仮想共有空間のためのドロネーオーバーレイネットワークの段階的構成法

大西　真晶[*1]　　西出　　亮[*2]　　上島　紳一[*3]

要　旨

本稿では，仮想共有空間実現のためのドロネーオーバーレイネットワークの段階的構成アルゴリズムを提案する．提案アルゴリズムでは，各ノードが他のノードに関する情報をもとにドロネー三角化手法を自律的に行い，局所的なドロネーネットワークを生成する．同時にノードが協調的に相互に情報を交換することにより，大域的なドロネーネットワークをベースネットワーク上に重畳的に構築することができる特徴を持つ．生成されたドロネーネットワークを用いて，ノード間の通信が行われ，遠隔ノード間についてはマルチホップ通信形態により通信が行われる．

本アルゴリズムは次の利点を持つ．つまり，(1)不必要な遠隔ノードを経由せずに地理的に局所的に存在するノード間で直接的に通信が行える，(2)ネットワークを常時更新続けることでノード数に関するスケーラビリティが得られる，(3)地図や地理空間などの２次元空間を段階的に構造化することができ，スーパーノードが地理的な領域を管理し，領域検索を可能とすることができる，(4)小さなサイズの経路表を構成して地理的ルーティング手法を構成できる，などの利点が得られる．本手法を用いることにより，ノードの地理的位置関係に基づいて，巨大な規模の仮想共有空間を用いた応用システムをスケーラブルに構築したり，位置指向情報システムを開発することが容易になるものと考えられる．

キーワード：仮想共有空間，ピアツーピア，計算幾何，逐次添加法，サーバーレス

[*1]関西大学大学院総合情報学研究科　[*2]関西大学大学院総合情報学研究科　[*3]関西大学総合情報学部

Incremental Construction of Delaunay Overlaid Network for Virtual Collaborative Space

Masaaki OHNISHI, Ryo NISHIDE, Shinichi UESHIMA
Graduate School of Informatics, Kansai University, Japan
{fa4d001,fa4d003,ueshima}@edu.kutc.kansai-u.ac.jp

Abstract

This paper proposes an incremental algorithm for constructing the Delaunay overlaid network for virtual collaborative space. In our algorithm, every node uses Delaunay triangulation, by using its knowledge of other nodes and cooperatively exchanging information among nodes, which generates and refines overlaid networks. On the resultant Delaunay network, nodes communicate with each other over virtual collaborative space, and employ multihopping communication among distant nodes. This algorithm structure has the following advantages: (1) utilization of only the localized nodes, omitting the use of unnecessary distant nodes for communication, (2) scalability through surveillance of nodes by refining and updating the network continuously, (3) two dimensional space organization which allows construction of super nodes that govern regional areas to process geographical range query, and (4) directional communication through a small routing table. The authors believe that this approach is applicable to large-scale virtual collaborative space and location aware system.

Keyterms: Virtual Collaborative Space, P2P, Computational Geometry, Incremental Method, Distributive Cooperation, Serverless

1. Introduction

Recently, much attention has been focused on construction of huge Virtual Collaborative Spaces (VCS) for communication and collaboration among nodes, e.g., human-to-human, human-to-machine, machine-to-machine, and so on[1,2,3]. For construction of VCS, two technical issues must be considered to realize the information infrastructure. One is to realize a mechanism for ad hoc communication among nodes distributed over a space, and the other is to find an efficient allocation scheme for CG data or location-dependent data gathered from sensors from outer environments. To build VCS efficiently, it is indispensable to build-in a geometry-based discussion.

In this paper, we propose a geometry-based P2P overlaid network as a scalable infrastructure for VCS, and show the incremental algorithm for constructing the overlaid network in a self-organizing manner. In VCS, nodes gather and build their ad hoc groups, communicate with other nodes in the groups, gathering information retrieved from their points of view, and take interaction mutually[4,5]. Thus, it is crucial to setup a geometry-based, efficient communication paths among nodes to reduce the communication cost on each node, since each node holds its own location and transmits information to other nodes in VCS. To the problem, an overlaid network is a preferable answer to avoid network congestion, since the total network load in a space becomes large. An efficient routing mechanism among nodes is required by considering the geometry of nodes, since nodes spread in a plane.

Furthermore, in designing behavior of nodes, we must take into consideration that all nodes autonomously generate node-to-node network, without their assumption that each node does not know all locations of other nodes distributed over the underlying space. Otherwise, we need a special node that has the whole view of the space, which does not guarantee scalability of the system. Therefore, node-to-node or peer-to-peer communication over geometrical spaces are desirable.

So far, many geographical routing mechanisms in node-to-node have been proposed to cope with flooding of requests among nodes distributed over two dimensional geographical plane[6,7,8,9,10,11,12]. In [6], they use storage abstraction in that (key,value) pair being hashed to geographical coordinates through a key, for data-centric storage supported by GPSR algorithm for actual storage. By defining the neighbor node as node to dictate the neighbor region, data is reachable by bounded number of hops in some practical scenarios. On the other hand, principles in computational geometry are introduced based on the underlying space, and geographical routing mechanisms are being proposed and built-in to some prototype systems[7,8,9]. These approaches focus on efficient networking or network formation of wireless and mobile communications. Some application-level overlays have been proposed for multicast, geographically-scoped queries in the context.

Our approach is classified in the latter. We employ Delaunay triangulation for nodes that reside in two dimensional geographical plane, where nodes represent machines, as well as autonomous resources, and users who work on terminals, such as mobiles, PDA's, and avatars[13]. We assume all nodes know their own location, and autonomously work on local triangulation by exchanging geometrical information of nodes to complete Delaunay triangulation for nodes in two dimensional plane incrementally.

We provide an incremental construction algorithm of the triangulation, which generates and refines overlaid networks over the base network. On the resultant Delaunay diagram, nodes communicate with each other, and employ multihopping communication among distant nodes. It should be noted that we assume no node knows the view of the entire space in our approach.

Our approach has the following advantages:

- **Localization:** Nodes communicate with only neighbor nodes, and organize incrementally in two dimensional space to obtain both Delaunay diagrams by themselves. The system localizes the influence of local behavior of nodes, and takes effects on distant nodes only when it is necessary, while this is difficult in server/client systems. In principle, communication among nodes nearby does not affect distant nodes in the space.

- **Scalability and dynamicity:** The overlaid network is scalable in nodes count. Our nodes are autonomous to keep on updating neighbor information in their routing tables by our incremental triangulation method, using the refinement of network only in node neighborhood. This guarantees the scalability of the network where nodes participate in network one by one according to the new nodes, user's log-in into the system, or node's ON/OFF, in the context of ad hoc network. In case all nodes are persistent and the network topology is static, our approach guarantees correctness to propagate information to the destination while the network is scalable throughout.

- **Space organization for efficient storage:** For a node in a plane, Delaunay triangular around the node shows and provides regional area to govern. Voronoi diagram, which is dual of Delaunay triangulation, can be handled simultaneously through node communication by our algorithm. The Delaunay node can be a node to store CG data, regional data of the node, and data gathered from outer environments. This node can also become the construction of super nodes that govern regional areas to process geographical range query.

- **Efficient directional routing:** The nodes that are spreaded in two dimensional plane need to create their routing tables for a limited number of neighbor nodes, and they communicate to all directions since the Delaunay diagram is a planar graph, which can reduce the communication load among nodes and network overloads. Nodes can communicate by constructing a small routing table of nearby nodes.

We believe our approach is applicable to (i) geographically distant world-wide database and (ii) location-based information system, such as GIS, navigation systems, and so on.

In Section 2, we describe the motivation of our approach, especially in the context of our project 'Digital Campus', that utilizes virtual collaborative space for educational point of view. In Section 3, we describe our basic idea and sketch of algorithm, and provide our incremental algorithm for Delaunay triangulation in two dimensional geographical spaces. Also, we show our simulated Delaunay triangulation in Section 4. Finally, we provide and compare the related works in Section 5.

2. Virtual Collaborative Space Built atop Overlaid Network

2.1. Motivation

We are constructing Digital Campus to be an educational platform for creating a new community over the network integrating the merits of both real and virtual collaborative spaces. It is contemplated as a one whole system accessible for both users on campus and in distant locations. We construct a precisely similar virtual space by mapping every element in real space to virtual space as components. We are trying to build different types of objects, artifacts, and users, etc. that reside in the space to work autonomously and collaboratively by using our layered cell modeling, which is an earlier version where we've tried to realize localized processing and communication among nodes[5]. Our goal is to realize a virtual space to accommodate with a flexible structure that incorporates all users, environments, etc. and enable intuitive and user-friendly interface, which correlates to the real space as well as accumulates information of objects, people, lectures, etc. from the real campus.

We construct node-to-node network over the virtual space to realize public sharing system for anyone to attend and utilize the resources, and users collaboration seamlessly for virtual communication in the university. In addition, by using node-to-node network technology, we try to realize scalability of the system in which there are no restrictions to number of participants. This system adapts a common sense of real university to users when utilizing virtual collaborative system, and enhances and supports activities of real university in virtual collaborative space.

Designing a structure to control user interface and communication software to create various communication groups is also taken into consideration. We realize university collaboration by connecting node-to-node network instantly for universities implementing this system, and merging them in virtual collaborate space.

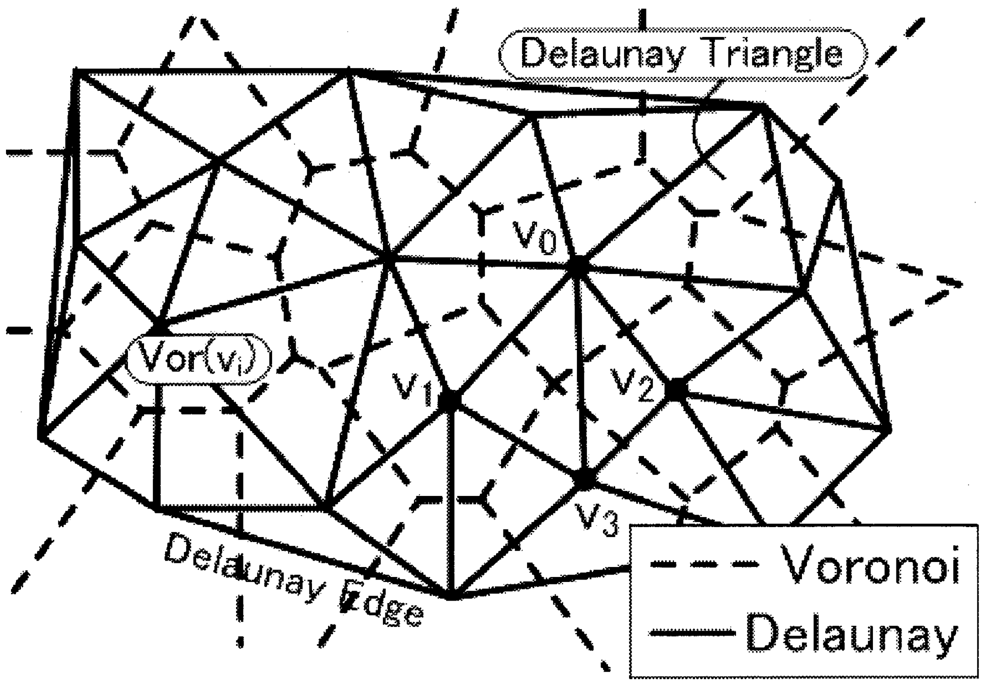


Figure 1. Voronoi and Delaunay Diagram (n=20)

2.2. Delaunay Triangulation and Voronoi Diagram

Here, we prepare terminologies and review briefly the concept of Delaunay triangulation and Voronoi diagram, as illustrated in Fig. 1. Let $V = \{v_1, v_2, \cdots, v_n\}$ be a set of nodes distributed in a plane. Assume there are no four nodes of V that are cocircular. A triangulation, denoted by Del(V), is a *Delaunay triangulation* if a circumcircle of each points of each of its triangles does not contain any other nodes of V in its interior. A triangular is called a *Delaunay triangle* if its circumcircle is empty of nodes of V inside. Edges of Delaunay triangular is called *Delaunay edges*. It is well-known that the Delaunay triangulation is a planar graph[13].

Voronoi diagram for V is a partition of plane into n Voronoi regions, one associated with each point v_i of V. Voronoi region, denoted by $Vor(v_i)$, associated with node v_i consists of all the points in the plane that are closer to v_i than any other node in V.

$$Vor(v_i) = \{x \mid d(v_i, x) < d(v_j, x), \text{ for } j \neq i\}$$

Furthermore, the followings are well-known facts: (i) $Vor(v_i)$ is a convex polygon (either bounded or unbounded) determined by bisectors of v_i and other nodes, and consists of O(n) line segments. (ii) Voronoi diagram for n nodes in a plane can be constructed in time complexity O(n).

The Delaunay triangulation and Voronoi Diagram are dual in the following sense. The Voronoi Diagram divides the plane into convex hulls corresponding to the number of nodes as the closest area of the node. The edges of the Voronoi polygon connect nodes in the neighboring area. That is, Delaunay edges provide neighbor nodes, while Voronoi region provides the space that the single node governs.

2.3. Advantages of the Proposed Method

By considering Delaunay triangulation and Voronoi diagram for VCS, the advantages beyond those merits that node-to-node paradigm can share are considered as follows:

- Coverage of a plane

  Full plane area can be divided by Delaunay Triangulation, and can reach to an arbitrary node on the plane by tracing along the Delaunay edges. Furthermore, any point or location in a plane belongs to $Vor(v_i)$ for some i. This implies that, when an arbitrary point in a plane is given, we can explicitly specify a node that governs the point. By this, location-dependent data in a plane can be stored in appropriate nodes in a load-balancing way.

- Directional routing

  Delaunay Diagram is exactly a planar graph, where a node can communicate to the neighboring nodes in any direction from an area as shown in Fig. 2 surrounding the node. Each node forwards messages to appropriate neighbors by comparing the location of neighbors and the destination. This enables to realize geographically-scoped multicast[7,10] by use of Euclidean minimum spanning tree, and geographically-scoped queries or geographical range queries in location-aware applications, such as sensor networks spread in large area, navigation systems that exploit GPS(Global Positioning System).

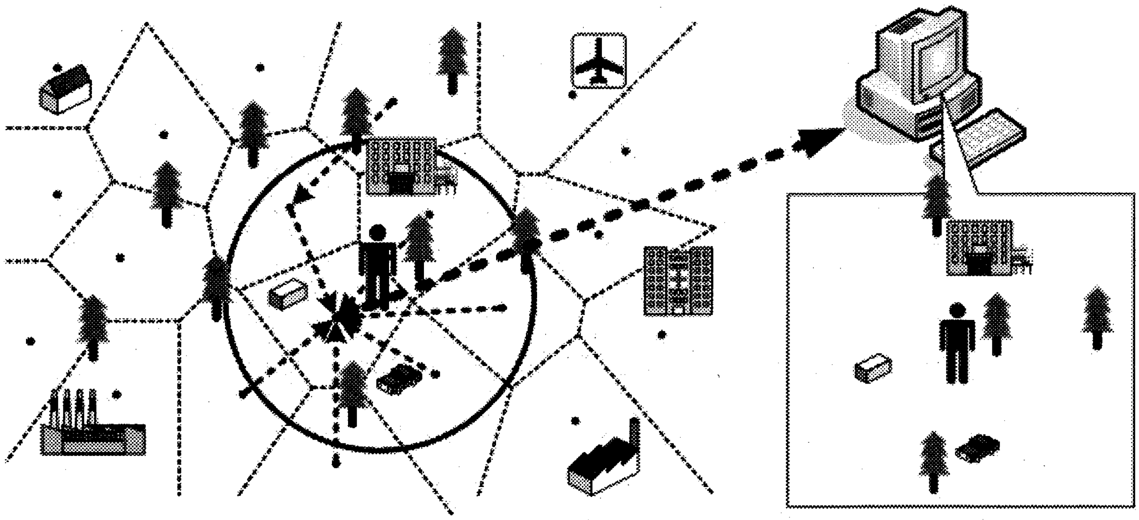


Figure 2. Delaunay Triangulation in Geographic Plane

- Reduction of the size of routing table

The total number of Delaunay edges is 3n - 3 - k, where n is the number of nodes and k is the number of points on the edge of the convex hull, respectively. The size of routing table of each node v_i to one-hop neighbor is equal to the number of Delaunay edges from v_i and O(1). It can also reach to the destination node using multihop communication by connecting each node to one of its neighboring nodes.

- Perimeter navigation of node

Suppose the neighbor of each node is sorted from the clockwise direction. Regions to govern by the node can be specified in two ways. One is triangulars defined by two adjacent neighbors and v_0. The other is a region defined by a perimeter, defined by all neighbors sequentially. When the node location information query is issued, the node absolutely exists in the area inside one step (one Delaunay edge) to the governed region, even if the node doesn't exist in that location.

- Super-node

As the designated area being clarified as a subdivision, each node can be used as Super-node, by which we can introduce multi-layered structure among nodes to cope with large number of nodes distributed over a plane.

Two nodes holding each other's information mutually are expressed as the edge in-between in Delaunay Diagram. Messages or data among nodes are sent along appropriate Delaunay edges to destination to establish connection between two or more nodes for communication in VCS.

2.4 Delaunay Network and Virtual Collaborative Space

The Delaunay network as an overlaid network provides the following advantages.

- Load-balancing and local data aggregation
  The constructing elements necessary for VCS, such as graphics data and avatar information, can be evenly stored at all nodes that are distributed in a plane. When allocating avatars in VCS which expresses the presence of users, or when user interaction in VCS is done through scopes of avatars, the system can visualize them with their surrounding environmental graphics, that are stored in consecutive regions. The path length for such requests propagation are expected to be short, and traffic to retrieve necessary data are also expected to be small consequently. In other words, requests propagation for retrieval of those data is localized. For any location-dependent or position-dependent events or data, there exists a node that can store and/or process them nearby the data. Hence VCS can be realized in a scalable way.

- Applicability to ad hoc grouping in VCS
  In the ad hoc network context, in case nodes join and leave or ON/OFF of power, Delaunay diagram can be updated since nodes are always exchanging location information with neighbors according to the incremental algorithm within the local area by our algorithm.

3. Incremental Construction of Delaunay Overlaid Network

3.1. Basic Ideas and Assumptions on Nodes

In this section, we describe our algorithm for incremental construction of Delaunay network as an overlaid network.

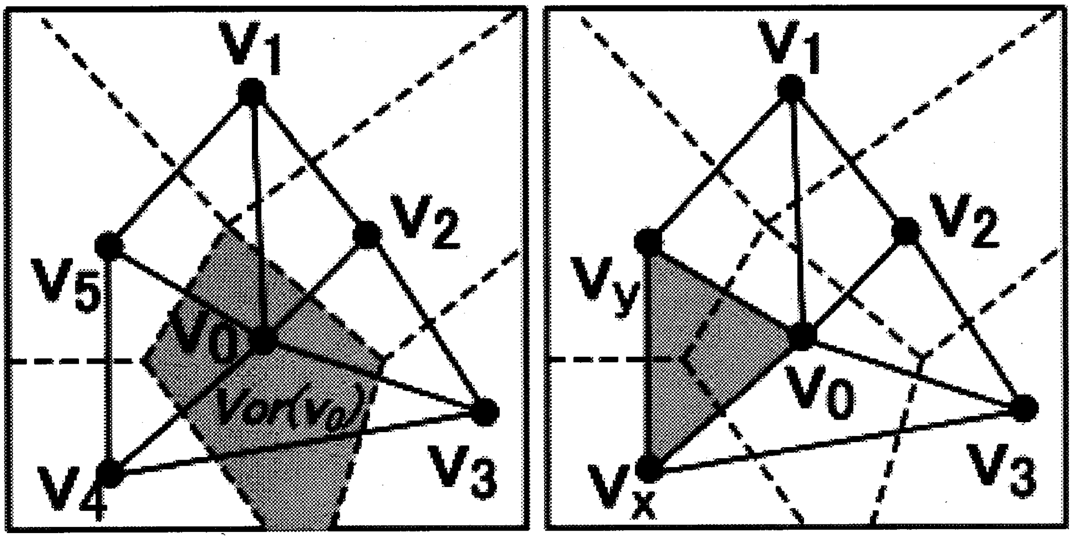
Nodes: We assume all nodes possess computational power, storage, network connectivity, know their absolute location on a plane, such as geographical location, and hold the list of other nodes' ID and location, to which each node can reach by one hop. More precisely, we introduce two assumptions on nodes' network formation to deploy practical scenarios to construct an overlaid network. That is,

- (A1) Each node is connected to the base network. We assume that two arbitrary points are symmetrically communicatable using base network in case receiver node's IDs are specified by sender node.

- (A2) Each node is provided multiple nodes that can be accessible by one hop, and holds them in its routing table.

  Under (A1) and (A2), we generate a triangular planar graph incrementally at each node in order to construct a whole Delaunay Diagram.

Basic ideas for incremental construction: In Delaunay triangulation, locations of all nodes are assumed to be known a priori. That is, Delaunay diagram cannot be constructed in a localized manner[13,14,15]. Hence, we must create a method and define a protocol that nodes employ for exchanging their location information among nodes in order to share advantages of Delaunay diagram in the context of practical node-to-node communication.



(a) Bounded Voronoi Region (b) Delaunay Triangle

Figure 3. Voronoi Region and Delaunay Triangle

To start with, we confirm the following observations in Delaunay diagram and Voronoi diagram as depicted in the left figure of Fig.3. The vicinity of v_0 is surrounded by all of the neighboring nodes' areas in the Voronoi diagram. Thus, v_0 is to be surrounded by continuous multiple sides (connected with the side of v_0) which connect nodes locating in the vicinity of v_0, similarly in the Delaunay diagram. This side (v_x, v_y) consequently constructs the Delaunay triangle (v_0, v_x, v_y) together with v_0 as illustrated the right figure of Fig.3.

According to the above observation, basic idea for our algorithm is that each node should autonomously select neighbor nodes to form a local Delaunay diagram, passing information of unnecessary nodes (those nodes that are not selected) to appropriate nodes, and repeat this procedure. In other words, information of unnecessary nodes travel to appropriate nodes step by step, while necessary information should be held in local regions. This approach is based on incremental method for Delaunay diagram generation in a centralized case[13], and revised to parallel and cooperative computation by autonomous nodes in a distributed network.

Here, we list some notations expressed in our algorithm:

- $\triangle v_a v_b v_c$: A triangle connecting nodes v_a, v_b, and v_c
- $C(v_a, v_b, v_c)$: A circumcircle of $\triangle v_a v_b v_c$
- AB * CD: Intersection of lines AB and CD
- $N(v_0)$: Neighbors of v_0, a set of nodes in routing table to which v_0 can reach each of $N(v_0)$ in one hop
- $LDN(v_0)$: A set of nodes that form local Delaunay diagram using a subset of $N(v_0)$ and v_0

Note that *Local Delaunay Neighbor* of v_0, $LDN(v_0)$, are nodes that satisfy the circumcircle condition.

To guarantee connectivity of two arbitrary nodes, we assume the following connectivity condition on the initial state of node connectivity.

- (Connectedness) In the initial state when nodes are given in a plane, nodes form a connected graph so that there are no isolated subgraph, nor no isolated node in it.

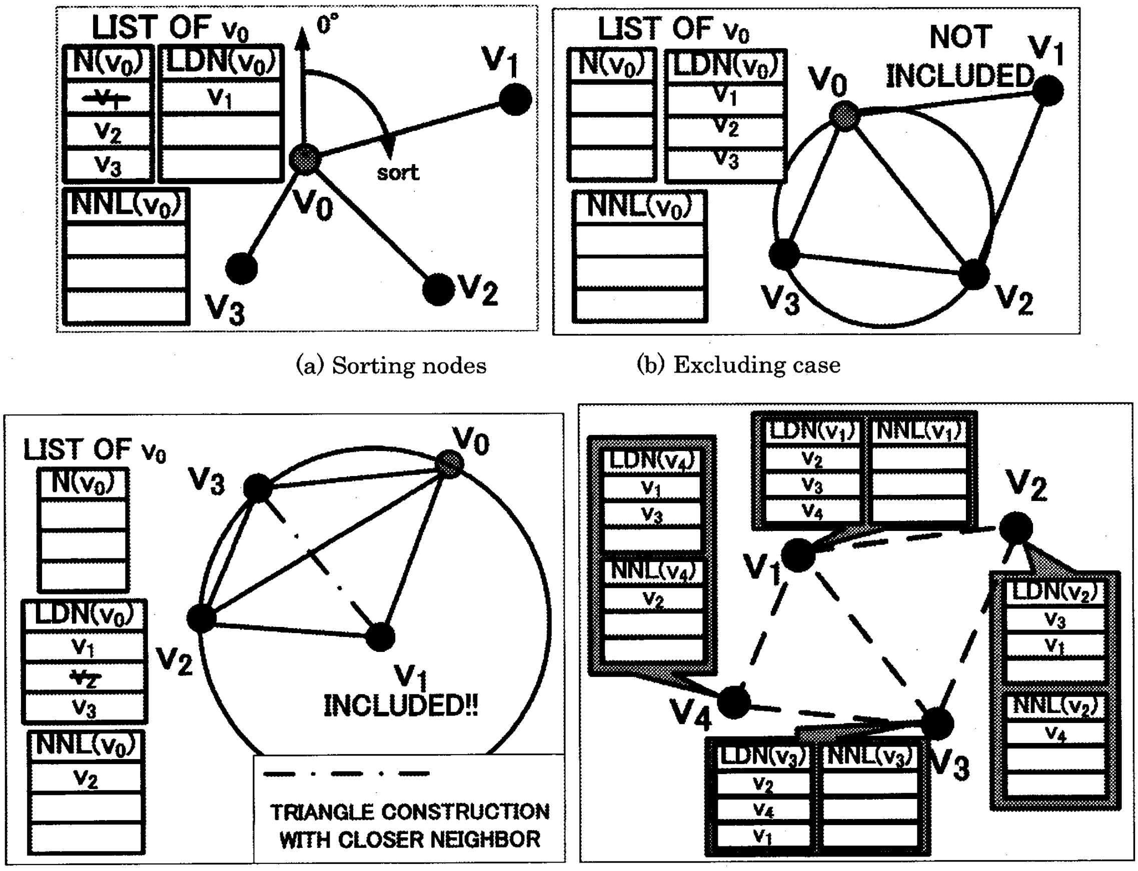
Recall the propositions in Delaunay triangulation as follows[13].

- **Proposition 1** v_0, v_1 and v_2 are the planar nodes of Delaunay graph if and only if no other nodes are included in $C(v_0, v_1, v_2)$. This condition corresponds with each Delaunay triangle in the plane.

Each node v_i forms a local Delaunay diagram satisfying the above necessary and sufficient condition given in Proposition 1 in a trial and error fashion, and passes information of unnecessary nodes to the nodes that should be holding it, other than v_i.

- **Proposition 2** Suppose two triangulars $\triangle v_i v_j v_k$ and $\triangle v_i v_j v_l$ share the edge $v_i v_j$. The edge $v_i v_j$ is illegal if and only if v_l is included inside $C(v_i, v_j, v_k)$. Further, one of $v_i v_j$ or $v_k v_l$ is illegal if v_i, v_j, v_k, v_l is not cocircular and form convex polygon.

Note that Proposition 2 is symmetric in that v_l is included inside $C(v_i, v_j, v_k)$ if and only if v_k is included inside $C(v_i, v_j, v_l)$. It is well-known that given an arbitrary triangulation for a set of nodes, we can obtain a legal triangulation by flipping the illegal edges, the sequence of which is shown in Fig. 4.



(a) Sorting nodes (b) Excluding case

(c) Including case (d) Neighboring relation

Figure 4. Resolution of Choosing Neighbors

Fig. 5 shows the brief process of recognizing each node as the Delaunay neighbor of v_0. The nodes suitable for Delaunay neighbor are transferred to LDN(v_0) list, while other nodes are transferred to other node list.
First, we create a list N(v_0) for the neighboring nodes of v_0. Then, we sort the nodes in the list N(v_0) clockwise according to the location from 0° north of v_0.

- All of the nodes in N(v_0) become the points of targeted Delaunay triangles.
- The other two points matching to v_0 are selected from the consecutive nodes in N(v_0), as $\triangle v_0v_1v_2$, $\triangle v_0v_2v_3$, $\triangle v_0v_3v_4$, ⋯ , $\triangle v_0v_{n-1}v_n$, $\triangle v_0v_nv_1$

Initially, in N(v_0), place all the nodes that v_0 has information. Then create a triangle with two consecutive nodes chosen from N(v_0).

Determine whether the nodes, prior to and following after the two consecutive nodes are included in the circumcircle of the triangle. If included, then the triangle is illegal. Remove the node or nodes of that triangle from the list. It is important to transfer the information of the node to be removed to one of Delaunay neighbors to guarantee the connectivity of the graph as shown in Fig. 6.

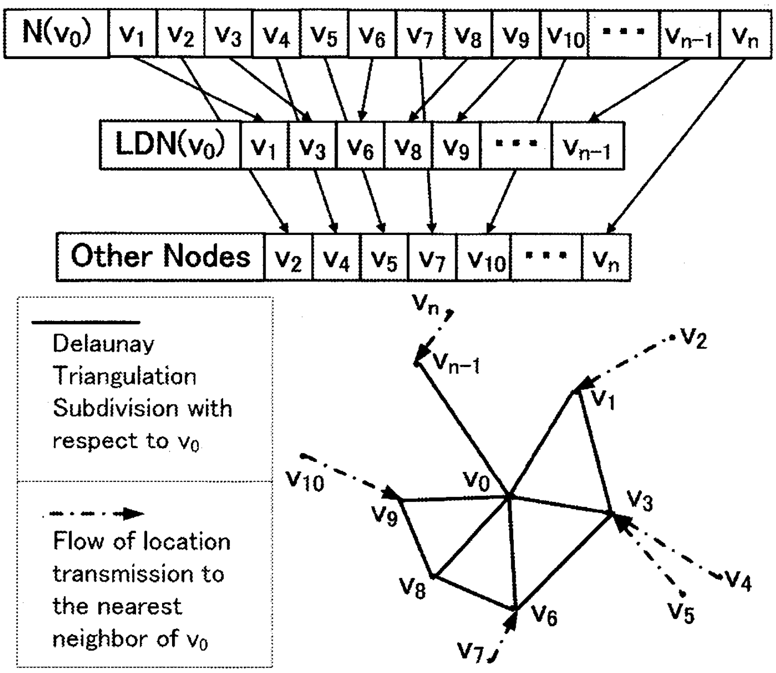


Figure 5. Knowing Their Delaunay Neighbors

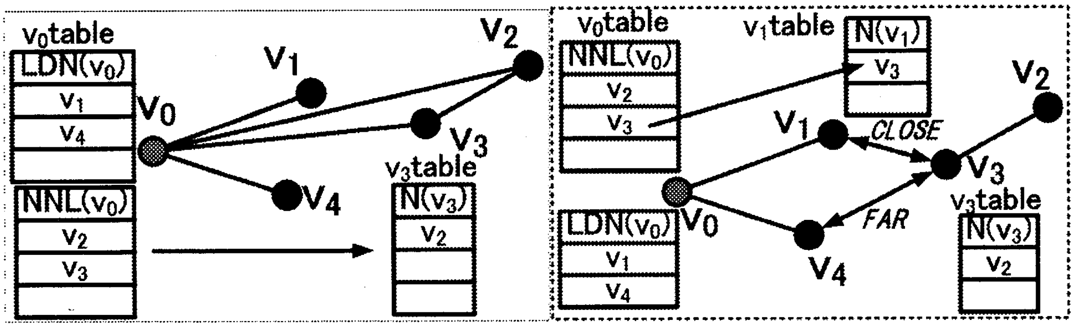


Figure 6. Transfer Info to Other Node

The list for neighbor nodes is considered complete when any circumcircle does not include other nodes of the list.

Consequently, we select LDN(v_0) for a given set of N(v_0), and unselected nodes are sent to other node list.

3.2. Algorithm

Here, we provide the algorithm.

Incrementing node in a plane: First, we briefly review the incremental method of the case that one single node is added to the existent Delaunay diagram, as shown in Fig. 7. Note that whole view of all nodes placed on a plane is known a priori.

1. Place the incrementing node v_{n+1} to the Delaunay diagram plane of n nodes
2. Randomly select one node v_x from n nodes
3. Select the nearest node to v_{n+1} from v_x and the Delaunay neighbor nodes of v_x
4. If the selected node is not v_x, then set the node as v_x, and send v_{n+1} node information to the node. Then, return to procedure on step 3. Otherwise, process steps 5 and 6 for nodes receiving v_{n+1} information.
5. Check if v_{n+1} is included in the circumcircles of all the triangles constructed by Delaunay neighbor nodes of v_x or v_x itself. If so, send v_{n+1} node information to the nodes constructing the triangle
6. Change the Delaunay neighbor node list of v_x from the node information held by v_x

With the above algorithm, a new node has been added in a plane to form a complete new Delaunay diagram. When these steps are processed correctly, the incremental algorithm over Delaunay diagram is reconstructed. But the receiver's process changes due to the transferred information through steps 4 and 5. It can be solved by (i) applying information recognizing which step (step 4 or 5) the information was transferred from, or (ii) process step 6 before step 3 and process step 5 if the transferred node information is the Delaunay neighbor of each node, if not, process steps 3 and 4. The comparison of incremental algorithm in computational geometry and P2P are shown in Tab. 1.

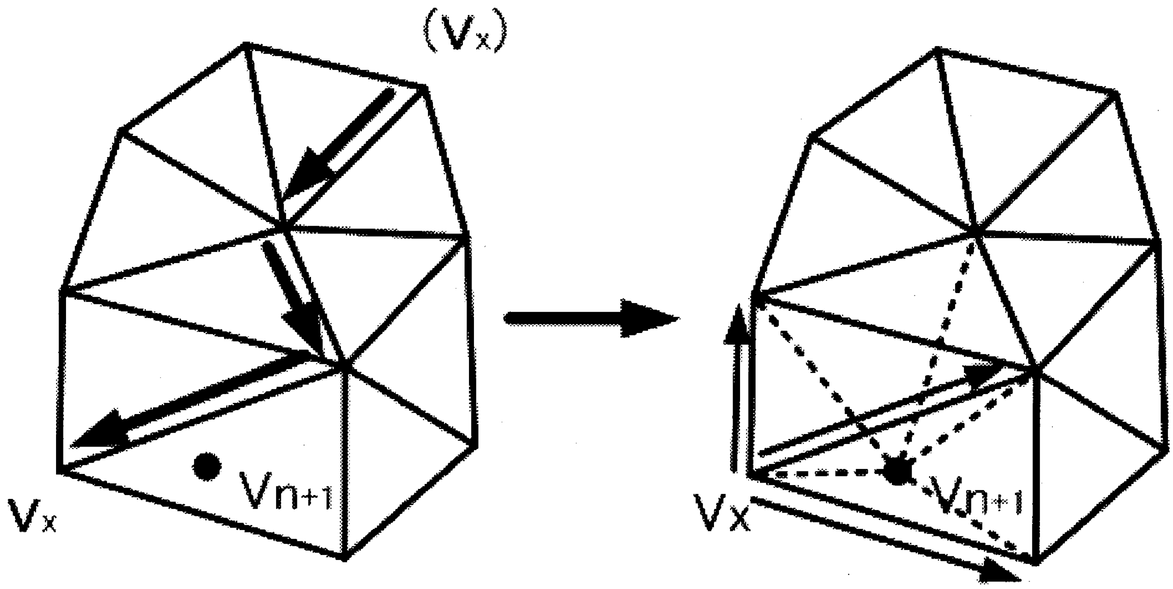


Figure 7. Incremental Algorithm

Table 1. Comparison of Incremental Algorithm in Computational Geometry and Proposed Method

Simple Method	P2P (Proposed Method)
Single node	Parallel and distributive nodes
Static	Dynamic
Whole view	Local view
Delaunay Triangulation	Local Delaunay Triangulation and transfer of location
by one-to-one communication	nodes collaboration

The incremental algorithm commonly used can only apply one node in the Delaunay graph at a time, and the network can not control all the nodes synchronously because each node only knows its neighbors. Therefore, it is necessary to come up with a solution of multiple node information being transferred from many directions synchronously, when multiple nodes participate in the Delaunay network at once. It can be solved by rejecting access of overwriting node information list by locking before entering this whole process. When the list is locked, the other node information can be cached, and can start processing with other node as soon as the process with previous node is finished and the list is unlocked.

Here, we list some notations used in our algorithm.

- $N(v_0)$: Node information list with respect to v_0
- $LDN(v_0)$: Neighbor node list of Delaunay Network with respect to v_0
- $NNL(v_0)$: Non-Neighbor node list of Delaunay Network with respect to v_0
- n : Number of nodes in the list
- $l_1, l_2, \cdots, l_n$: Nodes stored in sequential order in $NNL(v_0)$

These lists are used for records of their neighbors.

Here, we give our incremental algorithm for P2P system.

Step 1. The node information list is locked to reject access from other process.

Step 2. Construct $LDN(v_0)$ list from node information list.

Step 3. Send v_0 information to all of the neighbors in $LDN(v_0)$

Step 4. Transfer all of the node information of non-Delaunay neighbor list to each node closest in Delaunay neighbors, and remove the information from v_0 node information list.

Step 5. Send the node information in $LDN(v_0)$ to all of the Delaunay neighbor nodes, which constructs a Delaunay triangle with v_0 and v_0 neighbor.

Step 6. Unlock the node information list of v_0. When information of nodes are sent from other nodes, each node receives and stores the information, and is added into the node information list when it is unlocked.

Assume that a triangle (v_0, v_1, v_2) with points v_0 and two arbitrary points v_1 and v_2 that lie on a plane spanned by more than three elements of points V. The v_0 here refers to the main node for processing. From step 2, we prepare neighbor node list and non-neighbor node list. This step recognizes the Delaunay neighbor and categorizes every other nodes by constructing a list of Delaunay neighbor list and non-Delaunay neighbor list from $N(v_0)$.

Creating relation with nodes: The processes in step 2 in detail are as follows:

2-1. Sort the node in $N(v_0)$ clockwise according to the location from 0° north of v_0

2-2. Move the first information from $N(v_0)$ to the end of $LDN(v_0)$.

2-3. When the number of elements of $LDN(v_0)$ are more than three, then draw a circumcircle from $\triangle v_0 l_{n-1} l_n$ from $LDN(v_0)$.

2-4. If the following two conditions are satisfied:

Condition 1: $v_0\ l_{n-2} * l_{n-1}\ l_n$= "false"

Condition 2: $C(v_0, l_{n-1}, l_n) \supseteq l_{n-2}$

then move l_{n-1} to the $NNL(v_0)$ and repeat step 2-3.

2-5. If the node information still remains in $N(v_0)$, repeat step 2-2.

2-6. When the number of elements of $LDN(v_0)$ is more than three, then draw a circumcircle of $\triangle v_0 l_1 l_n$ from $LDN(v_0)$.

2-7. If the following two conditions are satisfied:

Condition 1: $v_0\ l_{n-1} * l_n\ l_1$ = "false"

Condition 2: $C(v_0, l_1, l_n) \supseteq l_{n-1}$

then move l_n to $NNL(v_0)$, and repeat step 2-6.

2-8. When the number of elements of $LDN(v_0)$ are more than three, then draw a circumcircle of $\triangle v_0 l_1 l_n$ from $LDN(v_0)$.

2-9. If the following two conditions are satisfied:

Condition 1: $v_0\ l_2 * l_n\ l_1$ = "false"

Condition 2: $C(v_0, l_1, l_n) \supseteq l_2$

then move l_1 to $NNL(v_0)$, and repeat step 2-8.

2-10. End the process

While the $LDN(v_0)$ is used for recognizing v_0's neighbors, we use $NNL(v_0)$ for process in step 4. It gives non-neighbor information before permanent removal. Here, we construct a copy list of $NNL(v_0)$, denoted as $CPL(v_0)$. The specified processes for step 4 are described as follows:

4-1. Create a list of copies of $NNL(v_0)$

4-2. Transmit l_1 in $CPL(v_0)$ to the closest node among all the nodes in $NNL(v_0)$ and

LDN(v_0) other than l_1

4-3. In case, the receiver is the node in the NNL(v_0), move l_1 to the node in the LDN(v_0) closest to the receiver node

4-4. Return to step 4-2 if the node information remains in CPL(v_0)

4-5. End the process

Remark 1: Convergence to Delaunay Diagram

When these steps are processed through all the nodes in the plane, the Delaunay triangle can be formed by connecting the neighbor nodes for each node.

In case all nodes are connected to form a connected graph at the initial stage by processing the above-mentioned steps to every nodes, all relative connections applied to nodes in the node list, which possess all the nodes, will eventually converge to a whole Delaunay connection.

An edge connecting two nodes, forms a small Delaunay network. If the new node information is added to any of these two nodes, it represents that the new node is added to this Delaunay network, which increases the number of nodes of the Delaunay network by one.

If the network is a connected graph, this small one edge Delaunay network merges other node that connects with each node, and gradually expands the Delaunay network. Consequently, the Delaunay network can be constructed by acquiring the whole network of the connected graph. Thus, it should be noted that our algorithm does not include a procedure to break connectivity of a connected graph through the whole process..

Every connected graph is formed by connecting multiple of one edge Delaunay networks. The whole Delaunay network expands to a size which equals to n in the connected graph. The shape of the Delaunay network corresponds to the locations of n nodes, and thus all the Delaunay networks are shaped in that way. Thus, it can be considered that the relative connection of all the nodes converges to Delaunay diagram connection.

Remark 2: If we assume a perfectly static network topology, our algorithm can deliver messages to a specified position by routing messages along Delaunay edges. Furthermore, it can deliver them even when network topology dynamically changes due to node failure or move, since every node always executes local Delaunay triangulation by exchanging location information with distant nodes.

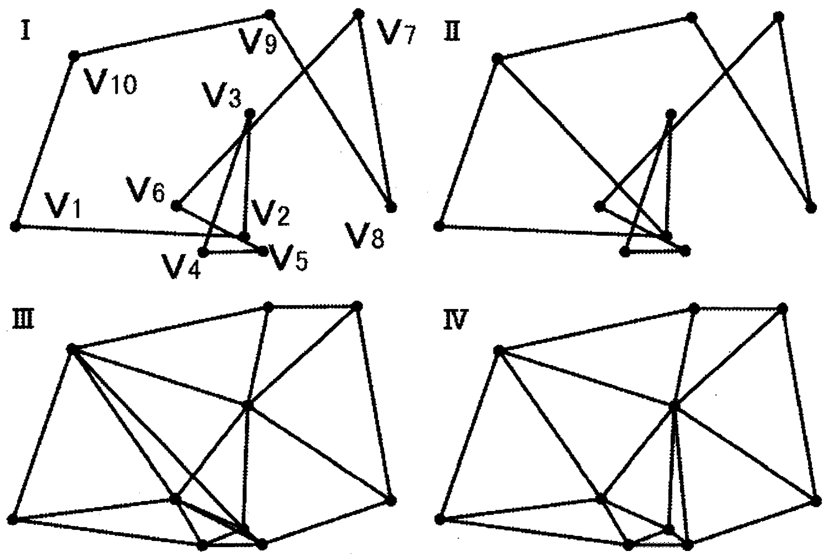


Figure 8. Simulated Delaunay Diagram (n=10)

3.3. Simulated Delaunay Diagram

We have implemented behavior of nodes according to our algorithm, and built up a simulator. Fig. 8 shows the stepwise process through (I) to (IV) of Delaunay triangulation for n=10. Nodes are allocated randomly on a plane and the initial connection of each node is set to 2. The figure shows lines that represent connectivity between nodes that intersect multiply. Nodes autonomously repeats local Delaunay triangulation while transferring unnecessary nodes information to appropriate nodes. Finally, all nodes complete Delaunay Triangulation where no lines intersect.

The simulator is coded by using Java2 Standard Edition(Ver.5) with IDE: Eclipse 3.0. Loads of class file in the whole simulator is 38KB. The simulator works on multithreaded environments, each threads an autonomous node working on the Delaunay Triangulation in a localized manner as described in 3.2.

The technical features of the simulators are as follows:

- It has a basic structure of P2P as each nodes can relay message to the neighboring node
- Although the nodes are constantly in operating condition, they can communicate only with the neighboring nodes
- Each node works independently, and as a result, it has a structure of Delaunay Diagram

4. Related Works

We here compare our approach with related works on node-to-node or peer-to-peer system distributed over geographical two dimensional plane. We describe a representative geographic routing mechanism called GHT(Geographic Hash Table) in the field of data dissemination required in the large-scale sensor networks[6].

The approach of GHT (Geographic Hash Table) for DCS (Data-Centric Storage) uses a data-centric canonical dissemination method, which a data object is correlated with a key and each node is designated to store a certain range of keys. This enables nodes to store or retrieve data according to their keys and thereby set-up a (key, value) based hash table. This key k is hashed into geographic coordinates. This provides storage abstraction, which guarantees invulnerability of storage even if nodes fail or move. The algorithm is developed out of GPSR in perimeter mode, which routed the packet to the vicinity of the destination. Algorithms of PRP (Perimeter Refresh Protocol) and SR (Structured Replication) are then described which allowed GHT to refresh the data stored for that key and which addressed the scaling problem, respectively.

Brad Karp, et al. proposed a routing protocol, a GPSR (Greedy Perimeter Stateless Routing) that forwards packet to its destination via geographically located routers [16]. Generally, GPSR is employed to realize real message packet routing mechanism. It is a geographical routing system, which uses a location server mapping all location of nodes for storing as well as for retrieving. All routers need to hold the state of local topology only and so each node knows all of its neighbors' position. Nodes' neighbor tables are always kept updated for those nodes within its own radio range. The algorithm consists of two phases; greedy forwarding and perimeter forwarding. During the process, when the greedy forwarding fails, it switches autonomously to perimeter mode.

Our approach is characterized by a planar graph, Delaunay diagram. This diagram is determined by the mutual location or position of nodes spread over two dimensional plane. We embed in the diagram our system principle that regional information near the node should be maintained nearby nodes, which is represented by the localized nature of the diagram. Furthermore, the diagram gives neighbors information of each node, at the same time. Each node routes messages according to the Delaunay edges, the number of which is limited as o(1). In our system, due to the move and/or retire of nodes, diagram must be updated, while nodes update the current diagram by themselves by exchanging routing tables with neighbors only in a limited one-hop neighborhood.

In GPSR, they employ perimeter forwarding and updation protocol in case the greedy forwarding does not work. While in our system, we do not use perimeter since the Delaunay graph itself is a planar graph, and each node is surrounded by a convex polygon in a plane.

Principles in computational geometry are introduced based on the geometry of the underlying space, and geographical routing mechanisms are being proposed and built-in some prototype systems. Regarding Delaunay triangulation, some researches employ the Delaunay diagram to cover geographical area, and to realize an efficient application-level networks. In [7,10,8,17], Delaunay triangulation of geographical space is employed for efficient topology and routing mechanism as overlaid networks. In [7], they provide local Delaunay triangulation algorithm by use of several messages to detect if the nodes be alive. They show numerically that GeoPeer is not affected by node density in a plane with their hop level mechanism for long range contact (LRC). They aim to construct applications for location-constrained queries, and information dissemination. [8] uses the similar approach for localized scatternet formation for multihop Bluetooth-based personal area ad hoc networks. [17] employs Delaunay Triangulation in application-layer for geographically-scoped multicast. These approaches discuss dynamic aspects of overlaid networks.

The proposed method can construct a Delaunay communication network enabling symmetric communication by structuring subdivision of space step by step, through exchanging location information between nodes locally. This enables flexible structured network allowing for new participation and retirement of nodes to search through whole available nodes in geographical space.

As for constructing Voronoi Diagram, incremental algorithm and recursive bisection algorithm are widely known. Our method differs from the point of constructing step by step Delaunay Triangulation Subdivision (paired with Voronoi Diagram) by transferring location information between nodes.

Li, et al. proposes an ad hoc network targeted for a short-range wireless communication device such as Bluetooth, when constructing an overlaid network. This method constructs partial Delaunay diagram from network edges expanding the network incrementally. This enables multihop connections between communication devices using efficient geographical routing by tracing the Delaunay sides on the plane. In this case, the base network becomes an overlaid network over local communication Bluetooth.

The precondition is that all the generatrices are given when the Delaunay triangulation subdivisions are constructed from the set of generatrices on the plane, considering the Delaunay diagram from the geometrical point of view. In other words, Delaunay diagram cannot be constructed in the localized condition. But there are no cases where all the node information is known beforehand as in the context of wireless ad hoc network, this approach has been proposed.

Our method constructs an ad hoc network by utilizing IP addresses as a global network. The Delaunay diagram of the whole plane cannot be constructed from only the local node information, as discussed above.

Therefore, our method constructs a Partial Delaunay Triangulation for each node as its center from the obtainable nodes information, applying the necessary and sufficient conditions for Delaunay diagram construction. At this point, it is important to process the unnecessary nodes in the structure. The algorithms are used here in order to send information to other node which is more appropriate than the main node, while maintaining connectibility with the Partial Delaunay Triangulation. This process is performed for all the nodes autonomically. Assuming that this process is one stage, the stage will be repeated until all of the information are transferred. Specifically, the algorithms are in a duplex structure. Each stage constructs a partial Delaunay diagram locally, while preparing for the information transfer to other nodes, and processes the information received from the next stage. The algorithms repeatingly process these steps.

The distinctive features of this method are to propose to construct a Delaunay diagram for a whole plane applying a distributive cooperative method, assuming that each nodes are autonomous. This allows efficient geographical routing by tracing triangle edges of the subdivided full planar region.

The base network has the structure of the overlaid network for global communication. A general overview comparison table of current proposed system to GPSR, GHT, and PDT (Partial Delaunay Triangulation) is shown in Tab. 2.

Table 2. Comparison of Related Systems and Proposed System

Items	GPSR	GHT	PDT	Proposed System
Node location information	Used	Used	Used	Used
Target for subdivided space	Geographical Space	Geographical Space	Geographical Space with sparse graph in 2D plane	Geographical space with map and CG plane
Unit of division	Planar graph	Planar graph	Planar subgraph	Convex polygonal Voronoi area
Work of Construction	Simple	Complicated	Complicated	Complicated
Scalability	Very Scalable	Very Scalable	Very Scalable	Very Scalable
Resilience of Network	Weak	Strong	Strong	Strong
Insertion/Deletion of node	Update nodes' neighbor tables of those nodes within its own radio range	Update the node information in the hash location of the key	Reconfigure the bluetree, a rooted spanning tree assigned with nodes of master-and-slaves roles	Update of local polygon through location information transfer to neighbor node (Delaunay Triangulation Subdivision)
Region search	Not Easy (Query to next hop neighbor node closest to destination, and follows successively to closer geographic hops)	Not Easy (Query the vicinity of destination initially)	Easy (Query to scatternet formations, which proceeds with iterations)	Easy (Query to optional geographical partition region)
Applicable area	Mobile network applications	Large-scale sensor network	Short-range wireless communication device	Geographical based Application (GIS, 3D CG)

5. Concluding Remarks

We have proposed an incremental algorithm of constructing the Delaunay overlaid network for virtual collaborative space. We have described our basic idea and sketch of algorithm, and provided our incremental algorithm for Delaunay triangulation in two dimensional geographical spaces. We have also shown the simulation results, and compared with the related works. As for future works, there are expansion of routing table, construction of long range contact (LRC), efficient data allocation such as CG data, and tracing avatar movement.

Acknowledgement

This work is supported in part by 'High-Tech Research Center' Project for Private Universities: matching fund subsidy from MEXT(Ministry of Education, Culture, Sports, Science and Technology) for ORC(Open Research Center) Project and CCC(Cyber Campus Consortium).

References

[1] Croquet Project, http://www.opencroquet.org/

[2] Andrew Rosenbloom, "SPECIAL ISSUE: Interactive immersion in 3D graphics", CACM, Vol. 47, Issue 8, pp.28-31 (2004.8)

[3] M. McCahill, J. Lombardi, "Enabling Social Dimensions of Learning through a Persistent, Unified, Massively Multi-User, and Self-Organizing Virtual Environment", Proc. Conference on Creating, Connecting and Collaborating through Computing (C5), pp.150-156 (IEEE CS Press) (2004.1).

[4] R. Nishide, S. Ueshima, "Digital Campus Project: A Dream University over the Internet Web", Proc. Conference on Creating, Connecting and Collaborating through Computing (C5), pp.104-111(IEEE CS Press) (2004.1).

[5] R. Nishide, I. Kimura, M. Ohnishi, S. Ueshima,"Modeling of Dynamic Behavior of Multi-Avatars In Virtual Collaborative Space", Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM2003) (2003.6)

[6] S. Ratnasamy, B.Karp, L.Yin, F.Yu, D.Estrin, R.Govindan, S.Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage", Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA'02), pp.78-87 (2002.8)

[7] F.Araujo, L. Rodrigues, "GeoPeer: A Location-Aware Peer-to-Peer System", Proceedings of the 3-rd IEEE International Symposium on Network Computing and Applications(NCA'04), pp.39-46, (2004.9)

[8] X.-Y.Li, I. Stojmenovic, Y. Wang, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation", IEEE Trans. Parallel and Distributed Systems, Vol.15 No.4, pp.350-361 (2004.4)

[9] X.-Y.Li, G. Calinescu, P. Wan, Y. Wang, "Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks", IEEE Trans. Parallel and Distributed Systems, Vol.14 No.10, pp.1035-1047 (2003.10)

[10] I.Stojmenovic, A.Prakash Ruhil, D.K.Lobiyal, "Voronoi diagram and convex hull based geocasting and routing in wireless networks", Proceedings of the 8-th IEEE International Symposium on Computers and Communication (ISCC'03), pp.51-56 (2003)

[11] Z. Shen, J. Thomas, "Location Triggered Routing Protocol in Mobile Ad Hoc Networks", Proceedings of the 3-rd IEEE International Symposium on Network Computing and Applications (NCA'04), pp.261-268, (2004.8-9)

[12] J. Liebeherr, M. Nahas, W. Si, "Application-layer multicasting with Delaunay triangulation overlays", IEEE Journal on Selected Areas in Communications, No. 8, pp. 1472-1488, (2002.10)

[13] M. de Berg, M., M. van Kreveld, M.Overmars, M., O. Schwarzkopf, "Computational Geometry : Algorithms and Applications", 2nd ed., Springer (2000)

[14] X.-Y.Li, "Algorithmic, Geometric and Graphs issues in Wireless Networks", Wireless Comm.

and Mobile Computing, Vol.3, no.2, pp.119-140 (2003)

[15] F.P.Preparata, M.I.Shamos,"Computational Geometry: An Introduction",Springer-Verlag, (1985)

[16] B. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000), pp.243-254, (2000.8)

[17] J. Liebeherr, Tyler K. Beam, "HyperCast: A Protocol for Maintaining Multicast Group Members in a Logical Hypercube Topology", Proceedings of First International Workshop on Networked Group Communication (NGC '99), in: Lecture Notes in Computer Science, Vol. 1736, pp. 72-89 (1999.7).

[18] S. Ratnasamy, P.Francis, M.Handley, R.Karp, S.Shenker, "A Scalable Content-Addressable Network", ACM SIGCOMM Computer Communication Review , Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, Volume 31 Issue 4, pp.161-172 (2001.8)

[19] S. Ratnasamy, M. Handley, R. Karp, S. Shenker", "Application-Level Multicast Using Content-Addressable Networks", Proceedings of Third International Workshop on Networked Group Communication (NGC '01), in: Lecture Notes in Computer Science, Vol. 2233, pp.14-28, (2001)

[20] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, "Geographic Routing without Location Information", Proceedings of the Ninth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2003), pp.96-108 (2003.9)

[21] I. Stoica, R. Morris, D.Karger, M. Kaashoek, H. Balakrishnan "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", Proceedings of the ACM SIGCOMM '01 Conference, pp.149-160 (2001.8)

[22] M. Kleis, X. Zhou, "A Placement Scheme for Peer-to-Peer Networks based on Principles from Geometry", Proceedings of the 4-th International Conference on Peer-to-Peer Computing (P2P' 04), pp. 134-141 (2004.8)

[23] C. Schmidt, M. Parashar, "Enabling Flexible Queries with Guarantees in P2P Systems", IEEE Internet Computing (IEEE CS), pp. 19-26, (2004.5-6)