Routing with Hexagonal Virtual Coordinates in Wireless Sensor Networks

Jang-Ping Sheu*, Ming-Lung Ding, and Kun-Ying Hsieh

*Department of Computer Science
National Tsing Hua University
Hsinchu 300 Taiwan
E-mail: *sheujp@csie.ncu.edu.tw

Department of Computer Science and Information Engineering
National Central University
Jhongli 320 Taiwan

Abstract

Using geographic routing, like GPSR, is efficient for ad hoc and wireless sensor networks, but it requires that nodes be aware of their physical positions. However, if there are holes in the network, routing across them using GPSR will lead to a lot of overloaded nodes on their boundaries. In this paper, we propose a distributed protocol, named Hexagonal Virtual Coordinate (HVC), for constructing a virtual coordinate system. After the HVC is constructed, the nodes in the network will be aware of the relative coordinates among the landmarks through the HVC chart. Based on the HVC chart, a source node can find an Auxiliary Routing Path to indicate the direction of the journey from the source to the destination. Simulation results show that our protocol can support geographic routing efficiently, and the landmarks found by our protocol are uniformly located in the network even if some holes exist within it. In addition, our protocol is resilient to various network shapes and can find a load balancing routing path to the destination even if this path comes across holes in the network.

Keywords: Geographic routing, localization, sensor networks

1. Introduction

A wireless sensor network is composed of a great number of sensor nodes used to gather interesting data everywhere in the network. If the load of the forwarding data is not fair to every node, overloaded nodes may exist. And if each sensor node has a constrained power supply, the overloaded nodes may die quickly, and we will lose a lot of interesting data from them. Designing a fair and efficient routing protocol to share the load of the overloaded nodes is an important issue. In geographic forwarding, a packet is greedily forwarded to its neighbor who is geographically closest to the destination. The nodes' locations are used as their addresses, and packets are forwarded in a greedy manner. The most well-known protocol is GPSR [10]. In a flat and regular region, if nodes are deployed densely and uniformly, geographical forwarding becomes an efficient and scalable scheme which can produce almost the shortest paths with little overhead.

Although geographic routing is efficient, it requires that the sensors be aware of their physical positions. This information can be obtained by equipping all the sensors with devices such as a Global Position System (GPS) [13]. However, a GPS is a costly device (in size, cost, and energy consumption) as opposed to the sensor node. Besides, greedy geographical forwarding runs into serious problems for sensor fields with complex geometry. Where there are holes (communication obstacles) within a sensor field, greedy forwarding may fail when all the neighbors are far from the destination. Greedy forwarding will use perimeter routing to route across the holes, but this is not good for wireless sensor networks due to the constrained power supply. The nodes in the boundaries of the holes will die quickly, and then the holes will become larger and larger, and will soon lose interesting information from the boundaries of the holes. Therefore, the virtual (or logical) coordinate system based on hop counts is proposed to give a solution to prevent the geographic forwarding from being blocked by obstacles in a complex environment. Nodes only need to maintain hop counts to some specific landmarks (or anchors) without being aware of their real positions. Previous works [11] have shown that the virtual coordinate system can support geographic routing efficiently in large scale sensor networks.

In this paper, we propose a distributed protocol to construct a virtual coordinate system by finding out which nodes should automatically be landmarks of the network. We also propose a mechanism for finding a routing path from the source node to the destination through the virtual coordinate system we constructed. Landmarks flood the control packets locally to assign each node a virtual coordinate. The virtual coordinate consists of hop counts to the nearest landmarks. On the other hand, every node has only local relative hops in relation to its nearest landmarks, and every landmark floods the control packet within a small region. Nodes can make greedy forwarding to the nearest ones locally, and they can make greedy forwarding to the farther ones by using relative relations of landmarks support. Simulation results show that the virtual coordinate system constructed by our protocol can support geographic routing efficiently, and the landmarks found by our protocol are located everywhere in the network uniformly even if some holes exist within it. In addition, our protocol is resilient to various network shapes and it can find a load balancing routing path to its destination even if this path comes across holes.

The rest of this paper is organized as follows. Section 2 presents previous work and our motivation. Section 3 describes our hexagonal virtual coordinate system and routing protocol. Section 4 evaluates the performance of our protocol in simulations. Finally, we draw the conclusions in Section 5.

2. Previous Work

Many algorithms are proposed to construct coordinate systems in wireless ad hoc and sensor networks. We can simply classify the algorithms into two categories: One is to find the real coordinates of the nodes [1], [9], [12], [14], [16]. The real coordinate system is to determine the real locations of all the nodes. And the other one is to find the virtual (or logical) coordinate [2], [3], [4], [6], [7], [8], [11], [15], [17]. The virtual coordinate system was constructed to find an embedding of the nodes into multi-dimensional space to reflect the underlying connectivity of the network. In this paper, we are interested in the virtual coordinate system.

The authors in [3] proposed a scalable logical coordinate framework in wireless sensor networks. Nodes in a network maintain hop counts to all the landmarks as their virtual coordinates, and run a greedy routing while transmitting a packet. The simulations show the effect of the number of landmarks and their positions on routing performance. Random landmark placement is compared to uniform placement at the network circumference. Since the number of landmarks and their locations are carefully chosen, it is not applicable in the sensor networks whose nodes are deployed randomly. The authors in [4] proposed an algorithm to find three nodes to be landmarks of the network and the authors in [17] proposed an algorithm to find four nodes near the corners of the network to be landmarks of the network. The landmarks in the above protocols are the global landmarks of the network where every node should be assigned virtual coordinates by all of them. In a correct logical coordinate space, the corresponding coordinates for the same landmark between any two neighboring nodes differ by one at most [3]. Thus, in a large scale sensor network, it will take a lot of time to exchange virtual coordinates between neighbors to reach their mutual neighbors, which differ by one at most. In addition, the global landmarks cannot reflect where the holes are by the virtual coordinates or landmarks.

In [6], the authors proposed a topology-enabled routing protocol. They partitioned the network into a lot of tiles by combinatorial Voronoi/Delaunary techniques. Each tile in the network had its home landmark, and the virtual coordinates of a node are assigned by the home landmark and its neighboring landmarks on the combinatorial Delaunay graph. The topology of the network could appear in the combinatorial Delaunay triangulation. But the landmarks near the holes of the network were chosen manually, and the others were chosen randomly. The landmarks here can be thought of as the local landmarks because of the virtual coordinates of the nodes were assigned by the closer landmarks, and these landmarks here reflected the topology of the network roughly. Since the virtual coordinate of each node was assigned by its local landmarks, it took less time to make the virtual coordinate system stable compared to the global landmarks.

The authors in [7] proposed a macroscopic geographic greedy routing (MGGR) protocol that combines advantages of GPSR [10] and GLIDER [6]. In GLIDER, if a packet cannot reach its final destination by the local coordinates, the packed will flood in whole tile. The authors in MGGR choose larger number of landmarks than GLIDER to create small tiles that can decrease the cost of flooding the packet in whole tile. Since the navigation between the tiles in GLIDER is based on the global availability of the combinatorial Delaunay graph, the maintenance and storage overhead of GLIDER grows with the number of tiles. To reduce the overhead, the MGGR uses geometric routing with location information to navigate between the tiles. However, the drawback of MGGR is the assumption that all the nodes in a wireless network are aware of their true geographic location.

In a large scale wireless sensor network, a lot of small and large holes might exist in the network. We do not know where the holes are without geographic location support or we detect them only after the nodes are deployed. To make our protocol scalable for real world application, we propose a distributed algorithm to build the virtual coordinate system automatically. Because local landmarks can reflect the topology of the network roughly and the virtual coordinate system can reach stability quickly, this protocol uses local landmarks to construct the virtual coordinate system. In addition, we wish our virtual coordinate system to be suitable for various network shapes with holes. This way of finding landmarks should come from inside to outside of network to suit various network shapes. Thus, we proposed a protocol, named Hexagonal Virtual Coordinate (HVC), to create a virtual coordinate system with load-balancing routing.

3. Routing Protocol with a Virtual Coordinate System

In this section, we present a novel virtual coordinate system, which depends only on node connectivity and does not rely on real node position. Our protocol consists of two phases. The first phase is constructing the virtual coordinate system, called Hexagonal Virtual Coordinate (HVC), by selecting some specific nodes to be the landmarks of the network. In the second phase, a routing scheme with the HVC is proposed.

3.1 Overview of HVC

Suppose that $G = (V, E)$ is a communication graph on the sensor nodes V, and the edges E present which pairs of nodes have direct communication with each other but not with the geometric distance between them. The graph distance between two nodes is estimated by hop distance to their identical nearest landmarks which are the common coordinates of the two nodes. The virtual coordinate of a node is a vector assigned by several nearest landmarks which represents the relative hop distance from it to them, and this enables nodes to make greedy forwarding to the nearest ones locally. Nodes with different nearest landmarks will have different virtual coordinate vectors, and communication between them should have the global view of the relative relation between landmarks, called the HVC chart. Suppose that $C(G) = (V', E')$ is the HVC chart. V' is a subset of nodes V, and is composed by the landmarks of the network. E' represents the hop distance between the pairs of landmarks in V', and the hop distance is less than some specific values to make routing efficient. Thus, we define two specific values in our protocol. The first one is R, which indicates the hop distance between the two adjacent landmarks we wish to find, and the value of it will influence the number of landmarks in the network. The second one is X, which indicates the maximum hops of forwarding control packets to assign virtual coordinates to nodes by landmarks.

To route the source node to the destination node, we flood the HVC chart to every node in the network to show them the global topology. Each node can find a shortest path, called the Auxiliary Routing Path (ARP), from it to the landmark nearest to destination node in the HVC chart to indicate the direction to the destination. The landmarks in the ARP will guide the packet to be greedy forwarded to its destination hop by hop. The landmarks in the ARP are similar to the pharoses; while we navigate in the dark ocean, they guide us to reach our destination sequentially. The HVC chart is similar to the nautical chart, which indicates where the pharoses are, as well as the ARP, which shows us the shortest path in the journey to the destination.

For example, the network in Fig. 1 has a large hole within it. The black triangle is the sink node, which initiates constructing the virtual coordinate system. The black dot is the landmark, and the black line represents the distance between two close landmarks. The HVC chart is composed of the landmarks and the black links between them. In Fig. 2, the path $S \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow D$ is the ARP found by the source node S to guide the direction to destination node D, where T_1, T_2, T_3, and T_4 are landmarks. Nodes can make greedy forwarding to their neighbors nearest to a landmark which is nearest to D in the ARP. Therefore, we can find a routing path from S to D, which is shown as sequences of red arrows in Fig. 2.

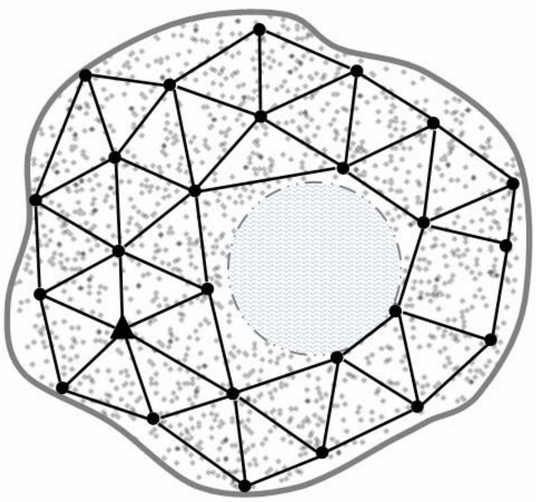

Figure 1: The HVC chart is constructed by the landmarks and the black lines.

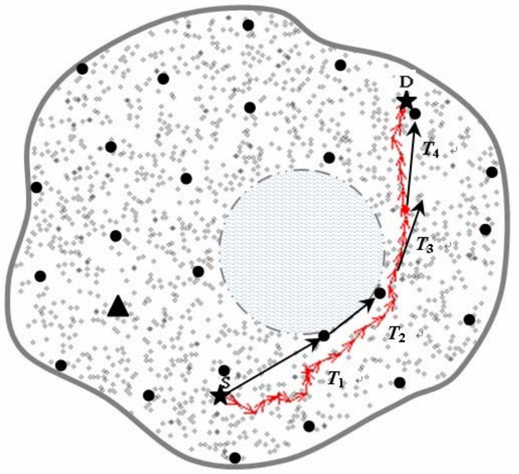

Figure 2: The ARP from the source node *S* to destination node *D* is $S \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow D$. The routing path from *S* to *D* is shown as sequences of red arrows.

3.2 HVC Construction Protocol

Here, we present our HVC construction protocol. In Fig. 3, we can see that the vertices of a hexagon with radius *R* are the intersection points of circles centered at each corresponding vertices with the same radius *R*. Therefore, we can construct a lot of hexagons from a specific point such as point *P* in Fig. 3. Note that, each vertex of a hexagon is the center of a circle. If we choose the centers of circles to be the landmarks of the network, we can obtain many landmarks which are uniformly distributed in the network. Since the virtual coordinates of nodes are assigned by the vertexes of hexagons and their centers in our protocol, we name the virtual coordinate system Hexagonal Virtual Coordinate (HVC).

Our HVC construction method can find landmarks in various network shapes and the landmarks are spread over the sensor network. In addition, this method can find landmarks surrounding holes and make an efficient routing to pass around the holes. We will validate it in our simulations. The HVC construction protocol runs as follows: Firstly, a specific node or sink node is assigned as the first landmark. Then we can find the second landmark by the flooding of the first assigned landmark. Through the first and second landmarks, we can recursively find all the other landmarks. Finally, the sink node will construct the HVC chart and broadcast it to all nodes in the network.

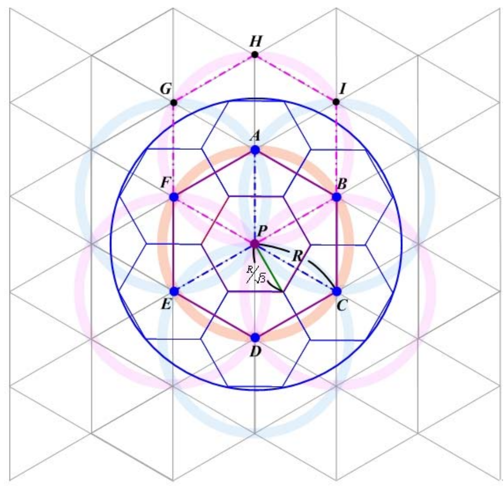

Figure 3: The vertices of hexagon *ABCDEF* and its center *P* are landmarks which are elected in the intersecting regions of the ring-shaped areas.

In our protocol, each node in the network stores a Virtual Coordinate Vector (VCV), which consists of no more than seven pairs of (*lm_id*, *lm_hop*), where *lm_id* is the identification (ID) of a landmark and *lm_hop* is the hop distance to the landmark. The VCV of a node records the hop distances between this node and seven nearest landmarks at most. We define the neighboring landmarks of a node to be all of the *lm_id*s in its VCV to indicate all its nearest landmarks. We assign a specific node or sink node located near the center of the network as the first landmark in which to begin our HVC construction protocol. Here, we let the first landmark be the sink of the network. Initially, the sink node sets *lm_id* = the ID of itself and *lm_hop* = 0 to be the first pair of (*lm_id*, *lm_hop*) in its VCV, and then it floods a control packet to the entire network. The purpose of flooding control packets by landmarks or sink is done so that virtual coordinates to nodes are assigned. The control packet includes the ID of the sink node, *lm_hop* = 1, a bit to indicate the packet is sent from the sink, and two specific values, *R* and *X*, where *R* indicates the hop distance between two adjacent landmarks we wish to find, and $X > R$ indicates the maximum hops for flooding the control packet in assigning the virtual coordinate to nodes by landmarks. A node receives the control packet, records the received (*lm_id*, *lm_hop*) in its VCV, and then increases *lm_hop* one in the control packets. After that, each node records the control packet was sent from sink and then forward it to the other nodes in its communication range. Nodes that receive the control packet again will drop it. Note that the control packet flooding by the sink node covers the entire network to inform every node who is the sink node but the control packet flooding by other landmarks covers only X hops.

After the control packet floods over the R hops, a ring-shaped area will exist in which the nodes in this area will have R hops distance to the sink node. In general, the ring-shaped area is not a perfect ring as shown in Fig. 4(a). However, for the convenience of presentation, the ring-shaped area is drawn as a perfect ring, as shown in Fig. 4(b). Assuming the ID of a node in the network is unique; the node with maximum ID in the ring-shaped area centered on the first landmark will be elected as the second landmark. To select the node with the maximum ID in this ring-shaped area, a node which has the distance to the sink node = R hops and maximum ID within its two-hop neighbors will flood a control packet with its ID in this ring-shaped area. Each node which has *lm_hop* = R assigned by the sink node receives this control packet, and will forward it if the ID in the packet is larger than the ID for itself. Otherwise, the control packet will be dropped. Finally, the control packet with the maximum ID will go back to the initiated node, and this node will become the second landmark. For example, assuming node P is the first landmark, as shown in Fig. 5, and the maximum ID in the ring-shaped area is node A, and node A will become the second landmark of the network. However, if the ring is broken by holes into disconnected segments, the node with maximum ID in each segment will become the landmark too.

When the second landmark is elected, the landmark records the second pair of (*lm_id*, *lm_hop*) in its VCV with *lm_id* = the ID of itself and *lm_hop* = 0. The second landmark then sends a control packet including *lm_id* = the ID of itself and *lm_hop* = 1 to nodes within X hops. Each node receives the control packet, records the received (*lm_id*, *lm_hop*) in its VCV, and forwards the control packet with an increasing *lm_hop* to the other nodes in its communication range. Each node that receives the control packet will forward it if the *lm_hop* in the control packet is less than or equal to X.

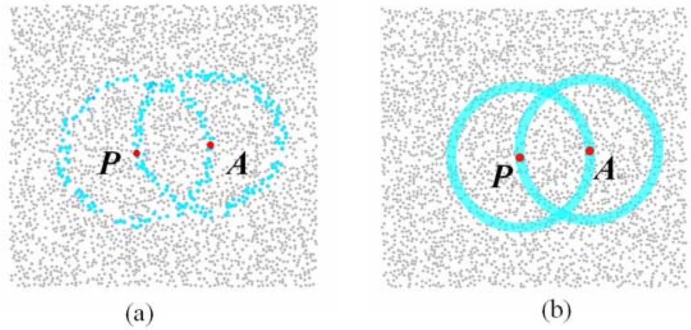

Figure 4: (a) The simulation results of nodes located at the R'th hop centered on red nodes A and P. (b) The ring-shaped areas are drawn as two perfect rings.

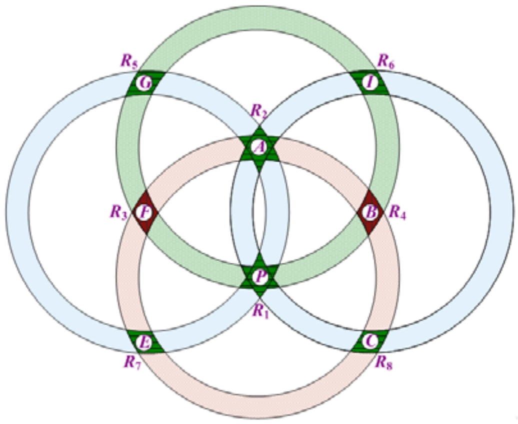

Figure 5: The perfect rings with radius = R hops. The landmarks are elected from the intersecting regions of the ring areas.

After the second landmark floods over R hops, the two rings centered at the first and second landmarks will have two intersection regions. For example, in Fig. 5, assume the first landmark P is located in region R_1 and the second landmark A is located in region R_2, then the intersection regions of the two rings with centers P and A are R_3 and R_4. Two nodes which have maximum ID within regions R_3 and R_4 will be elected as the third and fourth landmarks, respectively. A two-hop local flooding can be used to elect the landmark in each region. When the two landmarks are elected from regions R_3 and R_4, each landmark floods a control packet to assign the nodes within X hops a pair of (*lm_id*, *lm_hop*), just like the second landmark. Similarly, we can get the third and fourth ring areas centered at the third and fourth landmarks. The nodes in the third (fourth) ring have R-hop distance to the third (fourth) landmark. Thus, the third and fourth rings will intersect the first and second rings at four regions. For example, in Fig. 5, the first and second rings intersect the third and fourth rings at regions R_5, R_6, R_7, and R_8. We use the local flooding to elect a new landmark from each of the four regions. Note that, we ignore the intersection regions R_1 and R_2. This is because regions R_1 and R_2 have had landmarks elected before. This case can be easily checked by the VCV of those nodes within regions R_1 and R_2. The above procedure will continue until we cannot find any new landmark in the network.

An additional rule is added to enhance our HVC construction protocol. We do not want to elect a landmark which is close to any other existing landmarks. If a node which has one *lm_hop* is less than or equal to $R/\sqrt{3}$ in its VCV, it cannot be elected as a landmark, where the value of $R/\sqrt{3}$ is the farthest hop distance from the center node of a hexagon to one of its vertexes, as shown in Fig. 3. This rule guarantees that no new landmark will be elected in the hexagon centered at a landmark with radius $R/\sqrt{3}$ hops. For example, in Fig. 3, if node *P* is a landmark, we can be sure that there is no new landmark in the hexagon around landmark *P*. Note that, the VCV in a node has at most seven pairs of (*lm_id*, *lm_hop*). If a node receives more than seven pairs of (*lm_id*, *lm_hop*), it will drop the pair of (*lm_id*, *lm_hop*), which has the largest *lm_hop* in its VCV.

As mentioned above, the HVC chart is composed of the landmarks of the network and the hop distances between each pair of closer landmarks. The virtual coordinate of a node is assigned by its nearest landmarks with hop count $\leq X$. This means that each node in the network knows only the local hop distances from it to its nearest landmarks. Each node has no idea how to route from it to other nodes which are away from its local area. Thus, we introduce the HVC chart to make every node know about the relative position of the landmarks, as well as the route to everywhere in the network. How to find a routing path using the HVC chart will be discuss later. Here, we just describe how to construct the HVC chart.

Landmarks are the nodes elected to help us to construct the HVC. A landmark has at most seven pairs of (*lm_id*, *lm_hop*) in its VCV, one pair coming from itself and the others coming from the vertices of a hexagon inscribed in the circle, whose radius is *R* hops centered at that landmark. To gather the relative position of the landmarks, when a landmark receives seven pairs of (*lm_id*, *lm_hop*), it transmits a packet with its VCV to the sink node by flooding. If a landmark is located at the boundary of the network or the hole, it may receive less than seven pairs of (*lm_id*, *lm_hop*). We can set a threshold time T_b for a landmark to gather its seven pairs of (*lm_id*, *lm_hop*). When a landmark cannot receive enough pairs of (*lm_id*, *lm_hop*) within time T_b, it has to transmit a packet with its VCV to inform the sink node, too. After all the landmarks have transmitted the VCV packets to the sink, the sink can construct the HVC chart from them. The HVC chart can then be constructed by the sink, and the sink node will flood the HVC chart to all the nodes in the network to let them know the relative locations between the landmarks.

3.3 Analysis of the Flooding Radius of Landmarks

As mentioned above, we assume the maximum hops of flooding control packets by the landmarks in assigning virtual coordinates to nodes is *X*. The value of *X* is an important factor which will affect the routing path length. In the following, we will prove that if $X = 1.53R$, every node except in the boundaries of a network or holes can receive at least seven control packets for assigning virtual coordinates from landmarks in a dense network.

Theorem 1: If $X = 1.53R$, any node located in the network except in the boundaries of a network or holes can receive at least seven control packets for assigning virtual coordinates from landmarks in a dense network.

***Proof*:** In a dense network, we can assume that a landmark elected by our protocol is located at the center of a hexagon. In Fig. 6, the intersection points of the underlying gray lines can be treated as the landmarks founded by our HVC construction protocol. Consider a node located near the landmark *B*, which can receive control packets of assigning virtual coordinate from landmark *B* and its six closer landmarks, *A*, *C*, *D*, *E*, *K*, and *P*, if $X \geq R$. However, node *Q* near the vertex of the hexagon centered at landmark *B* is the farthest node from the landmarks *P* and *K* within the hexagon, as shown in Fig. 6. Therefore, if $X =$ the distance from *P* to *Q*, any node in the hexagon with center *B* can receive at least seven control packets for assigning virtual coordinate from landmarks. The distance from node *P* to node *Q* can be derived as follows. Since $\overline{AB} = R$ hops and $\angle BQA = 60^\circ$, we have $\overline{AQ} = \frac{2}{\sqrt{3}}\overline{AB} = \frac{2}{\sqrt{3}}R$ hops. Then we can get the distance from *P* to *Q*:

$$\overline{PQ} = \sqrt{\overline{AP}^2 + \overline{AQ}^2} = \sqrt{(R)^2 + \left(\frac{2}{\sqrt{3}}R\right)^2} = \sqrt{\frac{7}{3}}R = 1.53R$$

hops. The node Q is one of tip-top of a flooding area in which the landmark P would assign a coordinate to. Thus, the landmark P will flood $X = 1.53R$ hops to assign the farthest nodes virtual coordinates in its neighboring hexagons. In addition, each node in the network will be located in a hexagon. Thus, after the landmarks flood $1.53R$ hops to assign nodes to virtual coordinates, each node located in the network (except in the boundary of network or holes) will receive at least seven control packets for assigning virtual coordinates from landmarks.

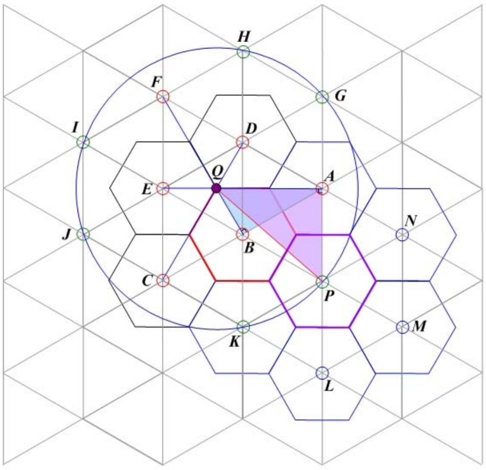

Figure 6: Node Q is a tip-top of the flooding area in which the landmark P would assign a virtual coordinate to. $\overline{PQ}$ is the radius of the flooding area by landmark P.

In Fig. 6, if we draw a circle centered at node Q, and enlarge it until reaches the landmark P, the circle can reach the landmarks B, D, and E first, and then reach A, C, and F, finally reach P, G, H, I, J, and K. The landmarks A, B, C, D, E, and F are the six closer neighboring landmarks of node Q. Since a node only requests seven nearest landmarks in our protocol, the last one is chosen from one of the landmarks P, G, H, I, J, and K randomly. Here, we try to further reduce the value of X to decrease the flooding overhead of our protocol. We do a simulation with 90,000 nodes in a 300 m × 300 m network. The distance between any two adjacent landmarks is about 50 m. In Fig. 7, the red nodes and node P are landmarks, and the nodes in the gray and purple areas are those with the seven nearest landmarks including landmark P. Node Q is one of the farthest nodes to landmark P in the purple area, and node Z is one of the farthest nodes to landmark P in the gray area. If the length of the line connected by node P and A is R hops, the length of the line connected by node P and node Q is $1.53R$ hops. By our simulation, if the value of X = the distance from P to Z, most nodes will receive control packets from seven landmarks except for the nodes located in the purple area, which can receive control packets only from six landmarks. This is because there are a few nodes, such as node Q, located near the vertices of a hexagon, as shown in Fig. 6. In our later simulation, when the value of X reduces from $1.53R$ to the distance between P and Z, it only has a little influence to our routing performance. In order to reduce the flooding overhead of landmarks, we can set X to be equal to the distance between node Z and landmark P. This distance is about $1.4R$ hops, which can be derived as follows.

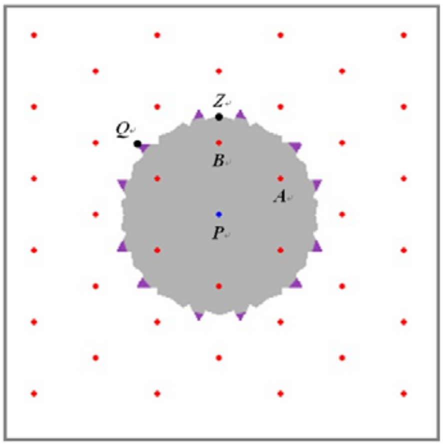

Figure 7: Two different tip-tops of forwarding areas are the nodes Q and Z.

Since the gray area in Fig. 7 can be treated as a circle centered at landmark P, we can randomly choose a node located on the boundary of this circle to derive its radius. In Fig. 8, we choose node Z located along the line segment $\overline{PB}$ as the boundary node Z in Fig. 7. Referring to Fig. 7, each node within the gray area has seven nearest landmarks including landmark P. Thus, node Z in Fig. 8 is the farthest node with seven nearest landmarks, A, B, C, D, E, P, and K, along the line segment $\overline{PB}$. Therefore, if we draw a circle centered at node Z with radius $\overline{PZ}$, nodes F and J are on the boundary of this circle. Here, we are going to calculate the distance between node Z and landmark P. In Fig. 8, the angles of $\angle EAP = 90°$ and $\angle ADR = 60°$ Thus, $\overline{AR} = \frac{\sqrt{3}}{2}\overline{AD} = \frac{\sqrt{3}}{2}R$ and

$\overline{PR} = \sqrt{\overline{AR}^2 + \overline{AP}^2} = \frac{\sqrt{7}}{2} R$. If the angle of $\angle APR = \alpha°$, we can get $\sin \alpha° = \frac{\overline{AR}}{\overline{PR}} = \sqrt{\frac{3}{7}}$ and $\cos \alpha° = \frac{\overline{AP}}{\overline{PR}} = \frac{2}{\sqrt{7}}$. Since the angles of $\angle PRZ = 90°$ and $\angle APB = 60°$, we can set the angle of $\angle RPZ = \beta° = 60° - \alpha°$. Thus, $\cos\beta° = \cos(60° - \alpha°) = \cos 60° \cos\alpha° + \sin 60° \sin\alpha° = \frac{1}{2} * \frac{2}{\sqrt{7}} + \frac{\sqrt{3}}{2} * \frac{\sqrt{3}}{\sqrt{7}} = \frac{5}{2\sqrt{7}}$ and $\cos\beta° = \frac{\overline{PR}}{\overline{PZ}} = \frac{\sqrt{7}}{2\overline{PZ}}$.
Therefore, we can get the length of $\overline{PZ} = 1.4\,R$.

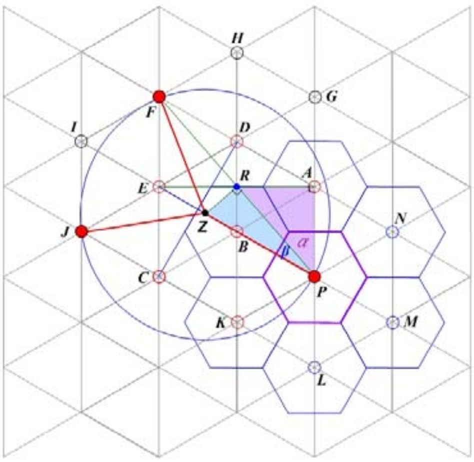

Figure 8: Node Z is another tip-top of flooding area that the landmark P can assign virtual coordinate to.

In summary, when the value of X is equals to 1.4 R, there exists a few nodes cannot receive seven control packets for assigning virtual coordinates from landmarks, but the effect to routing performance is little. Thus, we set $X = 1.4\,R$ in our later simulations.

3.4 HVC Routing Protocol

After the HVC is constructed, the nodes in the network become aware of the relative coordinates between the landmarks through the HVC chart. The HVC chart can point out where the destination is and where the landmarks are. We introduce an Auxiliary Routing Path (ARP) to indicate the direction in the journey from source to destination, so finally we can find a routing path to the destination with the ARP support. In our protocol, source node S will make greedy forwarding to destination node D. The landmarks shown in the ARP are merely to guide the packet to its destination, and we do not necessary forward a packet to reach any landmark as long as the next node is closer to the destination than the current one. Each landmark is treated as a standard node after the virtual coordinate system HVC is constructed. Note that, the last landmark in the ARP is the one nearest the destination node.

The ARP is a path made up by the source node, destination node, and some landmarks to indicate the direction of packet transmission. The neighboring landmarks of the source and destination nodes can be thought of as the outlets and inlets in which packets can deliver out and receive from, respectively. Through intuition, we may find the shortest path from source to destination to be the ARP. However, we assume the inlet for the packet to receive from is the landmark nearest to the destination node to increase the success rate of forwarding packets while the packet was forwarded to the destination directly. We only add one direction to guide the packet to be received accurately, but this does not increase the routing path. Thus, we can apply the shortest path algorithm, like Dijkstra's algorithm [5], to find the shortest path in HVC chart from the source node to a landmark nearest to the destination node as the ARP. For example, in Fig. 9, if T_5 is the nearest landmark of the destination node D, we can find the shortest path from S to T_5 by using Dijkstra's shortest path algorithm. If the hop distance of path $S \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow D$ is the shortest one found by Dijkstra's algorithm, it will become the ARP for the packet to deliver from S to D.

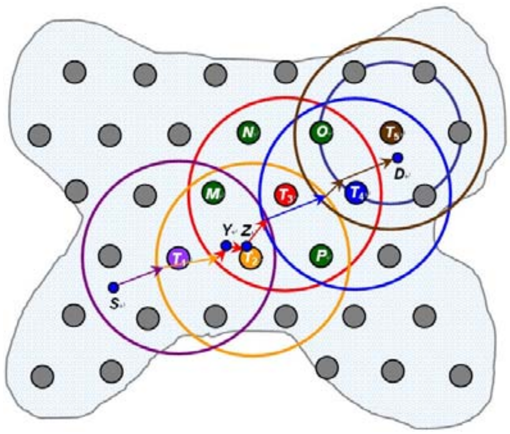

Figure 9: A diagram to demonstrate how to route source S to destination D. The ARP of this routing path is $S \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow D$. The nodes are the landmarks of the network except S, Y, Z, and D.

Since the distance between any pair of adjacent landmarks found by our protocol is almost equal, the ARP found by the source node may have many different choices due to the same path length. For example, in Fig. 9, there are six different paths from source node S to destination node D. The six paths aside from S and D are $T_1 \rightarrow M \rightarrow N \rightarrow O \rightarrow T_5$,

$T_1 \rightarrow M \rightarrow T_3 \rightarrow O \rightarrow T_5$, $T_1 \rightarrow M \rightarrow T_3 \rightarrow T_4 \rightarrow T_5$, $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow O \rightarrow T_5$, $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5$, and $T_1 \rightarrow T_2 \rightarrow P \rightarrow T_4 \rightarrow T_5$, respectively. Thus, we can randomly choose one of the paths to be our ARP to direct the packet forwarding direction for the journey from S to D. The multiple paths from the source to the destination can achieve better load balancing without hurting the quality of the packet transmission.

After an ARP is chosen by the source node, the ARP is added to the packet to guide the direction from the source to the destination. We face the next node (landmark) of S in the ARP and set it as the temporary destination to deliver the packet. We apply greedy forwarding over the virtual coordinate system. A node will choose one of its neighboring nodes with the least logical distance to a temporary destination as the next relay node. The logical distance of two nodes are defined as follows: Let virtual coordinates of nodes A and B be $(a_1, a_2, \ldots, a_m)$ and $(b_1, b_2, \ldots, b_n)$ which are assigned by the m and n landmarks, respectively, where $m \leq 7$ and $n \leq 7$. Assuming that landmarks l_1, l_2, …, and l_k are the common neighboring landmarks of nodes A and B, where $k \leq m$ and $k \leq n$. Let a_i' and b_i' denote the hop count from node A and node B to landmark l_i, respectively. The logical distance between nodes A and B is then defined as $D(A,B)=\sqrt{\sum_{i=1}^{k}(a_i'-b_i')^2}$. The nodes in the ARP will guide the packet to destination hop by hop until the packet is forwarded to its destination.

Without loss of generality, we assume there exists an ARP $S \rightarrow T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_k \rightarrow D$. An example is illustrated in Fig. 9. In the beginning, the source node sets the landmark T_1 to be the temporary destination and forwards the packet to a node which is closer to T_1. A node receiving the forwarding packet will look up its VCV first to check if there exists the farther landmark T_2 in its neighboring landmarks. If yes, it will set landmark T_2 to be a new temporary destination and forward the packet to a node which is closer to T_2. Otherwise, it will continuously forward the packet to a node which is closer to a temporary destination. In our later simulations, a packet can always make progress from a landmark to its next landmark until the last landmark T_k is set as a temporary destination. However, it may lose direction as it progresses towards the destination node D while the relay node is X hops away to T_k. To make the routing more reliable, a node will not forward a packet directly to destination node D unless the hop distance from it to T_k is less than R.

If an intermediate node cannot find a neighboring node which is closer to the temporary destination, the routing path can be said to be in a local minimum condition. This node will replace the temporary destination with a nearest landmark which is selected from its neighboring landmarks. When the intermediate node suffers from local minimum again, we can set the second nearest landmark to be its temporary destination. This procedure will go on until there is no local minimum, and we can set the original replaced landmark as the temporary destination again. For example, in Fig. 9, assuming that routing from S to D suffers from the local minimum in node Y, node Y will set the temporary destination from T_3 to landmark T_2. While we are reaching node Z, whose logical distance to T_3 is less than the preceding node, we are also setting T_3 as the temporary destination and continue our journey. While the landmark T_5 becomes the temporary destination, and the hop distance from transmitter to T_5 becomes less than R hops, this means that the destination node D is close. We can then deliver the packet to D directly.

4. Simulations

We used JAVA to implement our simulations. The aim of our simulations was to verify the correctness and feasibility of our protocols. We partitioned our simulations into two parts. The first part is to evaluate our HVC construction protocol in irregular network shapes. The second one is to evaluate the load-balancing and the path length in our routing protocol. In all the figures, the sink node and landmarks are marked as blue triangle and red circles, respectively, and the sensor nodes are shown as small gray circles.

4.1 Irregular Network Shapes

To validate if our protocol is resilient to various network shapes, we did simulations for different scenarios. We choose 10,000 pairs of source and destination randomly to evaluate the packet delivery ratio. In all the simulations, the communication range of each node is 10 m and X = 1.4 R. The first scenario is shown in Fig. 10. There are 2,500 nodes randomly distributed in a triangle area with one large hole = 50 m, where the base of this triangle is 500 m and altitude is 400 m. In average, each node has 8.5 neighbors. We set R = 9 hops. In our simulations, the packet delivery ratio from source to destination is 96.23%.

From our simulations, we found out that nodes located near the acute angles of a triangle network leads to routing failure due to the existence of only one landmark. We added one more landmark in each region of the vertex to increase our routing performance. Since each node has the hop distance to the sink, nodes located near the vertex of the network have a larger hop count than the other nodes. We selected the nodes with the maximum ID and the maximum hop distance to the sink within their two-hop neighbors as landmarks too. If we elect one more landmark in each region of the triangle vertex, the packet delivery ratio can reach up to 99.68%.

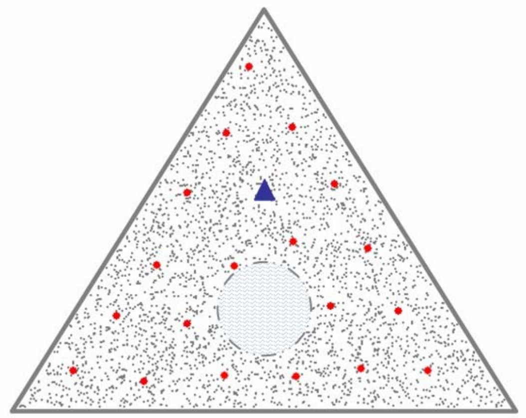

Figure 10: A triangle network with a large hole.

The second scenario is shown in Fig. 11. There are 5,000 nodes randomly distributed in a 500 m × 400 m rectangle area with four large holes in four corners, respectively, and three different shaped holes within the network. We set R = 8 hops. The packet delivery ratio is 99.28%. The third scenario is shown in Fig. 12. There are 1,500 nodes distributed randomly in an irregular network with a large circle hole = 60 m. The width and length of the network are 320 m and 300 m, respectively. We set R = 8 hops. The packet delivery ratio is 99.56%.

We have shown that the virtual coordinate system constructed by our protocol is resilient to various network shapes. No matter where the holes are located in the network, the virtual coordinate system can be constructed automatically and the packet delivery ratio is higher than 99%.

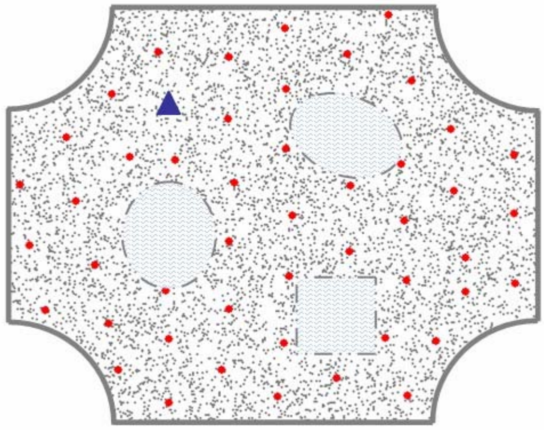

Figure 11: A rectangle network with four large holes in the corners and three large holes within the network.

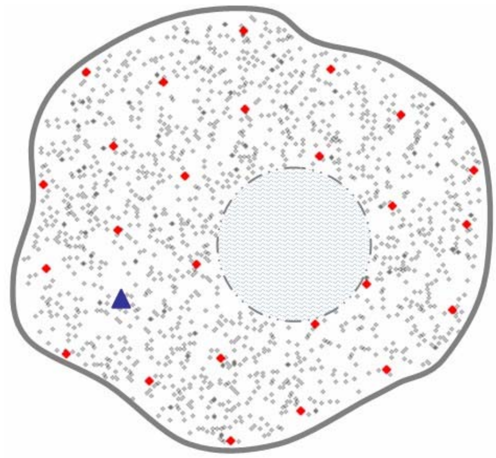

Figure 12: An irregular network with a large hole.

4.2 Load Balancing and Path Length

We are going to show that our protocol has the load-balancing routing ability to prevent some nodes from being overloaded. We simulated an irregular network as shown in Fig. 13. There are 1,500 nodes uniformly deployed in the network. Routing from the source to the destination may come across holes or it may not. We randomly choose 50 pairs of source and destination nodes with distance more than 20 hops away for each pair.

In our protocol, the distances between the closer landmarks are almost equal. Thus, the ARP found by the source node may have many different choices. Fig. 13 is an example. Routing from the source to the destination without coming across large holes may have many different paths. This means that the nodes in these paths can share the load of forwarding data from the source to the destination while the ARP is randomly chosen by the source. This is one of our advantages in finding nodes to be landmarks which are uniformly distributed in the network. In addition, the routing paths from the source to the destination and the reply from the destination to the source are different due to them having different ARPs. On the other hand, while routing across large holes, the boundary nodes were chosen as frequent relay nodes. Fig. 14 shows the hot spots in the network with 50 pairs of randomly chosen sources and destinations. Note that, the different colors represent different traffic loads: green (5-7 transit paths), pink (8-10 transit paths), red (11-13 transit paths), and black (≥ 13 transit paths). The black nodes are the most overloaded, and there are only 3 black nodes in our HVC protocol. However, the number of black nodes in the GPSR is 14, almost 5 times of our protocol. This simulation result shows that the nodes near the boundary also share the loads in our HVC protocol. When a packet gets stuck in the GPSR, it is forwarded along the boundary of a hole until greedy forwarding becomes possible again. Thus, the nodes in the boundaries of the holes are overloaded. But our HVC protocol can reflect the connectivity between nodes and the routing path, which is related to the landmarks in ARP. So the nodes near the boundaries of the holes can share the load of the forwarding data evenly. To sum up, our protocol can find a load-balancing routing path to its destination whether or not this path comes across holes or not.

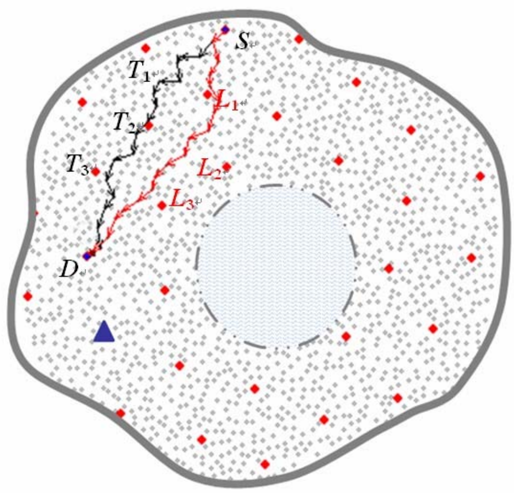

Figure 13: Load balancing routing while the routing path does not come across large holes. Routing from S to D can have many different ARPs, like $S \rightarrow T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow D$ or $S \rightarrow L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow D$.

In the first routing experiment, the average path length for our HVC protocol is 25.78 hops, and for the GPSR is 23.02 hops. Our protocol wasted 2.76 hops to correct the real direction to the destination. In the experiment of routing across holes, the average path length for our HVC protocol is 25.54 hops, and for the GPSR is 24.6 hops. Our protocol wasted only 0.94 hops more than the GPSR in sharing the loads of the boundary nodes. Although our protocol wasted more hops to forward a packet to its destination, we achieved load balancing routing to extend the network lifetime.

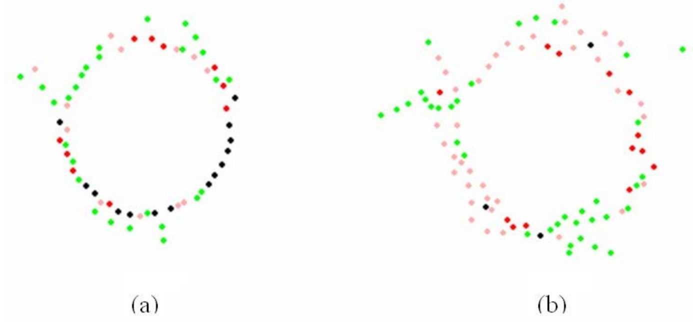

Figure 14: Load balancing routing while the routing path comes across holes. The hot spots in the network with the 50 pairs of randomly chosen sources and destinations: (a) traffic distribution map of GPSR (b) traffic distribution map of HVC.

5. Conclusions

We proposed a distributed protocol to create a virtual coordinate system and give load-balancing routing in wireless sensor networks. The simulations showed that our protocol is suitable for various network shapes, and the nodes in the network can share the load for forwarding data evenly. In addition, while forwarding a packet across holes, we made the routing path generated by our algorithm as short as possible, and the load could be shared by all the other nodes in the boundaries of the holes. However, while forwarding a packet across holes, the routing path generated by the GPSR would go to perimeter routing, and the load of forwarding data becomes overloaded in the nodes surrounding the holes. The balancing routing in our HVC protocol could extend the network lifetime, allowing it to gather more interesting data from nodes which died from the overhead in the GPSR. Even if there were holes in the network, we were able to find some nodes near the holes and identified them as landmarks to make the routing more balanced.

We have no idea on how many landmarks need to be found to have the best performance for routing. In our protocol, the less value of R is set, the less probable it is to suffer the local minimum. The memory overhead of maintaining HVC chart to each node, however, will increase. Analyzing the best value of R

for routing performance to suit various networks with different holes sizes and shapes remains to be done in future works.

References

[1] P. Bahl and V.N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," in *Proceedings of IEEE INFOCOM*, pp. 775-784, 2000.

[2] J. Bruck, J. Gao, and A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Networks," in Proceedings of ACM MOBICOM, pp. 88-102, 2005.

[3] Q. Cao, and T. Abdelzaher, "Scalable Logical Coordinates Framework for Routing in Wireless Sensor Networks," in Proceedings of IEEE International on Real-Time Systems Symposium, pp. 349 - 358, 2004.

[4] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks," in Proceedings of IEEE INFOCOM, pp. 150-160, 2005.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. "Introduction to Algorithms," Second Edition. MIT Press and McGraw-Hill, 2001.

[6] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang, "Glider: Gradient Landmark-based Distributed Routing for Sensor Networks," in Proceedings of IEEE INFOCOM, pp. 339-350, 2005.

[7] S. Funke, "Topological Hole Detection in Wireless Sensor Networks and its Application," in Proceedings of the Joint Workshop on Foundations of Mobile Computing, pp. 44-53, 2005.

[8] S. Funke and N. Milosavljević, "Guaranteed-delivery Geographic Routing under Uncertain Node Locations," in Proceedings of IEEE INFOCOM, pp. 1244-1252, 2007.

[9] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Location Schemes in Large Sensor Networks," in Proceedings of ACM MOBICOM , pp. 81-95, 2003.

[10] B. Karp and H. Kung, "Greedy Perimeter Stateless Routing," in Proceedings of ACM MOBICOM, pp. 243-254, 2000.

[11] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual Coordiantes for Ad Hoc and Sensor Networks," in Proceedings of the Joint Workshop on Foundations of Mobile Computing, pp. 8-16, 2004.

[12] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," in Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications, pp. 105-111, 2002.

[13] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," in Proceedings of IEEE GLOBECOM, Vol. 5, pp. 2926-2931, 2001.

[14] N. B. Priyantha, A. K.L. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-aware Mobile Applications," in Proceedings of ACM MOBICOM, pp. 1-14, 2001.

[15] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing Without Location Information," in Proceedings of ACM MOBICOM, pp. 96-108, 2003.

[16] Y. Shang, W. Rum, Y. Zhang, and M. Fromherz, "Localization from Mere Connectivity," in Proceedings of ACM MobiHoc, pp. 201-212, 2003.

[17] J.-P. Sheu, Y.-C. Chang and G.-H. Song, "Logical Coordinate Assignment for Geographic Routing in Wireless Sensor Networks," Journal of Pervasive Computing and Communications (to appear).