

# MAP: Medial axis based geometric routing in sensor networks\*

Jehoshua Bruck · Jie Gao · Anxiao (Andrew) Jiang

Published online: 23 October 2006  
© Springer Science + Business Media, LLC 2006

**Abstract** One of the challenging tasks in the deployment of dense wireless networks (like sensor networks) is in devising a routing scheme for node to node communication. Important consideration includes scalability, routing complexity, quality of communication paths and the load sharing of the routes. In this paper, we show that a compact and expressive abstraction of network connectivity by the medial axis enables efficient and localized routing. We propose MAP, a Medial Axis based naming and routing Protocol that does not require geographical locations, makes routing decisions locally, and achieves good load balancing. In its preprocessing phase, MAP constructs the medial axis of the sensor field, defined as the set of nodes with at least two closest bound-

ary nodes. The medial axis of the network captures both the complex geometry and non-trivial topology of the sensor field. It can be represented succinctly by a graph whose size is in the order of the complexity of the geometric features (e.g., the number of holes). Each node is then given a name related to its position with respect to the medial axis. The routing scheme is derived through local decisions based on the names of the source and destination nodes and guarantees delivery with reasonable and natural routes. We show by both theoretical analysis and simulations that our medial axis based geometric routing scheme is scalable, produces short routes, achieves excellent load balancing, and is very robust to variations in the network model.

---

\*A preliminary version appeared in ACM International Conference on Mobile Computing and Networking (MobiCom'05), August, 2005. This work was supported in part by the Lee Center for Advanced Networking at the California Institute of Technology, and by NSF grant CCR-TC-0209042.

---

Jie Gao's work was done at Center for the Mathematics of Information, California Institute of Technology, Pasadena, CA 91125

---

Anxiao (Andrew) Jiang's work was done at Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125

---

J. Bruck (✉)  
Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA  
e-mail: bruck@paradise.caltech.edu

J. Gao  
Computer Science Department, Stony Brook University,  
Stony Brook, NY 11794, USA  
e-mail: jgao@cs.sunysb.edu

A. (Andrew) Jiang  
Computer Science Department, Texas A&M University, College Station, TX 77843-3112, USA  
e-mail: ajiang@cs.tamu.edu

**Keywords** Medial axis · Routing · System design · Sensor networks

## 1 Introduction

Routing is elementary in all communication networks. The design of routing algorithms is tightly coupled with the design of auxiliary infrastructure that abstracts the network connectivity. For networks with stable links and powerful nodes, such as the Internet, infrastructures such as routing tables are constructed and maintained so that routing can be performed efficiently at each router by a routing table look-up, and routing paths are close to optimum. For networks with fragile links, constantly changing topologies, and nodes with less resourceful hardware, such as *ad hoc* mobile wireless networks, routing tends to be on-demand with no pre-computed infrastructures. However, without any auxiliary infrastructure, discovery of routes may have to rely on flooding the network.

In this paper we focus on routing in wireless sensor networks, where sensor nodes are stationary and deployed in a geometric space. Each sensor node has constrained power supply, thus energy conservation is an important consideration in the design of network protocols. For sensor networks that target at long-term usage such as environment monitoring, the demand for scalable point-to-point routing is increasing in order to support in-network information processing, content-based data storage and retrieval [25, 29], target tracking and detection [32], sensor tasking and control, and complex query mechanisms [16]. Depending on the applications, a node may specify the destination ID as in the case of sensor tasking, or in the case of target tracking, acquire the destination by a content-based storage structure such as the distributed hash table. Reactive routing protocols, which are designed mainly for ad hoc mobile networks and rely on flooding for route discovery, is too energy-expensive for sensor networks. It is also observed that wireless links for static sensor nodes are reasonably stable [17, 27]. Therefore it is desirable to perform moderate preprocessing and maintain some lightweight infrastructure so that more efficient and localized routing can be adopted.

A good intuition on how to build a lightweight and effective auxiliary routing infrastructure is that sensor networks are closely related to the geometric environment where they are deployed. Two nodes can directly communicate when they are geographically close. Thus geometric proximity information has high correlation with the network topology. This intuition has been used in geographical forwarding, a clever idea to effectively make routing decisions based on the geographical locations of destinations and the one-hop neighboring nodes—a packet is greedily forwarded to the one-hop neighbor that is geographically closest to the destination [6, 20, 23]. Such an abstraction of the network connectivity based on nodes' Euclidean coordinates has tremendously simplified the design of routing protocols and improved routing efficiency. For a sensor network with uniform and dense sensor deployment in a flat and regular region, geographical forwarding is an efficient and scalable scheme that produces almost shortest paths with very little overhead.

A natural question on the practicality of geographical routing is how to obtain the geographical locations of a large number of sensors. An essential part of the preprocessing overhead of building the infrastructure for geographical routing is to solve the localization problem, namely, finding the Euclidean coordinates of the sensors. Localization can be done by either hardware support such as Global Positioning Systems (GPS), or by algorithms that induce the locations of sensor nodes from their local interactions. In fact, if sensors are densely deployed in a flat regular region with simple geometry (e.g., a disk with no holes), greedy geographical routing is robust enough to localization errors and approximate locations suffice [8, 28].

Greedy geographical forwarding, however, runs into serious problems for sensor fields with complex geometry. In many of the real-world situations where sensor networks are deployed, such as metropolitan areas, warehouses, university campuses or airport terminals, the sensor field naturally has complex shape and/or many holes (regions where sensors are not deployed due to the existence of obstacles). When there are holes in a sensor field, greedy forwarding can fail when all the neighbors are further away from the destination. In other words, a route created by greedy forwarding tries to follow the straight line from source to destination, which is often blocked by obstacles in a complex environment. A number of ways have been devised to get around holes. For example, face routing or perimeter routing [6, 20, 23] deals with this case by routing a packet along the face of a planar subgraph until greedy forwarding can be performed again. If the sensor network has rich geometric features, perimeter routing has to be adopted frequently. There are several issues on applying face routing or perimeter routing in practice [21, 22]. The correct construction of the planar subgraph depends heavily on two impractical assumptions, the accurate location information and the unit disk graph model. Accurate location information is very hard to obtain. The communication graph is not a unit disk graph in practice [17]. Inaccurate location information or a slight deviation of the communication graph from the unit disk graph model may cause the routing subgraph to be disconnected or have crossing edges [21, 22, 30]. Further, perimeter routing produces awkward routing paths along the boundaries of holes. Overloading of nodes on the boundaries of holes exhausts the batteries of those nodes quickly, which will further enlarge the holes and eventually connect small holes to big holes or even partition the network into disconnected pieces. Notice that such an unbalance of loads is created not by traffic patterns but by the defect of routing algorithms.

The failure of greedy forwarding for sensor fields with complex geometry and/or non-trivial topology is mainly due to the mismatch of routing rules with the real network connectivity. The geographical locations, an abstraction on which routing rules are based, does not correlate well with the connectivity graph. Two nodes that are geographically close may actually be far away in the connectivity graph. A good abstraction for this case should not only takes into account the geometric proximity of the sensors, but also the global geometric shape and topological features of the sensor field. This intuition is validated by the observation that the global shape and the topological features of the layout mostly reflect the underlying structure of the environment (e.g. obstacles), and they are likely to remain stable. Nodes/links may come and go. But only when such changes are of large quantities and geographically correlated, can they possibly modify the global shape of the sensor field or destroy/create large-scale topological features. Thus we can afford to explicitly

compute an abstraction of the geometry of sensors and carry out proactive routing at this abstract level, such that these high-level combinatorial routes can be efficiently realized in the network by localized and decentralized protocols. A protocol that explicitly states the importance of topological information in routing in sensor networks with large holes, called GLIDER, was recently proposed by Fang et al. [12]. It is a landmark-based naming and routing scheme, where the global topology of the network is represented by a compact abstract Delaunay triangulation on a set of landmarks, and is used in a global planning step to guide routes around holes. However, the selection of landmarks has a critical impact on the performance of such landmark-based routing algorithms. Understanding how to select a good set of landmarks is a challenging task.

## 2 Overview

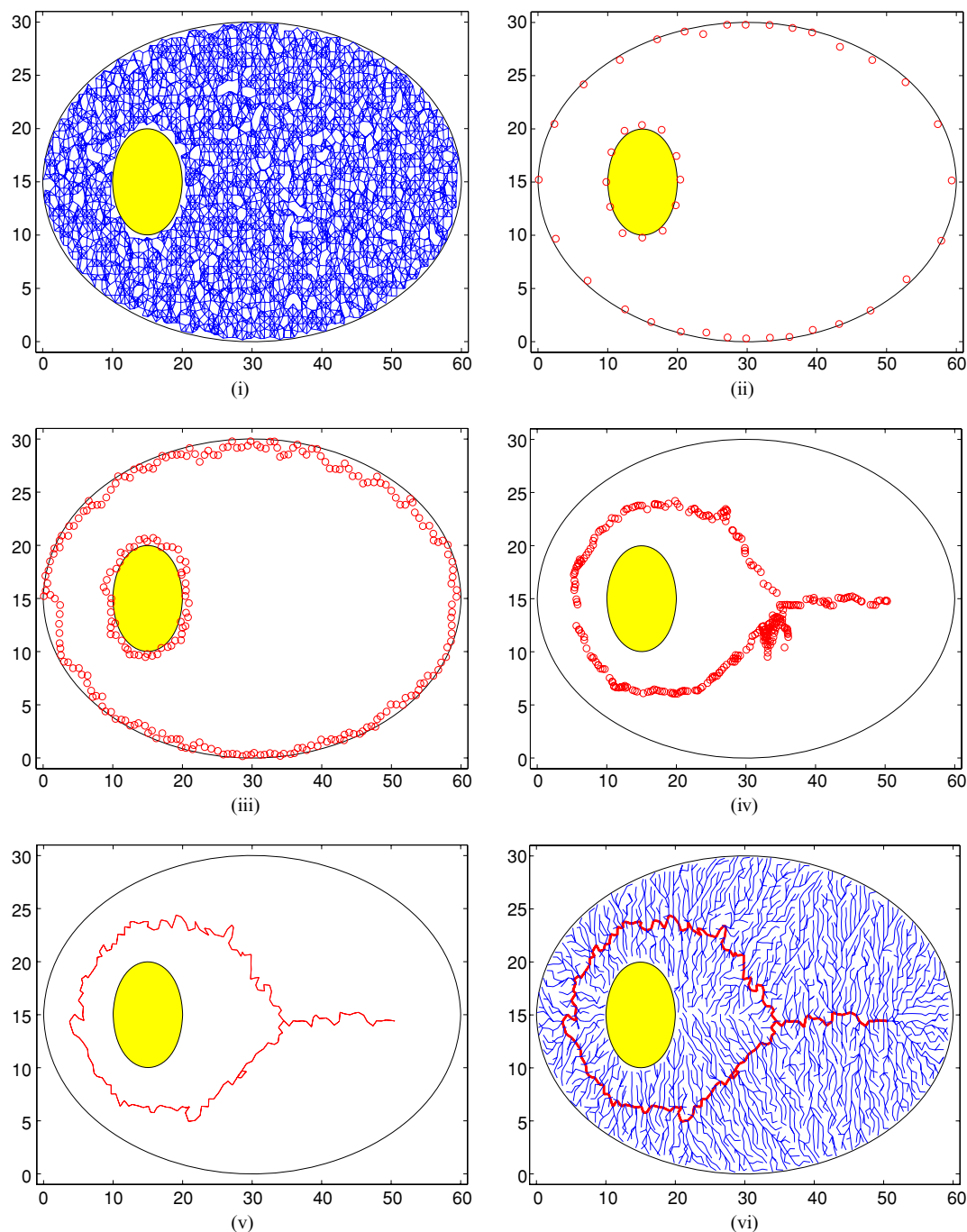
In this paper we explore an appropriate geometric abstraction of sensor networks that enables efficient and localized routing. The shapes of regions or surfaces have been studied extensively in computational geometry, and various structures have been proposed for efficient representation of shapes. One of them is the medial axis [3, 10], which is defined as the set of points with at least two closest neighbors on the boundaries of the shape. The medial axis is a ‘skeleton’ of a region that captures all the topological features. Therefore it has been used extensively to represent, reason about and explore properties of shapes, in areas such as robot path planning [19], surface reconstruction [1, 2] and shape classification [31]. In this paper we show that the medial axis of a sensor field is a good abstraction of the communication network. The medial axis can be constructed efficiently by using only the connectivity information, can be represented compactly by a graph whose size is proportional to the number of large geometric features, and enables an efficient gradient-descending routing algorithm.

We propose MAP, a medial axis based naming and routing protocol for sensor networks, in particular those with complicated geometric and topological features. Our MAP naming and routing methods are similar in spirit with GLIDER [12], in the way that we also take a compact abstraction of the global topology of the sensor field. The difference between MAP and GLIDER includes the choice of this abstraction—namely, the medial axis graph in MAP and the combinatorial Delaunay graph on landmarks in GLIDER—as well as how this abstraction is used to guide localized routing. Specifically, GLIDER relies on a carefully selected set of landmarks to discover the global topology of the network, while MAP eliminates such artifact and depends only on the inherent properties of the sensor field. MAP depends only on the connectivity graph and does not require the communication network be a unit

disk graph. MAP consists of two protocols: the Medial Axis Construction Protocol (MACP) that constructs the medial axis and the corresponding naming scheme at the initialization stage of the network; and the Medial Axis based Routing Protocol (MARP), which, with the help of a compact medial axis graph, routes packets by local gradient descent with only the names of source and destination nodes.

The construction of the medial axis has several steps. See Fig. 1 for an example. We first select a set of samples on hole boundaries (including the outer boundary) (Fig. 1 (ii)). By connecting nearby samples, we find most of the boundary nodes (Fig. 1 (iii)). The medial axis is identified as the set of nodes with at least two closest points on boundaries (Fig. 1 (iv) and (v)), and is represented by a medial axis graph which is a combinatorial graph with a size proportional to the number of large geometric features. This medial axis graph is very compact and is known to every sensor. The medial axis graph of Fig. 1 has two vertices, one edge and one self-loop. Each sensor is given a name with respect to the medial axis graph (Fig. 1 (vi)). In particular, for each sensor  $w$  on the medial axis, we define a chord as the shortest path (tree) from  $w$  to one of its closest sensor node on the boundary. A sensor’s name includes the chord on which it stays, and a normalized distance to its corresponding medial axis sensor. Such a naming scheme partitions the sensor field into canonical regions inside each of which a local Cartesian coordinate system is defined with one axis as an edge of the medial axis graph and the other axis as a chord of a vertex of that medial axis edge. The local Cartesian coordinate systems are glued together in exactly the same way as indicated by the edge adjacency of the medial axis graph, and provide a smooth and natural road system for the MAP routing algorithm.

MAP supports localized point-to-point routing with only the knowledge of the names of source and destination nodes. The name of the destination can be obtained with a distributed hash table or other location services (such as [25]). MAP routing has two stages. Routing is first planned on the abstract medial axis graph which is usually of a small size, and then realized in each canonical region by reactive local gradient descent routing. By using the medial axis graph in a global planning step, a source can find the reference path, defined as the shortest path in the medial axis graph, from the node in the medial axis corresponding to the source to the one corresponding to the destination. The actual routing rule is of manhattan-type, i.e., first trying to match the medial axis point with that of the destination and route in parallel with the reference path, and then trying to match the distance to the medial axis point with that of the destination and route along chords. Both routing in parallel with the medial axis and along chords can be realized by efficient local gradient descending in the local coordinate systems of the canonical regions. Although nodes on boundaries and the medial axis are important for representing the global geometry and

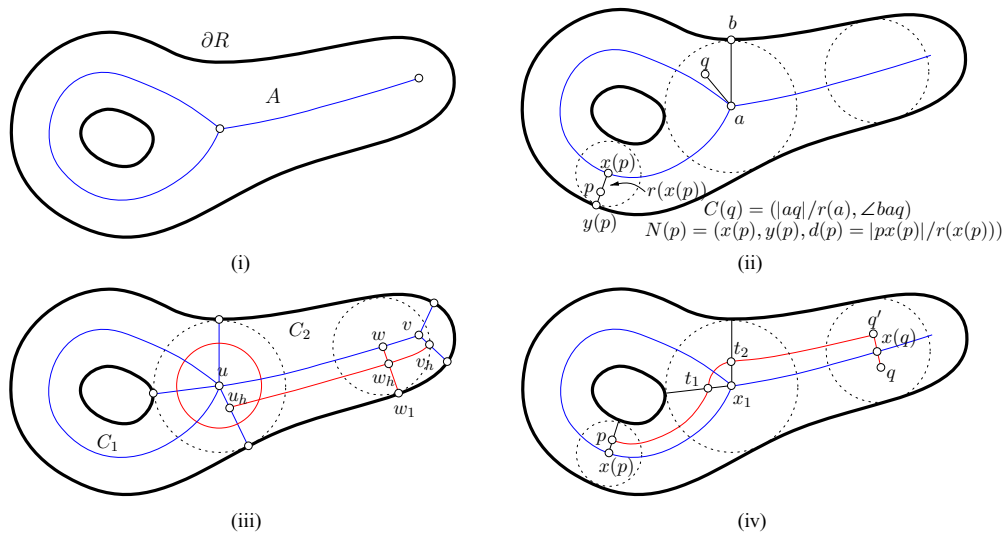


**Fig. 1** We create a scenario of a dense sensor network deployed inside an eclipse-shaped geometric region with a circular obstacle in the middle. There are 3000 sensor nodes, each with a communication radius of 1. The locations of the nodes are discarded after we create the scenario. The global geometry and topology of the network are inferred from the communication network. The sequence of figures illustrates how the medial axis of a sensor network is constructed. (i) A sensor network; (ii) A small sample of hand picked boundary nodes in the network; (iii) Discovery of more boundary nodes based on the sampled boundary nodes; (iv) Locally identified nodes on the medial axis; (v) The medial axis after a clean-up of noises; (vi) Balanced shortest-path trees rooted on the medial axis

topology, they play the same role as the other nodes in the actual MAP routing. The construction of the medial axis is only at the initialization stage with modest preprocessing overhead. Routing is implemented in a localized fashion and thus is scalable.

In summary, MAP has the following good properties:

- *Location-free*: no geographical location is required and only the connectivity graph suffices;



**Fig. 2** An example of the medial axis of the boundary of a closed region  $R$  (only the part of the medial axis in the interior of  $R$  is shown).  $\partial R$  is shown by thick curves. The medial axis  $A$  has a cycle, which means that the region  $R$  has a punched hole. (i) The medial axis of  $\partial R$  and two medial vertices; (ii) The naming scheme. (iii) The road system on  $R$ . Two canonical cells  $C_1$  and  $C_2$  may share a common medial vertex but no common chord; (iv) Routing from  $p$  to  $q$

- *Expressive*: our medial axis based infrastructure captures the large geometric and topological features of a sensor field;
- *Compact*: the medial axis can be represented by a graph of a size proportional to the complexity of large geometric and topological features;
- *Lightweight*: the construction and maintenance of the medial axis is lightweight;
- *Efficient*: the medial axis based routing algorithm uses local gradient descent and is localized;
- *Load balancing*: the medial axis based routing algorithm does not overload any nodes because of design defect;
- *Robust to network model*: MAP does not require that the network model is a unit disk graph and is very robust to variations in the network model.

### 3 Medial axis based naming and routing in continuous Euclidean domain

In this section we present the medial-axis-based naming and routing schemes for a continuous region in the Euclidean plane. All the concepts can be illustrated very nicely for the continuous case. In the next section, we will describe how to adapt these ideas to a discrete sensor field.

#### 3.1 Medial axis

We first review the definition of medial axis [5] for a continuous curve in the Euclidean plane. The *medial axis* of a curve  $F$  is a set of points in the plane which have two or

more (instead of one) closest points in  $F$ . In other words, for a point on the medial axis, if we grow a ball until it hits  $F$ , then the ball has two or more tangent points on  $F$ . The medial axis can be thought of as the Voronoi cell boundaries of the Voronoi diagram<sup>1</sup> defined on an infinite set of points on  $F$ . In this section, we study the medial axis of a bounded region in the Euclidean plane. Suppose  $R$  is a bounded open set in  $\mathbb{R}^2$ , we denote by  $\partial R$  the boundary curve of  $R$ . The medial axis of  $\partial R$  is denoted by  $A$ . It has been proved that for a piecewise analytic boundary in the plane, the medial axis is composed of a finite number of continuous curves [10]. Strictly speaking, the medial axis of  $\partial R$  has two parts in the interior and the exterior of  $R$  respectively. In this paper we focus on the part inside  $R$ . For each point  $a$  on the medial axis, we can draw a maximal disk inside  $R$  with two or more tangent points on  $\partial R$ . The line segment connecting a point  $a$  on the medial axis with its tangent point on  $\partial R$  is called a *chord* of  $a$ . We define a *medial ball*  $B_r(a)$  to be a (closed) ball centered at  $a \in A$  with radius  $r$ , which is tangent to  $\partial R$  at more than one point, and has no point of  $\partial R$  in its interior  $B_r(a) - \partial B_r(a)$ .  $r$  is called the *medial radius* of the point  $a \in A$ , denoted as  $r(a)$ . A point on  $A$  with at least three closest points in  $\partial R$  is called a *medial vertex*. A segment on the medial axis bounded by two medial vertices is denoted as a *medial edge*. Figure 2 shows an example.

<sup>1</sup> The Voronoi diagram of a set of points  $S$  in the Euclidean plane is a planar graph that partitions the plane into convex cells such that all the points inside a cell have the same closest neighbor in  $S$ . In particular, a point on the Voronoi cell boundary has equal distance to at least two points in  $S$ . Further information about Voronoi diagrams can be found in [11].

The medial axis  $A$  of  $\partial R$  retains all the topological information of the region  $R$ . To be precise, it has been shown that any bounded open subset in  $\mathbb{R}^k$  is homotopy equivalent<sup>2</sup> to its medial axis [10, 26]. Therefore, the medial axis can be used as a guideline on how to route from one point to another since it is a compact structure with exactly the same topological features as the underlying domain.

### 3.2 Naming scheme

Each point in  $R$  is assigned a name. We denote by  $|pq|$  the Euclidean distance between  $p, q$ . A point  $p$ 's name is a triple  $N(p) = (x(p), y(p), d(p))$ , where  $x(p) \in A, y(p) \in \partial R \cup \{\perp\}$ , with  $\perp$  representing 'invalid',  $d(p) \in [0, 1]$ . If a point  $p$  is on the medial axis  $A$ , we define its name as  $(p, \perp, 0)$ . If  $p$  is not a point on the medial axis, we define its name as  $N(p) = (x(p), y(p), d(p))$ , where  $p$  stays on a chord  $x(p)y(p)$ , with  $x(p) \in A, y(p) \in \partial R$  and  $d(p)$  being the normalized distance from  $p$  to  $x(p)$ . Namely,  $d(p) = |px(p)|/r(x(p))$ , with  $r(x(p))$  as the medial radius of  $x(p), r(x(p)) = |x(p)y(p)|$ . We also call the point  $x(p)$  the *medial point* of  $p$ , the point  $y(p)$  the *boundary point* of  $p$ , and  $d(p)$  its *height*,  $0 \leq d(p) \leq 1$ . See Fig. 2 (ii) for an example.

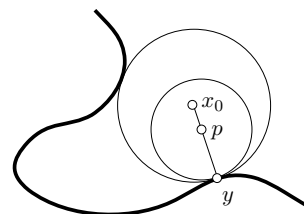
The naming scheme is a valid scheme, because each point has a unique name. To prove this, we first show a couple of lemmas.

**Lemma 3.1.** *For a point  $p$  not on the medial axis, if  $p$  is on a chord  $xy$ , with  $x \in A, y \in \partial R$ , then  $y$  is  $p$ 's only closest point on  $\partial R$ .*

**Proof:** Since  $p$  is not on the medial axis  $A$ ,  $p$  has only one closest point on  $\partial R$ . Assume  $p$ 's closest point on  $\partial R$  is  $y' \neq y$ . Then  $|xy'| \leq |xp| + |py'| < |xp| + |py| = |xy|$ , by triangular inequality. This leads to a contradiction to the fact that  $xy$  is a chord.  $\square$

**Lemma 3.2.** *If  $p$  is not on the medial axis, there is a unique chord through  $p$ .*

**Proof:** Since  $p$  is not on the medial axis,  $p$  has one and only one closest point on  $\partial R$ , denoted as  $y$ . Please see Fig. 3. The ball centered at  $p$  with radius  $|py|, B_{|py|}(p)$ , is tangent to  $\partial R$  at only one point  $y$  and has no other points of  $\partial R$  inside. Now we define a family of balls  $B_{|xy|}(x)$ , where  $x$  is on the line



**Fig. 3** Each point  $p$  not on the medial axis has exactly one chord through it

defined by  $p, y$  with  $p$  on the interior of the line segment  $xy$ . As  $x$  moves away from  $p$ , the ball  $B_{|xy|}(x)$  is enlarged. Take  $x_0$  as the first node such that  $B_{|x_0y|}(x_0)$  is tangent to at least two points on  $\partial R$ . Thus  $x_0$  is on the medial axis since it is tangent to at least two points on  $\partial R$ . So the line segment  $x_0y$  is a chord through  $p$ .

Next we show that  $p$  cannot stay on two chords. Assume otherwise,  $p$  is on two chords  $xy$  and  $x'y'$ . If  $y \neq y'$ , by Lemma 3.1,  $p$  has two closest points, which is not possible. If  $y = y'$ , then  $x, x', y = y'$  are collinear. Suppose that  $x$  is on the interior of  $x'y$ . Then the medial ball of  $x$  is completely inside the medial ball of  $x'$ . The medial ball of  $x$  has at least two points of  $\partial R$  on its boundary, one of which must be inside the medial ball of  $x'$ . That leads to a contradiction.  $\square$

Now we can prove Theorem 3.3.

**Theorem 3.3.** *Every point in  $R$  is assigned a unique name.*

**Proof:** When  $p \in A$ , the theorem is true. When  $p$  is not on the medial axis, there is a unique chord  $xy$  through  $p$ , by Lemma 3.2. By our naming scheme,  $p$ 's name is defined as  $N(p) = (x, y, |xp|/|xy|)$ . Suppose two points  $p, q$  have the same name,  $N(p) = N(q)$ , then they are on the same chord  $x(p)y(p)$  (same as  $x(q)y(q)$ ). Further  $d(p) = d(q)$ , so  $p, q$  must be the same point.  $\square$

The above naming scheme can be thought of as a Cartesian coordinate system aligned with the medial axis. For the purpose of routing, some of the points in  $R$  are also given local polar coordinates. For each medial vertex  $a$ ,  $a$  has at least three closest points on  $\partial R$ . We assign local polar coordinates to all the points within the medial ball  $B_r(a)$ . Specifically, we take  $b \in \partial R$  as one of the closest point of  $a$ . Assign a polar coordinate  $C(b) = (1, 0)$  to  $b$ . Each point  $p$  inside  $B_r(a)$  is assigned a polar coordinate  $C(p) = (|ap|/r, \angle bap)$  (an angle is measured counterclockwise) with respect to  $a$ . See Fig. 2 (ii) for an example. From now on we will call  $N(p)$  the name of  $p$  and  $C(p)$  its polar coordinates.

### 3.3 Road system

The naming scheme naturally produces a road system on the region  $R$ . The medial axis  $A$  and all the chords of medial

<sup>2</sup> Two maps  $f$  and  $g$  from  $X$  to  $Y$  are homotopic if there exists a continuous map  $H : X \times [0, 1] \rightarrow Y$  with  $H(x, 0) = f(x)$  and  $H(x, 1) = g(x)$ . Two spaces  $X$  and  $Y$  have the same homotopy type if there are continuous maps  $f : X \rightarrow Y$  and  $g : Y \rightarrow X$  such that  $g \circ f$  is homotopic to the identity map of  $X$  and  $f \circ g$  is homotopic to the identity map of  $Y$ . In other words, the maps  $f$  and  $g$  define a one-to-one correspondence of the topological features such as connected components, cycles, holes, tunnels, etc., and how these features are related.

vertices partition the region  $R$  into a set of canonical cells  $\{C_i\}, i = 1, \dots, m$ . Each canonical cell  $C$  is bounded by two chords, a medial edge and a segment of the boundary  $\partial R$ . Each medial edge belongs to two canonical cells. A point on the medial axis with  $k$  chords is adjacent to  $k$  canonical cells. We define an  $h$ -latitude curve as a collection of points in  $C$  with height  $h, 0 \leq h \leq 1$ , and an  $x$ -longitude curve as a chord in  $C$  with medial point  $x$ . It is easy to see that an  $x$ -longitude curve in  $C$  is a continuous line segment. The following theorem shows that the  $h$ -latitude curve is also continuous.

**Lemma 3.4.** *Inside a canonical cell, any two chords have no common intersection.*

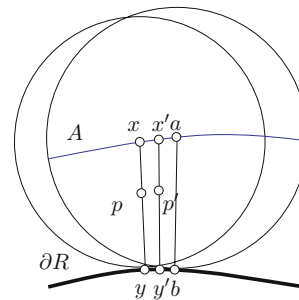
**Proof:** Suppose two chords  $xy, x'y'$  have a common intersection  $z$ , with  $x, x' \in A, y, y' \in \partial R$ . If  $z = x = x'$ , then  $x$  has two chords inside the same canonical cell, which is a contradiction with the definition of canonical cells. Otherwise,  $z$  is not on the medial axis and stays on two chords, which is a contradiction to Lemma 3.2. So the lemma is proved.  $\square$

Denote by  $S_R(p, q)$  the shortest path between  $p, q$  inside a domain  $R$ . The distance between two points  $p, q$  inside  $R$ , i.e., the length of the shortest path  $S_R(p, q)$ , is denoted as  $d_R(p, q)$ .

**Theorem 3.5.** *For a canonical cell  $C$  partitioned by the medial axis and all the chords of medial vertices, the collection of points with height  $h, 0 \leq h \leq 1$ , is a continuous curve.*

**Proof:** To prove this, we show that for any point  $p$  in  $C$  with height  $h$ , we can find a point  $p'$  with height  $h$  within distance  $\varepsilon$ , for any  $\varepsilon > 0$ . Suppose  $p$  stays on a chord  $xy$ , with  $x \in A, y \in \partial R$ . Both the medial axis  $A$  and the boundary  $\partial R$  are continuous, thus we can always find a chord  $x'y'$ , with  $x' \in A, y' \in \partial R$ , such that  $d_A(x, x') \leq \delta, d_{\partial R}(y, y') \leq \delta, \delta = \varepsilon/(1 + 2h)$ . This can be done as follows, we first pick a chord  $ab$  with  $a \in A, b \in \partial R$  such that  $d_A(x, a) \leq \delta$ . If  $d_{\partial R}(y, b) \leq \delta$ , then we take  $ab$  as the required chord  $x'y'$ . Otherwise, we take a point  $y'$  on  $\partial R$  such that  $y'$  is in between  $y$  and  $b$  and  $d_{\partial R}(y, y') \leq \delta$ . The chord  $y'$  stays on is denoted as  $x'y'$ . Then  $x'$  must be in between  $x$  and  $a$  on  $A$  and  $d_A(x, x') \leq d_A(x, a) \leq \delta$ , due to Lemma 3.4. See Fig. 4. By triangular inequality,  $|xy| - 2\delta \leq |x'y'| \leq |xy| + 2\delta$ . We take the point  $p'$  on the chord  $x'y'$  with height  $h$ . Then  $|xp| - 2h\delta \leq |x'p'| \leq |xp| + 2h\delta$ . Then  $|pp'| \leq ||xp| - |x'p'|| + d_A(x, x') \leq (1 + 2h)\delta = \varepsilon$ . So the theorem is proved.  $\square$

The latitude and longitude curves provide a Cartesian coordinate system for the points inside a cell  $C$ . Routing for two points inside the same cell can be done efficiently by first following the latitude curve to a point on the same chord



**Fig. 4.** The  $h$ -latitude curve in a canonical cell is continuous

as the destination, then following the longitude curve to the destination. The continuity of the  $h$ -latitude curve, proved by Theorem 3.5, implies that such routing can be implemented by local gradient descending. Specifically, routing following a latitude curve (longitude curve) is simply following the local gradient with the same height (the same medial point).

The coordinate system defined by latitude and longitude curves is a local one, i.e., defined only on points inside a canonical cell  $C$ . For routing across cells, we hope to transit from one local system to another one smoothly. If two cells  $C_1, C_2$  share a common chord  $xy$ , then the point on  $xy$  with height  $h$  is shared by the  $h$ -latitude curves in  $C_1$  and  $C_2$ . However, it is possible that a pair of medial edges  $e_1, e_2$  are adjacent (they share a common medial vertex), but the cells  $C_1, C_2$  do not share a common chord, where  $e_1$  is on the boundary of the cell  $C_1$  and  $e_2$  is on the boundary of the cell  $C_2$ . See Fig. 2 (iii) for an example. In order to transit from  $C_1$  to  $C_2$ , we use a ‘rotary’ element at each medial vertex, which is defined by the polar coordinate system inside the medial ball of a medial vertex.

For the points inside a medial ball of a medial vertex  $u$ , we define an  $\ell$ -angular curve as the set of points with polar coordinates  $(\ell, \cdot)$ , i.e., the circle centered at  $u$  with radius  $\ell$ , for  $0 \leq \ell \leq 1$ . The angular curves are co-centric circles with the same center  $u$ . If two medial edges  $e_1, e_2$  share a medial vertex  $u$  and  $C_1, C_2$  are two cells with  $e_1, e_2$  on the boundary, respectively, then the  $h$ -latitude curve in  $C_1$  can be connected to the  $h$ -latitude curve in  $C_2$  by the  $h$ -angular curve of the medial vertex  $u$ .

### 3.4 Routing scheme

Routing is performed by using only the names of the source  $p$  and the destination  $q$  and the medial axis  $A$  of the domain  $R$ . Suppose  $N(p) = (x(p), y(p), d(p)), N(q) = (x(q), y(q), d(q))$ , where  $x(p), x(q)$  are two points on the medial axis  $A$ . We first find the shortest path  $S_A(x(p), x(q))$  between  $x(p)$  and  $x(q)$  on  $A$ , which is denoted as the *reference path*. The length of the shortest path  $d_A(x(p), x(q))$  is denoted as the *reference distance*. An intuitive path between  $p, q$  is to route from  $p$  to  $x(p)$  on the medial axis  $A$ ,

follow  $S_A(x(p), x(q))$  until  $x(q)$  is reached, and route from  $x(q)$  to  $q$ . However, such paths use the medial axis as a routing ‘backbone’ and  $A$  will be heavily loaded. In our routing scheme, the routing path  $P(p, q)$  is a lifted up version of  $S_A(x(p), x(q))$ . Our routing scheme has two steps. First we route in parallel to  $S_A(x(p), x(q))$  until we reach a point  $t$  whose medial point is  $x(q)$ . The routing path produced is denoted as  $P_1(p, q)$ . Secondly, we follow the chords of  $x(q)$  or the angular curves of  $x(q)$  to reach the destination  $q$ . The routing path produced by the second step is denoted as  $P_2(p, q)$ . Figure 2 (iv) shows an example. In the following we present the two steps separately.

### 3.4.1 Routing in parallel to the medial axis

The routing path parallel to the reference path  $S_A(x(p), x(q))$  is realized by local gradient descent routing.  $S_A(x(p), x(q))$  is partitioned by medial vertices  $x_i$  to medial edges  $x_i x_{i-1}$  (the first segment and the last segment may be only part of medial edges),  $x_0 = x(p)$ ,  $x_k = x(q)$ .

Suppose that  $p$  is inside a cell  $C_0$ , with  $x_0 x_1$  on the boundary of  $C_0$ , the routing path follows the  $d(p)$ -latitude curve in  $C_0$  until it hits a chord on the boundary of  $C_0$ . Now we take a cell  $C_1$  with  $x_1 x_2$  on the boundary.  $C_0$  and  $C_1$  share at least a medial vertex  $x_1$ . If they share a chord  $x_1 y_1$ , then the  $d(p)$ -latitude curve in  $C_0$  is automatically connected to the  $d(p)$ -latitude curve in  $C_1$ . Otherwise, we use the  $d(p)$ -angular curve in the medial ball of  $x_1$  to transit from cell  $C_0$  to  $C_1$ . The  $d(p)$ -latitude curve in  $C_0$  is connected to the  $d(p)$ -latitude curve in  $C_1$  by the  $d(p)$ -angular curve of the medial vertex  $x_1$ . This procedure is continued until a node  $t$  with corresponding medial point  $x(q)$  is reached.

### 3.4.2 Routing on chords

The second part of the routing process starts with a node  $t$  that has the same medial point as the destination  $q$  and the same height as the source  $p$ , namely,  $x(t) = x(q)$ ,  $d(t) = d(p)$ . If  $x(q)$  is not a medial vertex, then  $x(q)$  has only two chords. Thus either  $t$  and  $q$  are on the same chord or they are on different chords. In the first case, routing along the  $x(q)$ -longitude curve inside the cell containing  $q$  will reach the destination  $q$ . In the second case,  $q$  and  $t$  are on different cells with  $x(q)$  on the boundary. We route first along the  $x(q)$ -longitude curve to the node  $x(q)$ , then follow the second chord to reach  $q$ . If  $x(q)$  is a medial vertex, then there is a local polar coordinate system around  $x(q)$ , thus we route from  $t$  along the chord through  $t$  to a node  $t'$  with height  $d(q)$ . Then we route from  $t'$  to  $q$  along the  $d(q)$ -angular curve in the local polar coordinate system of  $x(q)$ .

In summary, the naming system provides a natural road system on which routing can be done efficiently in a local-

ized manner. The routing scheme guarantees delivery for any source and destination pair.

## 4 Medial axis-based naming and routing in a discrete sensor field

The previous section explains the basic idea of using medial axis to build a naming and routing scheme in a continuous geometric region. All the concepts are clear and well-defined in the continuous case. For any two points, there is a natural route indicated by the medial axis between them. The adaptation of the concepts to a discrete sensor field, however, requires non-trivial re-design of the protocol due to the following reasons. First, sensors do not have geographical locations. The shape of a sensor field is not known and is only approximated by the connectivity of the sensor network. Also the proximity of two sensors is only approximated by the number of hops in the shortest path in the (unweighted) communication graph. Since the hop count is always an integer, the approximate distance has non-neglectable rounding errors. This brings more troubles to overcome since the exact medial axis is very sensitive to noises on boundaries—a small bump on the boundary may create a long branch in the medial axis [3].

In this section, we present a protocol that overcomes all these difficulties and constructs a robust medial axis and a routing algorithm for sensor networks, which requires no geographical locations and depends only on the connectivity graph of the sensor field. Our protocol computes in a preprocessing step a medial axis, and assigns names to all the sensors with respect to the medial axis. At the end of the preprocessing, a compact representation of the medial axis, called the *medial axis graph (MAG)*, whose size is proportion to the complexity of the geometric features of the sensor field, is stored at each node in the network. With the medial axis based naming infrastructure, routing is performed in two stages. In a global planning stage, the shortest path between the medial points of the source and the destination in MAG is found. The actual routing path is realized by local gradient descent routing either by routing in parallel to the medial edges on the globally planned path or on chords to the destination.

### 4.1 Construction of medial axis

The computation of the exact medial axis is only known in principle for semi-algebraic sets,<sup>3</sup> yet algebraic difficulties prevent efficient implementations even for shapes bounded

<sup>3</sup> Each element in a semi-algebraic set is the solution of a finite system of algebraic equations and inequalities.



by simple curves in the plane [3]. In practice, an approximation to the medial axis is computed by using the Voronoi diagram of a dense sampling on the boundary of a shape. To be precise, an  $\varepsilon$ -sample  $X$  of the boundary of a shape  $R$  is a finite set of points on the boundary  $\partial R$  such that each point on  $\partial R$  is less than distance  $\varepsilon$  away from a point in the  $\varepsilon$ -sample. For a shape in  $\mathbb{R}^2$ , the Voronoi edges and vertices of an  $\varepsilon$ -sample  $X$  that are completely inside the shape converge to the medial axis, when  $\varepsilon \rightarrow 0$  [7]. The construction of an approximate medial axis in a sensor network uses similar ideas. We first give a general outline of the protocol and then explain each step in detail. The Medial Axis Construction Protocol (MACP) runs as follows:

1. Detect boundaries of a sensor field;
2. Construct the medial axis graph, and broadcast it to every node in the network;
3. Name each node by local computation.

#### 4.1.1 Detect boundaries of a sensor field

The construction of the medial axis requires a sampling of nodes on the boundaries of the sensor field, including the outer boundary and the boundaries of interior holes. Each sample node knows to which hole boundary or outer boundary it belongs. These can be realized in different ways, either by manual identification of boundary nodes during deployment, or by automatic detection of holes. In particular, there are ways to detect some samples of sensors on the boundaries of holes by only the connectivity of the network. If the sensors are deployed uniformly densely in a field with large holes, sensors on the boundaries of holes usually have much smaller sensor density and can be detected as such [13]. Recently a topological method has been proposed to mark nodes on the boundaries of holes by detecting the breakage of wave propagation contours [15].

Given a reasonably dense sampling of boundary nodes, there are a number of techniques to detect the complete boundary from these samples. This is essentially the curve construction problem, namely, connecting the sampling nodes to a meaningful boundary. One way to tackle this problem is to apply the crust algorithm [1, 2] to discrete networks. In practice, a simple and effective algorithm is to ask the close-by sample nodes to discover each other through local flooding, and include all the nodes on the shortest paths between them as new boundary nodes. Such local discovery can be executed by different nodes simultaneously. The boundary nodes in Fig. 1 (iii) were detected in this way from the samples as shown in Fig. 1 (ii).

#### 4.1.2 Construct the medial axis graph

For a communication network represented by an unweighted graph  $G = (V, E)$  and a subset  $S \subseteq V$  on the boundaries of the sensor network, we define a node to be on the medial axis if it has equal hop counts to two closest boundary nodes. This node is called a *medial node*. The medial axis in the discrete case is defined as the subgraph  $G_M \subseteq G$  spanned by the medial nodes  $M$ .

The medial nodes defined above are noisy. It is known that in the continuous case the medial axis is sensitive to noises on boundaries [3]. In particular, a small bump on the boundary can incur a long branch in the medial axis. This instability is more severe for a discrete network. For example, all the nodes that are one-hop away from two boundary nodes are medial nodes by the definition, which are clearly undesirable. Similarly, a node that has equal distance to two close-by nodes on the same boundary is also unwanted. So we want to eliminate the unstable branches on the medial axis and keep those that correspond to genuine geometric features. In particular, we disregard those unstable medial nodes whose closest boundary nodes are on the same boundary and are within a small distance.

Medial nodes can be identified through local flooding. Specifically, each boundary node initiates the flooding of a message which contains its ID and the boundary it belongs to, as well as a counter that records how many hops the message has travelled. If a node receives a packet from a boundary node that is further away from its current nearest boundary node(s), it stops forwarding this packet. If the boundary nodes initiate their flooding at approximately the same time, and each packet travels at approximately the same speed, then a packet is dropped once it reaches the medial axis. This cuts down the total number of packets delivered and keeps the total communication cost low. As a result, each node learns its nearest boundary node(s) and can determine if it itself is a medial node or not. Due to integral rounding error, it is possible that the middle point of an edge has equal distance to two or more boundary nodes. In this case, we arbitrarily round it to an endpoint of the edge. Unstable medial nodes are ignored. Figure 1 (iv) shows a set of medial nodes. In the end, we have a set of medial nodes, which we classify into two types I and II. A medial node  $u$  of type I has equal distances (in hops) to two or more nodes on *different* boundaries. We call these boundaries the closest boundaries of  $u$ . The rest are of type II.

An important step in the construction of the medial axis is to guarantee that the medial nodes are connected in the correct way. For the continuous case, the medial axis inside the geometric region is connected with each minimal cycle surrounding exactly one obstacle. Our construction is based on this observation and that the discrete sensor field is a reasonably good approximation of the underlying shape.

For two boundaries  $i$  and  $j$ , if there exists a set of medial nodes of type I whose closest boundaries include  $i, j$ , we connect them into a short path  $P_{ij}$  and include all the nodes on this path in the medial axis. For any boundary  $i$ , we connect the paths  $P_{ij}$ , for all  $j$ , into a cycle. Two endpoints of different paths are called adjacent if they are connected on some cycle. For  $k$  endpoints  $a_1, a_2, \dots, a_k$ ,  $k \geq 3$ , respectively on different paths  $P_1, P_2, \dots, P_k$ , if  $a_i$  and  $a_{i+1}$  are adjacent, for  $i = 1, \dots, k - 1$ , then we use a star-like tree to connect them. We denote the root of the tree as a medial vertex of the medial axis graph. All of these operations can be implemented by a very small flooding in a local neighborhood in a progressive manner. Nearby medial nodes connect themselves into paths. Nearby paths connect themselves into cycles or by a tree rooted at medial vertices. Similarly, we use limited flooding to connect all type-II medial nodes into paths. These paths and the type-I cycles are connected by star-like trees into one connected piece. Finally we trim away short branches which may be caused by noise on the boundaries and are of no help for routing. Now the medial axis is composed of a number of cycles and paths nicely glued together. All the medial nodes with degree one or at least three are medial vertices. As an example, Fig. 1 (v) shows the medial axis after the medial nodes, as shown in (iv), are connected. At this point, we call all the nodes on the medial axis medial nodes, with a little abusing of the notation.

The medial axis constructed this way includes all the medial nodes and a set of edges that connect them in a meaningful way. It captures the geometric shape and the topological properties of a sensor network. We use the medial axis graph (MAG) as a combinatorial representation of the medial axis, which is made available to every sensor in the network. The set of vertices of MAG is the set of medial vertices. Each path between two medial vertices in the medial axis corresponds to an edge in MAG. The edges in MAG are weighted by the number of hops of the corresponding paths in the medial axis. The size of MAG is only proportional to the number of large geometric and topological features of the sensor field. For example, the MAG for the scenario in Fig. 1 (v) has only two vertices and two edges (one of them is a self-loop). The MAG for the scenario in Fig. 9 has only 17 vertices and 26 edges. Thus we can afford to let every sensor keep such a compact graph.

After the construction of the medial axis, we let a node flood the network, pull the information about the medial axis, and construct the abstracted medial axis graph. This graph is then broadcast to every sensor. In addition, each node in a medial edge remembers which medial edge it is in, its neighboring nodes in that medial edge, and how many hops it is from each endpoint of the medial edge.

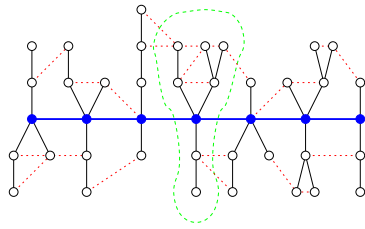
### 4.1.3 Cartesian coordinate systems

The medial axis of a sensor network is used as a reference to name every sensor node. In a continuous domain, a node is named by which chord it stays on. In a discrete sensor field, we use shortest path trees rooted at medial nodes to approximate chords. A node is named by which shortest path tree it stays on. To build the shortest path tree rooted at the medial axis, we start from the medial axis and progressively compute the closest medial node for each sensor. Every medial edge separates two canonical cells. So for a medial node  $v$  in the interior of a medial edge,  $v$  should have at most two shortest path trees rooted at itself, one on each side of the medial edge. Recall that each node knows its closest boundary and can decide which side of the medial edge it stays on<sup>4</sup>. Therefore, for each child  $u$  of  $v$  with a shortest path subtree  $T(u)$ , we perform a majority vote to assign  $T(u)$  to the side on which most of the nodes in  $T(u)$  agree. A medial vertex may have more than two shortest-path trees rooted on it corresponding to multiple canonical cells. Here we consider the pairs of trees corresponding to two canonical cells sharing the same medial edge. The nodes in the shortest path tree on one side of the medial edge are assigned positive height values. The nodes on the other side of the medial edge are assigned negative height values.

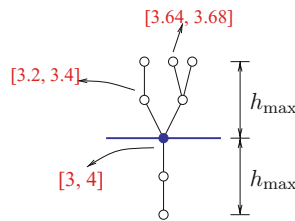
Each node is assigned a name by its relative position to the medial axis. Basically each node  $v$  is assigned an  $x$ -range  $[\ell(v), k(v)]$  and a height  $h(v)$ , where the  $x$ -range specifies to which part of the medial axis  $v$  corresponds, and the height specifies how far  $v$  is from the medial axis. In the medial axis graph, a medial edge actually corresponds to a medial axis path in the sensor network. See Fig. 5 as an example. Suppose the medial path has in total  $m$  nodes. Then the  $j$ -th node on the path is assigned an  $x$ -range  $[j - 1, j]$  and a height value 0.

Now we describe how to name the nodes in the same shortest path tree. The assignment of names starts from the root, a node on the medial axis, and propagates down its shortest path tree. See Fig. 6 as an example. Suppose a node  $v$  is given a name with an  $x$ -range  $[\ell(v), k(v)]$  and a height  $h(v)$ . Suppose  $v$  has  $c$  children. Then we uniformly partition the  $x$ -range of  $v$  into  $2c + 1$  subintervals and assign one even interval to each child. See Fig. 6. Notice that the collection of  $x$ -ranges of  $v$ 's children does not fully cover the  $x$ -range

<sup>4</sup> This is simple for a medial edge that has two different closest boundaries. For a medial edge whose closest boundaries belong to the same hole boundary, either the corresponding boundary nodes are in two connected pieces or in one connected piece. In the latter case, the medial edge has a medial vertex of degree one as its endpoint; we use any boundary node closest to that medial vertex to naturally partition the boundary nodes into two connected pieces. Then we treat the two pieces as two different boundaries.



**Fig. 5** Assign each node a name with respect to the medial axis. In this figure, the medial axis is drawn in thick blue horizontal line. Nodes on the medial axis are represented by solid circle. Black solid edges represent the balanced shortest path trees rooted at nodes on medial axis. The communication edges that are not tree edges are drawn in dotted line segments. The part bounded by the dashed curve are two shortest path trees rooted at a medial node, one on each side of the medial edge



**Fig. 6.** Assign each node a name with respect to the medial axis

of  $v$ . The purpose of these gaps is for easy node insertion and deletion, as we will describe in Section 4.3. Suppose the height of the shortest path tree is  $z$ . It is stretched to  $h_{max}$  in the naming system. ( $h_{max}$  is a parameter that can be adjusted.) Thus a node  $w$  that is  $i$  hops away from its root (a medial node) is assigned a height value of  $h(w)$  with  $|h(w)| = h_{max} \cdot i/z$ . The sign of  $h(w)$  depends on which side of the medial edge the shortest path tree containing  $w$  stays.

#### 4.1.4 Polar coordinate systems

Nodes around medial vertices are also assigned polar coordinates. Corresponding to each medial vertex of degree three or more in MAG, we construct a polar coordinate system. First of all, each medial vertex  $v$  with degree at least three is identified by several nearest boundary nodes. Such an identification is readily available in the process of constructing the medial axis graph. Suppose these boundary nodes are  $r$  hops away on average from  $v$ . The nodes within  $r$  hops from  $v$  are to be assigned polar coordinates, in addition to their original cartesian coordinates. The medial vertex  $v$  floods the  $r$ -hop neighborhood and constructs a shortest path tree rooted at  $v$ , denoted by  $T(v)$ . We assign polar coordinates to the nodes in  $T(v)$  in a way very similar to what we explained before. Specifically, the root  $v$  is given a polar coordinate with an angular range  $[0, 2\pi)$  and a radial coordinate 0. For a node  $u$ , its angular range is divided into  $2c + 1$  small intervals if  $u$  has  $c$  children in  $T(v)$ . Each child is assigned one interval.

The radial coordinate of a node is its hop count distance to the medial vertex  $v$ .

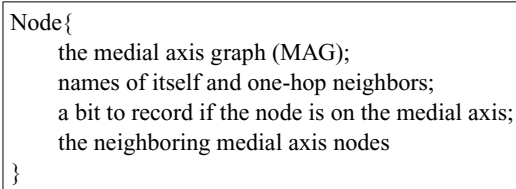
One non-trivial challenge in the assignment of polar coordinates is to align the branches in the shortest path tree  $T(v)$  in a way that is consistent with their true alignment in the physical network. The difficulty is that we do not know the orientations of the nodes in  $T(v)$  relative to the medial vertex  $v$ . For this purpose we identify all the maximal nodes in  $T(v)$  that do not have any neighbor in the sensor network with higher hop count distance from  $v$ . That set of nodes, which we denote by  $L(v)$ , naturally includes all the hop  $r$  nodes from  $v$ . Let the nodes in  $L(v)$  locally discover each other and connect themselves into a cycle. We start from the cycle to establish a good ordering of the nodes around  $v$ . Specifically, we first assign to each node  $w$  in the cycle an angle  $a(w)$  based on the circular ordering of those nodes in the cycle, and also assign  $w$  a weight that equals 1. Denote by  $N_k(v)$  the set of nodes that are  $k$  hops away from  $v$ . We use the ordering of the nodes in  $N_k(v)$  as an reference to establish an ordering of those nodes in  $N_{k-1}(v)$ , for  $k = 2, \dots, r$ . The ordering of nodes in  $N_r(v)$  naturally comes from the ordering in the cycle  $L(v)$ . A node  $w$  in  $N_{r-1}(v)$  is assigned an angle that is the weighted average<sup>5</sup> of the angles of the nodes in  $N_r(v)$  that are neighbors of  $w$  in the sensor network, if such neighboring nodes exist. Then a weight is assigned to  $w$  which is the sum of the weights of those neighbors in  $N_r(v)$ . Then for nodes in  $N_{r-2}(v)$ ,  $N_{r-3}(v)$ ,  $\dots$ ,  $N_1(v)$ , they are assigned angles and weights in a similar way. Essentially we propagate the ordering information from  $L(v)$  inwards. In the end, all the nodes in  $T(v)$  are assigned an angle each. Then we assign the angle ranges of the polar coordinates to nodes, starting with the root of  $T(v)$  and propagate outwards. Specifically, the nodes in  $N_1(v)$  are assigned angle ranges in a way that is consistent with the ordering of their angle values obtained above. Then the children of a node  $w \in N_1(v)$  are assigned intervals of  $w$ 's angle range in a way consistent with the ordering of their angle values, and so on.

To summarize, by the Medial Axis Construction Protocol, we construct a medial axis and its compact representation MAG, together with a shortest path forest rooted at medial axis nodes and some trees for polar coordinate systems. Each node in the sensor network only stores a small amount of information, namely, information about its one-hop neighbors and a small size medial axis graph, as shown in Fig. 7.

#### 4.2 Medial axis based routing

Our MAP routing scheme uses the names of the source and the destination. A source node can obtain the name of the destination node in several ways. If the source knows the

<sup>5</sup> We note here that the weighted average is taken modula  $2\pi$ .



**Fig. 7.** Information stored at a node

ID of the destination, we can use a scalable location service (such as [25]) to provide the ID-name correspondence for each sensor. A source node can also specify the name of the destination directly, since our naming scheme has an intuitive meaning. For applications such as content-based distributed hash table, sensors can hash data to a reservoir for a query to fetch. In this case, the source gets the name of the destination from the output of a hash function.

With the medial axis based naming infrastructure, the Medial Axis based Routing Protocol (MARP) runs as follows:

1. In the global planning step, find the shortest path  $S_A(x(p), x(q))$  in the medial axis graph  $A$  for the medial points of source  $p$  and destination  $q$ ;
2. Route in parallel to  $S_A(x(p), x(q))$  until a node with the same medial point as the destination  $q$  is reached;
3. Route in the shortest path tree rooted at that medial point to reach the destination  $q$ .

The global planning step is performed at the source node by a shortest path algorithm on MAG. In the following we will focus on the medial-axis based greedy routing in the sensor network. The basic idea is the same as in the continuous case. The reference path on the medial axis,  $S_A(x(p), x(q))$ , is partitioned into segments by medial vertices. Suppose we are currently at a node  $v$  in a shortest path tree whose root is in a medial edge  $x_i x_{i+1}$ ,  $1 \leq i \leq k-1$ , with  $x_i$  as a medial vertex,  $x_1 = x(p)$ ,  $x_k = x(q)$ . We hope to route in parallel with  $x_i x_{i+1}$ . For this purpose, we set a temporary routing goal as the node with height value  $h(p)$  in a shortest path tree rooted at  $x_{i+1}$ . We route toward the temporary routing goal in parallel with the medial edge  $x_i x_{i+1}$ ; when we reach the tree rooted at  $x_{i+1}$  that corresponds to the polar coordinate system (we actually don't have to reach the temporary routing goal), we route in the polar coordinate system of  $x_{i+1}$  to get to a node corresponding to the *next medial edge*  $x_{i+1} x_{i+2}$ , and update the temporary routing goal accordingly. Finally, when we reach a node whose shortest path tree has the same root as the destination  $q$  does, we can route easily in those shortest path trees to get to  $q$ .

Now we focus on how to route greedily towards the temporary goal in parallel with the medial edge  $x_i x_{i+1}$ . Suppose we are currently at a node  $v$ . We pick  $w$ , a neighbor of  $v$ , that is 'closer' to the temporary routing goal than  $v$  is as the next hop. The 'closeness' is defined in a way similar to the Eu-

clidean distance. We say that  $w$  is 'closer' to the temporary routing goal than  $v$  is if the following condition is satisfied:

- $(k(w) - \ell(x_{i+1}))^2 + (h(w) - h(p))^2 < (k(v) - \ell(x_{i+1}))^2 + (h(v) - h(p))^2$  if the  $x$ -range of  $v$ ,  $[\ell(v), k(v)]$  is smaller than that of  $x_{i+1}$ ,  $[\ell(x_{i+1}), k(x_{i+1})]$ ;
- $(\ell(w) - k(x_{i+1}))^2 + (h(w) - h(p))^2 < (\ell(v) - k(x_{i+1}))^2 + (h(v) - h(p))^2$  otherwise.

To avoid passing the same node twice, we always follow three rules:

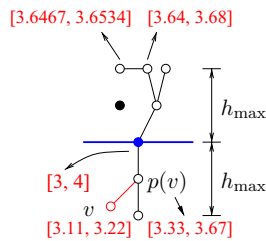
1. Never route to a descendant (to avoid going up and down in the same tree);
2. Never route to the other side of the medial edge (specifically,  $h(v)$  and  $h(w)$  should have the same sign);
3. Never route backward with respect to the medial edge (specifically, we should have  $k(w) \geq k(v)$  if the  $x$ -range of  $v$  is smaller than that of  $x_{i+1}$ , and  $\ell(w) \leq \ell(v)$  otherwise).

If no such neighbor  $w$  exists,—namely, when no greedy progress can be made,—we route to the parent of  $v$ . This routing method guarantees that we can reach a shortest path tree rooted at  $x_{i+1}$ , because at least in the medial edge we can always make greedy progress.

The above routing method is for routing inside a canonical cell under the Cartesian coordinates of  $x$  and height  $h$ . When a packet enters the polar coordinate range of a medial vertex, it is routed in the polar coordinate system. It remembers the angle where it starts and tries to route circularly around the medial vertex searching for a node in the next canonical cell. Again there are several basic rules. First a packet tries to keep the same hop distance to the medial vertex. It never routes to a node with a higher hop-distance value. If a packet cycles back and passes the angle where it started, it is forwarded 1-hop towards the medial vertex. In the worst case, the packet is delivered to the medial vertex and then forwarded to the correct canonical cell.

A small modification to the algorithm can substantially improve the performance of MAP. The idea is to keep a very small routing table, e.g., within 4 hops, at each node. This is because that the virtual coordinates generated by MAP may have a small mismatch with the real node positions and the connectivity in the communication graph. By using a small routing table, a node has more possibility to proceed greedily towards its (temporary) goal. That will not only shorten routing paths but also improve load balancing, because now there is less chance to routing toward the medial axis.

In summary, by the MAP routing protocol a source performs a global path planning in the very compact medial axis graph, and local greedy routing hop-by-hop guided by the globally planned path on MAG. Delivery is guaranteed since a packet never gets stuck.



**Fig. 8** If a new node  $v$  (in red) is added to the sensor network, we connect it to the shortest path tree of a node on the medial axis and assign it a name. If a node (in solid black) is deleted from the sensor network, its descendants find links to other parts of the tree and are assigned new names

### 4.3 Network dynamics

In a sensor network, links may come and go, nodes may be inserted and deleted. Our MAP naming and routing scheme can efficiently handle dynamics such that only local changes are necessary. The medial axis and the shortest path forest rooted on the medial axis are repaired. If a node also changes its name, it informs the location service of such changes.

Figure 8 illustrates how these changes are handled. Namely, when a new node  $v$  is inserted,  $v$  performs a one-hop broadcasting to find its one-hop neighbors. Then  $v$  is connected to the shortest path forest of the medial axis node by choosing its parent as its one-hop neighbor with smallest hop distance to the medial axis. Then  $v$  finds a gap on the  $x$ -coordinate range of its parent  $p(v)$ . In Fig. 8,  $v$  finds a gap  $[3, 3.33]$ , partitions the gap equally into three sub-intervals and stays on one of them, say  $[3.11, 3.22]$ . The  $x$ -coordinate ranges of all the other children of  $p(v)$  are not affected. If  $p(v)$  is of  $k$  hops away from the medial axis,  $v$  is of  $k + 1$  hops away. The height value of  $v$  is assigned according to its parent  $p(v)$ . Namely,  $h(v) = \frac{k+1}{k} \cdot h(p(v))$ .

When a node  $v$  is deleted from the network, we do nothing if  $v$  is a leaf on the shortest path tree to the medial axis. If  $v$  has descendants, then we connect their descendants to the shortest path trees through other nodes, in the same way as the node insertion scenario. If  $v$  also stays on the medial axis, its neighbors on the medial axis initiate a local flooding to find each other and include all the nodes that stay on their shortest paths in the medial axis. Then new names are assigned to these nodes involved in the update.

If a link between two nodes appears, nothing is changed. If a link between two nodes disappears, we do nothing as long as the link is neither a link between a node  $u$  and its parent on  $u$ 's shortest path tree, nor a link on the medial axis. If a link on the medial axis disappears, two medial nodes connect themselves by their shortest path and includes the nodes on this path to the medial axis. If a link between a node  $u$  and its parent disappears, then we may need to assign a new name to  $u$ . Similarly  $u$  finds among its one-hop neighbors the one with

smallest hop distance to the medial axis as its new parent.  $u$ 's name is updated according to its new parent.  $u$ 's descendants, if any, update their name accordingly.

We also note that the updates to the virtual coordinates are only performed upon permanent changes or when necessary. If the nodes or links go down temporarily, we do not update their coordinates. Instead, due to the greedy flavor of the routing algorithm, the packet will be delivered as long as there is one neighbor in the favorable direction. Thus the routing algorithm is robust to random temporary link failures.

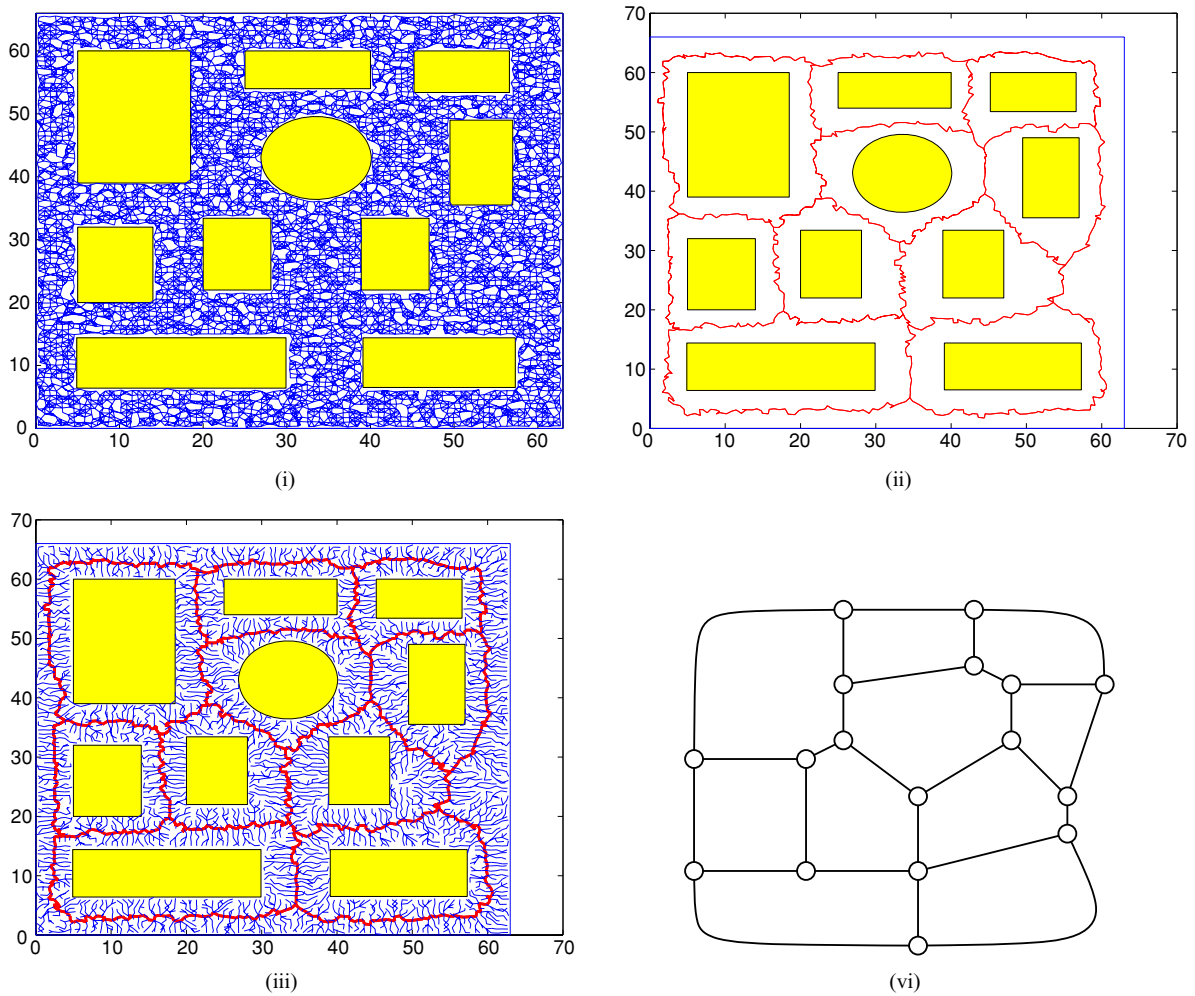
## 5 Simulation

We have implemented the MAP protocol and conducted extensive simulation for various types of environments. In this section we introduce two experiments for sensor networks deployed in a university campus and airport terminals. The university campus is a typical outdoor environment, as shown in Fig. 9, while airport terminals are representative of indoor scenarios, as shown in Fig. 10. Both of them have complex geometry and/or non-trivial topology.

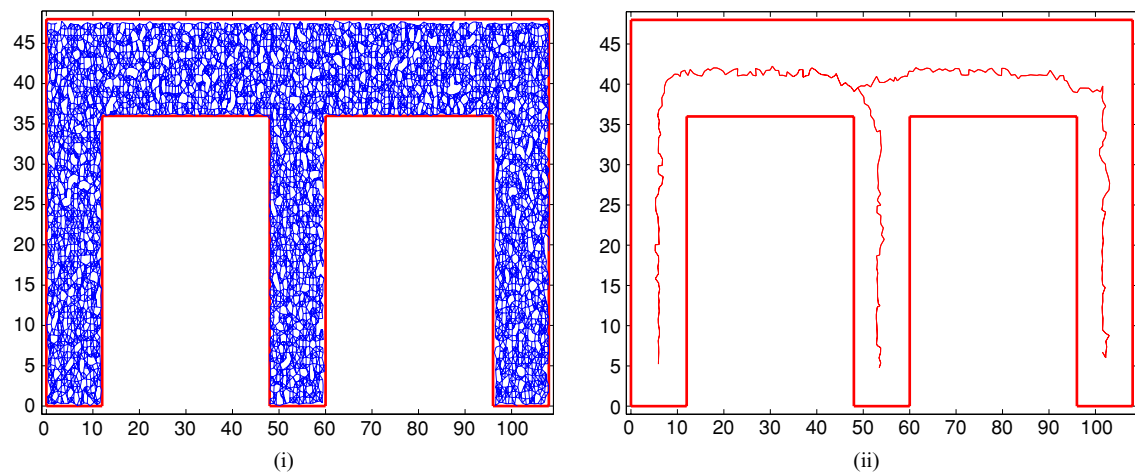
The university campus has a 620 m by 650 m rectangle as an outer boundary and 10 buildings inside. We deployed  $n$  ( $n = 5735$  in Fig. 9) sensors, each of which has a coverage radius of 10 meters. The sensors were deployed with a grid model with perturbation. Each sensor deviates from its grid position with a normal distribution with a standard deviation of  $\sigma = 2$  meters. We use the unit disk graph on the sensors as the graph model. We note that the communication graphs used in the experiments are actually quite sparse. The average degree of the communication graph is only 5.4067 in Fig. 9.

The airport terminals have a maximum  $x$ -span of 1080 meters and a  $y$ -span of 480 meters. It consists of three terminals that are connected by a corridor. We deployed  $n$  ( $n = 5204$  in Fig. 10) sensors, each of which has a coverage radius of 10 meters. Again the sensors were deployed with the same grid model with perturbation as before. The average degree of the communication graph is 5.4502 in Fig. 10.

In these experiments, our focus is to verify the validity of our ideas. We focus on the topological level of the MAP protocol and study its routing performance, load balancing as well as its robustness to network models. Many important issues on MAC layer, such as channel fading and acknowledgement packets, have not yet been considered in the current simulation and will be addressed in future work. We compared MAP with GPSR [20], a widely known geographic location-based routing protocol. GPSR makes routing decisions based on the geographical location information. It has two routing modes. In a greedy mode, a node routes the message to a one-hop neighbor whose geographical location (2D Euclidean coordinates in this case) is closest to the destination. If a node has no neighbor closer to the destination, it



**Fig. 9** A scenario of a university campus. Each sensor has a normalized coverage radius of 1. (i) A sensor network of 5735 nodes deployed on a campus; (ii) The medial axis; (iii) The shortest path forest rooted on the medial axis; (iv) The medial axis graph (MAG) stored at each sensor



**Fig. 10** A scenario of airport terminals. Each sensor has a normalized coverage radius of 1. (i) A sensor network of 5204 nodes; (ii) The medial axis

enters the perimeter routing mode. The packet is then routed by the ‘right-hand rule’ along the face of a planar subgraph such as Gabriel Graph (GG), Relative Neighborhood Graph (RNG) until it reaches a node where greedy routing can be performed again. In our simulation, we use the restricted Delaunay Graph (RDG) [18] as the underlying planar subgraph. A RDG is a graph spanner, i.e., the shortest path in RDG is at most a small constant factor times as long as that in the original communication graph. However, neither GG nor RNG in the original GPSR protocol is a graph spanner. It has been shown that GPSR based on RDG produces shorter paths than GPSR based on other non-spanner graphs such as GG or RNG, especially in the perimeter mode [18].

### 5.1 Quality of routing paths and load balancing

We measure the quality of routing paths in two ways, i.e., the number of hops and the total Euclidean length of the routing paths. We randomly pick 12000 source and destination pairs uniformly. Each experiment was performed 50 times. The results are shown in Fig. 11.

For the  $i$ th source and destination pair, we denote by  $h_i$  and  $H_i$  the numbers of hops in the routing paths produced by MAP and GPSR, respectively. We denote by  $\ell_i$  and  $L_i$  the Euclidean lengths of the routing paths produced by MAP and GPSR. In Fig. 11, the second column, *average ratio of hops*, is defined as  $\frac{1}{N} \sum_{i=1}^N h_i/H_i$ , where  $N = 50 \times 12000 = 6 \times 10^5$  is the total number of source/destination pairs. The third column is the standard deviation of  $h_i/H_i$  over the  $N$  pairs. Similarly, the *average ratio of lengths* is defined as  $\frac{1}{N} \sum_{i=1}^N \ell_i/L_i$ , and the standard deviation of the ratio of lengths is defined accordingly. The ratio of total hops and the ratio of total lengths are defined as  $\frac{\sum_i h_i}{\sum_i H_i}$  and  $\frac{\sum_i \ell_i}{\sum_i L_i}$ , respectively.

We have found that MAP generates routing paths of comparable lengths both in the number of hops and in the total

	routing on campus					
	ratio of hops		ratio of total hops	ratio of lengths		ratio of total lengths
	average	dev.		average	dev.	
$n = 5000$	1.011	0.386	0.874	1.040	0.398	0.904
$n = 5500$	1.014	0.412	0.880	1.049	0.424	0.919
$n = 6000$	1.045	0.411	0.912	1.085	0.437	0.963
$n = 6500$	1.078	0.430	0.935	1.122	0.440	0.996
$n = 7000$	1.087	0.432	0.948	1.143	0.441	1.012
	routing in airport terminals					
$n = 5000$	1.117	0.550	0.876	1.129	0.579	0.920
$n = 5500$	1.065	0.557	0.759	1.104	0.584	0.812
$n = 6000$	1.081	0.618	0.754	1.119	0.697	0.822
$n = 6500$	1.092	0.628	0.767	1.121	0.686	0.842
$n = 7000$	1.097	0.619	0.762	1.132	0.678	0.858

**Fig. 11** Performance of routing on a university campus and in airport terminals. Each experiment was performed 50 times. In each experiment, 12000 source and destination pairs were selected uniformly randomly

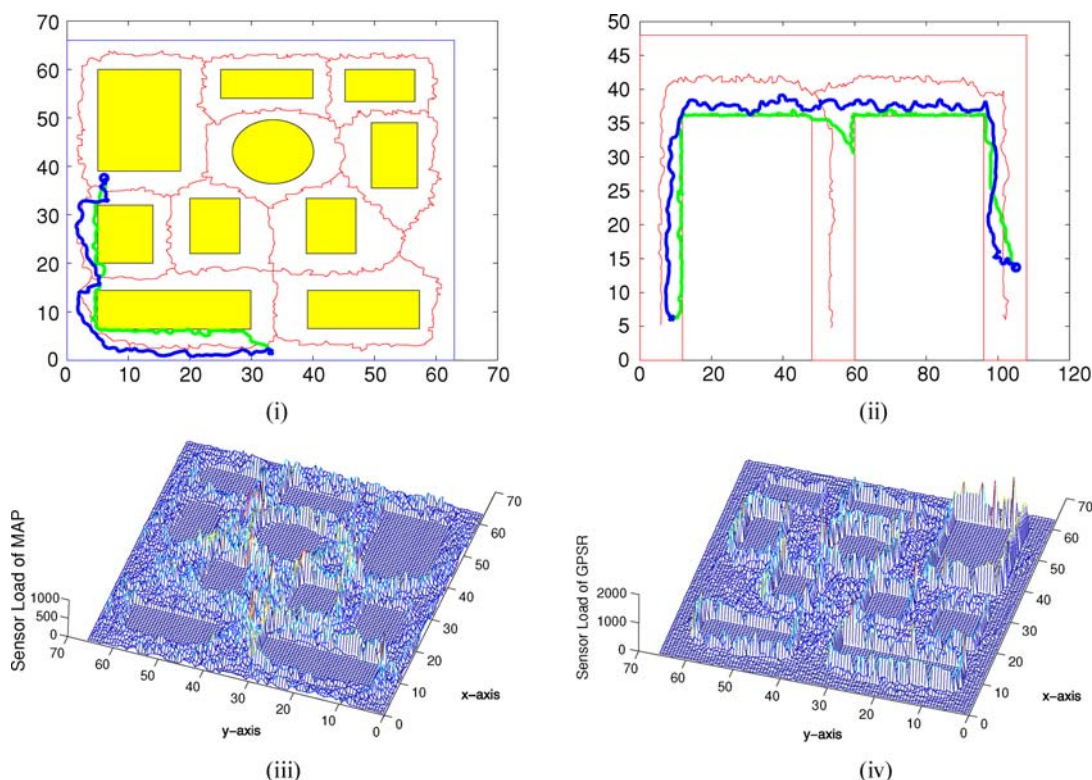
		$n = 5000$	5500	6000	6500	7000
Campus	MAP	1.522	1.787	1.855	1.954	1.096
	GPSR	2.524	2.850	2.940	3.025	3.104
Airport	MAP	2.856	3.102	3.405	3.729	3.819
	GPSR	3.773	4.307	4.658	4.817	4.972

**Fig. 12.** The normalized standard deviation of traffic load on sensors

Euclidean distance, as shown in Fig. 11. However, MAP achieves much better load balancing. In sensor networks, load balancing is a very important issue. Overloading a particular sensor node will drain its battery. Sharing of a common channel by many parties increases the possibility of conflict and the delay. We compared load balancing of MAP with GPSR. For the same family of source/destination pairs, we drew for each sensor  $v$  a bar with its height equal to the total number of routing paths though  $v$ . Figure 13 (iii) and (iv) show the histograms for MAP and GPSR respectively. It can be seen easily that the routing paths generated by GPSR concentrate heavily on the boundaries of holes, while sensors are more uniformly loaded by MAP. The reason is that geographical location-based routing schemes route greedily towards the destination based on Euclidean coordinates. As a result, routing paths tend to follow straight lines and can easily hit boundaries. When that happens, packets follow the boundary by ‘right-hand rule’ until greedy routing can be resumed. Such an operation heavily overloads nodes near boundaries. Comparatively, MAP captures the geometry of the environment and performs routing in parallel with the reference paths on the medial axis, thus it avoids the creation of heavily loaded area. Figure 13 (i) and (ii) show a couple of scenarios of the routing paths produced by MAP and GPSR. The normalized standard deviation, defined as the standard deviation divided by the mean value, of the load on individual nodes is shown in Figure 12, where MAP consistently has lower deviation values.

### 5.2 Robustness to network model

A commonly used model for wireless sensor networks is the unit disk graph model, where each node has a fixed communication radius. Two nodes can directly communicate if and only if they are within the communication radius. However, the unit disk graph model is a simplified model and is often far from the reality. The existence of communication links can significantly deviate from the unit distance constraint due to reasons such as multi-path fading. It is often observed that sensor nodes within a short communication range may not be able to communicate, while nodes with three or more times the distance can have stable links [17]. A routing scheme that heavily depends on the properties of unit disk graphs is thus not practical in reality. The MAP naming and routing scheme only depends on the connectivity graph. We show by



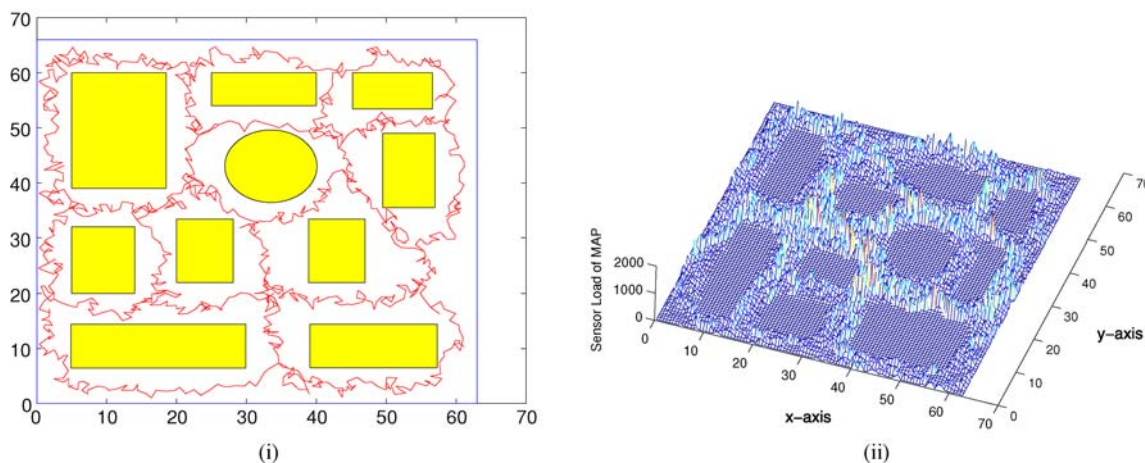
**Fig. 13** (i) and (ii) show the comparison of routing paths generated by MAP and GPSR between the same pair of source and destination nodes. Solid circle represents the source and hollow circle represents the destination. The darker path was generated by MAP and the lighter path was generated by GPSR. (iii) and (iv) show the comparison of load balancing of MAP and GPSR. 5500 sensors were deployed. (iii) is the load of MAP, (iv) is the load of GPSR

simulation that it is very robust to variations in the network model.

We have tested the robustness of MAP by using the Quasi-unit disk graph model [24]. Our Quasi-UDG model is characterized by a simple parameter  $\alpha$ . When two nodes are within distance  $1 - \alpha$ , a link between them always exists. If two nodes are more than  $1 + \alpha$  away, a link between them does

not exist. If the distance of two nodes is between  $1 - \alpha$  and  $1 + \alpha$ , a link between them exists with probability  $p$ . In order to keep the average node degree in the network approximately a constant, we choose  $p = (2 - \alpha)/4$ .

When  $\alpha$  is large, a quasi-UDG looks very different from a UDG at the local neighborhood level. But the medial axis still captures the correct topology. This can be seen from



**Fig. 14** MAP on the Quasi-UDG ( $\alpha = 0.8$ ) in a campus scenario. 5500 sensors were deployed. (i) The medial axis; (ii) A histogram of load by MAP on each node



	routing on campus			routing in airport terminals		
	ratio of total hops	ratio of total lengths	norm. dev. of load	ratio of total hops	ratio of total lengths	norm. dev. of load
$\alpha = 0.0$	1.000	1.000	1.787	1.000	1.000	3.102
$\alpha = 0.2$	1.224	1.316	1.608	1.238	1.367	1.927
$\alpha = 0.4$	1.109	1.376	1.602	1.092	1.411	2.024
$\alpha = 0.6$	1.045	1.415	1.630	0.948	1.376	2.086
$\alpha = 0.8$	0.918	1.428	1.687	0.950	1.469	2.108

**Fig. 15** Performance of MAP on sensor networks in a campus scenario under different graph models. 5500 sensors were deployed. Each experiment was performed 50 times. In each experiment, 12000 source and destination pairs were selected uniformly randomly

Fig. 14(i), where the medial axis in a Quasi-UDG with  $\alpha = 0.8$  is shown in the university campus scenario. We notice that the medial axis actually has self-crossings. However, we have found that MAP maintains stable performance on both the construction of medial axis and routing.

We also compared the routing performance under UDG and Quasi-UDG models in a number of scenarios. As a typical example, Fig. 15 shows the routing results with 5500 sensors deployed on a university campus. The *ratio of total hops (lengths)* is the ratio of the total number of hops (length) of the routing paths in the Quasi-UDGs to that in the UDGs (where  $\alpha = 0$ ). As before, we call the number of routing paths passing a sensor the load on the sensor. The fourth and the seventh columns of Fig. 15 are the normalized standard deviation of the load on sensors. Although the routing graphs cannot be the same for Quasi-UDG and UDG, they are both approximating the same underlying geometric domain. Note that in the Quasi-UDG model, the communication ranges of nodes are still small compared to the obstacles' sizes. We have found that the performance of routing is very stable for different  $\alpha$ , both in terms of routing path lengths and load balancing. It is shown by the results in Fig. 15. An example of the sensor load histogram for MAP with  $\alpha = 0.8$  is shown in 14 (ii).

## 6 Discussions

### 6.1 Routing on manifolds

The MAP naming and routing scheme can be extended to routing in other geometric spaces, in particular, a 2-dimensional manifold embedded in a 3-dimensional space. One such case is a sensor field deployed on an irregular terrain. One can define the medial axis of the manifold under the geodesic distance metric, which is a collection of continuous curves lying on the manifold. Any point can be given a name with respect to the medial axis in the same way as what MAP does. For a set of dense samples of nodes on such a manifold, the length of the shortest path between two nodes is a reasonably

good approximation of their geodesic distance [4]. Since we use the shortest path distance metric in the design of MAP, it is straightforward to extend the MAP naming and routing protocol to nodes on a manifold.

### 6.2 Geometric maps

One merit of MAP is that it requires no knowledge of the geographical locations of the nodes or the shape of the sensor field. In some cases, such as airport terminals or warehouses, the map of the sensor field is available and the construction of the medial axis can be simplified. When the geometric shape  $R$  of a sensor field is known, we can first construct the medial axis  $A$  of the boundary  $\partial R$ . This can be done by using standard techniques such as the crust algorithm [1]. Sensors near the geometric medial axis  $A$  can be marked by an examination of their proximity to  $A$ . The remaining part of MAP is the same.

### 6.3 Global load balancing

We have shown by simulation that the MAP routing protocol achieves very good load balancing. One reason is that on the lower level we route in parallel with the medial axis such that routes starting from different sources smoothly 'flow through'. Load balancing can be further improved at the abstract level. In the global planning stage, a routing path is selected by the shortest path routing algorithm on the medial axis graph (MAG). For a case where the sensor field has many 'corridors' with different widths, it is more desirable to route through a wide corridor than a narrow corridor. Therefore, we can attach a weight to each medial node which equals the maximum height (before normalization) of its shortest path tree. An edge on the medial axis graph has a total weight as the sum of weights of the medial nodes on the corresponding medial path. The weight of an edge can be understood as its 'capacity'. In this way the globally planned routing path can take into account the capacity of different edges such that the medial edge corresponding to a wide corridor is more preferable than the medial edge corresponding to a narrow one with the same length.

### 6.4 Location-free routing protocols

MAP belongs to the category of location-free routing protocols for sensor networks, where the routing rule is free of geographical information but rather based on virtual coordinates. We will give a quick overview of location-free protocols as well as a comparison of their design flavors.

Rao et al. [28] is the first paper on location-free protocols. They proposed an iterative scheme that embeds the connectivity graph in a 2D Euclidean plane such that the embedded virtual coordinates are used instead of the real geographical

locations for greedy geographical routing such as GPSR [20]. When packets get stuck at local minima, flooding is employed to deliver the packets. With a similar spirit, Bruck et al. [8] used local angle information to find a good embedding of the nodes in the 2D Euclidean plane. Further they showed that the local angle information is sufficient for finding a planar spanner of unit-disk graphs without the knowledge of geographical locations. In general such approaches compute a global embedding of sensor nodes in a 2-dimensional space. Finding such a global embedding is expensive in terms of both computation and communication, and will have large distortion if the sensors are actually deployed in a 3-dimensional space.

Fang et al. [12] took a different approach. Their scheme does not use any global embedding but rather takes an abstraction of the global topology of the sensor field. Such an abstraction, namely, a combinatorial graph on a subset of carefully selected landmarks, captures large topological features (such as where the holes are and how to get around them). The nodes are partitioned into tiles with respect to their closest landmarks. Each node is given a virtual coordinate, which is a function of the hop distances to only a subset of nearby landmarks. Both inter and intra-tile routing are performed in a greedy fashion under the virtual coordinate system. Using hop distances to landmarks to build virtual coordinates was also explored by Fonesca et al. [14] and Caruso et al. [9], where nodes are given global landmark-based virtual coordinates without the consideration of the topology of the sensor field.

MAP is similar to GLIDER in the way that MAP does not construct any global embedding either. The major difference is that MAP uses the medial axis of the sensor field, instead of the combinatorial Delaunay graph on landmarks, to represent the global topology. The main unsolved problem in GLIDER is a deeper understanding of the selection of landmarks and its effect on routing performance. The major drawback of MAP is that sensor fields are restricted to 2D manifolds. It is unclear how to extend the current scheme to sensors deployed in 3-dimensional space. It remains as interesting future work to conduct a thorough network-level comparison of all these location-free routing protocols [8, 9, 12, 14, 28] under various network topologies.

## References

1. N. Amenta, M. Bern and D. Eppstein, The crust and the  $\beta$ -Skeleton: combinatorial curve reconstruction, *Graphical Models and Image Processing* 60 (1998) 125–135.
2. N. Amenta, S. Choi and R. K. Kolluri, The power crust, unions of balls, and the medial axis transform, *Comput. Geom. Theory Appl.*, 19 (2001) 127–153.
3. D. Attali, J.-D. Boissonnat and H. Edelsbrunner, Stability and computation of the medial axis—a state-of-the-art report, in: *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. (Springer-Verlag, 2004).
4. M. Bernstein, V. de Silva, J. Langford and J. Tenenbaum, Graph approximations to geodesics on embedded manifolds, Technical report, Department of Psychology, Stanford University, 2000.
5. H. Blum. A transformation for extracting new descriptors of shape, in: W. Wathen-Dunn (ed.) *Models for the Perception of Speech and Visual Form*, MIT Press (1967) pp. 362–380.
6. P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in: *Proceedings of the 3rd Int. Workshop on Discrete Algorithms and methods for mobile computing and communications (DialM '99)* (1999) pp. 48–55.
7. J.W. Brandt, Convergence and continuity criteria for discrete approximations of the continuous planar skeletons, *CVGIP: Image Understanding* 59(1) (1994) 116–124.
8. J. Bruck, J. Gao and A. Jiang, Localization and routing in sensor networks by local angle information, in: *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05)* (May 2005) pp. 181–192.
9. A. Caruso, A. Urpi, S. Chessa and S. De, GPS Free coordinate assignment and routing in wireless sensor networks, in: *Proceedings of the 24th Conference of the IEEE Communication Society (INFOCOM)* (March 2005), Vol. 1, pp. 150–160.
10. H.I. Choi, S. W. Choi and H. P. Moon, Mathematical theory of medial axis transform, *Pacific Journal of Mathematics* 181(1) (1997) 57–88.
11. M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.
12. Q. Fang, J. Gao, L. Guibas, V. de Silva and L. Zhang, GLIDER: Gradient landmark-based distributed routing for sensor networks, in: *Proceedings of the 24th Conference of the IEEE Communication Society (INFOCOM)* (March 2005), Vol. 1 pp. 339–350.
13. S. P. Fekete, A. Kroeller, D. Pfisterer, S. Fischer and C. Buschmann, Neighborhood-based topology recognition in sensor networks, in: *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSOR)* (2004) pp. 123–136.
14. R. Fonesca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker and I. Stoica, Beacon vector routing: Scalable point-to-point routing in wireless sensor networks, in: *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, (May 2005).
15. S. Funke, Topological hole detection in wireless sensor networks and its applications, in: *DIALM-POMC '05: Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*, (New York, NY, USA, ACM Press, 2005) pp. 44–53.
16. D. Ganesan, D. Estrin, and J. Heidemann, DIMENSIONS: why do we need a new data handling architecture for sensor networks, in: *Proceedings ACM SIGCOMM Workshop on Hot Topics in Networks* 2002.
17. D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin and S. Wicker, Complex behavior at scale: An experimental study of low-power wireless sensor networks, Technical Report UCLA/CSD-TR 02-0013, UCLA, 2002.
18. J. Gao, L.J. Guibas, J. Hershberger, L. Zhang and A. Zhu, Geometric spanner for routing in mobile networks, in: *Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)* (2001) pp. 45–55.
19. L. Guibas, C. Holleman and L. E. Kavradi, A probabilistic roadmap planner for flexible objects with a workspace medial-axis based sampling approach, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Kyongju, Korea, IEEE Press, 1999) pp. 254–260.
20. B. Karp and H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: *Proceedings of the ACM/IEEE Interna-*

- tional Conference on Mobile Computing and Networking (MobiCom) (2000) pp. 243–254.
21. Y.-J. Kim, R. Govindan, B. Karp and S. Shenker, Geographic routing made practical, in: *Proceedings of the Second USENIX/ACM Symposium on Networked System Design and Implementation (NSDI 2005)*, May 2005.
  22. Y.-J. Kim, R. Govindan, B. Karp and S. Shenker, On the pitfalls of geographic face routing, in: *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, (New York, NY, USA, ACM Press, 2005) pp. 34–43.
  23. F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, Geometric ad-hoc routing: of theory and practice, in: *Proceedings 22nd ACM Int. Symposium on the Principles of Distributed Computing (PODC)* (2003) pp. 63–72.
  24. F. Kuhn and A. Zollinger, Ad-hoc networks beyond unit disk graphs, in: *Proceedings 2003 joint workshop on Foundations of mobile computing* (2003) pp. 69–78.
  25. J. Li, J. Jannotti, D. Decouto, D. Karger and R. Morris, A scalable location service for geographic ad-hoc routing, in: *Proceedings of 6th ACM/IEEE International Conference on Mobile Computing and Networking* (2000) pp. 120–130.
  26. A. Lieutier, Any open bounded subset of  $\mathbb{R}^n$  has the same homotopy type than its medial axis, in: *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, ACM Press (2003) pp. 65–75.
  27. M.A. Paskin and C.E. Guestrin, A robust architecture for distributed inference in sensor networks, Technical Report IRB-TR-03-039, Intel Research, 2004.
  28. A. Rao, C. Papadimitriou, S. Shenker and I. Stoica, Geographic routing without location information, in: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking* ACM Press (2003) pp. 96–108.
  29. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, GHT: A geographic hash table for data-centric storage in Sensornets, in: *Proceedings 1st ACM Workshop on Wireless Sensor Networks and Applications* (2002) pp. 78–87.
  30. K. Seada, A. Helmy and R. Govindan, On the effect of localization errors on geographic face routing in sensor networks, in: *IPSN'04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks* (ACM Press, 2004) pp. 71–80.
  31. D.J. Sheehy, C.G. Armstrong and D.J. Robinson, Shape description by medial surface construction, in: *IEEE Transactions on Visualization and Computer Graphics* (1996) vol. 2 No. 1, 62–72.
  32. F. Zhao, J. Shin and J. Reich, Information-driven dynamic sensor collaboration, *IEEE Signal Processing Magazine* 19(2) (2002) 61–72.



**Jehoshua Bruck** is the Gordon and Betty Moore Professor of Computation and Neural Systems and Electrical Engineering at the California

Institute of Technology. During 2003–2005 he served as the founding Director of Caltech's Information Science and Technology (IST) program.

He received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion, Israel Institute of Technology, in 1982 and 1985, respectively and the Ph.D. degree in Electrical Engineering from Stanford University in 1989.

His research combines work on the design of distributed information systems and the theoretical study of biological circuits and systems.

Dr. Bruck has an extensive industrial experience, including working with IBM Research for ten years where he participated in the design and implementation of the first IBM parallel computer. He was a co-founder and chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that focused on software products for management of network information systems.

He is an IEEE fellow, and his awards include the National Science Foundation Young Investigator award and the Sloan fellowship. He published more than 200 journal and conference papers and he holds 25 US patents. His papers were recognized in journals and conferences, including, winning the 2005 S. A. Schelkunoff Transactions prize paper award from the IEEE Antennas and Propagation society and the 2003 Best Paper Award in the 2003 Design Automation Conference.



**Jie Gao** received her Ph.D in computer science from Stanford University in 2004, and her BS degree from University of Science and Technology of China in 1999. She is currently an assistant professor at Computer Science department, State University of New York, Stony Brook. Her research interests include algorithms, ad hoc communication and sensor networks, and computational geometry.



**Anxiao (Andrew) Jiang** received the B.S. degree with honors in 1999 from the Department of Electronic Engineering, Tsinghua University, Beijing, China, and the M.S. and Ph.D. degrees in 2000 and 2004, respectively, from the Department of Electrical Engineering, California Institute of Technology. He is currently an assistant professor in the Department of Computer Science, Texas A&M University.

He was a recipient of the four-year Engineering Division Fellowship from the California Institute of Technology in 1999. His research interests include algorithm design, ad hoc communication and sensor networks, and file storage and retrieval.