Simple Robotic Routing in Ad Hoc Networks

Daejoong Kim, Nick Maxemchuk
Electrical Engineering Department, Columbia University
{daejoong, nick}@ee.columbia.edu

Abstract

Position-based routing protocols in ad hoc networks combine a forwarding strategy with a recovery algorithm. The former fails when there are void regions or physical obstacles that prevent transmission. Then, the recovery algorithm is used to detour the obstacles. To explore the obstacles and find a path around them, the earlier recovery approaches construct a planar graph to avoid routing loops. Distributed algorithms that find planar graphs require accurate knowledge on the location of nodes. The number of nodes on a recovery path increases as the node density increases. Our novel recovery technique operates on a grid model of a network. Obstacles are approximated by adjacent grid elements. We adopt the right-hand rule, which is common in robotics, to follow the perimeter of the discretized obstacle. We do not construct a planar graph. The grid structure reduces the positional accuracy required for nodes, and the recovery path length is independent of the node density.

1. Introduction

Two types of routing protocols have been developed for multi-hop mobile ad hoc networks; *topology-based routing* and *position-based routing*.

Topology-based routing uses the information about the links between nodes. A path between two nodes consists of a sequence of the links. Some topology-based routing techniques maintain paths to all destinations in the network, even if the paths are not currently used [1]. Other techniques find paths when they are needed by using flood messages [2, 3]. In mobile networks the paths between sources and destinations may change frequently. As paths change more frequently, techniques that maintain all paths must exchange more information, and techniques that only maintain active paths must flood the network more frequently. Besides, as the number of nodes or the path length increases in large networks, the topology-based techniques incur significant routing overhead.

Position-based routing, which we use interchangeably with *geographic routing*, can be an alternative technique for the large network or the rapidly changing network. In geographic routing, the routing decisions are made locally based on the physical position of the forwarding node's neighbors and not all of the nodes in the network. In addition, geographic routing is more robust as nodes move because it does not route through predetermined intermediate nodes. In the remainder of this paper, we focus on geographic routing techniques.

There are four distinct problems in geographic routing: 1) Determining the physical location of nodes. This can be accomplished by using the satellite Global Positioning System (GPS), or by using triangulation techniques among terrestrial nodes. It is difficult to measure precise locations because of noise and random movements of nodes. 2) Determining the location of the destination. Flood messages and location services [6] are two alternative techniques for locating a destination. 3) The forwarding strategy, which selects the next forwarding node that makes progress toward the destination [5, 7, 8, 9]. Greedy forwarding [5], which chooses a single-hop neighbor that is closest to the destination, is widely used to minimize the number of hops. 4) The recovery algorithm that is used to find a path when the forwarding strategy fails. Flooding is a simple recovery technique, but more efficient recovery algorithms have been proposed [4, 9, 10].

In this paper, we propose a forwarding rule and a recovery algorithm that are based on a novel model of the wireless ad hoc network. We assume that the first two problems are solved by any of the existing techniques.

In practical networks, the recovery algorithm dominates the performance of the geographic routing. In our simulations on the campus of Columbia University where there are buildings that block radio communications and temporary obstacles that occur when there are no forwarding nodes in a region, we found that about 95% of the messages experience local dead-ends, when the average number of neighbor nodes is

6.5. Without the recovery algorithm, they would fail to get to the destination.

In general, recovery algorithms partly adopt the Face routing technique [9, 10, 16]. The recovery algorithm takes over from the forwarding algorithm when the message is stuck at a local dead-end. The message circumnavigates the obstacle until a node is found that is closer to the destination than the local dead-end, then the forwarding algorithm resumes. This approach is similar to the navigation technique in robotics. However, in radio networks the simple algorithms [20] that are used to detour around obstacles in robotics may have routing loops. A robot may run along the perimeter of an obstacle touching the wall (with its right hand), while a message must jump from node to node.

Karp [4] has shown that the right-hand rule is loop-free in the multi-hop radio network if we apply the rule on a planar graph, which has no crossing edges. He has shown that either Relative Neighborhood Graph (RNG) or Gabriel Graph (GG) can be used to construct a planar graph in distributed fashion from non-planar graphs of the network. However, in order to construct the planar graph, we must accurately know the position of the nodes. It is very difficult to track the precise location of nodes in a mobile network. Recently, Seada [22] and Kim [21] showed that RNG or GG may not produce a correct planar graph in the practical network with location and link errors. Moreover, in dense networks the planar graphs lead the message to many intermediate nodes, which may be skipped, and produce unnecessarily long recovery paths. Recently, Gao [11] and Li [12] proposed using a Delaunay triangulation graph to construct a better planar graph that produces shorter recovery paths. However, this complicated technique requires communication overhead to construct and maintain the graph.

In this paper, we propose a model that eliminates the need to construct a planar graph. The network is logically partitioned into a grid, and obstacles are approximated by adjoining grid elements. In the recovery phase, the message circumnavigates the obstacle by simply following the outer grid elements of the discretized obstacle. Our *Grid-based Right-Hand Rule* (hereafter Grid RHR) enables the message to trace the boundary grid elements of the obstacle without routing loops. The advantages of this novel technique are: 1) It is less sensitive to the position of nodes than the earlier rules because the algorithm is applied to grid elements rather than individual nodes. 2) When the recovery rule transfers messages between adjacent grid elements, the Grid RHR is loop-free, and we do not have to explicitly construct a planar graph with the individual nodes. 3) The Grid RHR uses fewer hops to trace the boundary of an obstacle because we can transmit to nodes that are up to a transmission distance away. 4) The Grid RHR statistically distributes traffic between the nodes in a grid element, instead of always selecting the same nodes that are closest to the boundary.

The grid-based forwarding and recovery techniques are integrated into a geographic routing protocol, which we call *Robotic Routing Protocol* (RRP). We selected the name because the message in the network is analogous to a mobile robot that finds a good path to its goal in an unknown maze. RRP is a Greedy-Face-Greedy (GFG) type protocol that combines a greedy forwarding rule with Grid RHR. The greedy forwarding rule of RRP also reflects our network model. It determines the next forwarding cell, not a node. The message can be forwarded to any node in the next forwarding cell. In our simulations, RRP traverses half as many intermediate nodes as Greedy Perimeter Stateless Routing (GPSR) [4] over a broad range of node densities.

Since our grid network model makes it easy to estimate and process arbitrary obstacles, it can be used to study a new type of geographic routing that uses the information about obstacles in the network. For instance, better paths that avoid the obstacles can be found and used when multiple messages are transmitted between a source and destination. Just as a mouse in a maze can find shorter routes on successive runs, our messages can take shorter paths on successive runs. The shorter paths that consist of geographic landmarks to avoid obstacles are less affected by mobility than the topology-based shortest paths because the obstacles do not change as rapidly as the paths between individual nodes in the network.

This paper is organized as follows: Section 2 surveys related work. Section 3 presents our network model. Section 4 describes grid-based routing techniques including Grid RHR and Cell-based forwarding rules. Section 5 shows our simulation results. Section 6 discusses future work.

2. Related work

This section briefly surveys forwarding strategies and recovery algorithms of geographic routing.

Most forwarding rules decide the next forwarding node when given positions of the neighbors within radio transmission range. Most Forward within Radius (MFR) [7] chooses the node that makes the greatest progress on the line between the current node and the destination. Nearest with Forward Progress (NFP) [8] selects the nearest neighbor node with forward progress to minimize message conflicts. Greedy forwarding [5] selects the node that is closest to the destination to minimize the number of hops. Peculiarly, Beacon-Less Routing (BLR) [13] forwards the message without the positions of the neighbors. In BLR, the receivers compute the distance from the sender and determine the forwarding of the message with some delay. Other forwarding rules are found in [8, 9, 19]. These forwarding rules cannot guarantee the message delivery because of local dead-ends. Consequently, recovery algorithms that allow backward progress are required.

To recover the message from local dead-ends, the earlier researchers took simple approaches. Flooding is proposed in [14]. Existing topology-based routing techniques such as DSR [2] or AODV [3] are partially employed in [15]. These algorithms may increase the message delivery rate but they undermine the advantages of geographic routing.

Applying the right-hand rule on a planar graph has become a prevalent recovery technique since Karp [4] and Bose [16] independently proposed it. A planar graph produces the boundary path of obstacles that the message can follow without routing loops. This technique is similar to Face routing [9, 10, 16]. A key to these techniques is to efficiently construct and maintain a planar graph. Either RNG or GG constructs a planar graph using local information. The Restricted Delaunay Graph (RDG) that Gao proposes in [11] also makes a planar graph. RDG produces a shorter boundary path than that of RNG or GG, but many messages should be exchanged among neighbors to determine a local planar graph. These planar graph techniques are too location-sensitive to be used in practical networks where location errors and link losses often occur. In the practical network the constructed planar graph may have crossed edges or split the network. Recently, Kim [21] proposed Cross-Link Detection Protocol (CLDP), which constructs a planar graph by probing each link and removing crossed edges. This approach fixes some problems with the geometric planar graph, but the probe messages cause communication overhead, which increases severely in the dense network or in the rapidly changing network.

Some researchers have tried to shorten the recovery path length. Actually, the right-hand rule that always turns to the right-hand side of the obstacle may take a very long boundary path, even though there is a much shorter one on the other side. For this situation, Kuhn [17] proposes Bounded Face Routing (BFR). In BFR, if the message fails to escape from the local dead-end within a certain distance, it traces back to the other direction. Datta [18] proposes a shortcut algorithm in which the forwarding node predicts the next few hops on the path and sends the message directly to the last node within its transmission range. This technique increases the routing table size at each node.

Grid RHR proposed in this paper is a localized recovery algorithm that does not explicitly construct a planar graph. The grid structure is a planar graph in itself. The message can detour the obstacle without routing loops by following the perimeter of the approximated obstacle that consists of grid edges. Grid RHR most of the time finds a shorter path than the shortest boundary path in terms of hop count, independently of the node density.

3. Network model

In geographic routing, obstacles make the routing rules more complicated and degrade the performance of the routing protocol. Unfortunately, the obstacles are common situations in practical ad hoc networks because of geographic features, movements of nodes, and limited radio communications range. They are arbitrary in their shape and size. In this section, we present a network model that facilitates the estimating and processing of the obstacles.

We consider the wireless ad hoc network as a field that consists of transmission regions and obstacles. In the transmission region, the message can make progress to any direction. Obstacles are impediments that prevent messages from being forwarded.

Obstacles can be permanent or transient. *Permanent obstacles* may be geographical features of the landscape, such as mountains, rivers, or lakes, where forwarding nodes cannot be located, or obstructions, such as the wall of a building that inhibits radio communications. *Temporary obstacles* are regions where

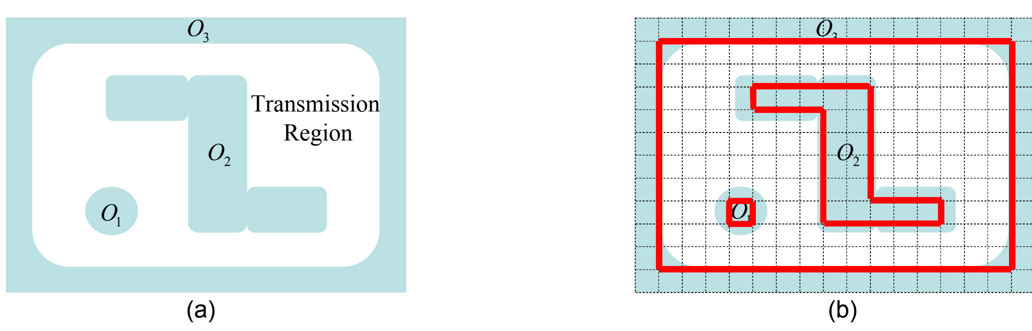


Figure 1. A network is comprised of obstacles and transmission regions. (a) Obstacles in the transmission region, (b) Approximating obstacles by a grid

there are currently no nodes that can forward messages. Permanent obstacles remain for the life of the network but temporary obstacles are continuously changing as the nodes move. There is always an obstacle that defines the perimeter of the network; O_3 in Figure 1.

The shape of an obstacle is defined by the position of the nodes that cannot forward messages in that direction. The obstacles are irregular, complex, and difficult to store and process. We will approximate obstacles on a grid structure as in Figure 1(b). The approximate structure is easier to store and process and is more stable as nodes at the edge of the obstacle move.

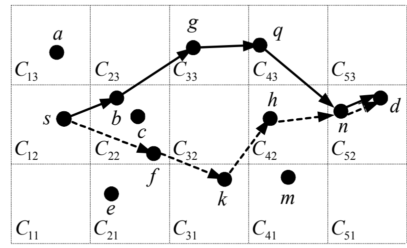
In a two-dimensional network the surface of the network is logically partitioned into squares with edge length d_G. In a three-dimensional network, as will occur in the buildings in a city, the volume is partitioned into cubes. In the remainder of this paper, we consider two-dimensional networks for simplicity.

It is assumed that nodes in the network have the same transmission radius, d_R, and any two nodes within d_R have a bidirectional link. In the two-dimensional network, $d_G \le d_R / \sqrt{2}$, so that all the nodes in a cell are one-hop neighbors and can communicate directly. A larger grid size approximates the obstacles roughly but we can handle the obstacles with less information.

A *wall* is defined as an edge between two cells, C_X and C_Y, if none of the nodes in C_X can communicate with any nodes in C_Y. A set of walls, if they are connected, represents an obstacle. The obstacle also can be represented by a set of adjacent empty cells where any link does not cross. Thus, an obstacle can be approximated to different graphs according to the definition.

Each node in the network maintains the routing information about its neighbor cells as well as its one-hop neighbor nodes. The node knows which neighbor cells are empty or occupied, and how to reach the occupied neighbor cells by associating them with the neighbor nodes. Some neighbor nodes may be located in the neighbor cells, and some can relay to a node in the neighbor cells. The nodes in the network can acquire the routing information by sending "Hello" messages periodically. The message contains the identities of cells that are the sender's adjacent occupied neighbor cells or ones that the sender can reach directly. The nodes that hear the message only save the necessary information. As a result, all nodes in the same cell have the common knowledge about the occupancy state of the 8 adjacent neighbor cells and the direct connections to the neighbor cells.

In this information model, each node maintains its single-hop neighbor nodes, but not two-hop neighbors, and the node does not store the position of the neighbors. The neighbor cell information maintained at each node reflects the existence of its two-hop neighbors as well as the one-hop neighbors.



(a)

s	*b*	*g*	*q*	*n*
C_{12}: *s* C_{13}: *a b* c** C_{22}: *b c f a* e** C_{21}: *e b* c* f** C_{31}: *c* f* e** C_{33}: *b* c* f**	C_{22}: *b c f* C_{12}: *s c* f* a* e** C_{13}: *a c* s** C_{33}: *g c* f** C_{21}: *e c* f* s** C_{43}: *g** C_{42}: *g** C_{31}: *c* f* e**	C_{33}: *g* C_{22}: *b c* C_{43}: *q* C_{42}: *h q** C_{41}: *h** C_{52}: *q* h** C_{13}: *b* c** C_{12}: *b* c** C_{21}: *b* c** C_{31}: *c* h**	C_{43}: *q* C_{33}: *g h** C_{42}: *h g* n** C_{52}: *n h** C_{22}: *g** C_{31}: *h** C_{41}: *h* n**	C_{52}: *n d* C_{43}: *q* C_{42}: *h q* m** C_{41}: *m h** C_{31}: *h* m** C_{33}: *q* h**

(b)

Figure 2. (a) Cell-based forwarding results in the solid path and position-based Greedy forwarding in the dotted path, (b) The routing information of intermediate nodes. * means a two-hop path. Node *s* determines the next cell C_{33} which is one of the closest cells to C_{52} of the destination *d*. There are three paths *b*, *c* and *f* to C_{33}. *s* randomly selects a path among them.

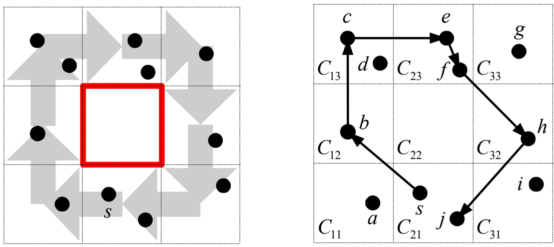
In the practical network with location errors, a node at nearby edges of the grid may have a discrepancy between its actual cell and the cell where it believes it belongs. However, such an error is not likely to directly affect the routing decision as long as the node has the same knowledge on the connections to the neighbor cells as the other nodes in the cell.

4. Robotic routing rules

This section presents robotic routing techniques based on the grid model described in section 3.

4.1. Cell-based forwarding

In the grid model, greedy forwarding strategy is applied to the grid cells rather than the individual nodes. In this *Cell-based forwarding*, a node in C_X determines the next forwarding cell C_N among the neighbor cells so that $DIST(C_N, C_D) = \min(DIST(C_U, C_D))$ for $C_U \in$ Neighbor_Cells, and $DIST(C_N, C_D) < DIST(C_X, C_D)$. C_D is the destination cell and *DIST* calculates the Euclidean distance between two centers of the cells. Once the next forwarding cell is determined, the message can take any path to the cell. Figure 2 shows an example of Cell-based forwarding.



(a) (b)

s	b	c	e	f	h
C_{21}: s j	C_{12}: b	C_{13}: c d	C_{23}: e f	C_{23}: f e	C_{32}: h
C_{11}: a b^* j^*	C_{11}: a s^*	C_{12}: b d^*	C_{13}: c d f^*	C_{13}: d e^*	C_{33}: g f^*
C_{12}: b a^*	C_{21}: s a^*	C_{23}: e d^*	C_{12}: c^* d^*	C_{12}: d^*	C_{21}: j i^*
C_{13}: b^*	C_{13}: c d	C_{11}: b^*	C_{32}: f^* g^*	C_{32}: h g^*	C_{11}: j^*
C_{31}: j^*	C_{23}: c^* d^*	C_{21}: b^*	C_{33}: g f^*	C_{33}: g e^* h^*	
C_{32}: j^*	C_{31}: s^*	C_{33}: e^*		C_{31}: h^*	
				C_{21}: h^*	

(c)

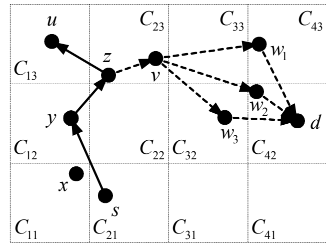
Figure 3. An example of the simple Grid RHR, (a) Logical movements, (b) Practical exploration, (c) The routing information of forwarding nodes.

Cell-based forwarding is similar to Random Forward Progress (RFP) [19] in that it randomly selects the next node, and similar to Greedy forwarding in that it determines the next cell so that the message can make the greatest progress toward the destination.

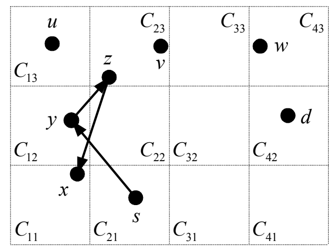
In Cell-based forwarding, the size of the routing table at each node is bounded to a certain number. When $d_G = d_R / \sqrt{2}$, if a node maintains its neighbor cells within two hops, the number of the potential neighbor cells is at most 45 independent of the node density. However, in the dense network where most cells are occupied, maintaining only the 8 adjacent neighbor cells is enough to route messages because the message can make progress to any direction.

4.2. Grid-based recovery algorithms

In radio networks, the right-hand rule that is used to follow the wall of obstacles in robotics may mislead the message into the wrong boundary path when straightforwardly applied to individual nodes. This problem can be resolved by predicting the boundary path with the extended information up to two-hop neighbors, or by taking a smaller step into the suspicious area where there may be a narrow path to the destination. In our grid model, however, the right-hand rule cannot be applied to individual nodes because the nodes do not store the positions. Instead, we apply the right-hand rule to the grid structure. For a brief introduction, consider a network logically partitioned into a grid. If two grid cells are respectively occupied and directly connected, the edges between the two cells are removed. Then, the remaining edges form graphs that approximate obstacles. Now, the message can trace the



(a)



(b)

Figure 4. (a) Simple Grid RHR fails in three cases (w_1,w_2,w_3), (b) Modified rules may produce undesired paths.

border of the approximated obstacle as a micro-mouse does in a maze. The remainder of this section explains how this Grid RHR works.

4.2.1. Simple Grid RHR. The graphs that approximate obstacles in the grid model consist of horizontal and vertical edges. The outer edges of the graph form a cycle if they allow a bi-directed walk. Therefore, when the message follows the outer edges, it can explore the obstacle without routing loops and eventually returns to the start place. Simple Grid RHR uses these inherent properties of the grid structure.

In the simple Grid RHR, the message is allowed to move in four directions: forward, backward, left, and right, as a micro-mouse moves in a maze. Figure 3 shows an example. The message moves logically as in Figure 3(a), but practical shortcut paths may be used as in Figure 3(b), which is possible because the forwarding nodes completely know the occupancy state of the adjacent neighbor cells.

This recovery technique is very simple but the message may fail to follow the boundary of the obstacle in specific topologies. In Figure 4(a), the message cannot move in the diagonal directions or jump over the empty cells, which result in a split of the network. To fix the problems, we can extend the *forwarding range* up to 20 adjacent neighbor cells, which covers all neighbor cells that the message can be directly transfered when $d_G = d_R / \sqrt{2}$. Then, the right-hand rule is applied to the centers of the occupied neighbor cells. However, the modified rules may produce undesired paths as in Figure 4(b), where node z sees C_{11} within its forwarding range and selects it

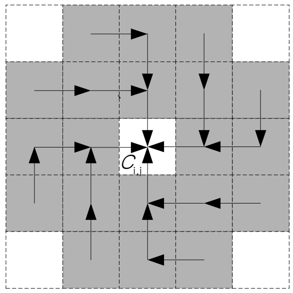


Figure 5. Message move patterns defined.

because it is the first cell counterclockwise from the previous cell C_{12}.

4.2.2 Grid RHR. We can fix the problems of the simple Grid RHR by 1) extending the forwarding range, and by 2) preventing the message from returning back to the old route.

1) In addition to the eight adjacent cells of $C_{i,j}$, the neighbor cells that are directly reachable from a node in $C_{i,j}$ are considered for message forwarding. All these neighbor cells define a *neighbor map* of $C_{i,j}$, which is a subset of the shaded region in Figure 5. This map contains all potential cells where the message came from and can be transferred. Thus, the message can move in any direction and jump over the empty cells.

2) The message remembers its explored path so that it cannot cross or return to the path later. The path is represented by a sequence of cells the message has passed. However, it is difficult to determine the passed cells because the message arbitrarily crosses over the grid cells. We tried to define the path as the sequence of the cells where the line between two centers of the previous and current cells intersects. The consequent path, however, often prohibits some potential next cells from being selected. For this reason, we define message move patterns as in Figure 5. For example, when a message is transferred from $C_{i+1, j-2}$ to $C_{i,j}$, it is considered to move through $C_{i+1, j-2}$, $C_{i, j-2}$, $C_{i, j-1}$, $C_{i,j}$, and this sequence is added to the explored path. The patterns are defined so that the path cannot exclude any potential forwarding cells. The defined patterns in Figure 5 are enough to describe any movements of the message because they cover all possible neighbor maps of $C_{i,j}$ when $d_G = d_R / \sqrt{2}$. If $d_G < d_R / \sqrt{2}$, new patterns should be defined for the increased set of potential neighbor cells.

In Grid RHR, the message is allowed to go back to the previous cell when it is the only path where the message can move. In this case, the old path is ignored because it may prevent the message from making progress, and the new path history begins at the node where the message turns back.

It is not necessary for the message to remember the entire explored path. The cells that are unreachable from the current cell do not affect the routing decision. Therefore, the message contains at most the latest cells corresponding to the shaded region in Figure 5.

After the message has explored the whole obstacle, it should be able to stop. For the stopping rule, we use the first two cells on the path. If the message returns to the first cell and is to be forwarded to the second cell on the path, then it stops the exploration. It means that the message has explored the whole circumference of the obstacle.

Grid RHR is summarized as follows. Given a message with its path history P, node x in C_X determines the next forwarding cell C_N by the following procedure:

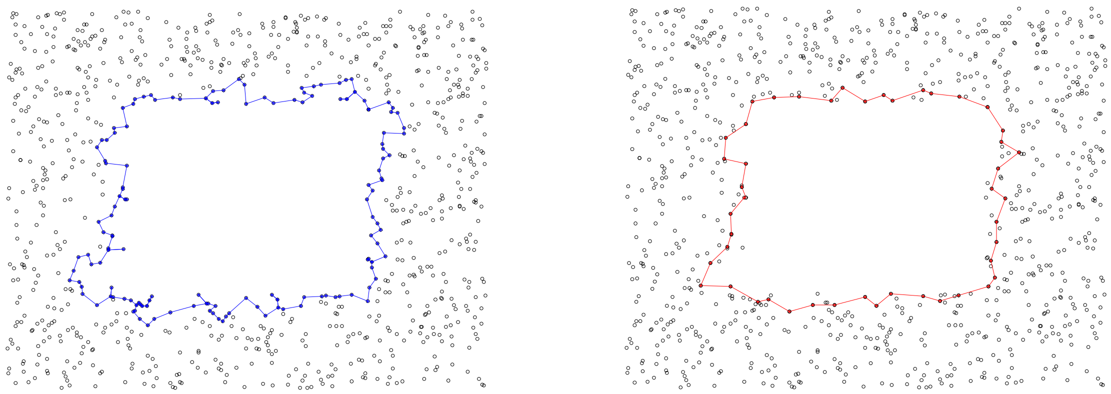
Step1. S=P, and a variable C_U is set to the latest cell on the path P.
Step2. Among the 8 adjacent cells, x determines the first cell C_V counterclockwise about C_X from C_U so that $\overline{C_X C_V}$ does not cross any cell in S.
Step3. If C_V has any node, $C_N = C_V$ and go to step 6.
Step4. If C_W exists that $\overline{C_X C_W}$ intersects C_V and not any cell in S, then $C_N = C_W$ and go to step 6.
Step5. $S = \varnothing$, $C_U = C_V$, and go to step 2.
Step6. If $C_N \in P$, $P = \varnothing$ and $P = \{C_X \to C_N\}$ else $P = Neighbor\_Map(P \cup \{C_X \to C_N\})$. Forward the message to C_N.

5. Simulation results

We design two experiments to test and evaluate grid-based techniques presented in the previous sections. Through these experiments, Grid RHR is compared with the well-known RNG planar graph based recovery algorithm (hereafter GPSR recovery algorithm). Some other recovery algorithms are qualitatively discussed. We analyze the simulation results mainly by two decisive metrics, the recovery path length and overhead needed to find the path. The message delivery success rate is another critical metric for the comparison. We confirm that both Grid RHR and GPSR recovery algorithm guarantee to find a recovery path in the connected static network.

For these simulations, we randomly deploy hundreds of nodes over the Columbia campus map, where buildings are permanent obstacles and temporary obstacles may occur too. To observe basic behaviors of the robotic routing techniques, the experiments are performed in the static network.

For the robotic routing rules, we assume that each node knows its neighbor map, which is defined in sec-



(a) GPSR recovery algorithm (161 hops) (b) Grid RHR (46 hops)

Figure 6. 900 nodes are deployed around a 500x500 square unit obstacle. The average number of neighbor nodes is 16. Grid RHR needs far fewer hops to go around the obstacle.

tion 4.2.2, as well as its single-hop neighbor nodes. The neighbor nodes are associated with the neighbor cells of the map. For GPSR, we assume that each node knows the physical position of itself and its single-hop neighbor nodes. Since the nodes in the network know their own physical position, the above assumptions for each protocol can be made at the same cost of "Hello" message exchanges. Therefore, it is not necessary to count those messages for the comparison, when we ignore a difference in the message length. For other assumptions, we follow the grid model presented in section 3.

5.1. Experiment 1

In this experiment, we measure how many hops recovery algorithms need to go around an obstacle randomly generated. The simulation results are compared by the ratio of the explored path length to the shortest boundary path in terms of hop count.

We experiment with 1000x1000 square unit networks where a permanent obstacle of 500x500 square units sits inside as in Figure 6. Nodes are randomly deployed around the obstacle so that the average number of neighbor nodes falls in between 3 and 20. A consequent obstacle may resemble the permanent obstacle, but most of the time the obstacle is more irregular and complex in the sparse network. The transmission radius d_R is set to 50 units and the grid size $d_G = d_R/\sqrt{2}$. Two messages leave from a node on the border and go around the obstacle by each recovery algorithm until they return to the start place.

The shortest boundary path, which is computed for a benchmark, is defined as the shortest perimeter that completely surrounds the obstacle and that has no nodes inside the perimeter. Each hop of the path should

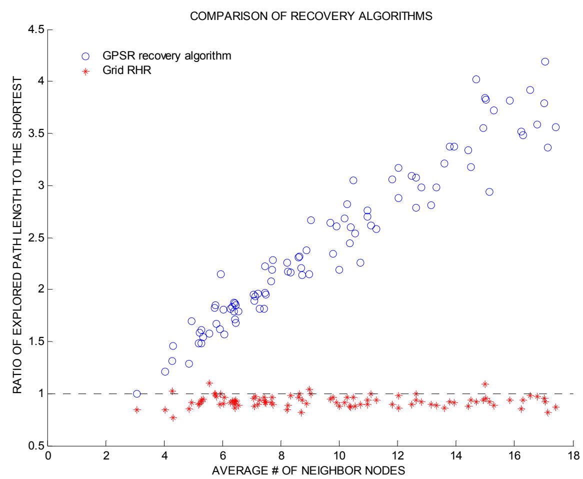


Figure 7. Comparison of the recovery path lengths

not be longer than d_R. Multiple shortest boundary paths may exist. We omit the shortest boundary path algorithm for lack of space.

Figure 7 says that most of the time, Grid RHR leads the message to circle the obstacle in a shorter path than the shortest boundary path. This is possible because the message is greedily forwarded from cell to cell by Cell-based forwarding rule, instead of stopping by every node on the boundary. This feature is also explained in Figure 6(b) where some nodes are found inside the path explored by Grid RHR, while the message in Figure 6(a) stops by every node in the vicinity of the border. As a result, Grid RHR uses far fewer messages to go around the obstacle than GPSR recovery algorithm. As seen in Figure 7, Grid RHR makes a stronger contrast with GPSR recovery algorithm as the average number of neighbors increases. At the densest simulation network, Grid RHR is about four times as

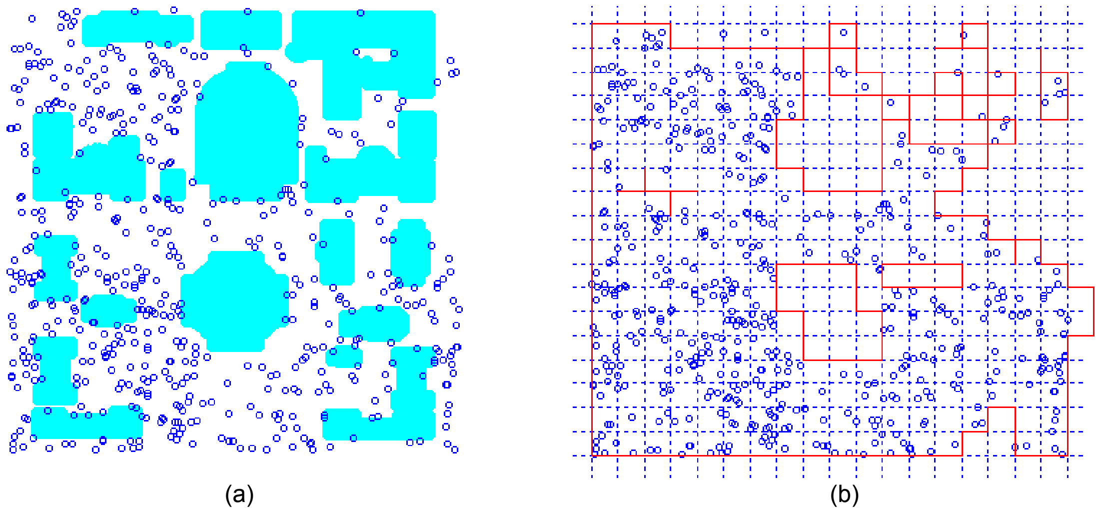


Figure 8. (a) 500 nodes are randomly distributed in the Columbia campus area. (b) Obstacles in the network are represented by grid edges.

efficient as GPSR recovery algorithm. One desirable feature of Grid RHR is that the recovery path length is proportional to the circumference of the obstacle, not the node density.

Grid RHR chooses hops with longer distances than GPSR recovery algorithm. It results in a shorter recovery path, but the longer transmission distances are more likely to be very weak in BER. Therefore, the Grid RHR messages are more likely to experience retransmissions than the GPSR messages in the real network under harsh wireless conditions.

As we expect, both Grid RHR and GPSR recovery algorithm guarantee to find a recovery path in the connected static network where no location errors occur. In the practical network, however, a certain level of location error is inevitable because the positioning system may not be that accurate, and nodes may move after sending a "Hello" message or the message may be lost. In Grid RHR, the location error may incur the wrong routing information about the neighbor cells. For example, an empty cell may be believed to be occupied, and vice versa. Nevertheless, an individual location error is not likely to directly affect the cell routing information because other nodes probably keep connections between the cells, especially in the dense network. Besides, the large-area cell covers a discrepancy in the location of nodes to some degree. The effect of location errors on Grid RHR will be quantitatively discussed in future.

Grid RHR can be compared to some other techniques that accomplish quite short recovery paths, such as Restricted Delaunay Graph (RDG) [11] and Localized Delaunay Triangulation (LDEL) [12], which use Delaunay triangulation graphs, or CLDP [21], which uses the right-hand rule. These techniques find shorter

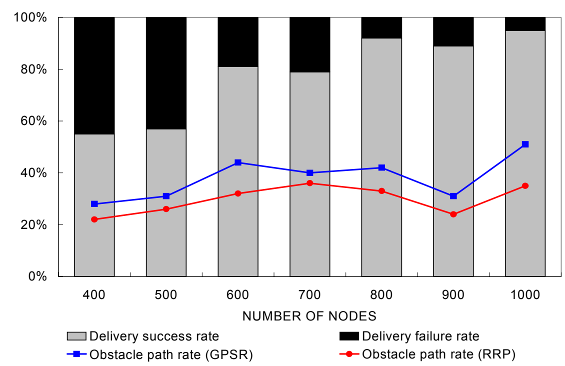
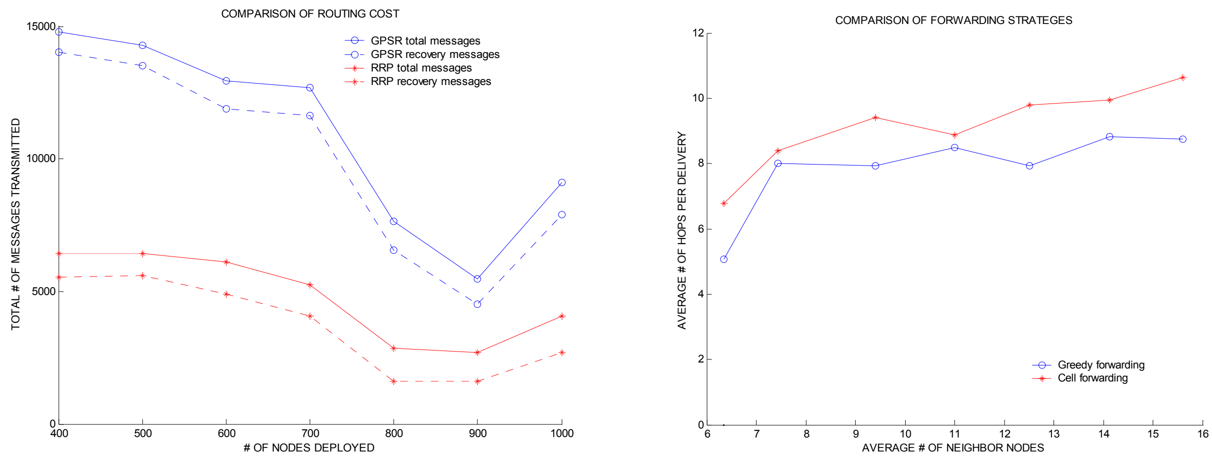


Figure 9. Message delivery rate and obstacle path rate

paths than GPSR recovery algorithm, but theoretically the path cannot be shorter than the shortest boundary path. Moreover, their protocol-based approaches bring communication overhead in the network. The overhead increases severely in the dense network where the constructed planar graph is rarely used. Note that Grid RHR causes no communication overhead, but it requires the message to contain a bit of the path information during the recovery phase.

5.2. Experiment 2

This experiment is more comprehensive than experiment 1. We implement two GFG-type protocols; GPSR, which combines Greedy forwarding with RNG planar graph based recovery algorithm, and RRP, which combines Cell-based forwarding with Grid RHR. We compare the routing cost of the two geographic routing protocols and investigate what largely affect the cost in the network with obstacles.



(a) (b)

Figure 10. (a) Comparison of the routing cost, (b) Comparison of forwarding strategies

The simulations are performed in the Columbia campus area. From 400 nodes to 1000 nodes are randomly distributed around the buildings as seen in Figure 8(a). The network size is 750x750 units, and d_R is set to 43 units so that d_G=30 units for convenience. Figure 8(b) shows the corresponding obstacles that the messages estimate during the recovery phase.

Two nodes are randomly chosen for a source and destination pair, then one node sends two messages to the other. One message is routed by GPSR and the other by RRP. We repeat this experiment extensively changing the network configuration every 20 messages. We also change the node density.

In Figure 9, GPSR and RRP show the same delivery success rate because their recovery algorithms guarantee the message delivery in the connected static network. The delivery failure occurs only when the source and destination are disconnected. The delivery success rate generally increases as the node density increases because more nodes get connected with the increased number of neighbor nodes. In the practical network, however, the message delivery is not guaranteed because of location errors and link losses.

Figure 10(a) compares the routing cost of the two protocols. It shows the total number of messages transmitted to deliver 100 messages in each network with the different node densities. It turns out that RRP is about two times as efficient as GPSR. The dotted curves represent the number of messages transmitted during the recovery mode. In the network with 400 nodes, 95% of the GPSR messages and 85% of the RRP messages are handled by each recovery algorithm. It means that the low routing cost of RRP is mainly due to Grid RHR. The routing cost of each protocol generally decreases as the node density increases because more nodes get connected and fewer messages go around obstacles to find a recovery path that does not exist. The sudden increase at 1000 nodes is due to the particular topology where the permanent and temporary obstacles form an irregular large obstacle, which is not likely to occur as the node density increases.

In the real networks, the routing cost curves will shift upward. GPSR messages are more likely to fail to get to the destination because GPSR is more sensitive to singular location errors than RRP. Meanwhile, RRP messages are more likely to experience retransmissions because RRP chooses hops with longer distances, which are weaker in BER.

If GPSR adopts the protocol-based improved recovery algorithm, such as LDEL or CLDP, its routing cost will be at least the sum of communication overhead and the routing cost required, assuming that it finds the shortest boundary path. Therefore, it is reasonably conjectured that the routing cost of the new GPSR is more than RRP's.

There is another simulation result in Figure 10(b) that compares Cell-based forwarding with position-based Greedy forwarding. In our simulation networks, Cell-based forwarding uses one or two more hops than Greedy forwarding, independent of the path length. However, the forwarding strategies hardly affect the overall efficiency of the routing protocol because they take a small portion of the total routing cost.

6. Discussion and future work

In our extensive simulations, RRP is proved to be more efficient than GPSR. Another interesting contrast between two protocols is that RRP is able to statistically distribute traffic loads among the nodes in a cell. RRP determines the next forwarding cell rather than a node, and any node in the cell can be used for the message forwarding. In contrast, GPSR protocol is deterministic about selecting the next forwarding node. If the source sends successive messages to the same destination, only some specific intermediate nodes will be heavily used. This may incur unbalanced power consumption.

One common problem with GFG-type routing protocols is that some messages explore the entire boundary of the outside obstacle in order to find a recovery path that does not exist. The situation may worsen because typical applications resend the message when there is no reply from the destination within a certain time. Therefore, if the message fails to find a path, the source needs to be informed of the result in order to prevent it from resending the message. Otherwise, the obstacle information may be spread over the network so that potential sources can decide whether the destination is reachable or not.

Another common problem is that the message cannot use the shortest path in the network with obstacles. This problem comes from the advantageous feature of geographic routing, that is, the local routing decision for the global message delivery. Without the global information about obstacles or links in the network, the message cannot avoid the obstacle in advance or make a wise decision when it is confronted by an obstacle.

To resolve these problems, the nodes can use the obstacle information. For example, if a source knows the shape and location of intervening obstacles between the destination and itself, the message can avoid the obstacles in advance by following some computed landmarks. Our on-going study includes this approach. We are designing a protocol that finds the geographic shortest path in the network with obstacles, by using the obstacle information.

7. Conclusion

In this paper, we have presented grid-based techniques for message routing in wireless ad hoc networks. We find that there is analogy between the message in the network and the mobile robot in a maze. Based on this connection, we have designed RRP that consists of two routing strategies, cell-based forwarding rules and grid-based recovery algorithms. RRP is less sensitive to location errors and requires less routing cost than the earlier geographic routing protocols. Our extensive simulations in the Columbia campus area show that RRP reduces the routing cost by about 50% when compared to GPSR.

Our novel network model makes it easy to represent and process the irregular and complex obstacles in the network. Future work will be focused on estimating the obstacles and the use of the information in order to further reduce the routing cost.

References

[1] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," In Proc. of the ACM SIGCOMM, Oct. 1994.

[2] D.B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, pp151-181, 1996

[3] C. Perkins, E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," In Proc. of IEEE WMCSA'99, pp90-100, 1999.

[4] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," In Proc. of ACM/IEEE MobiCom 2000, Aug. 2000.

[5] G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," ISI/RR-87-180, Mar. 1987.

[6] J. Li, J. Jannotti, D. Decouto, D. Karger, R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," In Proc. of ACM/IEEE MobiCom 2000, Aug. 2000.

[7] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," IEEE Commun., Vol 32, No 3, 1984.

[8] T. Hou, V. Li, "Transmission Range Control in Multihop Packet Radio Networks," IEEE Trans. Comm., Vol. 34, No. 1, 1986.

[9] E. Kranakis, H. Singh, J. Urrutia, "Compass Routing on Geometric Networks," In Proc. of the 11th Canadian Conf. on Computational Geometry (CCCG'99), Aug. 1999.

[10] F. Kuhn, R Wattenhofer, Y. Zhang, A. Zollinger, "Geometric Ad Hoc Routing: Of Theory and Practice," 22nd ACM Symposium on Principles of Distributed Computing, 2003.

[11] J. Gao, L. Guibas, J. Hershberger, L. Zhang, A. Zhu, "Geometric Spanner for Routing in Mobile Networks," In Proc. of ACM MobiHoc01, 2001.

[12] X. Li, G. Calinescu, P. Wan, Y. Wang, "Localized Delaunay Triangulation with Application in Wireless Ad Hoc Networks," IEEE Trans. Parallel and Distributed Sys., Vol. 14, No. 10, 2003.

[13] M. Heissenbuttel, T. Braun, "A Novel Position-based and Beacon-less Routing Algorithm for Mobile Ad-Hoc Networks," In Proc. of IEEE ASWN' 03, Bern, Switzerland, July 2003.

[14] I. Stojmenovic and X. Lin, "Loop-free Hybrid Single Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," IEEE Trans. Parallel and Distributed Sys., Vol. 12, No. 10, 2001.

[15] R. Jain, A. Puri, R. Sengupta, "Geographic Routing Using Partial Information for Wireless Ad Hoc Networks," IEEE Personal Communications, pp48-57, Feb. 2001.

[16] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, WA, Aug. 1999.

[17] F. Kuhn, R. Wattenhofer, A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," In Proc. of ACM MobiHoc03, 2003.

[18] S. Datta, I. Stojmenovic, J. Wu, "Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," In Proc. of IEEE Int. Conf. on Distributed Computing and Systems Workshops, Apr. 2001.

[19] R. Nelson, L. Kleinroch, "The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture," IEEE Trans. on Communications, Vol. 32, No. 6, Jun. 1984.

[20] A. Sankaranarayanan, M. Vidyasagar, "Path Planning for Moving a Point Object amidst Unknown Obstacles in a Plane: A New Algorithm and a General Theory for Algorithm Development," In Proc. of IEEE Conf. on Decision and Control, pp.1111-1119, Dec. 1990.

[21] Y. Kim, R. Govindan, B. Karp, S. Shenker, "Geographic Routing Made Practical," In Proc. of USENIX NSDI'05, 2005.

[22] K. Seada, A. Helmy, R. Govindan, "On the Effect of Localization Errors on Geographic Face Routing on Sensor Networks," In Proc. of IEEE/ACM IPSN'04, Apr. 2004.