selected from outside of the Delaunay neighbor nodes, and more distant entry has greater $\mathrm{h}(e)$.

The three methods described below use this value and each method calculates a value called the *rank in a routing table* $\mathrm{r}_s(e)$ in different ways.

Definition 3.2 (rank in a routing table): $\mathrm{r}_s(e)$ is entry e's rank in s's routing table.

This value describes s's estimation of the number of hops in a P2P Delaunay Network between s and e. In the proposed overlay, each node calculates $\mathrm{r}_s(e)$ and estimates the number of hops for their each routing table entry e.

Then, using $\mathrm{r}_s(e)$, the total order of the routing table set $\leq_{\mathrm{RT}}$ is defined in Sec. III-C.

2) Method I: In this method, we use the hop counts in a routing table as the rank in a routing table. That is, we define $\mathrm{r}_s(e) = \mathrm{h}(e)$. As mentioned earlier, the greater $\mathrm{h}(e)$ is, the more distant in P2P Delaunay Network e is from s .

3) Method II: In this method, we define the rank in a routing table as follows.

$$\mathrm{r}_s(e) = \#\{e^* \in D \cup S \cup \{s\} \mid \mathrm{h}_s(e^*) < \mathrm{h}_s(e)\}. \quad (1)$$

This expression means that $\mathrm{r}_s(e)$ is considered as a rank in all entries in ascending order of $\mathrm{h}(e)$.

For example in Fig. 2, the six entries with $\mathrm{h}(e) = 1$ (Delaunay neighbor nodes) have $\mathrm{r}_s(e) = 1$, the another six entries with $\mathrm{h}(e) = 2$ have $\mathrm{r}_s(e) = 7$, and the others with $\mathrm{h}(e) = 3$ have $\mathrm{r}_s(e) = 13$.

4) Method III: This method is a minor change of Method II. To define the rank in a routing table, we prepare a subset of a routing table as follows.

$$\mathrm{filter}(E, e) = \{e^* \in E | \angle ese^* < 90^\circ\}. \quad (2)$$

Where e is an entry of s's routing table E and $\angle ese^*$ is an Euclidean angle of several IDs. Using this, we define the rank in a routing table as follows.

$$\mathrm{r}_s(e) = \#\big(\mathrm{filter}(E, e) \cap \{e^* \in D \cup S \cup \{s\} \mid \mathrm{h}_s(e^*) < \mathrm{h}_s(e)\}\big). \quad (3)$$

Figure 3 shows an example of the calculation of $\mathrm{filter}(E, e)$. Differently from Method II, this method uses $\mathrm{filter}(E, e)$ and limits entries to calculate $\mathrm{r}_s(e)$.

C. $\leq_{\mathrm{RT}}$*:Total Order of the Routing Table Set*

In the proposed overlay, to define a total order of the routing table set $\leq_{\mathrm{RT}}$, we define the *gap of rank* as follows.

Definition 3.3 (gap of rank $\mathrm{f}(e)$*):* For each entry e in s's routing table, the gap of rank $\mathrm{f}(e)$ is defined as

$$\mathrm{f}(e) = \mid \mathrm{r}_s(e) - \mathrm{r}_e(s) \mid . \quad (4)$$

$\mathrm{f}(e)$ means an difference between $\mathrm{r}_s(e)$, an estimation of e's position in s's routing table and $\mathrm{r}_e(s)$, an estimation of s's position in e's routing table. This gap originates because both nodes construct their routing tables autonomously, and therefore the both estimations of the other's position become different.

The proposed overlay regard the gap $\mathrm{f}(e)$ as an error of the estimation of the number of hops between s and e. For example, we think the situation that $\mathrm{r}_s(e)$ is greater than $\mathrm{r}_e(s)$. This condition shows that s estimates extra hops between s and e comparing to e's estimation. In this way, each node can verify that it construct a shortcut link to a proper target.

From the above idea, we define that an ideal routing table has $\mathrm{f}(e) = 0$ for each entry. Hence, $\leq_{\mathrm{RT}}$ in the proposed overlay is defines as follows.

Definition 3.4 ($\leq_{\mathrm{RT}}$ *in proposed overlay):*

$$E \leq_{\mathrm{RT}} F \Leftrightarrow \mathrm{f}(E) \leq_{\mathrm{dic}} \mathrm{f}(F). \quad (5)$$

Where $\mathrm{f}(E)$ is a list $\{\mathrm{f}(e) \mid e \in E\}$ in which each $\mathrm{f}(e)$ are sorted in descending order and $\leq_{\mathrm{dic}}$ is lexicographical order.

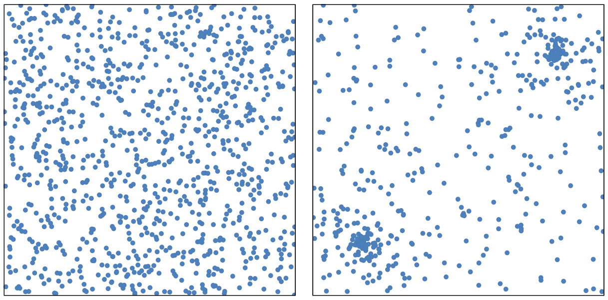


Fig. 4: Nodes' layout at uniform distribution and Zipf distribution ($\alpha = 1.0$).

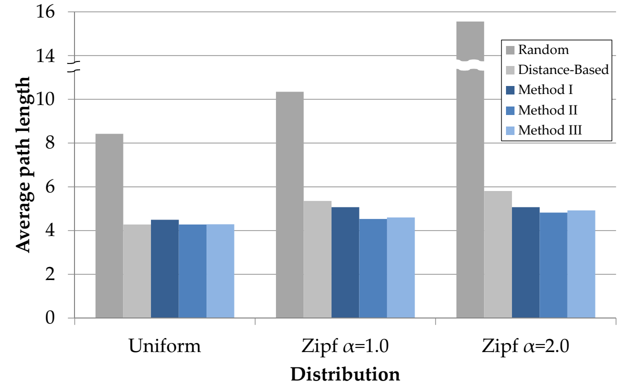


Fig. 5: Average path length.

D. Improvement of Routing Table based on FRT

1) Entry Leaning: The proposed overlay manages $\mathrm{r}_e(s)$ in addition to a pair of a node ID and its IP address in a routing table. In contrast to the ID and IP address pair, $\mathrm{r}_e(s)$ cannot be got from a result of various communications. Therefore, in entry learning of the proposed overlay, there is an additional communication to get $\mathrm{r}_e(s)$.

2) Entry Filtering: In entry filtering, an entry e^* filling below condition is deleted by using $\leq_{\mathrm{RT}}$ described above.

$$S \setminus \{e^*\} \leq_{\mathrm{RT}} S \setminus \{e\}, \quad e \in S. \tag{6}$$

IV. Evaluation

We implemented the proposed overlay on Overlay Weaver [11], [12], an overlay construction toolkit, and performed experiments.

To observe influence of the bias of node IDs to the number of path length, we set two situations of assignment of node IDs as follows.

1) *Uniform distribution*: All nodes distribute uniformly in a two-dimensional space.
2) *Zipf distribution*: 20% of nodes distribute uniformly in a two-dimensional space, and the others distribute around two concentration points following Zipf distribution ($\alpha = 1.0, 2, 0$).

The bias is larger in the order of Uniform distribution, Zipf distribution ($\alpha = 1.0$), Zipf distribution ($\alpha = 2.0$). Figure 4 shows nodes' layout in a two-dimensional space.

A. Comparison with Other Algorithms

1) Algorithms for Comparison: We brought two algorithms for comparison with the proposed overlay. Both algorithms construct a routing table by entry leaning and entry filtering similarly to the proposed overlay, but methods for entry filtering are different.

Random Filtering Algorithm: This algorithm does not use any indices and removes an entry randomly in entry filtering.

Distance-Based Filtering Algorithm: This algorithm uses Euclidean distance to select an entry to remove. It's designed to keep path length short under circumstances that nodes distribute uniformly in a two-dimensional ID space.

In Distance-Based Filtering Algorithm, let $\mathrm{d}(s, e)$ be Euclidean distance between a node s and s's entry e, and $E = \{e_i\}$ be s's routing table where each entry is sorted in ascending order of $\mathrm{d}(s, \cdot)$. We defined that an ideal routing table in this algorithm is its entries' distribution follows the probability density below:

$$\mathrm{f}(e_i) = C\frac{1}{\mathrm{d}(s, e_i)}. \tag{7}$$

Where C is a constant value for $\sum \mathrm{f}(e_i) = 1$.

Each node constructing an ideal routing table, Distance-Based Filtering Algorithm keeps path length within $O(\log^2 N/\log L)$, where N is a number of nodes in the whole of overlay. This feature is based on the theory of Small-World phenomenon [13].

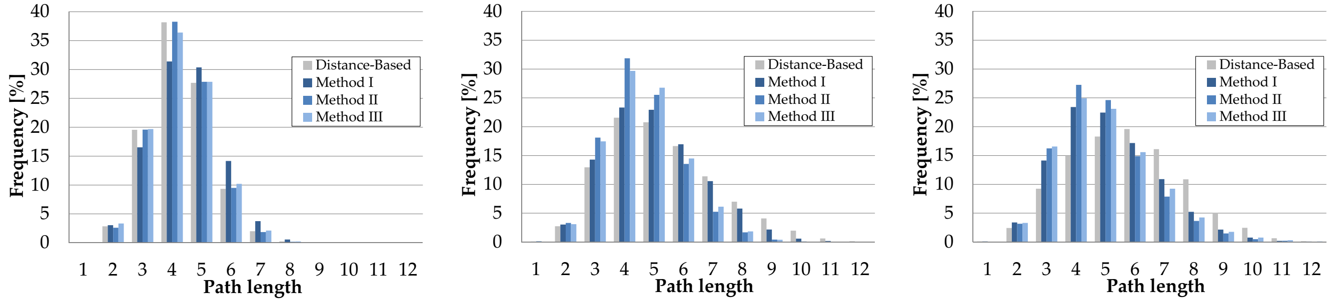
2) Experimental Results: We conducted the following experiment with the proposed overlay (Method I - III), Random Filtering Algorithm, and Distance-Based Filtering Algorithm.

1) Assign IDs in accordance with the above distributions to nodes and let them to construct a P2P Delaunay Network.
2) Let each node to repeat to execute entry learning by active lookups in accordance with the same distributions.
3) When the number of lookups of each node becomes 200 times, measure path length of the lookups.

In these experiments, the number of nodes is 1000 and the routing table size L is 20.

Figures 5 and Fig. 6 show the results of these experiments. First, taking notice of Random Filtering Algorithm, we found that this algorithm is no longer practicable to keep path length short.

Next, we take notice of Distance-Based Filtering Algorithm. The result shows that although this algorithm can keep path length short in the situation that nodes distribute uniformly, path length became longer as the bias of nodes' distribution increases. Average path length increased by 25% in Zipf distribution ($\alpha = 1.0$) and by 36% in Zipf distribution ($\alpha = 2.0$) than in uniform distribution. Moreover, frequency distributions (Fig. 6) show that as the bias increased, the frequency of short-path lookups became lower and the frequency of long-path lookups became greater. From this result, Distance-Based Filtering Algorithm is available only in the situation that nodes distribute uniformly.



(a) Uniform distribution. (b) Zipf distribution ($\alpha = 1.0$). (c) Zipf distribution ($\alpha = 2.0$).

Fig. 6: Frequency of each path length.

Now we observe the results of the three methods of the proposed overlay. We found that Method I is not effective to keep path length short and Method II is slightly better than Method III. In Method I, we use the hop counts in a routing table as an index to entry filtering, but in such a strategy to construct a routing table, a node appears to select other nodes in a particular direction. It causes to increase path length.

In what follows, we take notice Method II, the most efficient algorithm, and analyze it. In uniform distribution, the proposed overlay performs equivalent result to Distance-Based Filtering Algorithm. Hence, the proposed overlay can adapt to uniform distribution.

Increasing the bias of nodes' ID distribution, path length of the proposed overlay became longer, but the increased amounts are much smaller than ones of Distance-Based Filtering Algorithm. In fact, average path length increased by only 5% in Zipf distribution ($\alpha = 1.0$) and by 13% in Zipf distribution ($\alpha = 2.0$) than in uniform distribution. Frequency distributions (Fig. 6) show that in contrast to Distance-Based Filtering Algorithm, in the proposed overlay, the frequency of short-path lookups is kept high in any situations.

V. Conclusion

In this paper, we proposed a new structured overlay on a two-dimensional space that adapts to biased node ID distributions. The overlay provides unique and desirable features derived from FRT because it is designed based on FRT. The experiments showed efficiency of the three methods to select target nodes of shortcut links and they performed well as designed.

We revealed the most effecient method to keep path length short and confirmed that it can perform its routing efficiency even though nodes distribute ununiformly.

The proposed overlay works with node IDs assigned based on nodes' location. It enables geographical routing.

FRT-based design is the source of the desirable features. Flexibility of routing table entries supports geographic routing. With the flexibility a node can hold any node in its routing table. In other words, a node can select only communication-reachable nodes for its routing table entries. Though the current proposed overlay does not consider communication-reachability, it is straightforward to consider it.

Future work includes a logical and mathematical proof of the efficiency of the proposed overlay. Besides, we will extend the proposed overlay by utilizing the flexibility provided by FRT to consider communication-reachability.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 25700008 and 26540161.

References

[1] Y. Shu, B. C. Ooi, K.-L. Tan, and A. Zhou, "Supporting Multi-Dimensional Range Queries in Peer-to-Peer Systems," in *Proc. IEEE P2P 2005*. IEEE, 2005, pp. 173–180.

[2] T. Schütt, F. Schintke, and A. Reinefeld, "A Structured Overlay for Multi-dimensional Range Queries," in *Proc. Euro-Par 2007*. Springer, 2007, pp. 503–513.

[3] M. Ohnishi, R. Nishide, and S. Ueshima, "Incremental Construction of Delaunay Overlaid Network for Virtual Collaborative Space," in *Proc. C5 2005*. IEEE, 2005, pp. 75–82.

[4] H. Nagao and K. Shudo, "Flexible Routing Tables: Designing Routing Algorithms for Overlays Based on a Total Order on a Routing Table Set," in *Proc. IEEE P2P 2011*. IEEE, 2011, pp. 72–81.

[5] M. Hojo, H. Nagao, T. Miyao, and K. Shudo, "A Structured Overlay on a Two-Dimentional Space based on Flexible Routing Tables," *IPSJ Journal*, vol. 56, no. 2, pp. 439–447, 2015, (in Japanese).

[6] J. Aspnes and G. Shah, "Skip Graphs," *ACM Transactions on Algorithms*, vol. 3, no. 4, pp. 1–25, 2007.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proc. SIGCOMM 2001*. ACM, 2001, pp. 161–172.

[8] M. Ohnishi, S. Tsuboi, M. Hirayama, T. Eguchi, and S. Ueshima, "Distributive Generation Algorithm of Long Range Contact for Remote Spatial-data Access on P2P Delaunay Network," in *Proc. C5 2007*. IEEE, 2007, pp. 145–152.

[9] Y. Ando, H. Nagao, T. Miyao, and K. Shudo, "FRT-2-Chord: A DHT Supporting Seamless Transition between On-hop and Multi-hop Lookups with Symmetric Routing Table," in *Proc. ICOIN 2014*. IEEE, 2014, pp. 170–175.

[10] T. Miyao, H. Nagao, and K. Shudo, "A Method for Designing Proximity-aware Routing Algorithms for Structured Overlays," in *Proc. IEEE ISCC 2013*. IEEE, 2013, pp. 508–514.

[11] K. Shudo, Y. Tanaka, and S. Sekiguchi, "Overlay Weaver: An Overlay Construction Toolkit," *Computer Communications*, vol. 31, no. 2, pp. 402–412, 2008.

[12] K. Shudo, "Overlay Weaver," http://overlayweaver.sourceforge.net, 2006.

[13] J. Kleinberg, "The Small-World Phenomenon: An Algorithm Perspective," in *Proc. STOC 2000*. ACM, 2000, pp. 163–170.