DisTIN - A Distributed Spatial Index for P2P Environment

Oje Kwon†, Jung-Wook Moon‡, Ki-Joune Li‡

Department of Computer Science, Pusan National University, Gumjung-Gu, Pusan 609-735, South Korea

E-mail: †ojkwon@isel.cs.pusan.ac.kr, ‡{lik, jwmoon}@pnu.edu

Abstract In ubiquitous computing environments, the location-awareness of mobile devices is a fundamental functional requirement. Due to the lack of scalability, it is however impossible to store and manage a large number of mobile devices by a centralized server. In this paper, we propose DisTIN, an alternative method based on P2P method to index and handle the location data of mobile devices in fully distributed environment. Each mobile device stores the location and IP addresses of its neighbor devices determined by triangular network. And a query submitted to a mobile device can be processed via its neighbor devices linked by triangular network toward the query position without any centralized server. We also propose an update algorithm of DisTIN to handle the movement of mobile devices.

Keywords Ubiquitous Computing, P2P Environment, Spatial Index

1. Introduction

In ubiquitous computing environment, it is an important functional requirement to be aware of the context around each device. Location-awareness is a crucial part of context-awareness for mobile devices. In order to achieve the location-awareness, we need a method to store and handle the location data of mobile devices. By centralized approach, it is however nearly impossible to handle location data of a huge number of mobile devices in real time due to the lack of scalability. For example, it would be necessary to store and process the location data of more than several million vehicles in telematics application. With a centralized server or even a certain number of servers, it is impossible to update and process query on the location of several millions of vehicles in real-time.

A promising alternative approach is P2P environment, where data or algorithms are fully decentralized and each node directly communicates with others without centralized servers [1]. In P2P environments, an index is required to process query and retrieve data spread over nodes, otherwise an exhaustive flooding of query message over the entire network would be inevitable. An index in P2P environment allows applications to search for relevant data by specifying a predicate on the value. For this reason, a number of research have been done to provide distributed indexing structures such as DHT(Distributed Hash Table) [2-5] or super nodes [6]. Location data and the mobility of devices are not considered by these methods.

In this paper, we focus on the location data of mobile devices for indexing and query processing. We propose an distributed spatial index data structure, called DisTIN(Distributed Triangulated Irregular Network) for mobile devices based on triangulation, an updating mechanism and an in-network spatial query processing method for nearest neighbor by DisTIN.

This paper is organized as follows; the related work and motivation of this study are presented in section 2. And we propose a distributed data structure for indexing location data of mobile devices in P2P environment in section 3 as well as its updating mechanism. In section 4, an algorithm for processing *k*-nearest neighbor query is explained. We observe the performance of our method by experiments in section 5. And the conclusion and future work are given in section 6.

2. Related Work

A simple way to find data in P2P environment by specifying a predicate is to flood query messages over the networks. But this method results in a large number of messages and traffic overhead. Several distributed indexing methods have been proposed such as Chord [3] and CAN [2] to reduce traffic overhead and hop counts from the source node to the destination.

By Chord, the query message can be reached to the destination within $O(\log N)$ hop counts by using finger table, while CAN needs $O(N^{1/d})$ hop counts where N and d mean the number of nodes and the dimensionality of transformed space of CAN respectively. Other methods such as Pastry [7], Tapestry [8], and Bristle [9] are also based on DHT like CAN and Chord, although their DHTs and topologies are different.

While these methods are useful to process exact match or keyword match queries, they have two important problems. First, they cannot be used for range query processing. In order to overcome this problem, several extensions have been attempted, which are clearly summarized by [16]. For example, an extension of CAN [10] is proposed to handle range query, and Chord is extended to process range query [5]. In contrast with these methods, PePeR(Peer to Peer Range) [4] does not use any kind of DHT but distributed data structures and maintains links to the nodes with adjacent values. The links provide routing path to search a given query range in P2P environment. A similar method Skip graphs has been proposed, which is a generalization of skip lists, and support range queries in P2P environment with $O(\log n)$ hop counts and $O(\log n)$ messages [18].

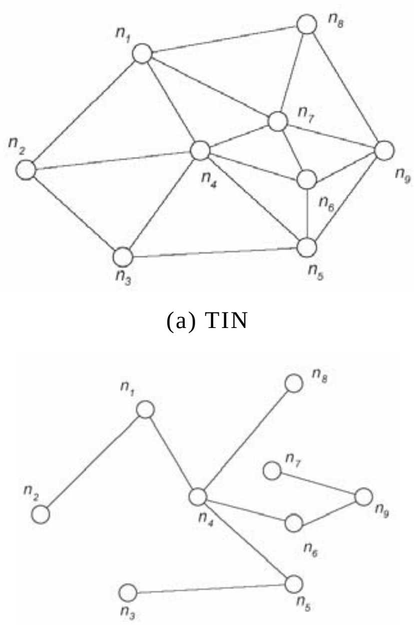
The second problem of distributed indices including PePeR and skip graphs, is that they do not support the mobility of nodes, although the mobility is an important requirement in many applications of P2P environments. In this paper, we focus on the mobility of nodes and dynamic updates of distributed spatial index in P2P environment.

In this paper, we however assume the P2Penvironment, where each mobile node is assigned an IP address of infrastructure network such as IPv6 address. The mobility of nodes has been widely studied for MANET (Mobile Ad-Hoc Network), and several routing mechanisms have been proposed based on geographic information [11-14]. MANET environment differs from P2P for several reasons. First, the information about neighbor nodes is not explicitly stored, while each node must specify its neighbor nodes in P2P. If a node receives a message from another node, it means that they are located in the vicinity and can communicate without specifying the destination node address. In this case, the vicinity can be considered as an implicit link between two nodes and spatial query processing is performed by routing via this implicit link. Second, a mobile node in MANET cannot communicate with other nodes locating outside of transmission area, while there is no communication restriction of distance between two nodes in P2P. Once an IP address of a node is known, a direct communication via infrastructure network is possible in P2P environment. It is an important advantage of P2P over MANET environment.

We propose a distributed spatial index for P2P environment in this paper. In particular, this index supports the movement of nodes with only small fraction of update costs.

3. Distributed Spatial Index by TIN

In order to develop a distributed spatial index of mobile devices in P2P environment, two requirements should be carefully considered. First, all nodes must be connected via the index. Second, spatial locality should be respected by the index. The second requirement means that a fraction of an index stored at a node must point to its neighbor nodes. It is evident since the nodes of interest for a given node are located at its vicinity in most cases.



(a) TIN

(b) Minimal Spanning Tree

Figure 1. Networks of TIN and MST

The connectivity of nodes with their neighbors constitutes a topology of nodes. A number of topologies have been proposed for sensor networks [19]. But these topologies are originally designed for ad-hoc network and inappropriate to P2P environment. Furthermore, they do not support efficiently the movement of nodes, even though some methods provide local modifications of topology.

In this paper, we propose a date data structure for distributed spatial index in P2P environment, called DisTIN (Distributed TIN) to support the mobility of nodes. Each node in DisTIN is connected to its neighbor nodes as a vertex of triangles as shown by figure 1. Note that the triangles of DisTIN do not respect delaunay triangulation unlike the methods proposed by [20, 21]. The condition for delaunay triangulation is strong that the dynamic property of index would reduce significantly.

Since the data structure for indexing should be fully distributed, TIN is distributed into nodes in P2P, where each mobile node has only a small fraction of TIN. We propose a distributed TIN, called DisTIN, that each node has a set of tuples (`IPaddress`, `position`) of its neighbor nodes connected by the edges of TIN as shown by figure 2.

Due to dynamic property of mobile nodes, the DisTIN must support the joining of a new node and movement of nodes. We will present the algorithms to support these dynamic properties of mobile nodes.

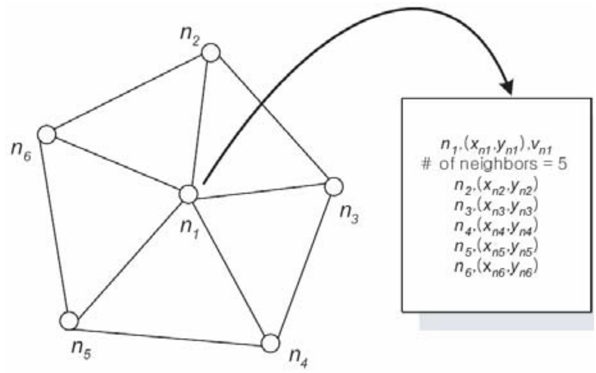


Figure 2 Data Structure of DisTIN for a node

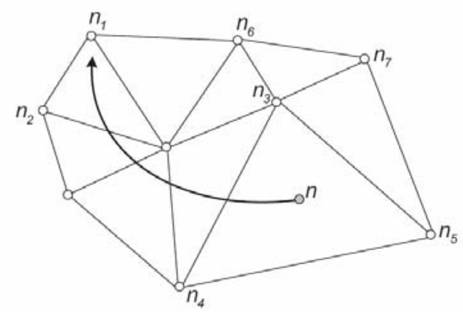
3.1. Joining a new node

In order to maintain DisTIN, we need a join algorithm for incremental growth, since we have no centralized data structure in P2P environment. It means that the triangulation should be performed in fully distributed way, when a new node joins the network.

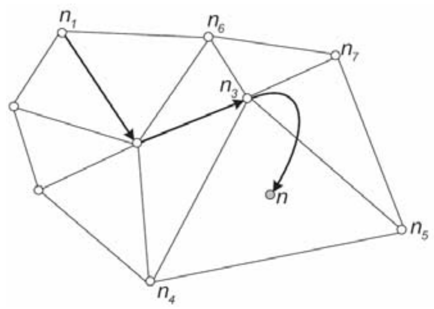
The join algorithm is simple and easy to implement by using the nearest query processing algorithm mentioned in the next section and consists of two steps as follows;

- step 1: Finding the triangle containing the new node.
- step 2: Local re-triangulation

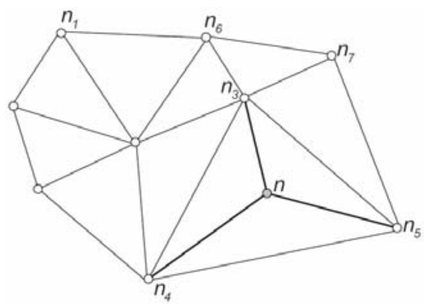
Figure 3 shows an example of join procedure. For step 1, the new node sends a `JOIN` message to a predefined node. We suppose that any new node knows the IP address of at least one node before joining the network. Then by the *k*-nearest neighbor query processing method presented in section 4, we can find the nearest node to the new node. And starting from the nearest neighbor node, we can find the triangle containing the new node. And for step 2, we insert three edges from the new node to each node of the triangle containing the new node in order to make new triangles.



(a) Sending `JOIN` message from node n to node n_1



(b) Routing to node n_3



(c) Re-triangulation

Figure 3. Example of Incremental Growth by TIN

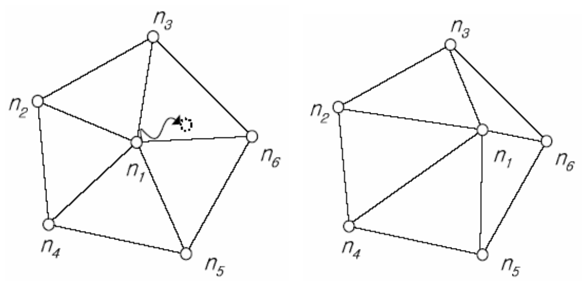
In fact, we apply delaunay triangulation to the local re-triangulation operation, which is slightly different from figure 3(c), to avoid narrow triangles, which degrade the performance of update and query processing. But note that the condition of delaunay triangulation is not globally respected by DisTIN due to its update algorithm, which will be explained in the next subsection.

3.2. Update for node movement

In order to support the mobility of devices, the triangular network should be updated. It is not only the location of mobile devices stored at the device itself and its neighbor devices, but also the triangular network. When a mobile device passes through a boundary line of triangle, the condition of triangulation is no longer respected and re-triangulation should be performed. For this reason, we consider the following two cases for the update algorithm of DisTIN;

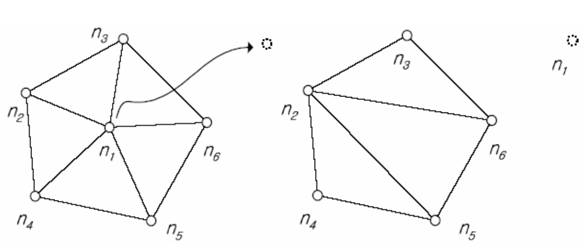
- case 1: the node stays in the same triangle. In this case, only the update on position data is required and no update of triangular network is needed as shown by figure 4. This node has only to send a `NEW_POSITION` message to its neighbor nodes and its neighbor nodes should update its position data of their distributed index.

- case 2: the node no longer stays in the same triangle. For this case, a relatively complicated update procedure is to be performed. First, we should update the triangle where the node no longer stays. A re-triangulation should be done by removing the node from the triangle, as explained by figure 5. And second, the new triangle where the node is entering must be updated. A re-triangulation should be done by adding this node to the triangle, as explained by figure 6.



(a) Before Update (b) After Update

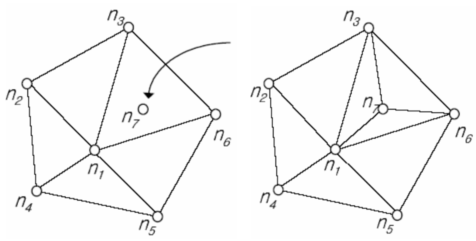
Figure 4. Example of Case 1- Node stays in a triangle



(a) Before Update (b) After Update

Figure 5. Re-triangulation of the former triangle

For the re-triangulation operation, we apply delaunay triangulation to avoid narrow triangles like the join algorithm. But DisTIN does not respect the property of delaunay triangulation due to the local movement of node within a triangle.



(a) Before Update (b) After Update

Figure 6. Re-triangulation of the new triangle

4. Spatial query processing by DisTIN

The query processing with DisTIN consists of two phases; *routing phase* and *evaluation phase*. The routing phase is to forward query message to the triangle containing the query position. This phase is performed via traversing edges of triangles. And the evaluation phase is to select the nodes satisfying query condition among the candidate nodes. In this paper, we present a query processing method for k-nearest neighbor search.

4.1. Routing Phase

In order to process spatial query, the query message has to be forwarded to one of three nodes of the triangle containing the query position. Depending on the selection policy of neighbor nodes for forwarding, two routing methods have been proposed in [15], *greedy method* and *half-moon method*.

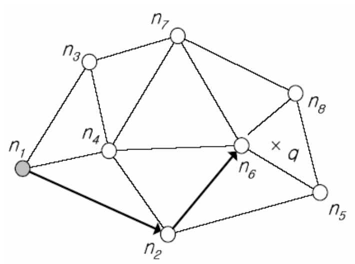


Figure 7. Routing by Greedy Method

The greedy method forwards one of neighbor nodes which is the closest to query point. Node n_k is selected among the nodes n_i linked to the source node, so that $\left|\overline{n_k p}\right| = \min\{\left|\overline{n_i p}\right|\}$. And the forwarding is repeated until the triangle containing query position will be found.

The half-moon method forwards query message to nodes located within the half-moon area towards query point as shown by figure 8. Node n_1 forwards query message to n_2, n_3 and n_4, since they are within the half-moon area.

By the same way, nodes n_2, n_3 and n_4 forward query message to their neighbors and the forwarding is repeated until the triangle containing query position will be found.

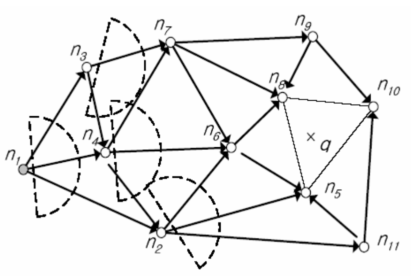


Figure 8. Routing by Half-Moon method

The greedy method can reduce the number of forwarding messages, but the routing path is not optimal. On the contrary, the half-moon method results in a relatively large number of forwarding messages, but finds much shorter routing path than the greedy method. To compare these two methods, results of experiments will be given in section 5.

4.2. Evaluation Phase

After arriving in the triangle containing the query position, we should select k nearest neighbor nodes. In order to evaluate the query, we first select the nearest node to the query position, which we call the *evaluating node*. And we expand the candidate area from the evaluating node in incremental way till the k nearest neighbor nodes are found. During the expansion, the evaluating node collects the information of the nodes within candidate area and decides whether the k nearest neighbor nodes are found. This procedure is composed of three steps and explained by figure 9.

- step 1(Finding the *candidate polygons*): First, the evaluating node collects the position data of its linked nodes and makes an initial candidate polygon(CP) with them.

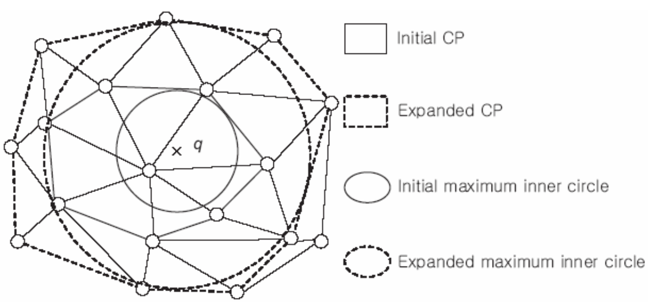


Figure 9. Expansion of candidate polygon(CP) and maximum inner circle

- step 2((Finding the *maximum inner circle*): Second, the evaluating node computes the maximum inner circle centered at the query position from the candidate polygon and find the nodes within this maximum inner circle. If more than or equal to k nodes are found within the maximum inner circle, the evaluating node refines the k nearest neighbor among them. Otherwise the third step is to be performed.
- step 3 (Expansion of CR and the maximum inner circle): If the number of nodes in the maximum inner circle is less than k, the evaluating node progressively expands the candidate polygon to the neighbors of the current candidate polygon till the maximum inner circle contains more than k nodes.

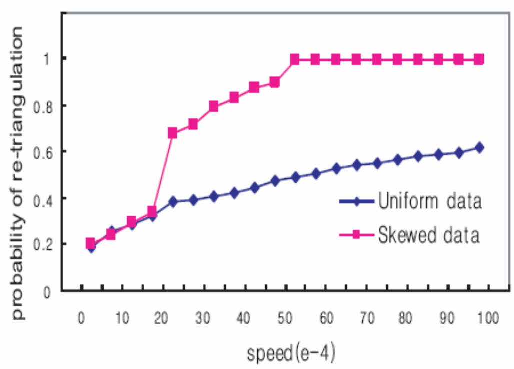
5. Experiments

We performed experiments to observe the cost of updates and query performance of DisTIN by simulation. We generate two synthetic data sets consisting of 10,000 nodes with uniform distribution and skewed distribution, respectively.

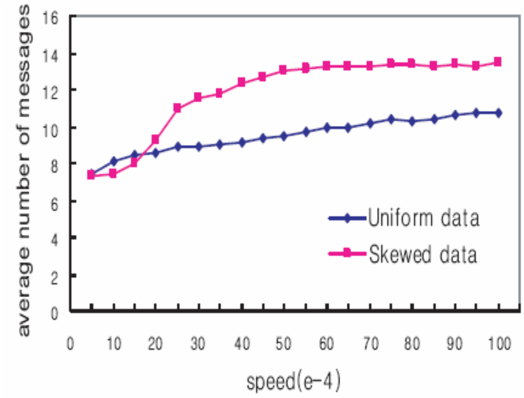
5.1. Experiments of DisTIN Updates

For the experiments of DisTIN updates, we observed the update cost with different node speeds. The speed is defined as the normalized distance of movement per one clock unit and the cost is measured in terms of the numbers of messages per one update. The results of experiments are given in figure 10.

First, we observed the probability that the node no more stays in the same triangle and re-triangulation should be consequently required as shown by figure 10(a). It is obvious that the probability increases as the speed becomes fast.



(a) Average number of employed neighbor nodes



(b) Average number of messages

Figure 10. DisTIN update cost

While the probability early approaches 1 with the skewed data, the probability is relatively small for the uniform data set. It is because the triangles of skewed data set are smaller than in uniform data set and it results in more frequent changes of triangles. Figure 10(b) shows the update performance of two data sets. This graph is strongly related with the graph of figure 10. While the number of messages for the skewed data becomes saturated around 14, it gradually increases for the uniform data.

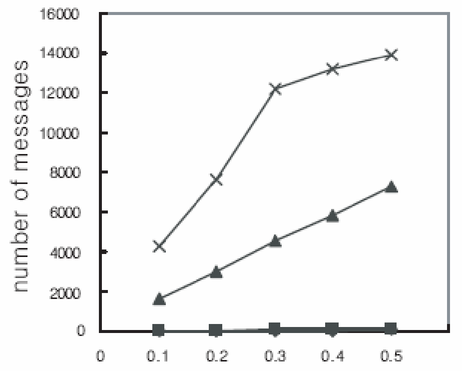
5.2. Experiments of k-Nearest Neighbor Query

The cost of spatial query by DisTIN is composed of the costs for routing and evaluation. But the cost of routing is more expensive than that of evaluation and we present the experiment results for routing phase. In the experiments, we compared the greedy method with the half-moon method in terms of three measures, the average number of messages, the employed nodes and the hop counts. We observed the performances for several query distances between the query position and source node where the query is submitted.

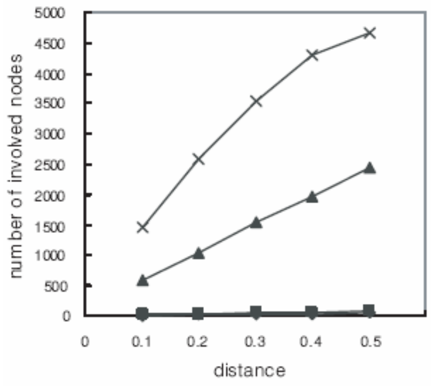
Figure 11(a) and figure 11(b) show that the number of messages and employed nodes of the half-moon method are considerably larger than the greedy method, even though the half-moon method gives better performances than the greedy method (see figure 11(c)). It means that it is preferable to apply the greedy method except the cases where the number of hop counts is extremely important.

6. Conclusion

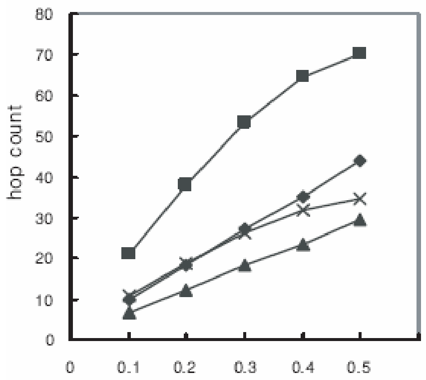
Although a number of methods have been proposed to process exact match or range queries, very few attention has been paid on spatial query process in P2P environment.



(a) number of messages



(b) number of employed nodes



(c) hop counts

Figure 11. Routing performances

This paper has two major contributions; 1) we propose a fully distributed spatial index for P2P environment, called DisTIN, and 2) this index supports the mobility of nodes. We also proposed a query processing method for *k*-nearest neighbor search in P2P environment by DisTIN.

DisTIN is based on a triangular network topology to respect two important requirements of distributed spatial index, dynamic property and spatial proximity. Our query processing method for *k*-nearest neighbor search is carried out by two phases; routing and evaluation phases. We expect that it can be easily extended to process other types of spatial queries in P2P environment such as range query.

To the best of our knowledge, it is one of the first distributed spatial indexing methods for P2P environment supporting the mobility of nodes. Future work includes several interesting research challenges. First, we need to extend the query types of DisTIN such spatial join. Second, it is necessary to consider the case where nodes move simultaneously, while we assume that only one node moves at a given time.

References

[1] Dimitrios Gunopulos, Vana Kalogeraki, and D. Zeinalipour-Yazti, “A Local Search Mechanism for Peer-to-Peer Networks”, pages 300–307. In Proceedings of ACM Conference on Information and Knowledge Management, 2002.

[2] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scoot Schenker, “A Scalable Content-Addressable Network”, pages 161–172. In Proceedings of ACM SIGCOMM Conference, 2001.

[3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakr-ishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, pages 149–160. In Proceedings of ACM SIGCOMM Conference, 2001.

[4] Antonios Daskos, Shahram Ghandeharizadeh, and Xinghua, “An. PePeR: A Distributed Range Addressing Space for Peer-to-Peer Systems”, pages 200–218. In Proceedings of Databases, Information Systems, and Peer-to-Peer Computing, 2003.

[5] Martin Frank, Jinbo Chen, Min Cai, and Pedro Szekely, “MAAN: A Multi-Attribute Addressable Network for Grid Information Services”, In Proceedings of Grid Computing, 2003.

[6] Ingo Brunkhorst, Hadhami Dhraief, Alfons Kemper, Wolfgang Nejdl, and Christian Wiesner, “Distributed Queries and Query Optimization in Schema-Based P2P-Systems”, pages 184–199. In Proceedings of the 1st International Workshop on Databases, Information Systems, and Peer-to-Peer Computing, 2003.

[7] Antony Rowstron, and Peter Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems”, In Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms, 2001.

[8] Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, Ben Y. Zhao, Ling Huang, and John Kubiatowicz, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing”, In IEEE Journal on Selected Areas in Communications, January 2004, Vol. 22, No. 1, pages 41–53, 2004.

[9] Chung-Ta King, and Hung-Chang Hsiao, “Bristle: A mobile structured peer-to-peer architecture”, In Proceedings of Symposium on Parallel and Distributed Processing, page 33.1, 2003.

[10] Artur Andrzejak, and Zhichen Xu, “Scalable, efficient range queries for grid information services”, In Proceedings of the Conference on Peer-to-Peer Computing, 2002.

[11] Jorg Widmer, Martin Mauve, and Hannes Hartenstein, “A Survey on Position-Based Routing in Mobile Ad-Hoc Networks”, volume 15, pages 30–39. In IEEE Network Magazine, 2001.

[12] Violet R. Syrotiuk, Stefano Basagni, Imrich Chlamtac, and Barry A. Woodward, “A Distance Routing Effect Algorithm for Mobility(DREAM)”, pages 76–84. In Proceedings of Mobile computing and networking, 1998.

[13] Young Ko, and Nitin H. Vaidya, “Location-Aided Routing(LAR) in Mobile Ad Hoc Networks”, pages 66–75. In Proceedings of Conference on Mobile computing and networking, 1998.

[14] 14. Brad Karp, and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless networks”, pages 243–254. In Proceedings of Conference on Mobile computing and networking, 2000.

[15] Bog-Ja Lim, Hye-Young Kang, and Ki-Joune Li, “P2P spatial query processing by delaunay triangulation”, In Proceedings of Workshop on Web and Wireless GIS, pages 136–150, 2004

[16] Prakash Linga, Adina Crainiceanu, Johannes Gehrke, and Jayavel Shanmugasudaram, “Guaranteeing Correctness and Availability in P2P Range Indices”, pages 323-335. In Proceedings of SIGMOD, 2005

[17] Karl Aberer, “P-grid: A self-organizing access structure for P2P”, page 179-194, In Proceedings of *CoopIS*, 2001.

[18] James Aspnes and Gauri Shah, “Skip Graphs”, page 384-393. In Proceedings of SODA Conference 2003.

[19] Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder, “Geometric Spanners for Wireless Ad hoc Networks”, pages 408-421, In IEEE Transaction on Parallel and Distributed Systems, Vol.14 No.4, 2003

[20] Xiang-Yang Li, Gruia Calinescu, Peng-Jun Wan, Yu Wang, “Local Delaunay Triangulation with Application in Ad Hoc Wireless Networks”, pages 1035-1047, In IEEE Transaction on Parallel and Distributed Systems, Vol.14 No.10, 2003.

[21] Xiang-Yang Li, Ivan Stojmenovic, and Yu Wan, “Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation”, pages 350-361, In IEEE Transaction on Parallel and Distributed Systems, Vol.15 No.4, 2004