WRGP: Weight-Aware Route Guiding Protocol for Wireless Sensor Networks with Obstacles

Chih-Yung Chang*, Wei-Cheng Ju*, Chao-Tsun Chang+, Yu-Chieh Chen*
*Tamkang University, Tamsui, Taiwan, +Hsiuping Institute Technology, Taichung, Taiwan
cychang@mail.tku.edu.tw, 491191291@s91.tku.edu.tw, cctas@mail.hit.edu.tw, 896410122@s96.tku.edu.tw

***Abstract*—The greedy forwarding routing protocol has been widely used for constructing a route with low control overheads in wireless sensor networks. However, its performance drops significantly when obstacles exist. This paper proposes a novel mechanism, named WRGP, which removes the impact of obstacles on the greedy forwarding routing. The proposed WRGP initially applies the previous research to specify the border nodes that surround the obstacle. Then the border nodes in the concave region of the obstacle initiate the weight assigning process and establish a forbidden region to prevent the packets from entering the concave region. Finally WRGP specifies some border nodes to act as the effective border nodes for constructing the optimal routes from themselves to the sink node. Comparing with the existing obstacles-resisting protocols, the proposed WRGP avoids the ping-pong effect and guides the packets moving along the shortest path from the encountered effective border node to the sink node. In addition, the M-WRGP is further developed to cope with the multi-obstacle problem. Simulation results show that both WRGP and M-WRGP outperform the existing protocol PAGER in terms of control overheads and average route length.**

Keywords-sensor networks; route guiding; protocol; obstacles.

I. INTRODUCTION

In a WSN, the greedy forwarding routing protocol has been widely applied for constructing routes between sensors and sink nodes. In the greedy forwarding routing protocol [1][3], a node delivers the message to the neighbor closest to the destination until the message is arrived at the destination node. With the good use of location information, the greedy forwarding routing creates a small amount of control packets and usually constructs an efficient route. However, obstacles can be dynamically formed in a WSN due to several reasons. First, the random deployment causes a non-uniform distribution of WSN where some region might not contain any sensor node. Second, some sensor nodes exhaust their energy after working for a relatively long period of time, resulting in some region without functionality of sensing and communication. Third, physical obstacle such as mountains or buildings also can be treated as an obstacle which potentially blocks the communications. Fourth, all sensor nodes in some region could not work normally owing to the animus signal interference. Finally, groups of animals-passing or strong breeze blowing also cause some sensor nodes failure or sweeping away, and hence dynamically form an obstacle. Since the greedy forwarding routing protocol did not take obstacles into consideration, packets that encounter a concave region of an obstacle will be propagated or blocked in the region where greedy forwarding is impossible.

In literature, many routing protocols that take obstacles into consideration have been addressed for the WSN. Depending on the timing that copes with the obstacle problem, these researches can be classified into passive and active categories. In the passive approaches [2][3], obstacle-resisting mechanism is involved in the design of routing protocols and will be adopted when the packets encounter an obstacle. However, there is no preprocessing for the obstacle as it forms and the obstacle information is usually not maintained by the WSN. Packets will be routed around the perimeter of the obstacle according to the rules predefined in the protocols to prevent the transmission of packet from blocking. Since the passive approaches lack for global obstacle information, packets can be routed to the destination successfully by an inefficient path rather than the shortest path.

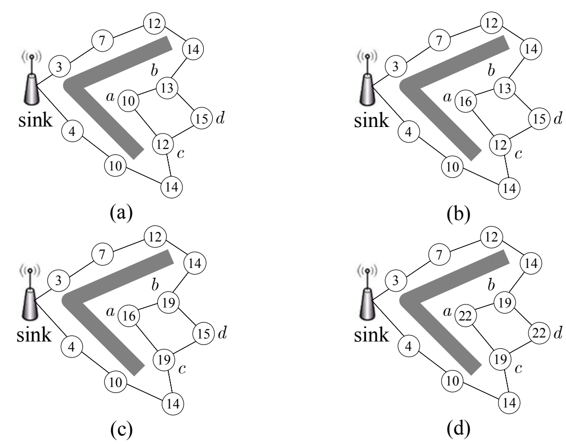


Fig. 1: An example of ping-pong effect in [4]. The number in each node represents the weight cost of that node. Node a changes its weight cost twice as shown from (a) to (d).

Alternatively, the active approaches [4] automatically detect and maintain the obstacle information. In PAGER[4], each node that locates outside the concave region is assigned with a cost in a distributed manner so that node closer to the sink node has a smaller cost. On the contrary, in the concave region, node closer to the sink node has a larger cost value. The cost-aware routing is achieved by each node forwarding the received packet to the neighbor with the lowest cost. As a result, the packets are prevented from entering a concave region where greedy forwarding is impossible. However, the weight-update procedure raises the ping-pong effect which creates significant control overheads, as shown in Fig. 1. In Fig. 1(a), since node a is closer to the sink node, its weight is smaller than weights of nodes b and c. To guarantee that sensing information can be delivered to the sink node, node a should change its weight to 16, as shown in Fig. 1(b). However, this operation raises a problem that nodes b and c can not deliver their information to the sink node since their weights are smaller than all neighbors' weights. Similar to node a, nodes b and c change their weights to 19 in a distributed manner, as shown in Fig. 1(c). This cause that node a again changes its weight since its weight is again smaller than those of nodes b and c. Consequently, the ping-pong effect for updating weight raises significant control overheads. In addition, packet routing based on the weight value might lead to an inefficient route and hence increase the energy consumption and end-to-end delay. Moreover, PAGER only considered the single obstacle environment. When there are multiple obstacles dynamically formed in the WSN, PAGER might lead to an inefficient route. Figure 2 depicts an example of the inefficient route by applying PAGER. In Fig. 2, the cost value of node b is smaller than that of node c because that l_2 is smaller than l_1. Since the routing policy in the cost-aware routing is to forward the received packet to the neighbor with the smallest cost, node a will forward the received packet to node b, which results in an inefficient route.

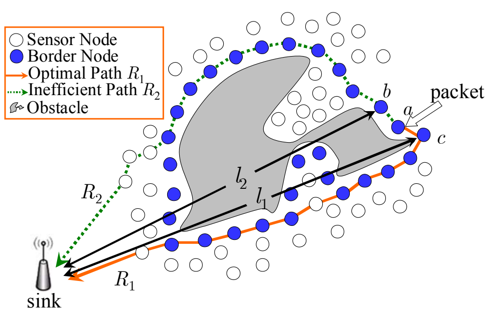


Fig. 2: The WRGP constructs an efficient route R_1 while PAGER [4] and construct an inefficient route R_2.

This paper proposes a Weight-aware Route Guiding Protocol (WRGP) that develops the weight-based guiding algorithm to resist the obstacle dynamically formed in the WSN. The proposed protocol actively establish a forbidden region by reassigning the weight value of each node inside the concave region of the obstacle. When packets arrives border nodes, WRGP guides the packets moving along the shortest path to the sink based on the preprocessed obstacle information.

II. WRGP: The Weight-aware Route Guiding Protocol

A. Network Environment and Problem Statement

In the present network model, all sensor nodes are stationary and each sensor node is aware of its own and neighbors' location information. Initially, each sensor node maintains a weight value which indicates the logical distance or cost for routing a packet from itself to the sink node [4]. The weight value increases with the distance between itself and the sink node. This implies that the greedy forwarding routing can be applied according to the node's weight. The packet sent from any sensor node to the sink node will be delivered to the neighbor with the smallest weight value. Herein, we aim to develop a route guiding protocol that detects and maintains the obstacle information as soon as the obstacle is formed. The developed guiding protocol intends to prevent the packets from entering the concave region and guide the packets to the best route when the greedy forwarding routing is failed.

B. The WRGP Protocol

The WRGP will be executed when the WSN has been initially deployed or whenever the neighboring information is significantly changed. In general, the WRGP mainly consists of the Concave Region Identification, Weight Modification, and Packet Guiding phases. Initially, the mechanism proposed in [5] for finding the stuck node can be utilized in our algorithm to identify the border nodes that surround the obstacle. Then WRGP executes the following three phases..

Phase I: Concave Region Identification Phase

This phase aims to identify the concave region which blocks the transmission when the greedy forwarding routing protocol is applied. Depending on the location of the sink node, the concave region might not impact the greedy forwarding routing. We define that an *effective concave node* is the border node that there exists no neighboring node closer to the sink node than itself. The concave region that contains at least one effective concave node is referred to *effective concave region*. The proposed WRGP will be only applied to the effective concave regions.

Phase II: Weight Modification Phase

This phase constructs the forbidden region by reassigning the weight value for each sensor node inside the effective concave region. Different from the pervious work [4], the weight reassigning process avoids the ping-pong effect, and thus saves significant energy consumptions. Let the original weight of any node u be denoted by $w(u)$ which is calculated according to the distance between node u and sink node. Let point p be the farthest point to the sink node in the effective concave region. That is, $w(p)$ has the largest value in the effective concave region. Here, we assume that each node in the effective concave region knows this value. The effective concave node v will reassign its weight with a new value $w'(v)$ according to expression (1).

$$w'(v) = 2w(p) - w(v) \quad (1)$$

After reassigning the new weight, node v abandons its border node role and notifies all neighbors about its new weight by broadcasting a Weight Modification packet (or *WM* for short). Upon receiving the *WM* packet, each sensor node, say x, that plays the border node role updates the sender's weight in its cache and further checks whether or not it is an effective concave node after removing the sender from its neighboring set. If it is the case, node x repeats the same operation what node v has done. Otherwise, node x plays the border node role. When node p receives the WM packet, the weight modification phase is terminated.

In the effective concave region, let the original weights of the sensor nodes $s_1, s_2, \ldots, s_{n\text{-}1}, s_x, p$ satisfy the condition $w(s_1) \leq w(s_2) \leq \ldots \leq w(s_{n\text{-}1}) \leq w(s_n) \leq w(p)$, where node p is the farthest point to the sink node. After the execution of this Phase, we have $p \leq w'(s_n) \leq \ldots \leq w'(s_2) \leq w'(s_1)$. This also indicates that the concave region has been constructed as a forbidden region which prevents data packet from entering the region. The progress of the modification implies that each sensor node in the effective concave region reassigns its weight exactly one time and avoids ping-pong effect.

Phase III: Packet Guiding Phase

This phase aims to guide the packets for overcoming the existing obstacle and therefore route the packet along the efficient path. Let the *Farthest Border Node* of a given obstacle be the border node that is farthest to the sink node and the node is denoted by notation b_{max}. Furthermore, let line L denote the line connecting sink node and node b_{max}. In each side of the line L, the *Critical Border Node* refers to the border node whose vertical distance to line L is largest. Notations c_n and c_s denote the two critical border nodes located at the north and south sides of line L, respectively. All border nodes that are lying on the path from c_n through b_{max} to c_s are called *Effective Border Nodes*.

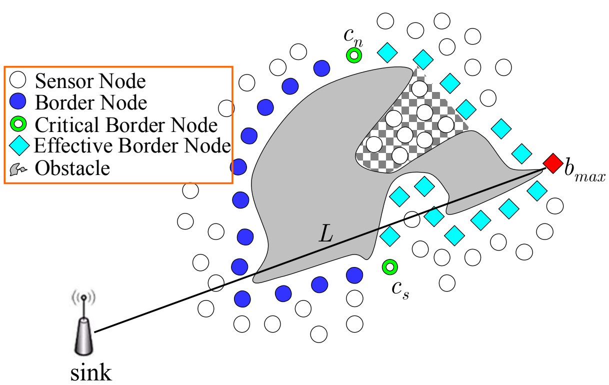


Fig. 3: An example of the farthest, critical, and effective border nodes.

Figure 3 is an example that illustrates the abovementioned three definitions. Note that one important property of the effective border node is that it might guide the packet to an inefficient route if the greedy forwarding routing is applied. In other words, all non-effective border nodes can forward the packet to a closer neighbor by applying the greedy forwarding routing. Therefore, in this phase, only the effective border nodes need to evaluate the cost of the two possible paths and hence derive a better route for packet guiding.

The identification of b_{max}, c_n and c_s can be achieved by initiating some exploiting packets that traverse all border nodes. Due to the limited of space, we omit these details. The most important task in this phase is that each effective border node locally evaluates the cost of the two possible routes and guides the packet to the best route when they receive data packets in the near future. Let $dist(a, b)$ and $hop(a, b)$ denote the distance and the number of hops between nodes a and b, respectively. Note that each effective border node exactly has two possible paths from itself to the sink node: one path passes through the critical border node c_n and the other one passes through the critical border node c_s. For each effective border node b, the cost for each path can be measured by the expression hop(b, *critical border node*) + hop(*critical border node*, *sink*). When every effective border node finishes the evaluation of its cost according to the above expression, the third phase will be terminated. Then the greedy forwarding can be applied in a manner that the packets will be moved along the shortest path even though the obstacle is encountered.

III. Multi-obstacle Route Guiding Protocol

This section considers the existence of multiple obstacles in the WSN and modifies the proposed WRGP to cope with the multiple obstacles problem. Here, we use the *S-WRGP* and the *M-WRGP* to denote the protocols of WRGP developed for single obstacle and multiple obstacles, respectively.

Figure 4 is an example for illustrating the inefficient route constructed by S-WRGP in a multi-obstacle environment. By applying the S-WRGP, the source sensor node located nearby the obstacle O_1 will guide the packet along the path R_3 for overcoming the obstacle O_1 since path R_3 is better than R_1. However, the packet along the path will encounter obstacle O_2 and then the S-WRGP is applied again. The S-WRGP further guides the packet along route R_2 to overcome the new obstacle O_2. In fact, if we consider the obstacles O_1 and O_2 together, the best route would be R_1. Since each effective border node only maintains the information of the nearby obstacle, S-WRGP can not look ahead the other obstacles and thus results in an inefficient route.

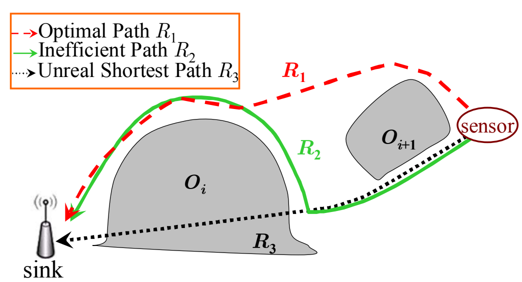
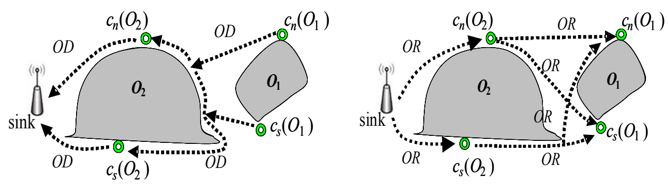


Fig. 4: An example that *S-WRGP* results in an inefficient route in the environment that contains multiple obstacles.

In the multi-obstacle environment, the information of new obstacles is required to be notified to all effective border nodes of those obstacles that are farther to the sink node for determining the shortest path. Flooding the information to all nodes is a simple way but will lead to significant control overhead. The *M-WRGP* aims to minimize the maintenance cost for new obstacles. The proposed *M-WRGP* uses the *Obstacle Discovery* (or *OD* for short) packet to discover the new forming obstacles which are closer to the sink node and uses the *Obstacle Reply* (or *OR* for short) packet to reply the detail information of the discovered obstacle.



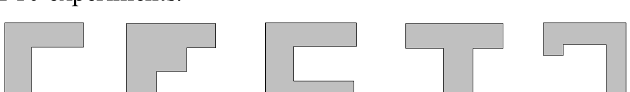
(a) (b)

Fig. 5: The new obstacle exploitation and the optimal path construction designed in M-WRGP. (a)The *OD* packets are initiated by the critical border nodes of O_1 for exploiting the obstacles closer to the sink node than O_1. (b) The *OR* packets are replied by the sink and the critical border nodes for modifying the weight value of effective border nodes of each obstacle.

In the following, we use obstacles O_1 and O_2 to illustrate the operations of M-WRGP where O_2 is closer to sink node than O_1. The critical border nodes of obstacles O_i are denoted by $c_s(O_i)$ and $c_n(O_i)$. Each node of $c_s(O_1)$ and $c_n(O_1)$ sends an *OD* packet to the sink for exploring all possible obstacles that are closer to sink than O_1. As shown in Fig. 5(a), the *OD* packet will encounter some effective border node of obstacle O_2. This effective border node further sends the *OD* packet to sink node by passing through the critical border nodes $c_s(O_2)$ or $c_n(O_2)$. As shown in Fig. 5(b), upon receiving the *OD* packet, the sink node creates an *OR* packet. The *OR* packet aims to collect information of $c_s(O_2)$ and $c_n(O_2)$ and send to the critical border nodes $c_s(O_1)$ and $c_n(O_1)$. Upon receiving the *OR* packet which contains the information of $c_s(O_2)$ and $c_n(O_2)$, the $c_s(O_1)$ or $c_n(O_1)$ nodes further forward the information to all effective border nodes of obstacle O_1. All effective border nodes will recalculate its optimal path from itself to the sink node according to the information of $c_s(O_1)$, $c_n(O_1)$, $c_s(O_2)$ and $c_n(O_2)$. This recalculation takes the obstacles O_1 and O_2 into consideration and hence is able to create an optimal path. Then the *M-WRGP* is finished.

IV. Performance Studies

This section examines the performance improvement of S-WRGP and M-WRGPagainst the existing work PAGER in terms of the average number of hops for routing and the control overheads for establishing the forbidden region. The simulator TinyVIZ is used to implement the compared protocols. The network size is 1000m×1000m. The number of sensors deployed in the WSN is a constant 500. Performance measures considered herein include the number of control packets for establishing the forbidden region and the average routing length. Each experimental result was obtained from the average results of 10 experiments.



(a) (b) (c) (d) (e)

Fig. 6: The shapes of the obstacles in our simulation environment.

Figure 6 shows the considered shapes of an obstacle with the same size. The communication range is controlled such that the average number of neighboring sensors is 10 and the sink node is arranged at the left-top corner. This environment is used to measure the performance results in Figs. 7 and 8. Figure 7 shows the impact of obstacle shapes on the control overhead and route efficiency of S-WRGP. The S-WRGP creates control overhead for constructing the forbidden region in a concave region. Therefore, the number of sensors located in the concave region will be the key factor that impacts the control overheads.

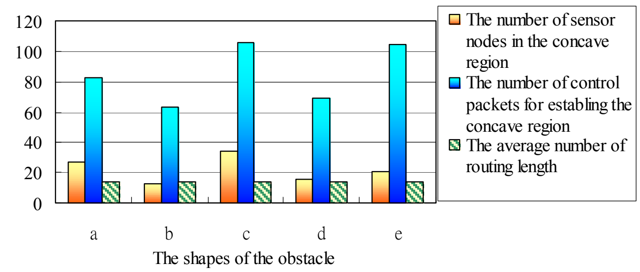


Fig. 7: The performances of S-WRGP examined in various shapes of the obstacle. Different shapes of the obstacle result in different numbers of sensors in the concave region and hence cause different control overheads.

Figure 7 depicts the number of sensors located in the concave region under different obstacle shapes. The shape type (c) has largest number of sensors while the shape type (b) has the least number of sensors in their concave regions. These results also impact the control overhead of S-WRGP as shown in Fig. 7. The control overhead increases with the number of sensors in that the concave region. As a result, the control overheads of obstacles with shape types (c) and (b) are maximal and minimal, respectively. This is because that all sensors in the concave region participates the operations of Phase III for constructing the forbidden region. Since the S-WRGP constructs the forbidden region to prevent the packet from entering concave region, the routing lengths of five obstacle shapes are similar as shown in Fig. 7.

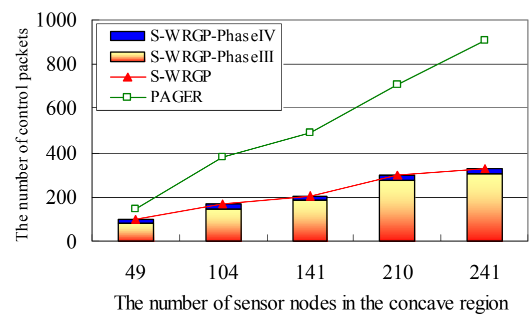
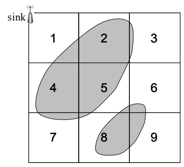
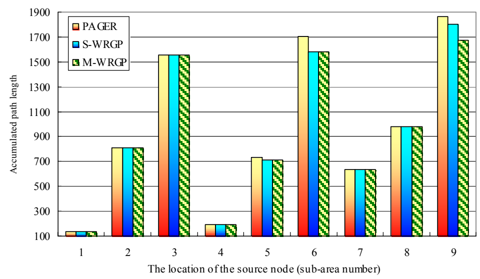


Fig. 8: Comparison of PAGER and S-WRGP in terms of the control overhead in the environment shown in Fig. 6(a).

Figure 8 compares the control overheads of PAGER and S-WRGP for establishing the forbidden region. The obstacle shape shown in Fig. 6(a) is applied and the number of sensors deployed in the concave region is varied from 49 to 241. Since S-WRGP avoids the ping-pong effect which is the major drawback in PAGER, the S-WRGP significantly outperforms PAGER in terms of the control overhead.

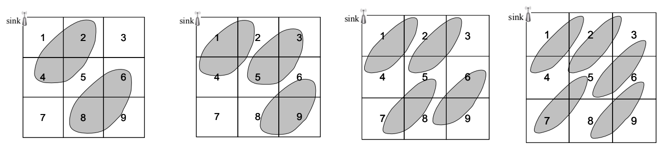
Figure 9 further investigates the impact of source location on accumulated routing length which represents the summation of all route lengths in each run. The obstacle environment shown in Fig. 9(a) is applied and the number of the neighboring nodes is set at ten. As shown in Fig. 9(b), compared with PAGER, the proposed M-WRGP has significant improvement when the source nodes are located in sub-areas six and nine. For example, if the source node located in the sub-area six initiates a packet, PAGER will forward the packet along the path passing through sub-areas 6, 5, 3, 2 and 1 while M-WRGP efficiently forwards the packet along the path passing through sub-areas 6, 3, 2 and 1.





(a) (b)

Fig. 9: (a)The deployment of two obstacles in our simulation environment. (b)Comparison of PAGER, S-WRGP and M-WRGP in terms of path length. The source location is varied at various sub-areas as shown in Fig. 9(a).



(a) (b) (c) (d)

Fig. 10: The obstacle distributions with various numbers of obstacles.

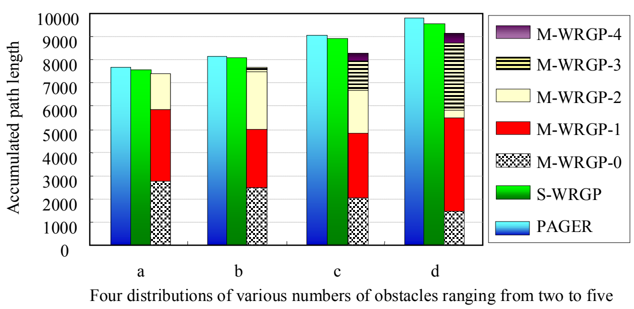


Fig. 11: Performance study of accumulated path length by varying the number of obstacles. The proposed M-WRGP significantly outperforms PAGER and S-WRGP in all cases.

The PAGER did not consider the multi-obstacle and thus constructs the forbidden region for each obstacle individually. The M-WRGP looks ahead the other obstacles in a multi-obstacle environment and helps to guide the packet along the shortest route. Increasing the number obstacles also increases the opportunity of packet encountering obstacles. Figure 10 depicts four distributions with the number of obstacles ranging from two to five. As show in Fig. 11, the obstacle distribution shown in Fig. 10(d) causes the largest accumulated path length since most packets encounter multiple obstacles and hence increase the routing length. Note that the legend M-WRGP-k denotes that k obstacles are encountered by applying the M-WRGP. For example, the accumulated path length of those packets that encounter one obstacle can be found in M-WRGP-1 in Fig. 11. In general, the M-WRGP outperforms PAGER in all cases.

V. CONCLUSIONS

In the WSN environment, obstacles might significantly drop the performance of existing location-aware routing protocols. Existing passive routing protocols can only move the packets away from the obstacles, but cannot guide the packet to the best route. This paper proposes a protocol that actively constructs the forbidden region for the concave region as soon as the obstacle is formed, avoids the ping-pong effect and reduces the control overheads. The selected effective border nodes not only prevent the packets from entering the forbidden regions but also guide the packets to the best route. The proposed S-WRGP actively tackles the obstacle problems, and thereby improves the communication efficiency for packet transmissions. In addition, the M-WRGP is developed to cope with the multi-obstacle problem. Simulation results show that both the S-WRGP and M-WRGP outperform PAGER in terms of control overhead and route length.

REFERENCES

[1] I. Stojmenovic and X. Lin, “GEDIR: Loop-Free Location Based Routing in Wireless Networks,” *International Conference on Parallel and Distributed Computing and Systems*, pp. 1025-1028, Nov. 1999.

[2] C. Y. Chang, C. T. Chang, S. C. Tu, “Obstacle-Free Geocasting Protocols for Single/Multi-Destination Short Message Services in Ad Hoc Networks,“ *ACM Wireless Networks*, vol. 9, no. 2, pp. 143-155, March 2003.

[3] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” *Mobicom 2000*, pp. 243-254, August 2000.

[4] L. Zou, M. Lu, and Z. Xiong, “A Distributed Algorithm for the Dead End Problem of Location Based Routing in Sensor Networks,” *TVT 2005*, vol. 54, no. 4, pp. 1509-1522, July 2005.

[5] Q. Fang, J. Gao, and L. J. Guibas, ”Locating and Bypassing Routing Holes in Sensor Networks,” *INFOCOM 2004*, vol. 4, pp. 2458-2468, March 2004