INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
*Int. J. Commun. Syst.* 2007; **20**:767–789
Published online 18 September 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.844

# Efficient Delaunay-based localized routing for wireless sensor networks

Yu Wang[1,*,†] and Xiang-Yang Li[2,‡]

[1] *Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, U.S.A.*
[2] *Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Chicago, IL 60616, U.S.A.*

## SUMMARY

Consider a wireless sensor network consisting of $n$ wireless sensors randomly distributed in a two-dimensional plane. In this paper, we show that *with high probability* we can locally find a path for any pair of sensors such that the length of the path is no more than a constant factor of the minimum. By assuming each sensor knows its position, our new routing method decides where to forward the message purely based on the position of current node, its neighbours, and the positions of the source and the target. Our method is based on a novel structure called localized Delaunay triangulation and a geometric routing method that guarantees that the distance travelled by the packets is no more than a small constant factor of the minimum when the Delaunay triangulation of sensor nodes are known. Our experiments show that the delivery rates of existing localized routing protocols are increased when localized Delaunay triangulation is used instead of several previously proposed topologies, and our localized routing protocol based on Delaunay triangulation works well in practice. We also conducted extensive simulations of another localized routing protocol, *face routing*. The path found by this protocol is also reasonably good compared with previous one although it cannot guarantee a constant approximation on the length of the path travelled theoretically. Copyright © 2006 John Wiley & Sons, Ltd.

Received 19 January 2006; Revised 14 June 2006; Accepted 30 June 2006

KEY WORDS: wireless sensor networks; Delaunay triangulation; localized routing; power efficiency

*Correspondence to: Yu Wang, Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, U.S.A.
† E-mail: ywang32@uncc.edu
‡ E-mail: xli@cs.iit.edu

Contract/grant sponsor: U.S. National Science Foundation; contract/grant number: CCR-0311174
Contract/grant sponsor: University of North Carolina at Charlotte

## 1. INTRODUCTION

Due to its potential applications in various situations such as battlefield, emergency relief, environment monitoring, and so on, wireless sensor network has recently emerged as a premier research topic. Sensor networks consist of a set of sensor nodes which are spread over a geographical area. These nodes are able to perform processing as well as sensing and are additionally capable of communicating with each other. With co-ordination among these sensor nodes, the network together can achieve a larger sensing task both in urban environments and in inhospitable terrain. There are two common types of sensor networks: *sink-based sensor networks* and *ad hoc sensor networks*. In a sink-based sensor network, there is one or multiple sink nodes (or called base-stations) which are in charge of collecting data from all sensor nodes and managing the whole network. The traffic model in such sensor networks is usually one-to-all/all-to-one or many-to-all/all-to-many, i.e. the sink is either the sender or the receiver of all traffic. On the other hand, in an *ad hoc* sensor network there are no such sink nodes. All sensor nodes are equal in terms of roles in both communication and network management. It is a pure *ad hoc* network, and the traffic can be any peer-to-peer communications. While the sink-based sensor network is widely used in environment monitoring, the *ad hoc* sensor network has its own application, such as in battlefield and emergency relief. For example, in the battlefield, the sensor nodes with wireless devices are carried by soldiers, and the network should allow any pair of soldiers exchange information and communicate with each other, thus this scenario requires an *ad hoc* sensor network. In this paper, we will mainly focus on designing routing protocols for *ad hoc sensor networks*, but our proposed protocol can also be used in *sink-based sensor networks*.

One of the central challenges in the design of *ad hoc* sensor networks is the development of dynamic routing protocols that can efficiently find routes between two communication sensors. In recent years, a variety of routing protocols [1–6] targeted specifically for *ad hoc* environment have been developed. For the review of the state of the art in *ad hoc* routing protocols, see surveys [7–9].

Several researchers proposed another set of *ad hoc* routing protocols, namely the localized routing, which select the next node to forward the packets based on the information in the packet header, and the position of its local neighbours. Bose and Morin [10] showed that several localized routing protocols guarantee to deliver the packets if the underlying network topology is the Delaunay triangulation of all wireless nodes. They also gave a localized routing protocol based on the Delaunay triangulation such that the total distance travelled by the packet is no more than a small constant factor of the distance between the source and the destination. However, it is expensive to construct the Delaunay triangulation in a distributed manner, and routing based on it might not be possible since the Delaunay triangulation can contain links longer than the transmission radius of the wireless devices. Then, several researchers proposed to use some *planar* network topologies, that can be constructed efficiently in a distributed manner, as the routing topology. Lin *et al.* [11], Bose *et al.* [12] and Karp *et al.* [13] proposed to use the Gabriel graph [14] as the underlying routing topology. Routing according to the right-hand rule (or called face routing), which guarantees delivery in planar graphs [10], is used when simple greedy-based routing heuristics fail.

Using Gabriel graph although can achieve the guarantee of the delivery of the packets with the help of the right-hand rule, however, the distance travelled by the packet could be much larger than the minimum required [15–17]. In other words, Gabriel graph is not a good



DOI: 10.1002/dac

approximation of the communication graph (usually modelled by an unit disk graph which we will define later) in terms of the pairwise distance between wireless nodes. This is true even when the points are randomly and uniformly distributed in a unit square [15]. Formally, given a graph $H$, a spanning subgraph $G$ of $H$ is a $t$-spanner if the length of the shortest path connecting any two points in $G$ is no more than $t$ times the length of the shortest path connecting the two points in $H$. In Reference [18], Li *et al.* designed a localized algorithm that constructs a planar $t$-spanner for the unit-disk graph, such that some of the localized routing protocols can be applied on it. They obtained a value of approximately 2.5 for the constant $t$. They called the constructed graph planarized *local Delaunay triangulation* [18], denoted by PLDel.

Applying the routing methods proposed in References [12, 13] on the planarized localized Delaunay graph PLDel, a better performance is expected because the localized Delaunay triangulation is denser compared to the Gabriel graph, but still with $O(n)$ edges. However, these two methods do not guarantee that the ratio between the distance travelled by the packets to the minimum possible. The method proposed by Bose and Morrin [10] does guarantee this distance ratio, but that needs the construction of the Delaunay triangulation, which cannot be constructed and updated efficiently in a distributed manner.

Hence, in this paper, we are interested in studying the performances of several routing protocols on localized Delaunay triangulation. We prove that the localized Delaunay triangulation almost surely contains the Delaunay triangulation of the set $n$ of randomly distributed wireless sensor nodes when the transmission range $r_n$ satisfies $n\pi r_n^2 \geqslant 4(\ln n + c(n))/n$, where $c(n) \to \infty$ as $n$ goes infinity. Note that, Gupta and Kumar [19] showed that the unit disk graph is connected with high probability if the transmission range $r_n$ satisfies $\pi \cdot r_n^2 \geqslant (\ln n + c(n))/n$ for any $c(n)$ with $c(n) \to \infty$ as $n$ goes infinity. When the unit disk graph is connected, then with high probability, we can construct the Delaunay triangulation $\mathrm{Del}(V)$ by constructing the local Delaunay triangulation instead.

We then present a localized routing method that guarantees that the distance travelled by the packet is no more than a small constant factor of the minimum using the property of Delaunay triangulation. We study the performance of this localized routing method by simulations in which results show the delivery is guaranteed and the ratio of the length travelled by packet to the minimum is small. Our simulations also show that the delivery rates of several localized routing protocols are also increased when the localized Delaunay triangulation is used. In our experiments, several simple local routing heuristics, applied on the localized Delaunay triangulation, have always successfully delivered the packets, while other heuristics were successful in over 90% of the random instances. Moreover, because the constructed topology is planar, a localized routing algorithm using the right-hand rule guarantees the delivery of the packets from source node to the destination when simple heuristics fail. The experiments also show that several localized routing algorithms (notably, compass routing [20] and greedy routing) also result in a path whose length is within a small constant factor of the shortest path; we already know such a path exists since the localized Delaunay triangulation is a $t$-spanner.

In summary, the main unique contributions of this paper are as follows: (1) We show that, given a set of randomly distributed sensor nodes over a region with node density $n$, when the transmission range $r_n$ satisfies $\pi r_n^2 \geqslant 8 \log n/n$, the localized Delaunay triangulation equals the Delaunay triangulation with probability at least $1 - (1/n)$. This implies that with high probability we can construct the Delaunay triangulation using the localized Delaunay triangulation if the network is connected. We also conduct simulation of random graphs to confirm our analysis results. (2) We present a complete and detailed localized routing method



DOI: 10.1002/dac

using Delaunay triangulation based on the method from Reference [10], which guarantees that the distance travelled by the packets is no more than a small constant factor of the minimum. In Reference [10], the authors did not provide enough technique details to implement the routing method. (3) Putting together two results above with the planarized *local Delaunay triangulation* from Reference [18], we give an efficient localized routing method based on local Delaunay triangulation, and with high probability, it can find a path whose length is within a small constant factor of the minimum. To our best knowledge, this is the first localized routing method that can guarantee such property. (4) We also conduct experiments to study the performance of different localized routing protocols on various previously proposed topologies. Note that since our proposed method has theoretical bound for minimum distance travelled by the packets between any pair nodes in *ad hoc* sensor networks, it will also guarantee the bound for packets' travelling distance between any sensor and the sink in sink-based sensor networks.

The remainder of the paper is organized as follows. In Section 2, we review some definitions, some related geometry structures, and previously known localized routing protocols for wireless networks. We then show a fully localized routing algorithm that, with high probability, guarantees that the distance travelled by the packets is no more than a small constant factor of the minimum in Section 3. We study the performance of the localized routing algorithm based on Delaunay triangulation and various routing protocols on various structures in Section 4. In Section 5, we also discuss variations and possible improvements for our proposed routing algorithms. Section 6 gives a brief conclusion of our paper.

## 2. PRELIMINARIES

A wireless *ad hoc* sensor network consists of a set $V$ of $n$ wireless sensor nodes distributed in a two-dimensional plane. For simplicity, we assume that each node has the same *maximum* transmission range, denoted by $r_n$. These wireless sensor nodes define a *unit disk graph* UDG($V$) in which there is an edge between two nodes if and only if their Euclidean distance is at most $r_u$. In other words, we assume that two nodes can always receive the signal from each other directly if the Euclidean distance between them is no more than the maximum transmission range. We also use $G(V, r_n)$ to denote such induced unit disk graph. Hereafter, UDG($V$) is always assumed to be connected. All sensor nodes have distinctive identities and each node knows its position information either through a low-power Global Position System (GPS) receiver or location service provide by some nodes or localization approaches. Most of our results actually only requires that each node knows the relative positions of its neighbours, which can be achieved by using the angle of arrival of the signal or the strength of the signal. By one-hop broadcasting, each node $u$ can gather the location information of all nodes within its transmission range. Hereafter, a *broadcast* means a node sends out a message which will be received by all nodes within its transmission range.

### *2.1. Spanner and spanning ratio*

In an *ad hoc* sensor network, two far-apart nodes can communicate with each other through the relay of intermediate nodes; hence, each node only needs to set small transmission ranges which reduces the signal interference and saves power for transmissions. To guarantee the advantage, a good network topology should be energy efficient for routing. Since the power consumed to



DOI: 10.1002/dac

support a link is proportional to the length of that line (i.e. the power consumed by link $uv$ is $\|uv\|^{\beta}$, where $\beta$ is a constant between 2 and 6 depended on the environment and $\|uv\|$ is the Euclidean distance between $u$ and $v$), that is to say, the length of the shortest path between any two nodes in the constructed routing topology should not exceed a constant factor of the length of the shortest path in original network. Let $\Pi_G(u,v)$ be the shortest path connecting $u$ and $v$ in a weighted graph $G$, and $\|\Pi_G(u,v)\|$ be the length of $\Pi_G(u,v)$. A graph $G$ is a $t$-spanner of a graph $H$ if $G$ is a subgraph of $H$ and, for any two nodes $u$ and $v$, $\|\Pi_G(u,v)\| \leqslant t\|\Pi_H(u,v)\|$. With $H$ understood, we also call $t$ the *length stretch factor* of the spanner $G$. Notice that if a routing topology $G$ is a $t$-spanner in term of length, the power consumed by the shortest path in $G$ is at most $t^{\beta}$ of the minimum in the original network. The proof can be found in Reference [21].

Let $\varrho_G(u,v)$ be the path found by a unicasting routing method $\varrho$ from node $u$ to $v$ in a weighted graph $G$, and $\|\varrho_G(u,v)\|$ be the length of the path. Here, the length of a path $P$ is the summation of the length of all edges in $P$, i.e. $\|P\| = \sum_{uv \in P} \|uv\|$. The *spanning ratio* achieved by a routing method $\varrho$ is defined as $\max_{u,v}(\|\varrho_G(u,v)\|/\|uv\|)$. Note that the spanning ratio achieved by a specific routing method could be much larger than the length stretch factor of the underlying structure. Nonetheless, a structure with a small stretch factor is necessary for some routing method to possibly perform well. In this paper, we will give a localized routing method who can achieve constant spanning ratio. In other words, the length of path is no more than a constant factor of the minimum.

### 2.2. Voronoi diagram and Delaunay triangulation

Delaunay triangulation is a well-known planar length spanner. We continue with definitions of Delaunay triangulation and its geometric dual, Voronoi diagram. We assume that there are no four sensor nodes of $V$ that are co-circular. A triangulation of $V$ is a *Delaunay triangulation*, denoted by $\mathrm{Del}(V)$, if the circumcircle of each of its triangles does not contain any other nodes of $V$ in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of nodes of $V$. The *Voronoi region*, denoted by $\mathrm{Vor}(p)$, of a node $p$ in $V$ is the collection of two-dimensional points such that every point is closer to $p$ than to any other node of $V$. The *Voronoi diagram* for $V$, denoted by $\mathrm{Vor}(V)$, is the union of all Voronoi regions $\mathrm{Vor}(p)$, where $p \in V$. The Delaunay triangulation $\mathrm{Del}(V)$ is also the dual of the Voronoi diagram: two nodes $p$ and $q$ are connected in $\mathrm{Del}(V)$ if and only if $\mathrm{Vor}(p)$ and $\mathrm{Vor}(q)$ share a common boundary. The shared boundary of two Voronoi regions $\mathrm{Vor}(p)$ and $\mathrm{Vor}(q)$ is on the perpendicular bisector line of segment $pq$. The boundary segment of a Voronoi region is called the *Voronoi edge*. The intersection point of two Voronoi edge is called the *Voronoi vertex*. Each Voronoi vertex is the circumcentre of some Delaunay triangle. See Figure 1 for an illustration of the dual relation between $\mathrm{Vor}(V)$ and $\mathrm{Del}(V)$. It is well-known that the Delaunay triangulation $\mathrm{Del}(V)$ is a planar $t$-spanner of the completed Euclidean graph $K(V)$. The best known upper bound on $t$ is $2\pi/(3\cos\frac{\pi}{6}) \approx 2.42$ [22, 23] and the best known lower bound is $\pi/2$ [24].

### 2.3. Proximity graphs

The following geometric structures or graphs have been used as routing topologies for *ad hoc* networks. For convenience, let $disk(u,v)$ be the closed disk with diameter $uv$ and $disk(u,v,w)$ be the circumcircle defined by the triangle $\triangle uvw$. The *relative neighbourhood graph* [25], denoted by $\mathrm{RNG}(V)$, consists of all edges $uv$ such that $\|uv\| \leqslant r_u$ and there is no point $w \in V$ such that $\|uw\| < \|uv\|$, and $\|wv\| < \|uv\|$. The *Gabriel graph* [14], denoted by $\mathrm{GG}(V)$, consists of all edges $uv$

 
DOI: 10.1002/dac

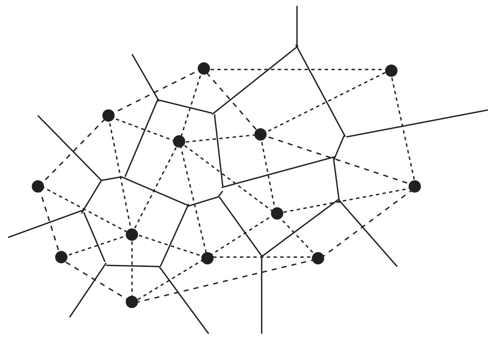

Figure 1. Voronoi diagram and Delaunay triangulation (dash lines) of a set of two-dimensional nodes.

such that $\|uv\| \leqslant r_u$ and $disk(u, v)$ does not contain any node from $V$. The *Yao graph* [26] with an integer parameter $k \geqslant 6$, denoted by $YG_k(V)$, is defined as follows. At each node $u$, any $k$ equal-separated rays originated at $u$ define $k$ cones. In each cone, choose the closest node $v$ to $u$ with distance at most $r_u$, if there is any, and add a directed link $\overrightarrow{uv}$. Ties are broken arbitrarily.

Bose *et al.* [15] showed that the length stretch factor of RNG($V$) is at most $n-1$ and the length stretch factor of GG($V$) is at most $4\pi\sqrt{2n-4}/3$. Recently, Wang *et al.* [17] showed that the length stretch factor of GG($V$) is precisely $\sqrt{n-1}$ actually. Bose *et al.* [15] also showed that the length stretch factor of Gabriel graph on a uniformly random $n$ points set in a square is almost surely at least $O(\sqrt{\log n/\log \log n})$. Several papers [21, 27, 28] showed that the Yao graph $\mathrm{YG}_k(V)$ has length stretch factor at most $1/(1-2\sin\frac{\pi}{k})$. Some researchers [21, 29, 30] proposed to construct the wireless network topology based on Yao graph. However, the Yao graph is not guaranteed to be planar. The relative neighbourhood graph and the Gabriel graph are planar graphs and have been used as the routing topology for *ad hoc* networks [12, 13, 31–33], but they are not a spanner for the unit-disk graph.

### 2.4. Localized algorithms

Let $N_k(u)$ be the set of nodes of $V$ that are within $k$ hops distance of $u$ in UDG($V$). A node $v \in N_k(u)$ is called the $k$-neighbour of the node $u$. In this paper, we always assume that each node $u$ of $V$ knows its location and identity. Then, after one successful transmission by every node, each node $u$ of $V$ knows the location and identity information of all nodes in $N_1(u)$. The total communication cost of all nodes to do so is $O(n \log n)$ bits. A distributed algorithm is a *localized algorithm* if it uses only the information of all $k$-neighbours of each node plus the information of a constant number of additional nodes. In this paper, we concentrate on the case $k=1$. That is, a node uses only the information of the 1-hop neighbours. Note that to collect the information of $N_k(u)$ when $k>1$ may need several rounds of communication with number of messages. A topology $G$ can be constructed *locally* in the *ad hoc* wireless environment if each node $u$ can compute the edges of $G$ incident on $u$ by using only the location information of all its $k$-neighbours. A routing algorithm is a *localized* routing if the decision of which node it forwards packet to is based only on local information plus the source and the destination.



DOI: 10.1002/dac

### 2.5. Previous localized routing methods

The geometric nature of the multi-hop *ad hoc* networks allows a promising idea: localized routing protocols. A routing protocol is *localized* if the decision to which node to forward a packet is based only on:

1. The information in the header of the packet. This information includes the source and the destination of the packet, but more data could be included, provided that its total length is bounded.
2. The local information gathered by the node from a small neighbourhood. This information includes the set of 1-hop neighbours of the node, but a larger neighbourhood set could be used provided it can be collected efficiently.

Randomization is also used in designing the protocols. A routing is said to be *memory-less* if the decision to which node to forward a packet is solely based on the destination, current node and its neighbours within some constant hops. Localized routing is sometimes called in the literature *stateless* [13], *online* [10, 34], or *distributed* [11].

We summarize some localized routing protocols proposed in the networking and computational geometry literature. Let $u$ be the current node and $t$ be the destination node.

- Compass Routing (Cmp) [20]: Current node $u$ finds the next relay node $v$ such that the angle $\angle vut$ is the smallest among all neighbours of $u$ in a given topology.
- Random Compass Routing (RCmp) [20]: Let $v_1$ be the node on the above of line $ut$ such that $\angle v_1ut$ is the smallest among all such neighbours of $u$. Similarly, we define $v_2$ to be nodes below line $ut$ that minimizes the angle $\angle v_2ut$. Then node $u$ randomly choose $v_1$ or $v_2$ to forward the packet.
- Greedy Routing (Grdy) [12]: Current node $u$ finds the next relay node $v$ such that the distance $\|vt\|$ is the smallest among all neighbours of $u$ in a given topology.
- Most Forwarding Routing (MFR) [11]: Current node $u$ finds the next relay node $v$ such that $\|v't\|$ is the smallest among all neighbours of $u$ in a given topology, where $v'$ is the projection of $v$ on segment $ut$.
- Nearest Neighbour Routing (NN): Given a parameter angle $\alpha$, node $u$ finds the nearest node $v$ as forwarding node among all neighbours of $u$ in a given topology such that $\angle vut \leqslant \alpha$.
- Farthest Neighbour Routing (FN): Given a parameter angle $\alpha$, node $u$ finds the farthest node $v$ as forwarding node among all neighbours of $u$ in a given topology such that $\angle vut \leqslant \alpha$.
- Greedy-Compass (GCmp) [10, 35]: Current node $u$ first finds the neighbours $v_1$ and $v_2$ such that $v_1$ forms the smallest counter-clockwise angle $\angle tuv_1$ and $v_2$ forms the smallest clockwise angle $\angle tuv_2$ among all neighbours of $u$ with the segment $ut$. The packet is forwarded to the node of $\{v_1, v_2\}$ with minimum distance to $t$.

It was shown in References [12, 20] that the compass routing, random compass routing and the greedy routing guarantee to deliver the packets from the source to the destination if Delaunay triangulation is used as network topology. They proved this by showing that the distance from the selected forwarding node $v$ to the destination node $t$ is less than the distance from current node $u$ to $t$. However, the same proof cannot be carried over when the network topology is Yao graph, Gabriel graph, relative neighbourhood graph, and even the localized



DOI: 10.1002/dac

Delaunay triangulation. When the underlying network topology is a planar graph, the right hand rule is often used to guarantee the packet delivery after simple localized routing heuristics fail [11–13, 36, 37]. Morin proved the following results in Reference [35]. The greedy routing guarantees the delivery of the packets if the Delaunay triangulation is used as the underlying structure. The compass routing guarantees the delivery of the packets if the regular triangulation is used as the underlying structure. Delaunay triangulation is a special regular triangulation. There are triangulations (not Delaunay) that defeat these two schemes. The greedy-compass routing works for all triangulations, i.e. it guarantees the delivery of the packets as long as there is a triangulation used as the underlying structure. Every oblivious routing method is defeated by some convex subdivisions.

Applying the right hand rule in planar graphs, a routing protocol called *face routing* is proposed by Kranakis *et al*. [20] (in the paper they call the algorithm *Compass Routing II*). The authors [20, 38] also proved that the face routing algorithm guarantees to reach the destination $t$ after traversing at most $O(n)$ edges where $n$ is the number of nodes when the underlying network topology is a planar graph. Though face routing terminates in linear time, it is not satisfactory, since already a very simple flooding algorithm will terminate in $O(n)$ steps. Then Kuhn *et al*. [36, 38] proposed two new methods *adaptive face routing* and *other adaptive face routing*, in which, restricted search areas are used to avoid exploring the complete boundary of faces.

Although some of the localized routing protocols guarantee the delivery of the packet if some special geometry structures are used as the routing topology, none of these guarantees the ratio of the distance travelled by the packets over the minimum possible. Bose and Morrin [10] proposed a method to bound this ratio using the Delaunay triangulation. They showed that the distance travelled by the packet is within a constant factor of the distance between the source and the destination. Note that constructing Delaunay triangulation in a distributed manner is communication expensive.

### *2.6. Location service*

In order to make the localized routing work, the source node has to learn the current (or approximately current) location of the destination node. In some applications of sensor networks which use sensor networks to collect data, the destination node is often fixed and called the sink node, thus, location service is not needed in these applications. However, the help of a *location service* is needed in most other application scenarios. Mobile nodes register their locations to the location service. When a source node does not know the position of the destination node, it queries the location service to get that information. In cellular networks, there are dedicated position severs. It will be difficult to implement the centralized approach of location services in *ad hoc* sensor networks. Therefore, many distributed location service systems have been proposed [39–42]. For more detailed, please refer these references.

## 3. LOCALIZED ROUTING WORKS

In this section, we propose a fully localized routing algorithm that, with high probability, guarantees that the distance travelled by the packets is no more than a small constant factor of the minimum. The localized routing algorithm is based on an efficient Delaunay-based routing method proposed in Reference [10] and uses localized Delaunay triangulation [18] as the routing



DOI: 10.1002/dac

topology. We will answer the following questions in this section: When and how the Delaunay triangulation could be constructed locally? How to do efficient localized routing on the Delaunay triangulation? First, we show the Delaunay triangulation could be constructed locally with high probability when the nodes are randomly distributed. Then, we review how to construct it locally by novel localized algorithms. At last, we adopt the method in Reference [10] to give our first localized routing algorithm that, with high probability, guarantees that bounds the distance travelled by the packets by a small constant factor of the minimum.

### *3.1. When Delaunay could be constructed locally?*

Although the method [10] works perfectly if the Delaunay triangulation of the set of nodes is known in advance, it is communication-intensive to construct the Delaunay triangulation in a distributed manner in the worst case. We will show that the Delaunay triangulation can be constructed using some localized approach with high probability when the nodes are randomly distributed and the transmission range is larger than some threshold (with high probability we can do so when the network is connected). Gupta and Kumar [19] showed that the unit disk graph is connected with high probability if the transmission range $r_n$ satisfies $\pi \cdot r_n^2 \geqslant (\ln n + c(n))/n$ for any $c(n)$ with $c(n) \to \infty$ as $n$ goes infinity. Our construction is based on the local Delaunay triangulation by showing that all edges in the Delaunay triangulation is no more than the transmission radius with high probability when the nodes are randomly and uniformly distributed.

We assume that the wireless sensor nodes are randomly and uniformly distributed in a unit area disk. It was proved in several papers [19, 43] that the random point process bears the same stochastic property as the homogeneous Poisson point process. The standard probabilistic model of *homogeneous Poisson process* is characterized by the property that the number of nodes in a region is a random variable depending only on the area (or volume in higher dimensions) of the region and the density of the process. Let $\lambda$ be the density.

- The probability that there are exactly $k$ nodes appearing in any region $\Psi$ of area $A$ is $\frac{(\lambda A)^k}{k!} \cdot \mathrm{e}^{-\lambda A}$.
- For any region $\Psi$, the conditional distribution of nodes in $\Psi$ given that exactly $k$ nodes in the region is *joint uniform*.

Hereafter, we let $\mathscr{P}_n$ be a homogeneous Poisson process of intensity $n$ on the unit area disk. We will consider the homogeneous Poisson point process instead of the random point process in our proof.

Let $D$ be the variable denoting the length of the longest edge $pq$ of the Delaunay triangulation of all wireless sensor nodes generated by a homogeneous Poisson process with density $n$. Consider any edge $e$ with length $\ell$ contained in some triangle $\triangle pqs$. Then the circumcircle of triangle $\triangle pqs$ has area at least $\pi\ell^2/4$. This circumcircle must contain no other nodes inside from the property of the Delaunay triangulation. The probability, denoted by $p_1$, that this circumcircle is empty of nodes is $\frac{(n\pi\ell^2/4)^0}{0!} \cdot \mathrm{e}^{-n\pi\ell^2/4} = \mathrm{e}^{-n\pi\ell^2/4}$. The probability that the longest edge of the Delaunay triangulation $T$ is $d_n$ satisfies

$$Pr(D \geqslant d_n) = Pr\left(\bigcup_{e \in T} e \geqslant d_n\right) \leqslant \sum_{e \in T} Pr(e \geqslant d_n) \leqslant 3n \cdot \mathrm{e}^{-n\pi d_n^2/4}$$



DOI: 10.1002/dac

Note that, there are at most $3n$ edges in the two-dimensional Delaunay triangulation of $n$ nodes. By solving the inequality $3n \cdot e^{-n\pi d_n^2/4} \leqslant 1/\beta$, we know that, with probability at most $1/\beta$, the longest edge of the Delaunay triangulation has length $d_n$, where $\pi d_n^2 \geqslant 4(\ln n + \ln \beta + \ln 3)/n$. In other words, with probability at least $1 - 1/\beta$, the longest edge of the Delaunay triangulation has length $d_n$, where

$$\pi d_n^2 \leqslant 4 \cdot \frac{\ln n + \ln \beta + \ln 3}{n}$$

Penrose [44] showed that the longest edge of the minimum spanning tree of homogeneous Poisson point process $\mathscr{P}_n$ is at most $M_n$ with probability $e^{-e^{-\alpha}}$, where $n\pi M_n^2 \leqslant \ln n + \alpha$. In other words, if the transmission radius $r_n$ satisfies

$$\pi r_n^2 \geqslant \frac{\ln n + \alpha}{n}$$

then the induced graph $G(V, r_n)$ is connected with probability at least $e^{-e^{-\alpha}}$ when $n$ goes infinity. By substituting $e^{\alpha} = \gamma$, we know that, with probability at least $1 - 1/\gamma$, the induced unit disk graph is connected if the transmission range $r_n$ of every node satisfies that

$$\pi r_n^2 \geqslant \frac{\ln n + \ln \gamma}{n}$$

Combining the above analysis, the induced network is a connected graph with probability at least $1 - (1/n^7)$ if $\pi r_n^2 \geqslant 8 \ln n/n$; meanwhile, with probability at least $1 - (1/n)$, the longest edge $d_n$ of the Delaunay triangulation is at most $r_n$. Note that to make the induced network connected with probability $1 - (1/n)$, we need set the transmission radius $r_n$ satisfies $\pi r_n^2 \geqslant (2 \ln n)/n$. In other words, the required transmission range so that local Delaunay triangulation equals the Delaunay triangulation is just twice of the minimum transmission range to have a connected network with high probability. Practically, the transmission range is often larger than the minimum requirement to get connectivity with high probability.

In the previous analysis, we did not consider the boundary effects. Our simulation results will show that the Delaunay edges near the domain boundary is often larger than the expected value of theoretical analysis for the domain without boundary. This is due to two reasons. First, our theoretical analysis holds only when $n$ is large enough. Second, when the geometry domain in which the wireless sensor nodes are distributed is bounded, the circumcircle of the Delaunay triangle near the domain boundary is not fully contained in the geometry domain. Thus, the probability that the circumcircle is empty of other nodes does not depend on the area of the circumcircle; instead, it depends on the area of the intersection of the circumcircle with the geometry domain.

### *3.2. How to construct Delaunay locally?*

Recall that a triangle $\triangle uvw$ belongs to the Delaunay triangulation $\mathrm{Del}(V)$ if its circumcircle $\mathrm{disk}(u, v, w)$ does not contain any other node of $V$ in its interior. It is easy to show that nodes $u$, $v$ and $w$ together cannot decide if they can form a triangle $\triangle uvw$ in $\mathrm{Del}(V)$ by using only their local information. For example, there is a node $y$ inside the circumcircle $\mathrm{disk}(u, v, w)$ but the distances between $y$ and $u$, $v$, $w$ are all larger than 1 (i.e. nodes $u$, $v$, $w$ cannot see node $y$). Since constructing Delaunay triangulation in a distributed manner is communication-intensive, we



DOI: 10.1002/dac

will rely on some localized construction method, more specifically, localized Delaunay triangulation [18]. For completeness of presentation, we give a brief review of the definition of localized Delaunay triangulation.

*Definition 3.1*
A triangle $\triangle uvw$ satisfies *k-localized Delaunay property* if (1) the interior of disk$(u, v, w)$ does not contain any node of $V$ that is a $k$-neighbour of $u$, $v$, or $w$; (2) all edges of the triangle $\triangle uvw$ have length no more than one unit. Triangle $\triangle uvw$ is called a *k-localized Delaunay triangle*.

*Definition 3.2*
The *k-localized Delaunay triangulation* over a node set $V$, denoted by $\mathrm{LDel}^{(k)}(V)$, has exactly Gabriel edges (edges in Gabriel graph) and edges of $k$-localized Delaunay triangles.

Li *et al.* [18] first proved that $\mathrm{LDel}^{(k)}(V)$ is a length spanner for all $k$ and is a planar graph for $k \geqslant 2$. Then they presented a localized algorithm to construct $\mathrm{LDel}^{(1)}(V)$ and remove the intersections in $\mathrm{LDel}^{(1)}(V)$. The algorithm generates a planar graph PLDel$(V)$ with communication cost of $O(n\log n)$ bits. They proved that PLDel$(V)$ is also a $t$-spanner of UDG $(V)$. The constant behind the $O()$ in the communication cost is bounded by 37 [18], i.e. the total communication cost to compute the local Delanauy triangulation is at most $37n \log n$ bits. Note that $\log n$ is the number of bits required to represent a node ID. In the construction algorithm, each node first constructs Delaunay triangulation based on 1-hop neighbour information and proposes to add the adjacent Delaunay edges, then nodes exchange the proposed Delaunay edges with neighbours and only keep those consistent Delaunay edges. Due to space limit, we do not review the construction algorithm. The reader interested in the detailed algorithm can find it in Reference [18]. Note that if the longest edge of the Delaunay triangulation is at most $r_u$, obviously, PLDel is the Delaunay triangulation actually.

Gao *et al.* [45] also proposed another structure, called *restricted Delaunay graph* RDG to locally approximate Delaunay triangulation. They showed that it has constant stretch factor properties and can be maintained locally. Our Delaunay-based localized routing can also be well performed on the restricted Delaunay graph since when the transmission range satisfies the condition derived from previous subsection the restricted Delaunay graph also equals the Delaunay triangulation with high probability.

### *3.3. How to do efficient routing on Delaunay?*

Bose and Morrin [10] have proposed a method to route the packets using the Delaunay triangulation. Their routing strategy is based on a remarkable proof by Dobkin *et al.* [46] that the Delaunay triangulation is a spanner. However, there are plenty of technique details left to be discussed. In this subsection, we present a complete and detailed localized routing method using the Delaunay triangulation.

To discuss the localized routing algorithm, we need a quick review of the proof by Dobkin *et al.* [46]. They proved that the Delaunay triangulation is a $t$-spanner by constructing a path $\Pi_{dfs}(u, v)$ in Del$(V)$ with length no more $(1 + \sqrt{5})\pi\|uv\|/2$. The constructed path consists of at most two parts: one is some *direct DT* paths, the other is some *shortcut* subpaths.

Given two nodes $u$ and $v$, let $b_0 = u, b_1, b_2, \ldots, b_{m-1}, b_m = v$ be the nodes corresponding to the sequence of Voronoi regions traversed by walking from $u$ to $v$ along the segment $uv$. See Figure 2



DOI: 10.1002/dac

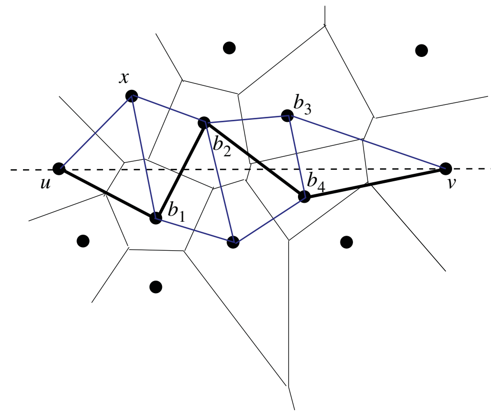


Figure 2. The direct DT path $ub_1b_2b_3b_4v$ between $u$ and $v$ shown by thickest lines. The tunnel $T(u, v)$ is shown by shaded lines. The thin lines represent the Voronoi diagram.

for an illustration. If a Voronoi edge or a Voronoi vertex happens to lie on the segment $uv$, then choose the Voronoi region lying above $uv$. Assume that the line $uv$ is the $x$-axis. Let $x(v)$ and $y(v)$ be the value of the $x$-co-ordinate and $y$-co-ordinate of a node $v$, respectively. The sequence of nodes $b_i$, $0 \leqslant i \leqslant m$, defines a path from $u$ to $v$. In general, they [46] refer to the path constructed this way between some nodes $u$ and $v$ as the *direct DT path* from $u$ to $v$. If the direct DT path connecting $u$ and $v$ is lying entirely above or entirely below the segment $uv$, it is called *one-sided*.

Define the *tunnel*, denoted by $T(u, v)$, of segment $uv$ as the set of triangles in the Delaunay triangulation, whose interior intersects the segment $uv$. The triangles illustrated in Figure 2 is the tunnel $T(u, v)$ defined for nodes $u$ and $v$.

The path constructed by Dobkin *et al.* uses the direct DT path as long as it is above the $x$-axis. Assume that the path constructed so far has brought us to some node $b_i$ such that $y(b_i) \geqslant 0$, $b_i \neq v$, and $y(b_{i+1}) < 0$. Let $j$ be the least integer larger than $i$ such that $y(b_j) \geqslant 0$. Note that here $j$ exists because $y(b_m) = 0$ by assuming that $uv$ is the $x$-axis. Then the path constructed by Dobkin *et al.* uses either the direct DT path from $b_i$ to $b_j$ or takes a *shortcut*, which is the upper boundary of the tunnel $T(u, v)$ that connects $b_i$ and $b_j$. See Reference [46] for more detail about the condition when to choose the direct DT path from $b_i$ to $b_j$ and when to choose the shortcut path from $b_i$ to $b_j$. Let $c_{dfs} = (1 + \sqrt{5})\pi/2$. It was proved in Reference [46] that either the length of the direct DT path from $b_i$ to $b_j$ is at most $c_{dfs}(x(b_j) - x(b_i))$ or the length of the shortcut between $b_i$ and $b_j$ is at most $c_{dfs}(x(b_j) - x(b_i))$. For example, in Figure 2, node $b_2$ is below the axis $uv$. Thus, node $u$ either takes path $ub_2b_3$ or path $uxb_3$ to node $b_3$. Path $ub_2b_3$ is the direct DT path, which is below the axis. Path $uxb_3$ is the shortcut path from $u$ to $b_3$.

Routing the packets along the direct DT path is a localized routing method, but it is not competitive on its own for all Delaunay triangulations. Bose and Morin [10] presented an example such that the distance travelled in this approach could be arbitrarily larger than the minimum. The routing strategy by Bose and Morin uses the direct DT path as long as it is above the $x$-axis. When the direct DT path lead us to an edge $b_ib_{i+1}$ that intersects $uv$, it either continues to use the direct DT path or the shortcut to node $b_j$. The difficulty occurs as the strategy does not know prior which of these two paths is shorter. Their solution is to simulate exploring both paths in a parallel manner whenever the first one reaches node $b_j$. However,

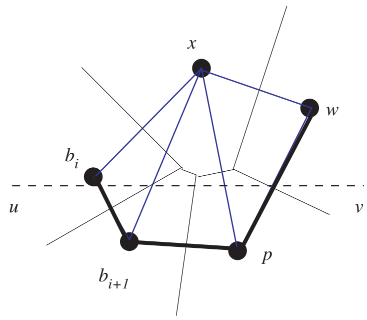


Figure 3. Find the next neighbour of $b_i$ in the direct DT path or the neighbour of $x$ in the shortcut path.

many technique details need to be filled so it can be implemented. Basically, we have to answer the following questions: (1) how to find the neighbour in the direct DT path locally, (2) how to find the neighbour in the shortcut path locally, and (3) how to determine whether node $b_j$ is reached. We call this routing method Delaunay triangulation based routing, denoted by DTR.

For simplicity, let $v_1 = u, v_2, \ldots, v_{k-1}, v_k = v$ be the $k$ vertices of all $b_i$'s that is on or above the segment $uv$.

First, we study how to find the neighbour of a node $b_i$ in the direct DT path locally. Note that, by definition, the Voronoi region of a vertex is always a convex region. Thus, line segment $uv$ only intersects at most *two* Voronoi edges of a Voronoi region. In other words, the direct DT path is uniquely and well defined. Assume that the current vertex $b_i$ wants to find its next neighbour in the direct DT path. Then node $b_i$ can compute Vor($b_i$) locally since it knows all Delaunay edges incident on $b_i$ and the Voronoi diagram is a dual of the Delaunay triangulation. Then node $b_{i+1}$ is the node that (1) shares the Voronoi edge of $Vor(b_i)$ that is intersected by $uv$, and (2) has larger $x$-co-ordinate than node $b_i$. See Figure 3 for an illustration.

Second, we show how to find the next neighbour of a node $x$ in the shortcut path locally. Remember that the shortcut path is the boundary segments of $T(u, v)$, which connects two consecutive vertices $v_i$ and $v_{i+1}$, of tunnel $T(u, v)$. Vertex $x$ first sorts all Delaunay edges incident on $x$ in count-clockwise order. Then $x$ finds the incident neighbour vertex $w$ such that $xw$ does not intersect the segment $uv$, but the previous Delaunay edge intersects $uv$. See Figure 3 for an illustration. Here node $w$ is the next node on the short-cut path.

Thirdly, we reach the node $b_j$ if the following conditions hold: (1) the Voronoi diagram of the current node intersects $uv$, (2) the $y$-co-ordinate is not negative, and (3) if we are exploring the shortcut path, then the Voronoi Diagram of the previous visited node does not intersect $uv$; if we are exploring the direct shortcut path, then the $y$-co-ordinate of the previous visited node is negative. In Figure 3, node $w$ will be that node $b_j$.

Node $u$ computes its Voronoi region as follows when knowing all Delaunay triangles incident on $u$. For each incident Delaunay triangle $\triangle vuw$, $u$ computes the circumcentre $c$ of $\triangle vuw$. Node $u$ connects two circumcentres if the corresponding Delaunay triangles share a common edge. All such edges connected circumcentres form the Voronoi region of $u$. If $u$ is a boundary node, special treatment is needed to compute the Voronoi region. Note that, $u$ is a boundary node if and only if the Delaunay triangles incident on $u$ only covers at most half of the space



DOI: 10.1002/dac

surrendering $u$. Assume that edges $uv$ and $uw$ are the two boundary Delaunay edges incident on $u$. Let $x$ be the circumcentre of the unique triangle $\triangle uvp$; $y$ be the circumcentre of the unique triangle $\triangle uvq$. Then a ray starting at $x$ and is perpendicular to $uv$ is drawn, and a ray starting at $y$ and is perpendicular to $uw$ is drawn. These two rays, together with all segments connecting the circumcentres of the Delaunay triangles incident on $u$ form the Voronoi region of $u$.

The routing algorithm works as following. Let $v_0 = u$ and $i = 0$. Let node $v_{i+1}$ be the node returned by EXPLORE($v_i$). If $v_{i+1}$ is not node $v$, then increase $i$ by one and continue EXPLORE($v_i$). Algorithm 1 is the detailed description of the algorithm EXPLORE($v_i$).

---

**Algorithm 1** EXPLORE($v_i$)

---

Let $p_0$ be the next neighbour of $v_i$ in the direct DT path, and $q_0$ be the next neighbour of $v_i$ in the shortcut path. Let $j = 0$ and $l_0 = \min(\|v_i p_0\|, \|v_i q_0\|)$.
**repeat**
  **if** $\|v_i p_j\| \leqslant \|v_i q_j\|$ **then**
    EXPLORE DIRECT DT PATH:
    Route the packet along the direct DT path from node $v_i$ until reaching a node $v_{i+1}$, which is on the direct DT path and is above the segment $uv$, or reaching a node $p_{j+1}$, such that the distance travelled from $p_0$ to $p_{j+1}$ is larger than $2l_j$ for the first time.
    **if** node $v_{i+1}$ is reached **then**
      return $v_{i+1}$ and quit.
    **else**
      set $j = j + 1$ and $l_j$ be the distance travelled from $p_0$ to $p_{j+1}$
      travel back to node $v_i$.
    **end if**
  **else**
    EXPLORE SHORTCUT PATH:
    Route the packet along the shortcut path from node $v_i$ until reaching node $v_{i+1}$, which is on the direct DT path and is above the segment $uv$, or reaching a node $q_{j+1}$, such that the distance travelled from $q_0$ to $q_{j+1}$ is larger than $2l_j$ for the first time.
    **if** node $v_{i+1}$ is reached **then**
      return $v_{i+1}$ and quit.
    **else**
      set $j = j + 1$ and $l_j$ be the distance travelled from $q_0$ to $q_{j+1}$
      travel back to node $v_i$.
    **end if**
  **end if**
**until** $v_{i+1}$ is reached.

---

Note that, originally, Bose and Morin [10] always start exploring the shortcut path first. However, this may lead to a long travelling distance when the first edge of the shortcut path is much longer than the direct DT path. Morin [35] proved the following theorem.

*Theorem 3.3*
The distance travelled by the above routing strategy is $9c_{dfs}$-competitive.



DOI: 10.1002/dac

*Proof*
Assume that the EXPLORE algorithm starts from node $b_i$ and ends with node $b_j$. It was proved in Reference [46] that either the length of the direct DT path from $b_i$ to $b_j$ is at most $c_{dfs}(x(b_j) - x(b_i))$ or the length of the shortcut between $b_i$ and $b_j$ is at most $c_{dfs}(x(b_j) - x(b_i))$. We only have to show that the actual distance travelled by the EXPLORE algorithm is at most 9 times the distance between $b_i$ and $b_j$, denoted by $L$. Note that, $l_j \leqslant 2^j l_0$ and the distance from $p_0$ to $p_j$ is travelled back and forth. The total distance travelled by exploring the direct DT path is at most $\sum_{j=0}^{k} 2l_j \leqslant \sum_{j=0}^{k} 2 \cdot 2^j l_0 \leqslant 4L$, where $k$ is the maximum integer such that $l_k < L$. Similarly, the total distance travelled by exploring the shortcut path is at most $4L$. At last, it travels distance $L$ when node $b_j$ is reached. Thus, total travelled distance is at most $9L$. The theorem follows from $L \leqslant c_{dfs}(x(b_j) - x(b_i))$. □

Note that so far the Delaunay-based method is the only localized routing method that can guarantee the total *distance* travelled by packets is constant competitive even in the worst case scenario. The above theorem only bounds the total *distance* travelled by the Delaunay routing. If we consider the total *power* consumed by the route comparing with the optimal least-power path in the original network, unfortunately, the constant bound does not exist. In other words, for routing, if a routing scheme finds a path that has constant length spanning ratio, the path not necessarily has constant power-spanning ratio. We prove this as Theorem 3.4. Note that this is different with the stretch factor of topologies. If a topology has a constant length stretch factor, it must have a constant power stretch factor. See Lemma 2 in Reference [21] for the proof of this claim.

*Theorem 3.4*
The total power consumed by the route obtained by the above routing strategy could be sufficiently larger than the total power consumed by the optimal least-power path in the original network (unit disk graph). In other words, the ratio of the total power consumed to the optimal is not bounded by a constant, could be as bad as $O(n)$ where $n$ is the total number of nodes. Here the total power consumed by a route $P$ is the summation of the power consumed by each link on $P$, i.e. $p(P) = \sum_{uv \in P} p(uv)$.

*Proof*
We prove the theorem by constructing examples. Note the total power consumed by the route is depended on the definition of the energy model for each link.

If the energy model is that the power $p(uv)$ consumed by a link $uv$ is $||uv||^\beta$, then the following example proves the theorem. Assume a unit link $uv$ is part of Delaunay, and there is another path connecting $uv$ with almost infinite number of nodes $w_1, w_2, \ldots, w_n$ (no $w_i$ is inside the circle formed by $uv$). Assume the length of path $w_1 w_2 \cdots w_n$ is a constant time of the length of $uv$ (say $l$) and nodes $w_i$ are evenly distributed. Delaunay-based routing will use $uv$, however the best energy path is $w_1 w_2 \cdots w_n$ with total energy almost $n(l/n)^2 = l^2/n$ (if $\beta = 2$) of that by $uv$ ($1^2 = 1$). Therefore, the power consumed by Delaunay routing could be sufficiently larger ($O(n)$) than the optimal least-power path.

If the energy model is that the power $p(uv)$ consumed by a link $uv$ is $||uv||^\beta + c$ where $c$ is a constant, then the following example will prove the theorem. Assume that there is a unit link $uv$ and all $n$ nodes $w_i$ are evenly distributed on $uv$. Delaunay routing will take the path $w_1 w_2 \cdots w_n$ which has energy cost about $n(1/n)^2 + nc = 1/n + nc$ and the direct path from $u$ to $v$ has



DOI: 10.1002/dac

cost of $1^2 + c = 1 + c$. When $n$ is sufficiently large, the former could be sufficiently larger than the later. □

Note that the examples shown in the above proof happen rarely in a randomly deployed network. In Section 5, we further discuss some variations of our routing method to improve its power efficiency or other performances.

## 4. SIMULATIONS

In this section, we evaluate the performance of our routing method by conducting simulations with random networks. Before testing our routing algorithm, we first study the transition phenomena of the longest edge of the Delaunay triangulation.

In our experiments, three different geometry regions $\Omega$: disk with radius 200 m, square with side 400 m, and unbounded region of grids (with unit 400 m), are tested. The node density $n$ is 50, 100, 200, 300, 400, and 500. For each choice of $\Omega$ and $n$, 10 000 sample of $n$ points is generated, and the longest Delaunay edge is generated for each sample. Left figures of Figure 4 illustrate the longest Delaunay edge length $D_n$ distribution, while the right figures illustrate its transition phenomena. The statistics for $D_n$ is from 0 to 400 m, using 4 m increment. Interestingly, for square region, varying density $n$ does not change the distribution and transition at all statistically. The transition in the circular region is slower than the counterpart in the unbounded region. We found $D_n \leqslant 130$ m almost surely for circular region with $n = 100$.

We then present our experiments of various routing methods on different topologies. We choose 100 nodes distributed randomly in a circular area with radius 100 m. Each node is specified by a random $x$, $y$ co-ordinate, with transmission radius 30 m. Figure 5 illustrates some discussed topologies. We randomly select 20% of nodes as source; and for each source, we randomly choose 20% of nodes as destination. The statistics are computed over 10 different node sets. We found that $LDel^{(2)}(V)$ and $PLDel(V)$ are almost the same as $Del(V)$. The differences lie near the boundary. These two graphs are preferred over the Yao graph because we can apply the right hand rule when the simple heuristic localized routing fails.

Interestingly, we found that when the underlying network topology is Yao graph, $Del(V)$, $LDel^{(2)}(V)$, or $PLDel(V)$, the compass routing, random compass routing and the greedy routing delivered the packets in all our experiments. Note that it was proved that the Delaunay triangulation guarantees the delivery of the packets for these three routing methods. We also found that the local Delaunay triangulation and the planarized local Delaunay triangulation are almost the same as the Delaunay triangulation. The only differences lye near the domain boundary, which does not affect the localized routing too much. Thus, as we expected, the compass routing, random compass routing and the greedy routing delivered the packets in all our simulations for Delaunay related structures. The reason they also delivered the packets when Yao structure is used as the underlying topology could be there is a node within the transmission range in the direction of the destination with high probability when the number of nodes within transmission range is large enough.

Table I illustrates the delivery rates of different localized routing protocols on various network topologies. For nearest neighbour routing and farthest neighbour routing, we choose the angle $\alpha = \pi/3$. In other words, we only choose the nearest node or the farthest node within $\pi/3$ of the destination direction. The $LDel^{(2)}(V)$ and $PLDel(V)$ graphs are preferred over the



DOI: 10.1002/dac

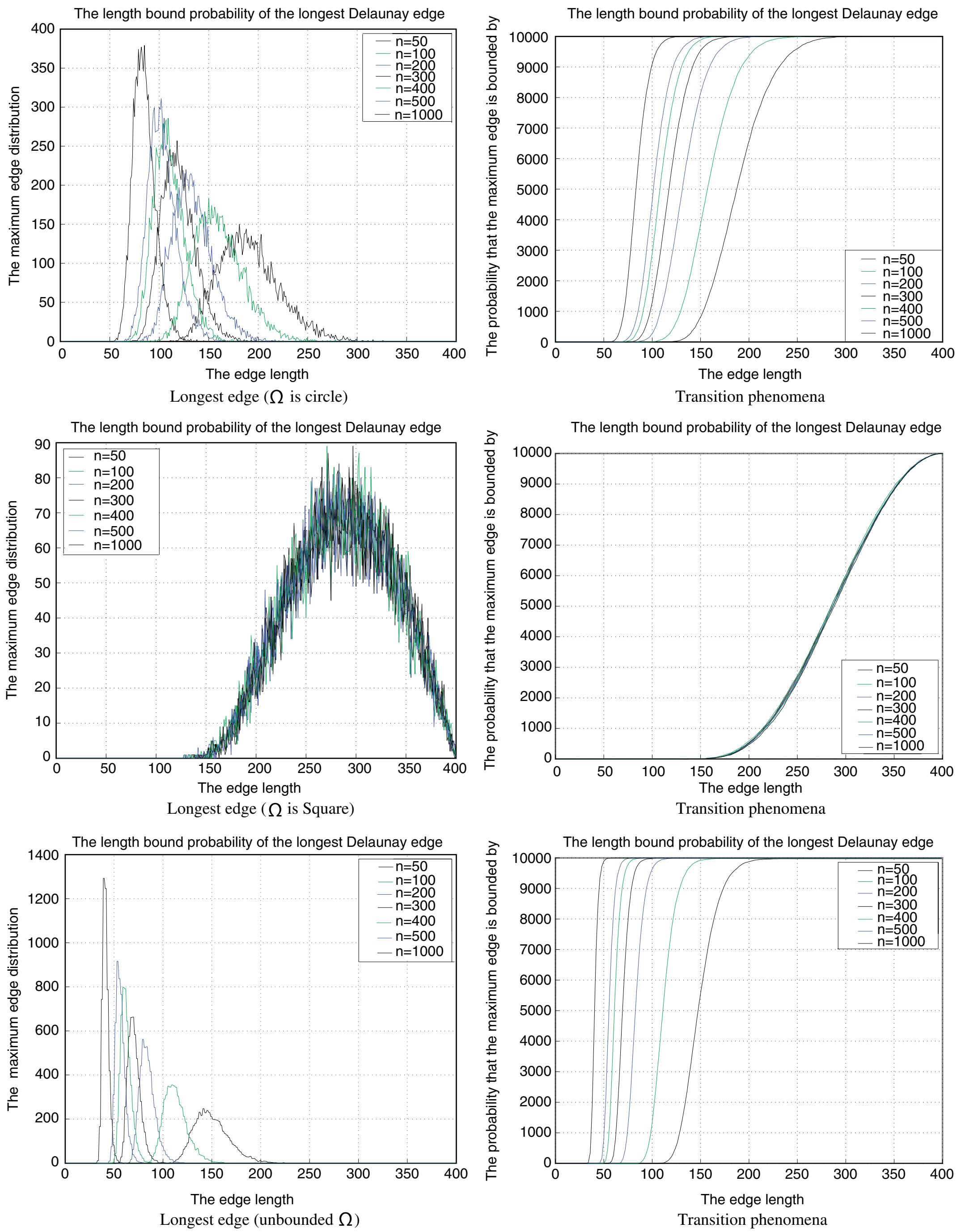


Figure 4. Transition phenomena of $D_n$ when Ω is circle, square, and unbounded.



DOI: 10.1002/dac

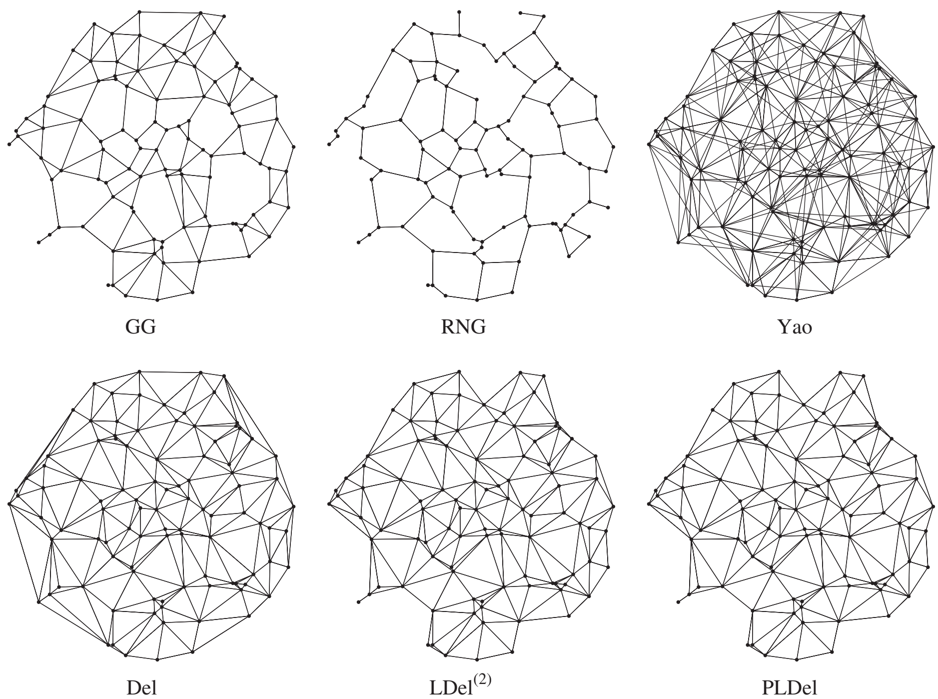


Figure 5. Various planar network topologies (except Yao).

Table I. Delivery rate.

| | Yao | RNG | GG | Del | $LDel^{(2)}$ | PLDel |
|---|---|---|---|---|---|---|
| NN | 98.7 | 44.9 | 83.2 | 99.1 | 97.8 | 98.3 |
| FN | 97.5 | 49 | 81.7 | 92.1 | 97 | 97.6 |
| MFR | 98.5 | 78.5 | 96.6 | 95.2 | 96.6 | 99.7 |
| Cmp | 100 | 86.6 | 99.6 | 100 | 100 | 100 |
| RCmp | 100 | 91.7 | 99.9 | 100 | 100 | 100 |
| Grdy | 100 | 87.5 | 99.6 | 100 | 100 | 100 |
| GCmp | 93 | 95.5 | 99.9 | 100 | 100 | 100 |
| DTR | | | | 100 | 100 | 100 |

Yao graph because we can apply the right-hand rule when previous simple heuristic localized routing fails. Both References [12, 13] use the greedy routing on Gabriel graph and use the right hand rule when greedy fails.

Table II illustrates the maximum spanning ratios of $\|\Pi(s,t)\|/\|st\|$, where $\Pi(s,t)$ is the path traversed by the packet using different localized routing protocols on various network topologies from source $s$ to destination $t$. Because the localized Delaunay triangulation is much dense than all previous known planar network topologies such as Gabriel graph and the relative neighbourhood graph, the delivery rates of many online routing methods are near or equal 100%. However, we have to admit that the travelled distance by the Delaunay based routing



DOI: 10.1002/dac

Table II. Maximum spanning ratio.

| | Yao | RNG | GG | Del | LDel$^{(2)}$ | PLDel |
|---|---|---|---|---|---|---|
| NN | 1.9 | 2.1 | 1.9 | 1.7 | 1.8 | 1.9 |
| FN | 4.2 | 2.8 | 2.7 | 5.2 | 3.4 | 3.1 |
| MFR | 4.8 | 3.2 | 2.4 | 4.5 | 3.9 | 4.1 |
| Cmp | 3.3 | 2.9 | 2.8 | 1.6 | 1.8 | 2.0 |
| RCmp | 2.7 | 3.0 | 2.4 | 1.7 | 2.0 | 1.8 |
| Grdy | 2.1 | 3.5 | 2.2 | 2.0 | 1.9 | 1.9 |
| GCmp | 2.8 | 3.2 | 2.6 | 1.7 | 1.8 | 2.0 |
| DTR | | | | 6.4 | 6.4 | 6.5 |

Table III. Maximum spanning ratio of face routing: (GG/LDel).

| $n$ | $r = 30$ | $r = 40$ | $r = 50$ | $r = 60$ | $r = 70$ |
|---|---|---|---|---|---|
| 20 | 64.9/16.2 | 30.7/12.4 | 66.0/17.4 | 46.4/12.3 | 18.9/13.3 |
| 30 | 14.2/11.0 | 15.3/13.6 | 15.7/11.9 | 15.1/13.0 | 15.3/13.4 |
| 40 | 39.8/18.4 | 36.2/13.7 | 31.1/13.1 | 33.6/12.4 | 15.4/12.6 |
| 50 | 12.7/13.2 | 33.7/13.9 | 33.6/13.4 | 47.6/12.5 | 28.5/11.1 |
| 60 | 26.6/13.6 | 42.6/13.1 | 27.5/13.7 | 34.1/13.7 | 41.5/13.2 |
| 70 | 49.8/16.7 | 31.8/13.6 | 29.3/18.2 | 71.4/14.4 | 32.2/14.3 |
| 80 | 31.4/14.0 | 34.3/14.9 | 67.5/16.0 | 52.2/14.8 | 26.5/13.0 |
| 90 | 41.5/14.0 | 44.4/13.9 | 39.1/14.3 | 60.2/16.3 | 33.4/14.8 |
| 100 | 41.7/18.4 | 57.3/14.7 | 50.6/15.0 | 79.2/19.2 | 86.1/14.3 |

Table IV. Average spanning ratio of face routing: (GG/LDel).

| $n$ | $r = 30$ | $r = 40$ | $r = 50$ | $r = 60$ | $r = 70$ |
|---|---|---|---|---|---|
| 20 | 3.2/2.9 | 3.0/2.8 | 2.9/2.9 | 2.9/2.7 | 2.9/2.7 |
| 30 | 4.7/4.4 | 4.8/4.5 | 4.6/4.4 | 4.4/4.5 | 4.3/4.1 |
| 40 | 5.0/5.2 | 5.2/5.5 | 5.1/4.8 | 5.0/4.8 | 5.1/4.9 |
| 50 | 5.5/4.9 | 5.9/5.3 | 5.9/5.4 | 5.7/5.5 | 5.3/5.3 |
| 60 | 6.1/5.3 | 6.1/5.7 | 6.1/5.4 | 6.3/5.7 | 6.0/6.1 |
| 70 | 6.5/5.9 | 6.5/5.6 | 6.6/6.1 | 6.4/6.2 | 6.6/5.8 |
| 80 | 6.9/5.9 | 6.7/6.1 | 7.1/6.4 | 6.9/5.9 | 6.6/5.8 |
| 90 | 7.0/6.0 | 7.1/6.4 | 7.4/6.5 | 7.5/6.4 | 7.1/6.0 |
| 100 | 7.3/6.4 | 7.4/6.5 | 7.7/6.8 | 7.3/6.6 | 7.3/6.5 |

method DTR is larger than that by most previous methods for most source and destination pairs, although the actual distance of the travelled path is at the same level. Remember that, Delaunay based routing method has to travel some path back and forth to explore a better path. Nevertheless, Delaunay based routing is the only method known that can guarantee that the total travelled distance by the packet is within a constant factor of the minimum in any case.

We also conducted extensive simulations of the face routing on Gabriel graph and the local Delaunay triangulation $LDel^1(V)$. We choose $n = 20, 30, \ldots, 90, 100$ nodes randomly and uniformly distributed in a square of length 100 m. The uniform transmission range of nodes are set as $r$, where $r$ varies from 30, 40, 50, 60 to 70 m. See Tables III and IV for the maximum and



DOI: 10.1002/dac

the averaged spanning ratio achieved. The maximum and the average is computed for all pair of nodes. Given $n$ and $r$, we generate 10 sets of random $n$ points. We found that the spanning ratio of the face routing is significantly less when LDel is used instead of GG. It may be due to LDel has more edges, thus the faces traversed by the face routing is often smaller when LDel is used than the case when GG is used.

## 5. DISCUSSION

In this paper, we proposed a Delaunay-based localized routing method. Our Delaunay-based routing can be combined with greedy routing to make the routing protocol simpler and more efficient. We can apply our Delaunay based routing only when greedy routing fails. If greedy routing works, we will keep using it until the local minimum happens. Thus, the performance measurement in this paper is only to compare the improvement of the routing part when the greedy routing fails. Note that the total cost of this kind of combined routing (such as those in [12, 13]) is composed of two parts: the cost of greedy routing and the cost of path found to get out of local minimum when greedy routing fails. Here, we focused on comparing the second part.

For our proposed Delaunay-based routing, actually, some improvements can also be made. Assume that $uv$ and $vw$ are links in Delaunay and Delaunay routing selects both links $uv$ and $vw$ for routing. If $u$ and $w$ are within the transmission range of each other (clearly they may not be Delaunay neighbours), we can short-cut the route by replacing $uv$ and $vw$ with direct link $uw$ only. Similar improvement can be made till no such improvement exists. This will reduce the hop number of the found path. We can also add another criterion to decide whether to do short-cut: we do short-cut only if it saves more energy under the considered energy model. Note that depending on the energy model, short-cut may save energy, or may not save energy.

Another variation of the proposed Delaunay-based routing is that instead of using the binary search to find the route we can use the face routing on localized Delaunay to find the route. Note that the binary search method may not work when using localized Delaunay instead of Delaunay (although the probability that this happens is small) while face routing can always work on all planar graphs. If compared with greedy face routing [12, 13] on Gabriel graph, the face routing on localized Delaunay always uses less links, since localized Delaunay is denser and contains Gabriel graph as a subgraph.

## 6. CONCLUSION

In this paper, we showed that, given a set of randomly distributed wireless sensor nodes over a unit-area region with node density $n$, when the transmission range $r_n$ satisfies $\pi r_n^2 \geqslant 8 \log n/n$, the localized Delaunay triangulation equals the Delaunay triangulation with probability at least $1-(1/n)$. If $\pi r_n^2 \geqslant 8 \log n/n$, the induced network topology is connected with probability at least $1-(1/n^7)$. In other words, with high probability, we can construct the Delaunay triangulation using the localized Delaunay triangulation if the network is connected. Thus, we can apply a localized routing protocol [10] that guarantees that the distance travelled by the packets is no more than a small constant factor of the minimum. We also conducted simulations to show that the delivery rates of existing localized routing protocols are increased when localized Delaunay triangulation is used instead of several previously proposed topologies.



DOI: 10.1002/dac

Note that the Delaunay based routing method DTR works only when a Delaunay triangulation is obtained. Though in this paper we showed with high probability localized Delaunay triangulation can be used instead of Delaunay triangulation, there are still cases Delaunay triangulation is needed. Currently, when we found that Delaunay triangulation is not constructed or cannot be approximated by localized Delaunay triangulation, we rely on other heuristic to route the packets. We leave it as a future work to design a localized routing protocol that can guarantee the travelled distance using only the localized Delaunay triangulation. Also we are interested in designing a localized routing protocol such that the found path consumes energy within a constant factor of the optimum with high probability, since Theorem 3.4 showed that the path found by Delaunay-based routing may be not power efficient.

## ACKNOWLEDGEMENTS

The work of X.-Y. Li was partially supported by U.S. National Science Foundation CCR-0311174. The work of Y. Wang was supported, in part, by funds provided by the University of North Carolina at Charlotte. The authors also greatly appreciate the anonymous reviewers and the editors for their constructive comments for improving the paper.

## REFERENCES

1. Johnson DB, Maltz DA. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, Imielinski T, Korth HF (eds). Kluwer Academic Publishers: Dordrecht, 1996.
2. Murthy S, Garcia-Luna-Aceves J. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications Journal*, *Special issue on Routing in Mobile Communication Networks* 1996; **1**(2):183–197.
3. Park V, Corson M. A highly adaptive distributed routing algorithm for mobile wireless networks. *IEEE Infocom*, Kobe, Japan, 1997.
4. Perkins C. *Ad-hoc* on-demand distance vector routing. *Proceedings of MILCOM'97*, Monterey, CA, U.S.A., November 1997.
5. Perkins C, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing. *Proceedings of the ACM SIGCOMM*, London, U.K., October 1994.
6. Sinha P, Sivakumar R, Bharghavan V. Cedar: core extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications* 1999; **17**(8):1454–1465.
7. Royer E, Toh C. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, April 1999.
8. Ramanathan S, Steenstrup M. A survey of routing techniques for mobile communication networks. *ACM/Baltzer Mobile Networks and Applications* 1996; 89–104.
9. Mauve M, Widmer J, Harenstein H. A survey on position-based routing in mobile *ad hoc* networks. *IEEE Network Magazine* 2001; **15**(6):30–39.
10. Bose P, Morin P. Online routing in triangulations. *Proceedings of the 10th Annual International Symposium on Algorithms and Computation ISAAC*, 1999.
11. Stojmenovic I, Lin X. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems* 2001; **12**(10):1023–1032.
12. Bose P, Morin P, Stojmenovic I, Urrutia J. Routing with guaranteed delivery in *ad hoc* wireless networks. *ACM/Kluwer Wireless Networks* 2001; **7**(6):609–616.
13. Karp B, Kung HT. GPSR: greedy perimeter stateless routing for wireless networks. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking* (*MobiCom*), Boston, MA, U.S.A., 2000.
14. Gabriel KR, Sokal RR. A new statistical approach to geographic variation analysis. *Systematic Zoology* 1969; **18**:259–278.
15. Bose P, Devroye L, Evans W, Kirkpatrick D. On the spanning ratio of gabriel graphs and beta-skeletons. *Proceedings of the Latin American Theoretical Infocomatics* (*LATIN*), Cancun, Mexico, 2002.
16. Eppstein D. Beta-skeletons have unbounded dilation. *Technical Report ICS-TR-96-15*, University of California, Irvine, 1996.



DOI: 10.1002/dac

17. Wang W, Li X-Y, Moaveni-Nejad K, Wang Y, Song W-Z. The spanning ratios of beta-skeletons. *Canadian Conference on Computational Geometry* (*CCCG 2003*), Halifax, Canada, 2003.
18. Li X-Y, Calinescu G, Wan P-J, Wang Y. Localized Delaunay triangulation with application in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Processing* 2003; **14**(10):1035–1047.
19. Gupta P, Kumar PR. Critical power for asymptotic connectivity in wireless networks. *Stochastic Analysis*, *Control*, *Optimization and Applications*: *A Volume in Honor of W. H. Fleming*, McEneaney WM, Yin G, Zhang Q (eds). Birkhauser: Boston, 1998.
20. Kranakis E, Singh H, Urrutia J. Compass routing on geometric networks. *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, 1999; 51–54.
21. Li X-Y, Wan P-J, Wang Y. Power efficient and sparse spanner for wireless ad hoc networks. *Proceedings of IEEE International Conference on Computer Communications and Networks* (*ICCCN01*), Los Alamitos, California, 2001.
22. Keil JM, Gutwin CA. The Delaunay triangulation closely approximates the complete euclidean graph. *Proceedings of 1st Workshop on Algorithms and Data Structures*, Ottawa, Canada, 1989.
23. Keil JM, Gutwin CA. Classes of graphs which approximate the complete euclidean graph. *Discrete Computational Geometry* 1992; **7**(1):13–28.
24. Chew PL. There is a planar graph as good as the complete graph. *Proceedings of the 2nd Symposium on Computational Geometry*, Yorktown Heights, New York, U.S.A., 1986; 169–177.
25. Toussaint GT. The relative neighborhood graph of a finite planar set. *Pattern Recognition* 1980; **12**(4):261–268.
26. Yao AC-C. On constructing minimum spanning trees in *k*-dimensional spaces and related problems. *SIAM Journal of Computing* 1982; **11**:721–736.
27. Lukovszki T. New results on geometric spanners and their applications. *Ph.D. Thesis*, University of Paderborn, 1999.
28. Fischer M, Lukovszki T, Ziegler M. Partitioned neighborhood spanners of minimal outdegree. *Technical Report*, Heinz Nixdore Institute, Germany, 1999.
29. Wattenhofer R, Li L, Bahl P, Wang Y-M. Distributed topology control for wireless multihop ad-hoc networks. *IEEE INFOCOM'01*, Anchorage, Alaska, U.S.A., 2001.
30. Li L, Halpern JY, Bahl P, Wang YM, Wattenhofer R. Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks. *ACM Symposium on Principle of Distributed Computing* (*PODC*), Newport, Rhode Island, U.S.A., 2001.
31. Datta S, Stojmenovic I, Wu J. Internal node and shortcut based routing with guaranteed delivery in wireless networks. *Cluster Computing* 2002; **5**(2):169–178.
32. Stojmenovic I, Datta S. Power and cost aware localized routing with guaranteed delivery in wireless networks. *Proceedings of Seventh IEEE Symposium on Computers and Communications ISCC*, Taormina, Italy, 2002.
33. Seddigh M, Solano Gonzalez J, Stojmenovic I. RNG and internal node based broadcasting algorithms for wireless one-to-one networks. *ACM Mobile Computing and Communications Review* 2002; **5**(2):37–44.
34. Bose P, Brodnik A, Carlsson S, Demaine ED, Fleischer R, Lopez-Ortiz A, Morin P, Munro JI. Online routing in convex subdivisions. *International Symposium on Algorithms and Computation*, Taipei, Taiwan, 2000; 47–59.
35. Morin P. Online routing in geometric graphs. *Ph.D. Thesis*, Carleton University School of Computer Science, 2001.
36. Kuhn F, Wattenhofer R, Zollinger A. Worst-case optimal and average-case efficient geometric *ad-hoc* routing. *Proceedings of 4th ACM International Symposium on Mobile Ad-Hoc Networking and Computing* (*MobiHoc*), Annapolis, Maryland, U.S.A., 2003.
37. Kuhn F, Wattenhofer R, Zhang Y, Zollinger A. Geometric *ad-hoc* routing: of theory and practice. *Proceedings of 22nd ACM International Symposium on the Principles of Distributed Computing* (*PODC*), Boston, Massachusetts, U.S.A., 2003.
38. Kuhn F, Wattenhofer R, Zollinger A. Asymptotically optimal geometric mobile ad-hoc routing. *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications* (*Dial-M*). ACM Press: New York, 2002; 24–33.
39. Basagni S, Chlamtac I, Syrotiuk VR, Woodward BA. A distance routing effect algorithm for mobility (dream). *Proceedings of ACM/IEEE MobiCom'98*, Dallas, Texas, U.S.A., 1998.
40. Haas Z, Liang B. *Ad-hoc* mobility management with uniform quorum systems. *IEEE/ACM Transactions on Networking* 1999; **7**(2):228–240.
41. Stojmenovic I. A routing strategy and quorum based location update scheme for *ad hoc* wireless networks. *Technical Report TR-99-09*, Computer Science, SITE, University of Ottawa, 1999.
42. Amouris KN, Papavassiliou S, Li M. A position based multi-zone routing protocol for wide area mobile ad-hoc networks. *Proceedings of 49th IEEE Vehicular Technology Conference*, 1999; 1365–1369.
43. Penrose M. On *k*-connectivity for a geometric random graph. *Random Structures and Algorithms* 1999; **15**:145–164.
44. Penrose M. The longest edge of the random minimal spanning tree. *Annals of Applied Probability* 1997; **7**:340–361.
45. Gao J, Guibas LJ, Hershburger J, Zhang L, Zhu A. Geometric spanner for routing in mobile networks. *Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc 01*), Long Beach, California, U.S.A., 2001.
46. Dobkin DP, Friedman SJ, Supowit KJ. Delaunay graphs are almost as good as complete graphs. *Discrete Computational Geometry* 1990; **5**(4):399–407.




DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

**Yu Wang** is an assistant professor in Department of Computer Science, the University of North Carolina at Charlotte. He received his PhD degree in computer science from Illinois Institute of Technology in 2004, his BS degree and MS degree in computer science from Tsinghua University, China, in 1998 and 2000. His current research interests include computer networks, wireless networks, mobile computing, algorithm design, and artificial intelligence. He is a member of the ACM, IEEE, and SIAM.

**Xiang-Yang Li** has been with Department of Computer Science at the Illinois Institute of Technology since 2000, where he is an associate professor now. He received MS (2000) and PhD (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received his Bachelor degrees of Computer Science and Business Management from Tsinghua University, China in 1995. His research interests span the wireless *ad hoc* networks, game theory, computational geometry, and cryptography and network security. He is a Member of the ACM, IEEE, and IEEE Communication Society.



DOI: 10.1002/dac