数理解析研究所講究録
第 1641 巻 2009 年 135-143

Geometric Problems on Ad-Hoc Network Design

Takeshi Tokuyama *

Abstract

In this talk, we consider two geometric problems on the design of ad-hoc networks: Geometric routing problem and interference minimization problem. Both are formulated as computational geometric problems, and familiar tools such as local neighbor graph and epsilon nets can be applied to design nice algorithms. We introduce recent results on these problems, and discuss open problems that are attractive to computational geometers.

1 Introduction

Ad hoc wireless network is a vital infrastructure in recent information society, and an emerging area of active research. From the viewpoint of theoretical computer science, design of efficient algorithms on several network problems on an ad-hoc wireless network is an important subject. A model of a wireless network is as follows: Given a set $S = \{p_1, p_2, \ldots, p_n\}$ of n points in a plane corresponding to the router nodes of the network. Each router node p_i has its transmission radius r_i, and it can send message to a node in its transmission area, that is, the disk of radius r_i around p_i. Thus, two router nodes can directly communicate each other if they are in the transmission disk of each other.

For simplicity, we assume that the maximum transmission radius of each router node is 1. Hence, if each router node uses its maximum power, $r_i = 1$ for each i, and the communication network forms a graph called unit disk graph $UDG(S)$.

The unit disk graph is a classical object in computational geometry, and its property is well studied. $UDG(S)$ is often very dense if the points are near to each other, and we usually use its sparce subgraph to realize the network connection. The selection and construction of the subgraph depends on applications and objective of the ad-hoc network. Since energy is the limiting factor for the operability and lifetime of these networks, various mechanisms have been developed to conserve energy. These are collectively called topology control, and the most popular subgraph is the minimum spanning tree.

However, it is often advantageous to use other types of networks. In this paper, we focus on the *geometric routing* problem and *interference minimization* problem in an ad-hoc wireless network, and demonstrate that local neighbor graphs are useful in ad-hod network design.

*GSIS, Tohoku University, Japan. (tokuyama@dais.is.tohoku.ac.jp

2 Geometric routing problem

If one would like to send a packet from a given node s to its destination t in a network, one need to find a path from s towards t. In the ad-hoc network, the packet is sent along a path P in the graph $UDG(S)$ from s to t. If the whole graph structure is known to each node, the problem is just the shortest path finding problem (or its variant). However, in an ad-hoc network, each node knows little about the whole network. Thus, basically each packet explore its own path independently. No additional space is given to each packet to store its history of exploration. A famous method is called *flooding*, for which router nodes send message packets to find a path, which often need to create a lot of messages on the network.

In the current information infrastructure, it is relatively easy to find the geometric locatin of a routers; GPS is such a system. Thus, it is reasonable to find the route by using geometric information of locations of routers. The *geometric routing* problem is given as follows: The geometric location of the source s and t are known when we send a packet, but the information of the path P is not given in advance. Each node v knows its exact geometric location, and can find location of nodes in its transmission radius by sending a control message and receive answers. At each node v on the path, the packet is forwarded to its adjacent node according to a protocol. The routing should be done by using only local information of each node. In a mobile ad-hoc network, the set S changes due to insertion, deletion, failure, and movement.

One popular approach is the *greedy routing*, in which the packet is sent to the neighbor of v that is nearest to t. An improved method is to define a connected spanning subgraph H (such as Delauney triangulation and Gabriel graph) of $G(S)$ and do routing on this graph. Unfortunately, the greedy routing fails if there is no neighbor node that is nearer to t than v. There are several research to design a geometric routing that surely send a packet. Bose et al [2] gave a method named *face-routing*, which always correctly send a packet on a planar graph H. Since the original face-routing method is quite slow, Kuhn et al [6] proposed the adaptive face-routing method to give improvement.

A practically better method is a hybrid method of the greedy routing and the adaptive face routing. We apply some greedy method, and if it gets stuck, we call an exceptional routine (face routing) to complete the routing. GOAFR$^+$ proposed by Khun et al [5] is such a system, in which the Gabriel graph is considered as H. One defect about the Gabriel graph is that it has too few edges and it calls the exceptional routine often. Another defect is that it is not cheap to maintain it in the dynamic situation.

In [?], we proposed a new routing scheme using random local neighbors. The scheme consists of a kind of greedy routing and an exceptional routine called *one-face* routing. It has a feature that it is easy to maintain dynamically and also we can be analyze its theoretical performance under a weak assumption of the distribution of the points. Our graph is not always connected even if the point set is distributed well. Although it may look strange to do routing on a disconnected graph, we overcome the problem of disconnectedness by additional control using the property of wireless communication. This enables us to prove that the greedy routing on our graph successes for a good point distribution. Precisely speaking, the routing never goes into the exceptional routine if each node on the route has its nearest local neighbor (in the direction of the destination node t) within its transmission radius. Note that a node can decide whether t is in its tranmission radius by just locally computing the distance between t and it; thus without sending any message.

For a general point distribution, the method may get stuck, and need a detour. Each node on the route can detect whether it needs an exceptional routine to find a detour and continue the routing. The exceptional routine is the one-face routing (that is a variant of face routing that only searches in one face) using the relative neighbor graph.

We theoretically guarantee that the routing does not fail. Moreover, we give theoretical analysis and experimental evaluation of the number of hops on the route and also the data space at each node for the dynamic structure under an assumption of the distribution of points. Moreover, the feature that we connect to random local neighbors enable us efficient maintainance of the network when each points moves or updated by insertion/deletion.

2.1 Directional Routing using Generalized Local Neighbor Graph

Let $D(v,r) = \{x \in \mathrm{R}^2 : d(x,v) < r\}$ be the disk of radius r around a point v in the plane, and we write $D(v)$ for the unit disk $D(v,1)$. The region $P(v,k) = \{p \neq v : (k-1)\pi/3 \leq \arg(vp) < k\pi/3\}$ $(k = 1,2,..,6)$ is called the k-th wedge around v, where $\arg(vp)$ is the argument angle of the vector vp. $Q(v,k) = P(v,k) \cap D(v)$ is called the k-th sextant around v.

We say that the point configuration S is *local-neighbor perfect* if $Q(v,k) \cap S \neq \emptyset$ whenever $P(v,k) \cap S \neq \emptyset$ for any v and k. By definition, any point configuration becomes local-neighbor perfect if the transmission radius is large enough such that the nearest point (if any) of $P(v,k) \cap S$ is in the transmission disk for each v and k.

Given a set S of n points in the plane, we define $F = (f_1, f_2, \ldots f_6)$ such that f_k is a map from S to $S \cup \{*\}$ such that $f_k(v) \in Q(v,k)$ if $Q(v,k) \neq \emptyset$ and $f_k(v) = *$ otherwise. The asterisk symbol means "none". The 6 points (or possibly $*$) $f_k(v)$ $(k = 1,2,..,6)$ are called the *local neighbors* of v associated with F. The *generalized local neighbor graph* associated with F is a directed graph on the vertex set S with directed edges $(v, f_k(v))$ for each $v \in S$ and $k = 1,2,\ldots,6$, in which we ignore edges towards $*$. The graph depends on F and is denoted by $\mathrm{GLNG}_F(S)$ formally; however, it is denoted by $\mathrm{GLNG}(S)$ if F is not explicitly considered. It is clear that $\mathrm{GLNG}(S)$ can be constructed in a distributed fashion by only using local information within $D(v)$ for each $v \in S$.

Example 1. If S is local neighbor perfect and $f_k(v)$ is the nearest point in $Q(v,k)$ to v, $\mathrm{GLNG}(S)$ becomes the *local neighbor graph* $\mathrm{LNG}(S)$, which is a popular graphs in computational geometry [8, 7, 4], and it has been used to

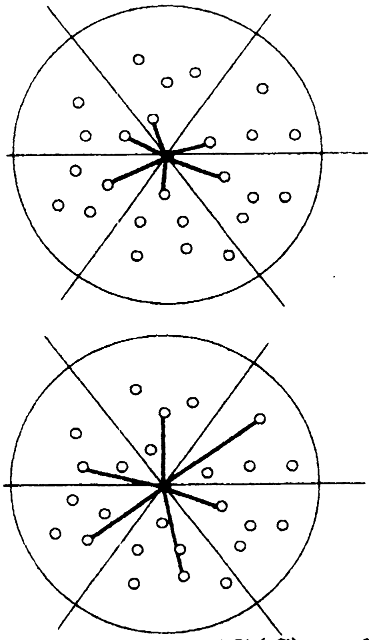

Figure 1: Neighbors in LNG(S) and RLNG(S)

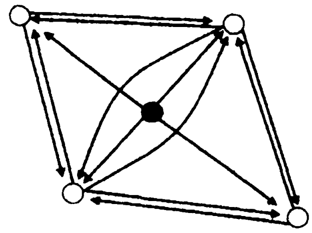

Figure 2: Disconnected GLNG(S)

design an ad-hoc network with low interference [3].

Example 2. If we choose $f_k(v)$ randomly from $Q(v, k)$ for each $v \in S$ and $1 \leq k \leq 6$, $GLNG(S)$ is called a random local neighbor graph and denoted by RLNG(S) (see Figure 1).

GLNG(S) is not always strongly connected as a directed graph even if S is local neighbor perfect as shown in the following counter example: Consider the configuration of five nodes colored into black and white given in Figure 2, where points are near to each other such that UDG(S) is the complete graph, and hence S is local neighbor perfect. Any white node has a white local neighbor in each nonempty sextant. Thus, we may choose white $f_k(v)$ for each white node v and each k, and obtain a graph shown in the picture. Thus, there is no directed path from a white node to the black node, and hence the graph GLNG(S) is not strongly connected.

One may feel absurd to consider a routing method using GLNG(S), since the black node in the above counter example is unreachable from other nodes in the graph. Nevertheless, utilizing the nature of a wireless network and geometric information, we can design the following routing strategy.

Directional routing algorithm:

1. Initialize $v = s$
2. If $t \in D(v)$, send the packet to t and complete the process.
3. Find k such that $t \in P(v, k)$. If $f_k(v) = *$, halt;
   otherwise send the packet to $f_k(v)$, set $v = f_k(v)$ and go to Step 2.

Note that the last hop (corresponding to Step 2 of the algorithm) in the directional routing need not be an edge of GLNG(S); this restores the connectivity of the network.

Lemma 2.1. *If the directional routing algorithm does not halt in Step 3, it successes to send the packet to t in a finite step.*

Proof. Consider any node v on the route. If $t \in D(v)$, the routing successes. Otherwise, the packet is sent to $w = f_k(v)$. Now, consider the distance $d(v,t)$ and $d(w,t)$. Since both are in $P(v,k)$ and $d(v,t) > d(v,w)$, we have $d(v,t) > d(w,t)$. Since the distance strictly decreases and S is a finite set, the packet is sent to t eventually. □

If S is local-neighbor perfect, $t \in P(v,k)$ implies $Q(v,k) \neq \emptyset$; therefore, the directional routing algorithm never halts, and we have the following:

Corollary 2.2. *If S is local-neighbor perfect, the directional routing method on* GLNG(S) *always successfully sends the packet from any source to any destination.*

2.2 Detour finding if the directional routing halts.

Let us consider the case where S is not local-neighbor perfect. The directional routing algorithm may halt by finding $f_k(v) = *$. For the case, we need an exceptional routine to continue the routing. We note that such a detour-finding routine is inevitable for any known geometric routing strategy.

The relative neighborhood graph is an undirected graph on the vertex set S such that it has an edge (u,v) if and only if there is no other node of S in the *lens* $D(u,r) \cap D(v,r)$ where $r = d(u,v)$. We consider the graph H(S) that consists of all the edges shorter than 1 in the relative neighborhood graph of S. The relative neighborhood graph contains a minimum spanning tree (MST) of S [1, 7], and accordingly, H(S) contains the MST, since MST only uses edges shorter than 1 if UDG is connected. Also, all the neighbors of a vertex v in H(S) are relative neighbors in LNG(S), and hence the degree is at most 6.

We name the following method the *one-face routing* on H(S). An advantage of the method to previous face routing algorithms [2, 5] is that we can give a theoretical bound of number of faces traced during sending a packet.

Suppose that the directional routing algorithm halts at v. Let R be the unique face of the planar graph H(S) such that R has v on its boundary and R intersect the segment vt in the neighbor of v. Let ∂R be the boundary chain of R. We trace on ∂R to find a vertex q called *exit point* sufficiently nearer to t than v. We may use the bidirectional tracing method given in [5].

Although the intersection point of ∂R and vt is nearer to t than v, it is not trivial to show that there is a vertex of ∂R nearer to t. Indeed, it is false for a general planar graph, and we need properties of H(S) and our assumption that UDG(S) is (hence, H(S) is) connected.

Consider a quadrant Δ of the unit disk $D(v)$ such that vt is the halving halfline of Δ. Let D be the disk of a radius $\max(r - \delta, 1)$ around t, where

$r = d(v,t)$ and $\delta \leq 1/2$ is a parameter. Naturally, if we find a point in D, then either the distance towards the target is decreased by δ or t is found in the next step. We can prove the following lemma:

Lemma 2.3. *If $\Delta \setminus D$ contains no point of S, there exists a vertex of ∂R in D.*

Proof. Connectivity of $H(S)$ assures that t is not an interior point of R, and there is an edge $e = (u,w)$ of ∂R intersecting vt. If there is a no vertex of ∂R in D, the edge e cuts D into two connected components as seen in Figure 3. Since e is an edge of $H(S)$, either $d(u,v) \geq d(u,w)$ or $d(w,v) \geq d(u,w)$ (otherwise the lens corresponding to e contains v contradicting the definition of H). We can assume that the former case occurs, and hence u is in the same side of w with respect to the perpendicular bisector of vw. The segment uw and the center t of D must lie in the same side of the bisector. Also, from the above fact and $w \notin \Delta \cup D$, we have $\angle wvt > \pi/4$. Thus, by elementary geometry, we can see that $d(u,w)$ is greater than the radius of D. Thus, the length of e is greater than 1, contradicting the definition of $H(S)$. □

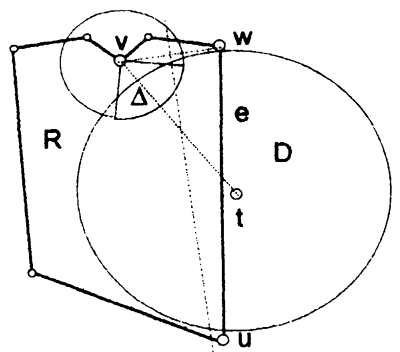

Figure 3: Proof of Lemma 2.3

If $\delta = \alpha/2$, we can see that $\Delta \setminus D$ is a subset of the disk of radius α around v, and hence contains no point of S. (Recall that α is a lower bound of the minimum distance between nodes). Hence, each call of one-face routing decreases the distance by at least $\alpha/2$, and we have an $O(\alpha^{-1}d(s,t))$ upper bound of the number of calls of one-face routing.

We can modify the detour-finding strategy to improve this bound: If the directional routing halts, we first examine whether Δ has a point of S or not. We can find all points of S in Δ by using the neighbor-find operation (see the remark below if one would like to avoid it). If Δ has a point q of S, we send the packet to q without calling the one-face routing, and the distance is decreased similarly to the usual step of directional routing, since the angle $\angle qvt$ is less than $\pi/4$, and hence less than $\pi/3$. If Δ has no point of S, we can set $\delta = 1/2$, and the one-face routing always finds a point q to decrease the distance to the target t by at least $1/2$. Thus, we have the following:

Theorem 2.4. *By the above modification, we can reduce the number of calls of one-face routing to $O(d(s,t))$.*

2.3 Analysis of number of hops

Now, let us consider the number of hops on a route. In the following, we ignore the hops required in the detour-finding routine.

Clearly, $d(s,t)$ hops are necessary for any routing method. If S is uniformly distributed, the greedy routing on UDG(S) uses $O(d(s,t))$ hops, which is the best possible. However, greedy routing on UDG(S) has several defects discussed in the introduction. The Gabriel graph is considered in GOAFR$^+$ [5], but seems to be inferior in terms of number of hops. See Section ?? to find experimental results to support the above discussion.

Moreover, theoretical asymptotic bound is not known for those methods for a general S. We begin with the following rather weak bound for the general case.

Lemma 2.5. *For any choice of F, the number of hops in the greedy routing for sending a packet from s to t is $O(d(s,t)\alpha^{-1})$.*

Proof. If t is far enough (say, distance is more than $1+\alpha$) from a current node v, we can observe that the distance is reduced by $c\alpha$ for a constant c by a hop. If the distance to t is less than $1+\alpha$, there are at most $O(\alpha^{-1})$ nodes whose distance to t is between 1 and $1+\alpha$, thus we reach the unit circle around t in $O(\alpha^{-1})$ additional steps. This proves the lemma. □

The $O(\alpha^{-1}d(s,t))$ bound is tight for LNG(S) even for a uniform distribution. Also the same lower bound exists for the greedy routing on the Gabriel graph. RLNG(S) has generally longer edges than LNG(S), and thus advantageous in number of hops. If S is uniformly distributed and $f_k(v)$ is randomly chosen, the bound of the expected number of hops becomes $O(d(s,t))$, since we can expect a constant improvement of the distance in each step.

Theoretically, it is interesting to consider the performance of the algorithm if $f_k(v)$ is randomly chosen and S is the worst distribution. There is an instance such that the expected number of hops becomes $\Omega(d(s,t)\log\alpha^{-1})$. We conjecture that this lower bound is tight, but currently, we only have a weaker upper bound given in the following theorem.

Theorem 2.6. *For any point set S with the distance condition and for any pair $s,t \in S$, the expected number of hops in the directional routing on* RLNG(S) *is $O(d(s,t)\log^2\alpha^{-1}+\alpha^{-1})$.*

Proof. We only give an intuitive outline. Suppose that we sort the points in $Q(v,k)$ with respect to the distance towards v. If we randomly pick w, it is located in the middle in the sorted list. Now, we can observe $Q(w,k)\cap D(v) \subseteq Q(v,k)$, and thus we consume all the points of $Q(v,k)$ within $O(\log N)$ hops if there are N points. This argument cheats a little, since it is not always true that the target t is located in $Q(w,k)$. But we can prove that we need to consider at most two directions (k and $k+1$, or k and $k-1$) while we remains in $D(v)$. This gives $O(\log^2 N)$ bound, and since $N = O(\alpha^{-2})$, we have the theorem. □

3 Interference Minimization

Another primary issue in wireless communication is *interference*, where communication between two parties is affected by transmissions from a third party. High interference increases the probability of packet collisions and therefore packet retransmission, which can significantly affect the effectiveness of the system and the energy use. Therefore, it is desirable to keep a low interference at every node.

Traditionally, interference has been implicitly minimized by reducing the density or the node degrees in the communication network. By keeping the transmission radii small, we then not only reduce the power consumption but also the density, and intuitively also the interference. Burkhart et al. [?], however, showed that low interference is not implied by sparseness. Also, that networks constructed from nearest-neighbor connections can fail dismally to bound the interference. On the other hand, they gave experimental results that indicate that graph spanners do help reduce interference in practice. Their work prompted the explicit study of interference minimization.

We introduce the result of [3] that use the local neighborhood graph and an ϵ-net of isothetic regular triangles to design a theoretically nice network to minimize the interference, and discuss some open problems.

Acknowledgement

This is a survey based on joint works with M. Halldorsson, H. Onuma, and K. Sato.

References

[1] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[3] M. M. Halldórsson and T. Tokuyama. Minimizing interference of a wireless ad-hoc network in a plane. *Theor. Comput. Sci.*, 402:29–42, 2008.

[4] N. Katoh, T. Tokuyama, and K. Iwano. On minimum and maximum spanning trees of linearly moving points. *Discrete & Computational Geometry*, 13:161–176, 1995.

[5] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *Proc. 22nd ACM Symposium on Principles of Distributed Computing*, pages 63–72, 2003.

[6] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. 6th Int. Workshop, DIAL-M2002*, pages 24–33, 2002.

[7] F. P. Preparata and M. I. Shamos. *Computational Geometry - An Introduction.* Springer, 1985.

[8] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J Comput*, pages 721–736, 1982.