Active Route-Guiding Protocols for Resisting Obstacles in Wireless Sensor Networks

Chih-Yung Chang, *Associate Member, IEEE*, Chao-Tsun Chang, Yu-Chieh Chen, and Shih-Chieh Lee

***Abstract*—In wireless sensor networks, a geographic region without the functionality of sensing and communication can generally be treated as an obstacle, which significantly impacts the performance of existing location-based routing. An obstacle can dynamically be formed due to unbalanced deployment, sensor failure, or power exhaustion, animus interference, or physical obstacles such as mountains or buildings. This paper proposes novel algorithms that enable the existing location-based routing protocols that resist obstacles. Applying the proposed active route-guiding protocol for single obstacles (S-RGP), border nodes that surround the obstacles will actively establish a forbidden region for concave obstacles and make the obstacle information transparent. Then, packets will be guided to overcome the obstacle and move along the shorter path from the encountered border node to the sink node. In addition, the proposed active route-guiding protocol for multiple obstacles (M-RGP) takes multiple obstacles into consideration and integrates their information to help the packets overcome multiple obstacles. Simulation results show that the proposed S-RGP and M-RGP create low overhead and significantly reduce the average route length, and, therefore, improve the energy consumption and end-to-end delay for a wireless sensor network with obstacles.**

***Index Terms*—Data aggregation, geographic routing, location-aware routing, wireless sensor networks (WSNs).**

I. Introduction

Wireless sensor networks (WSNs) have a wide range of potential applications, including environmental monitoring, military operations, smart homes, and remote medical systems. A WSN comprises a sink node and an extremely large number of sensor nodes that communicate with each other to perform a broader sensing task. A sensor node is a tiny device that is capable of sensing, data processing and storing, and communication but is constrained in energy.

Recently, a much research [1], [2], [4], [5] has focused on establishing efficient routing paths from the sensor node to the sink node. Shortening the route length can help reduce the power consumption and delay time and increase the successful transmission rate. However, the existence of obstacles may block the establishment of a routing path or increase the route length. In general, a geographic region without the functionality of sensing and communication can be treated as an obstacle. In WSNs, obstacles can dynamically be formed due to several reasons. First, the random deployment causes a nonuniform distribution of WSNs, where some regions might not contain any sensor node. In addition, some sensor nodes exhaust their energy after working for a relative long period of time, and thus, some regions lose the functionality of sensing and communication. Furthermore, physical obstacles such as mountains or buildings can block the communications. Another reason for the forming of an obstacle is that all sensor nodes in some regions could not normally work due to the animus signal interference. Finally, groups of animals that pass a region or a strong breeze that blows the WSN cause some sensor nodes to fail or be swept away, dynamically forming an obstacle. After the appearance of the obstacles, packet routing will be blocked or inefficient.

A number of routing protocols that consider obstacles have been proposed for the WSN in the last few years. Depending on the timing for handling the obstacles, these works can be classified into passive and active categories. In passive approaches [3], there is no preprocessing for the obstacle when it is formed. Therefore, the obstacle information is not maintained by the WSN. However, the obstacle-handling mechanism is involved in the design of routing protocol and will be adopted when the packets encounter an obstacle. Packets will be routed around the perimeter of the obstacle according to the rules predefined in the protocols to prevent the transmission of packets from blocking. Because the passive approaches lack global obstacle information, packets can only successfully route to the destination rather than along the shorter path. Alternatively, the active approaches [1], [2], [4], [5], [9], [10] automatically detect and maintain the obstacle information. To minimize the impact of obstacles on existing routing protocols, obstacle detection [6]–[8] is an essential task. Sensors near the obstacle cooperatively maintain the obstacle information when the obstacle is formed. With the information of obstacle shape and size, the active approaches guide packets toward the shorter route and prevent the packets from entering a concave region, where greedy forwarding is impossible.

Because the existing active routing protocols do not consider the obstacle shape, the packets that encounter the obstacle might be guided to an inefficient route, which increases the energy consumption and end-to-end latency. In addition, the existing active protocols have mainly focused on how a packet can be guided to a better route from the single-obstacle point of view. However, the monitoring region might contain multiple

Manuscript received June 11, 2009; revised October 21, 2009, April 2, 2010 and July 31, 2010; accepted August 8, 2010. Date of publication August 19, 2010; date of current version November 12, 2010. The review of this paper was coordinated by Prof. H. Hassanein.

C.-Y. Chang and Y.-C. Chen are with the Department of Computer Science and Information Engineering, Tamkang University, Tamsui 25137, Taiwan (e-mail: cychang@mail.tku.edu.tw; ycchen@wireless.cs.tku.edu.tw).

C.-T. Chang is with the Department of Information Management, Hsiuping Institute of Technology, Taichung 41280, Taiwan (e-mail: cctas@mail.hit.edu.tw).

S.-C. Lee was with the Department of Computer Science and Information Engineering, Tamkang University, Tamsui 25137, Taiwan. He is now with Altek Corporation, Hsinchu 30078, Taiwan (e-mail: aaron@wireless.cs.tku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2068065

obstacles. Obstacles that are closer to the sink node will block the packets that have successfully overcome the obstacles that are farther the sink nodes. As a result, a large number of packets will encounter several obstacles, increasing the route length. How we can consider the obstacle shape and multiobstacle scenario and also guide the packets to efficient routes are important issues. This paper proposes distributed algorithms that consider the existence of obstacles in a WSN. Sensors around the obstacles will actively establish a forbidden-region boundary for concave regions in an offline manner and make the obstacle information transparent, preventing the packets from visiting a forbidden region. When packets are close to the concave obstacle, sensors around the obstacle would guide the packet to overcome the obstacles and move to the sink node along the better path. In addition, this paper considers the environment that contains multiple obstacles. The proposed route-guiding protocols that deal with the multiple obstacles can integrate the information of newly and previously formed obstacles and then dynamically guide the packets to the best routes.

The rest of this paper is organized as follows. Section II discusses the related work and the basic concepts of the proposed protocols. Sections III and IV present details of the proposed active route-guiding protocol for a single obstacle (S-RGP) and the active route-guiding protocol for multiple obstacles (M-RGP), which deal with single and multiple obstacles, respectively. Section V describes the analysis on the routing length of S-RGP and M-RGP. Section VI examines the performance of S-RGP and M-RGP compared with existing works. Finally, a summary of the proposed protocol and future works are drawn.

II. Related Work and Basic Concepts

This section initially reviews several active location-aware routing protocols that take obstacles into account. Then, the key concepts of the proposed protocols are presented. In the literature, some studies [1], [2] proposed face-routing protocols, named greedy–face–greedy (GFG) and greedy perimeter stateless routing (GPSR). GPSR [2] combines greedy forwarding and face routing to reduce the path length. Similar to GFG, GPSR initially constructs faces for a given WSN. Each forwarding node chooses the neighbor that is closest to the destination as the next hop forwarder. In case that the packet encounters the obstacle and the transmission is blocked in the concave region, the forwarding node will apply the right-hand rule (i.e., it will select the first counterclockwise neighbor as the next forwarder). Face-routing protocols such as GFG and GPSR can overcome the obstacle and successfully forward the packet to the destination. However, the routing path constructed by face routing might be inefficient due to the lack of global obstacle information. It is well known that GPSR has widely been applied in related studies and has been considered a fundamental operation for network routing. Therefore, an improvement of GPSR will have significant contributions, particularly for the existing works that apply GPSR as their main routing protocol.

Some other researchers developed weight-based routing algorithms [9], [10], actively maintaining the obstacle information and making it transparent to the border nodes that are geographically near the obstacle. Therefore, border nodes can avoid the packet that visits the concave regions of an obstacle. One previous work [9] proposed a distributed algorithm, called partial-partition-avoiding geographic routing (PAGER), to establish a shadow region for the concave obstacle. Each node outside the shadow region is assigned with a cost in a distributed manner so that the node closer to the sink node has a smaller cost. On the contrary, in the shadow region, the node closer to the sink node has a larger cost value. The cost-aware routing is achieved by each node that forwards the received packet to the neighbor with the lowest cost. Although this approach can actively overcome the concave obstacle, all shadow nodes are required to communicate with each other to set up their costs. In the worst case, all sensors in the shadow area need to regulate their costs when a sensor exhausts its energy. Therefore, a considerable control overhead is introduced for maintaining the nodes' costs.

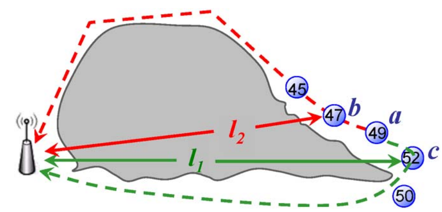


Fig. 1. Major disadvantage of PAGER. The packet can be guided to a long routing path.

Another major problem of PAGER is that the cost value maintained by each sensor only represents the distance between itself and the sink node. It cannot guarantee that packets move along the shorter path to the sink node. As shown in Fig. 1, the cost of node b is smaller than the cost of node c, because l_2 is smaller than l_1. Therefore, node a will forward the received packet to node b and, hence, construct an inefficient route. Similarly, [10] proposes a cost-aware routing protocol but extends with a cost-adapting phase to balance the difference of costs among the neighborhood sensors, preventing the packet from visiting the concave region of an obstacle. However, [10] has the same drawbacks of PAGER.

To cope with the aforementioned problems, this paper proposes the route-guiding protocols S-RGP and M-RGP, which aim at maintaining the obstacle information and guiding the packets to the shorter path. The basic idea of the S-RGP is given as follows. The S-RGP mainly consists of four phases. First, the obstacle detection phase aims at locally selecting a few sensors to serve as the border nodes. The selected border nodes guarantee that they are connected and surround the obstacle. Second, the normalization phase further selects some border nodes that serve as the corner nodes to regularize the obstacle shape. Fig. 2 depicts the selected border and corner nodes. Third, the forbidden-region construction phase employs these corner nodes to collaboratively establish the convex forbidden region to prevent the packets from visiting the concave region. As shown in Fig. 3, the shadow regions are the constructed forbidden regions. Last, the packet-guiding phase guides the packets to the shorter path. As shown in Fig. 3, the packet will be guided to move along the shorter path $R1$, rather than path $R2$, to detour the obstacle. As a result, the S-RGP helps reduce

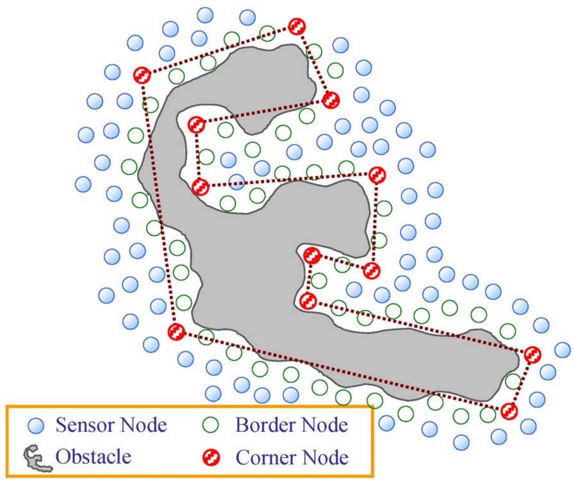


Fig. 2. S-RGP selects the border and corner nodes in the first two phases.

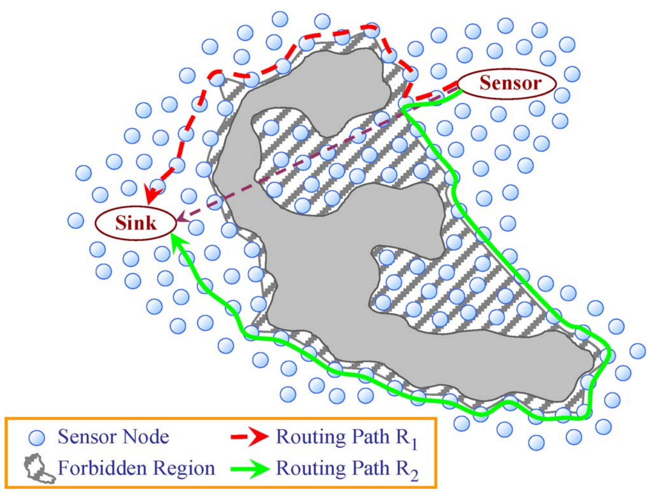


Fig. 3. S-RGP constructs a forbidden region for concave regions and guides packets to the shorter path.

the route length and improves the network performance in terms of power consumption and end-to-end delay. In addition to overcoming the single obstacle, the M-RGP is proposed to integrate the information of multiple obstacles and guide the packets to the shorter route.

III. S-RGP: Active Route-Guiding Protocol for a Single Obstacle

In the present network model, all sensor nodes are stationary. Each sensor node is aware of its own location information based on either a low-power Global Positioning System (GPS) receiver or localization methods [11] and exchanges location information with one-hop neighbors through beacons. Sensors are synchronized using the GPS or other mechanisms such as time-stamp exchange [12]. The occurrence of an obstacle is unpredictable, and the change of signal to noise is ignored. Details of S-RGP are described as follows.

S-RGP will be executed as the network has initially been deployed or the neighboring information has largely been changed. In general, S-RGP mainly consists of the obstacle detection, normalization, forbidden-region construction, and packet-guiding phases.

A. Phase 1: Obstacle Detection

This phase aims at locally selecting some sensor nodes near the obstacle to serve as border nodes. There should be as few selected border nodes as possible, but they should be connected and surrounding the obstacle. To completely represent the obstacle shape, the selected border nodes for an obstacle should satisfy the following two properties.

(P1) ***Surrounding property.*** The border nodes are connected and can surround the obstacle. In other words, the area formed by linking the selected border nodes has to cover the whole obstacle.

(P2) ***Minimal property.*** The selected border node for a given obstacle should be as few as possible.

Fewer border nodes can reduce the computation and communication overhead in executing the S-RGP task. However, an unlimited reduction in the number of border nodes may not capture the whole obstacle information. Properties (P1) and (P2) aim to set up a proper number of border nodes.

In this phase, a few sensor nodes that surround the obstacle are selected as the border nodes. We propose the following distributed border selection rules for each sensor node. First, each node s locally constructs its Voronoi diagram. The Voronoi diagram [15] is used to partition the field into many polygons. The partitioned polygon contains exactly one sensor. Every point in a given polygon is closer to the sensor located in it than to any other. The simple procedure to construct the Voronoi Diagram is given as follows. Initially, each sensor and its one-hop neighbors exchange their location information. Then, each sensor computes the medial axis of each neighbor according to the exchanged information. In the end, each sensor can get a polygon that is surrounded by the calculated media axis for its one-hop neighbors. Then, the Voronoi neighbors of each node, e.g., s, are ordered counterclockwise and denoted as an ordered list $n_1, n_2, \ldots, n_k$, where k is the number of Voronoi neighbors of node s. Let symbol $v^s_{n_i,n_{i+1}}$ denote the Voronoi vertex of s constructed by s and its two adjacent Voronoi neighbors n_i and n_{i+1}. Recall that the obstacle region is formed for two major reasons: 1) the sparse deployment, so that there is no forwarding node closer to the sink node than itself, and 2) the physical obstacle or animus signal interference. If node s is located near the obstacle region, it will serve as the border node. Therefore, each node s uses the constructed Voronoi diagram to check if there is any adjacent Voronoi neighbor that satisfies the aforementioned two cases.

We propose check rules $R1$ and $R2$ to determine the border node in a distributed manner. $R1$ is mainly designed for finding the border node that is located near the concave region of the obstacle, whereas $R2$ is designed for identifying the border node that is located near the convex region of the obstacle.

$$\textbf{\textit{Check rule R1}}: \textit{Node } s \textit{ should serve as a border node if it has any Voronoi edge with infinite length.} \quad \square$$

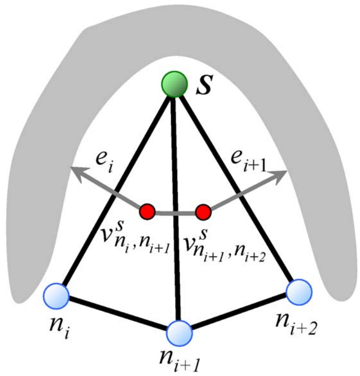


Fig. 4. Node s is a border node because it has two Voronoi edges e_i and e_{i+1} with infinite length.

Fig. 4 shows an example that illustrates the concept behind $R1$. Because node s has two Voronoi edges e_i and e_{i+1} with infinite length, node s would play a border-node role. The reason is described as follows. Because node s has an edge with infinite length, node s cannot construct a Voronoi cell, and hence, it is implied that s is located near an obstacle or a coverage hole. The following discussion further explains that node s is located near the concave region of the obstacle. As shown in Fig. 4, the Voronoi neighbors of s are n_i, n_{i+1}, and n_{i+2}. Node s is located outside the triangular constructed by these three Voronoi neighbors. This case also implies that all Voronoi neighbors of s are located at the same side of s, which verifies that there is no Voronoi neighbor located at the other side of s. Hence, s is located near the concave region of the obstacle.

$R2$ intends to use a Voronoi vertex to detect both the hole and the physical obstacle, i.e.,

$$\textbf{\textit{Check rule R2}}: \textit{Node s should serve as a border node if any Voronoi vertex } v^s_{n_i,n_{i+1}} \textit{ of s satisfies that } v^{n_i}_{s,n_i+1} \textit{ does not belong to } n_i \textit{ or } v^{n_{i+1}}_{s,n_i} \textit{does not belong to } n_{i+1}. \quad \square$$

The case that n_i cannot construct $v^{n_i}_{s,n_{i+1}}$ implies that n_i cannot communicate with n_{i+1}, which might be due to the following two reasons. One reason is that the distance between n_i and n_{i+1} is larger than the communication range. In this case, there is a coverage hole between n_i and n_{i+1}. Because the hole is also treated as an obstacle, it is implied that node s is located near the obstacle. The other reason is that the distance between n_i and n_{i+1} is smaller than the communication range, but a physical obstacle blocks their communication. This case also implies that node s is located near the physical obstacle. As a result, node s should serve as a border node.

Consider Fig. 5(a). A physical obstacle blocks the communication of nodes n_i and n_{i+1}. Therefore, nodes n_i and n_{i+1} do not have Voronoi vertices $v^{n_i}_{s,n_{i+1}}$ and $v^{n_{i+1}}_{s,n_i}$. Applying $R2$, node s will locally identify that it is a border node. Fig. 5(b) and (c) shows two examples to further examine $R2$. Let the communication range of any two sensor nodes be r. In Fig. 5(b), we assume that the communication range is twice the sensing range. In this example, there exists a small sensing hole that does not block the communication between n_1 and n_2. Hence, node s should not serve as a border node. By applying $R2$, sensor s will not serve as a border node because none of its Voronoi neighbors satisfies $R2$. In Fig. 5(c), nodes s, n_1, n_2, and p are within each other's communication range, and a physical obstacle blocks the communication between s and p and between n_1 and n_2. Let v and v' denote the Voronoi vertices of s and n_2, respectively. Because the Voronoi vertex v of s is constructed by s, n_1, and n_2, we have $v = v^s_{n_1,n_2}$. Then, we will check whether vertex $v^{n_2}_{s,n_1}$ belongs to n_2. If vertex $v^{n_2}_{s,n_1}$ does not belong to n_2, s satisfies $R2$ and should serve as a border node. Because the Voronoi vertex v' of n_2 is constructed by n_2, s, and p, we have $v' = v^{n_2}_{s,p}$. Because $v^{n_2}_{s,n_1}$ and $v^{n_2}_{s,p}$ are different in the second subscript, $v^{n_2}_{s,n_1}$ does not belong to n_2, and $R2$ holds. As a result, s will serve as a border node, which again verifies that $R2$ works in this case.

Lemma 1 and Theorem 1 present that the border nodes selected by the proposed Rules are connected and have the surrounding property.

Lemma 1: If s is a border node, at least two Voronoi neighbors of s serve as border nodes.

Proof: The proof is by contradiction. Assume that all Voronoi neighbors of s are not border nodes; hence, there does not exist any Voronoi neighbor that satisfies $R2$. Because s is a border node, s has to satisfy $R2$. Hence, there must exist two Voronoi neighbors, e.g., n_i and n_{i+1}, such that the distance of n_i and n_{i+1} is larger than the transmission range or there exists some obstacle between n_i and n_{i+1}. Because n_i and n_{i+1} are adjacent to the obstacle, n_i and n_{i+1} will identify themselves as border nodes by applying $R2$. Hence, two Voronoi neighbors of s serve as border nodes. ■

Theorem 1: The border nodes selected by the proposed rule are connected and can surround the obstacle.

Proof: Because the sensor nodes around the obstacles are connected, by applying check rules $R1$ and $R2$, sensor nodes around the obstacles will serve as border nodes. According to Lemma 1, each border node has two neighboring border nodes, and these border nodes are selected from the connected sensor node set around the obstacle. Hence, these selected border nodes are connected. ■

By applying check rules $R1$ and $R2$, any sensor node, e.g., s, can determinate whether it is a border by constructing the Voronoi diagram and collecting the Voronoi vertex sets of its Voronoi neighbors.

B. Phase 2: Normalization

This phase further selects some border nodes to serve as the corner nodes for identifying the obstacle shape. Herein, some representative border nodes that can outline the obstacle's shape are called *corner nodes*. The corner nodes are denoted by c_i, and the nearest corner nodes at the left- and right-hand sides of c_i are called the neighboring corner nodes of c_i and are denoted as c_{i+1} and c_{i-1}, respectively.

To capture the obstacle shape, a border node should play the role of a corner node at the location that the obstacle shape has a significant change. We define the turning and accumulated angles to measure the shape variation of a given border node as follows.

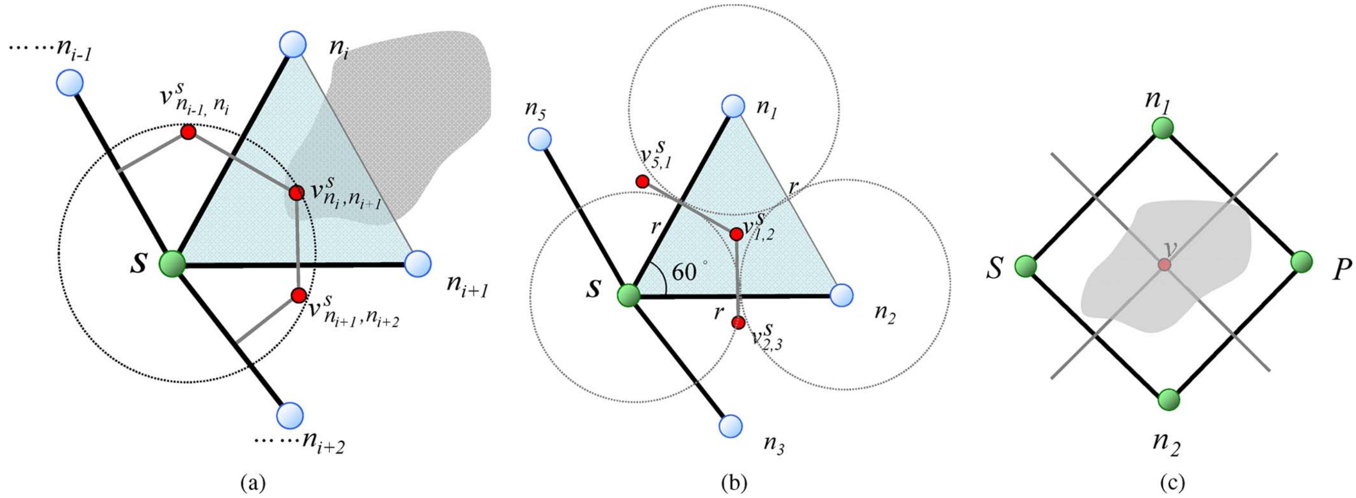


Fig. 5. Examples for illustrating *R*2. (a) Node s serves as a border node because a physical obstacle blocks the communication of nodes n_i and n_{i+1}. (b) Node s is not a border node because the small sensing hole does not block the communication between n_1 and n_2. (c) Node s is a border node because the physical obstacle blocks the communication between n_1 and n_2.

Definition—Turning Angle: For a given border node b_i, the *turning angle* β_i is defined as the angle that sweeps from b_{i-1} to b_{i+1} counterclockwise. That is, $\beta_i = \angle b_{i-1}b_ib_{i+1} - \pi$.

Definition—Accumulated Angle: The accumulate angle $\beta_{i,j}$ of a border node b_j is the sum of the turning angles of a set of successive border nodes $b_i, b_{i+1}, \ldots, b_j$. That is, $\beta_{i,j} = \sum_{y=i}^{j} \beta_y$.

If $\angle b_{i-1}b_ib_{i+1}$ is smaller than 180°, β_i is positive. Otherwise, β_i is negative. Two types of corner nodes are identified in this phase. The first type of corner nodes aims to capture the significant shape change due to the concave or convex shape. If the turning angle of the border node b_i is larger than 180°, node b_i is located beside the convex shape and should serve as a corner node. If the two neighboring border nodes of b_i are within each other's communication range but cannot communicate with each other, node b_i is beside the concave shape and should serve as a corner node.

Another type of corner node aims at capturing the significant shape change due to the accumulation of several small shape changes. To determine the second-type corner node, each first-type corner node, e.g., c_i, sends rightward an *accumulated angle packet* to inform the neighboring corner node c_{i+1} in a multihop manner. For each border node b_j that lies on the path from c_i to c_{i+1}, the accumulated angle is computed and compared with a threshold β. If the accumulated angle $\beta_{i,j}$ of node b_j is larger than β, node b_j becomes a corner node. As shown in Fig. 6, the corner node B transmits an accumulated angle packet to the corner node D. Upon receiving the accumulated angle packet from the border node R, the border node S computes the accumulated angle R^0 and then compares with the threshold β. Because the angle R^0 is smaller than β, the border node S does not become a corner node. Then, the border node S continuously transmits the accumulated packet that contains the accumulated angle R^0 to the border node C. Similarly, the border node C computes the accumulated angle, and hence, the accumulated angle is equal to $R^0 + S^0$. Because the accumulated angle is larger than the threshold β, the border node C servers as a corner node. When a new second-type

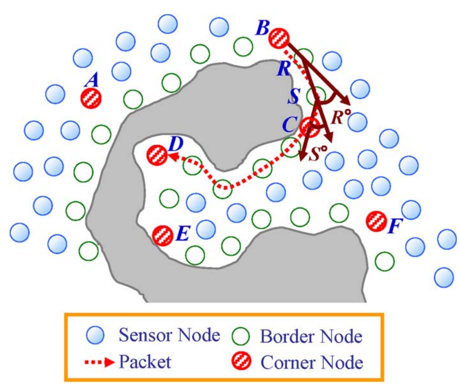


Fig. 6. Example of finding the corner node based on the accumulated angle.

corner node is found, the accumulated angle is reset to zero, and the new corner node initiates a new *accumulated angle packet* to c_{i+1} to find another second-type corner node and sends an acknowledgment (ACK) packet to the corner node that initials the accumulated packet in a multihop manner. The procedure of finding second-type corner nodes will be terminated when c_{i+1} receives the accumulated angle packet sent from c_i.

C. Phase 3: Forbidden-Region Construction

In this phase, the forbidden region is established to prevent the packets from visiting the concave region. After waiting for a random time period, corner node c_i informs c_{i-1} to construct a route to c_{i+1} by using a greedy forwarding routing protocol [1]. In the meantime, c_i, c_{i-1}, and c_{i+1} will be locked to keep them from constructing more than one forbidden region at a time. The constructed route will be treated as the new boundary of the forbidden region. Then, nodes c_{i-1} and c_{i+1} will update the information of their neighboring corner nodes from $\{c_{i-2}, c_i\}$ and $\{c_i, c_{i+2}\}$ to $\{c_{i-2}, c_{i+1}\}$ and $\{c_{i-1}, c_{i+2}\}$, respectively. Then, c_{i-1} and c_{i+1} repeat the forbidden-region construction procedure until both the neighboring corner nodes of each c_i no longer change. If another corner node is in the route from c_{i-1} to c_{i+1}, this run of forbidden-region construction will be

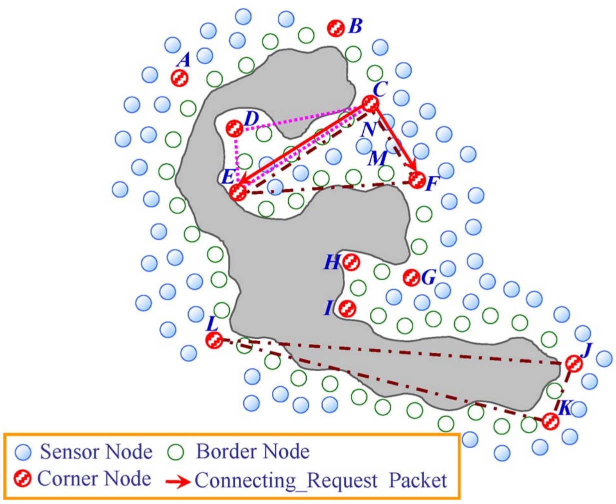


Fig. 7. Example of forbidden-region construction.

terminated. Finally, all sensor nodes on the forbidden-region boundary will play the role of the *forbidden-region boundary node*.

Fig. 7 shows an example for constructing the forbidden region. After waiting for a random time period, corner node D notifies its neighboring corner node C to construct a route to corner node E using the greedy forwarding routing protocol. When the route has been constructed, corner nodes C and E update the information of their neighboring corner nodes from $\{B, D\}$ and $\{D, F\}$ to $\{B, E\}$ and $\{C, F\}$, respectively. Then, node E repeats the same procedure, asking neighboring corner node C to construct a route to F. Afterward, a new forbidden-region boundary from C to F is constructed, and the neighboring corner nodes for nodes C and F are $\{B, F\}$ and $\{C, G\}$, respectively. Finally, sensor nodes N and M become forbidden-region boundary nodes. At the same time, corner node K starts the procedure and notifies J to construct a route to L. During the route construction, the routing path passes through node K, terminating the route-construction procedure, because corner nodes J, K, and L are not located at the concave region.

D. Phase 4: Packet Guiding

In this phase, each border node or forbidden-region boundary node will locally derive the best route from itself to the sink node for packet guiding in the future. At the beginning of this phase, each corner node determines whether it is one of the extreme east, west, south, and north corner nodes by comparing its location with the location of two neighboring corner nodes. Corner nodes at the extreme east, west, south, or north locations are called *extreme corner nodes* and are denoted C_E, C_W, C_S, and C_N. The local maximal problem will not occur while each corner node compares its location with the locations of its two neighboring corner nodes because the concave regions in the obstacle have been transformed to the convex forbidden region in the previous phase.

Then, C_N sends a *guide probe packet* that traverses all border nodes and forbidden-region boundary nodes. The guide probe packet consists of fields, including the sender ID, hop count, east node, west node, and south node. The initial values of the sender ID, hop count, east node, west node, and south node fields in the guide probe packet are C_N, 0, null, null, and null, respectively. When the packet traverses a border node or a forbidden-region boundary node, the hop count value is increased by 1. When C_E, C_W, or C_S receives the guide probe packet, the information {ID, location, hop count} of extreme corner nodes C_E, C_W, and C_S will be recorded in the east node, west node, and south node fields, respectively. When the guide probe packet arrives at node C_N, the packet transmission will be stopped. Next, C_N will send the collected information of the extreme corner nodes to all border nodes and forbidden-region boundary nodes. Each border node or forbidden-region boundary node can further use this information to derive the best route from itself to the sink node and can therefore guide the packet to the best route.

Each border node or boundary node should make a decision for the packet-forwarding direction, which can be either left or right. Because each border node or forbidden-region boundary node knows the closest left and right extreme corner nodes, they can evaluate the distance between each of the two closest extreme corner nodes and the sink node and the distance between itself and each of the two closest extreme corner nodes.

Suppose that the location of the sink node X is (x, y) and the locations of C_W, C_E, C_S, and C_N are (x_W, y_W), (x_E, y_E), (x_S, y_S), and (x_N, y_N), respectively. The next step in this phase aims at finding the two extreme corner nodes that are closest to the sink node from the left and right directions of every border node and forbidden-region boundary node. First, a point will be selected as the reference point within the area surrounded by the four extreme corner nodes. The simplest way is to select the gravity, e.g., Z, of the four extreme corner nodes as the reference point, whose location can easily be derived by

$$Z = ((x_W + x_E + x_S + x_N +)/4, \quad (x_W + x_E + x_S + x_N +)/4). \quad (1)$$

Connecting the reference point to each of the four extreme corner nodes can partition the whole WSN region into four disjoint subregions. Assume that the sink node falls in the subregion spanned by the angle $\angle C_A Z C_B$, where C_A and C_B are two adjacent extreme corner nodes, and Z is the gravity. The two extreme corner nodes C_A and C_B are defined as the *end corner nodes*. The end corner nodes C_A and C_B can be obtained from $\{C_W, C_E, C_S, C_N\}$ by checking whether the two adjacent extreme corner nodes satisfy the following formula:

$$\angle C_A Z X + \angle C_B Z X = \angle C_A Z C_B. \quad (2)$$

Therefore, nodes C_A and C_B are the nearest two corner nodes to the sink node from the left and right routing directions. The best route from any border node or forbidden-region boundary node, e.g., b, to the sink node would be either $b \rightarrow C_A \rightarrow X$ or $b \rightarrow C_B \rightarrow X$. When the best route has been derived, the packet-forwarding direction can be determined. To derive the best route, node b individually calculates the hop count from itself to each of the nodes C_A and C_B and then evaluates the distance between each end corner node and the

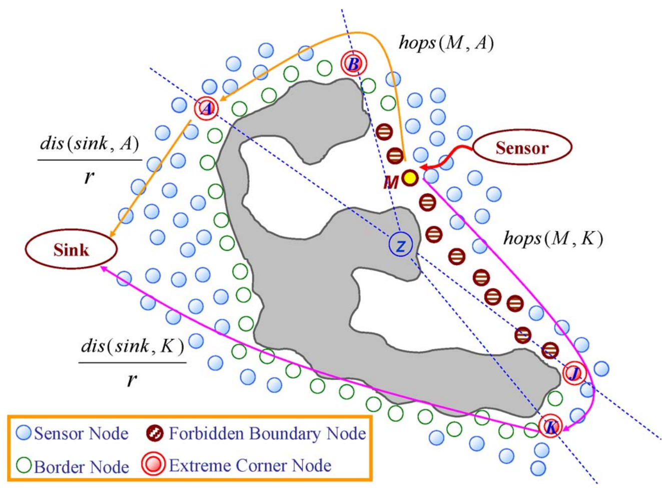


Fig. 8. In the packet-guiding phase, S-RGP uses reference point Z and four extremely corner nodes to derive the best route and the packet-forwarding direction.

sink node. Finally, node b selects one of the two end corner nodes as the intermediate destination of the best route and determines the best forwarding direction from the candidates of the left and right directions. The following formula helps node b select the intermediate destination node from nodes C_A and C_B:

$$Minimum\left(hops(b, c_i) + \frac{dis(sink, c_i)}{r},\right.$$
$$\left. hops(b, c_j) + \frac{dis(\text{sink}, c_j)}{r}\right) \quad (3)$$

where $dis(P, Q)$ denotes the distance between sensor nodes P and Q, and r denotes the communication range of each sensor node.

Fig. 8 depicts an example of this phase. Assume that point Z is the gravity of the extreme corner nodes B, J, K, and A. Because the sink node is located within the subregion spanned by the angle $\angle AZK$, nodes A and K will be the end corner nodes. Any border or boundary node M can calculate the hop count from itself to end corner nodes A and K and the distance from each of the end corner nodes A and K to the sink node. By applying (3), node M determines that corner node A is the intermediate destination of the best route and that the right direction is the data-forwarding direction. In the future, when node M receives any data packet, it can guide the packet to node A by forwarding the packet to the right neighboring border node. As a result, every border node or forbidden-region boundary node can guide the data packet to the best route.

IV. Route-Guiding Protocol for Multiple Obstacles (M-RGP) Active Route-Guiding Protocol for Multiple Obstacles

This section considers the existence of multiple obstacles in the given WSN. If more than one obstacle is dynamically formed in the WSN, the proposed S-RGP, which deals with a single obstacle, might lead to a longer route. Here, we use M-RGP to denote the developed route-guiding protocol for multiple obstacles. Fig. 9 shows an example of the impact of multiple obstacles on the route construction. If there is only one obstacle O_i in the WSN, the proposed S-RGP might choose routing path R_2, which is the shorter path from the source sensor to the sink node. However, if one more obstacle O_{i+1} is dynamically formed in the WSN, the data packet will encounter obstacle O_{i+1}, and the proposed S-RGP might locally guide the data packet to route R_4. However, by integrating the two obstacles, the best route is R_1. Because each border node only maintains the information of a single obstacle, the route guiding cannot look ahead of the other obstacles and, thus, results in an inefficient route.

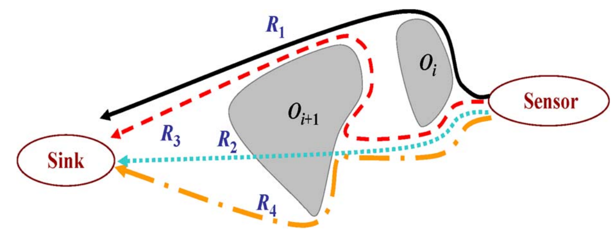


Fig. 9. Example that S-RGP cannot guide the packet to the best route.

The following notations help in presenting the details of M-RGP:

O_i	obstacle (a smaller i represents the obstacle closer to the sink node);
$C_N(O_i)$	north extreme corner node of the obstacle O_i;
$C_S(O_i)$	south extreme corner node of the obstacle O_i;
$C_E(O_i)$	east extreme corner node of the obstacle O_i.
$C_W(O_i)$	west extreme corner node of the obstacle O_i;
$b(O_i)$	any border or boundary node of obstacle O_i;
$C_A(O_i), C_B(O_i)$	two extreme corner nodes closest to the sink node.

Because obstacles are dynamically formed with time, the border nodes of previously formed obstacles cannot know the information of newly formed obstacles. Sending the information of new obstacles to all nodes in the WSNs by flooding is a simple way but might lead to high maintenance cost. Therefore, a better approach is to send the information of new obstacles to all the border nodes that surround the old obstacles when the data packets encounter the new obstacle. That is, the maintenance of multiobstacle information is passive. The concept behind our design is that the maintenance of the single obstacle is active so that the packet that applies the greedy forwarding route will not enter the concave region. However, to reduce the maintenance cost of multiobstacle information, a passive policy is adopted. The information of the new obstacle will be maintained by the border nodes of the old obstacle when the packet encounters the new obstacle. In the following discussion, obstacles O_i and O_{i+1} denote the previously and newly formed obstacles in the given WSN. By applying

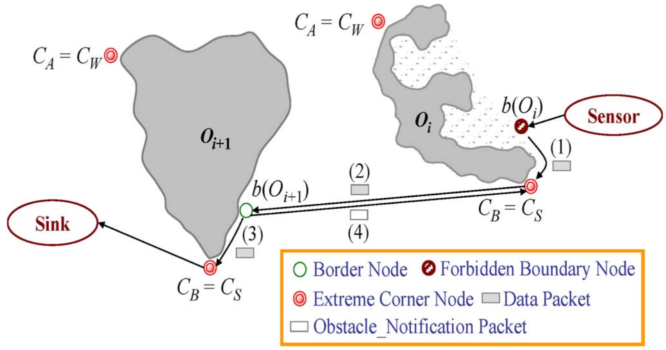


Fig. 10. Example of applying the M-RGP operations for overcoming multiple obstacles. (1) Border node $b(O_i)$ selects C_B as the intermediate destination node. (2) Packet applies the greedy forwarding routing protocol and encounters another obstacle O_{i+1}. (3) Border node $b(O_{i+1})$ guides the data packet to $C_B(O_{i+1})$ by applying S-RGP. (4) Border node $b(O_{i+1})$ prepares an *obstacle_notification* packet and sends it to $C_B(O_i)$.

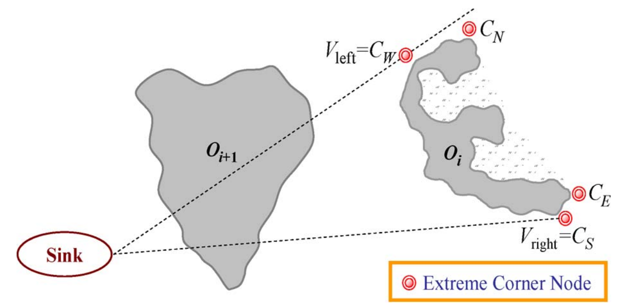


Fig. 11. Example of a selection of the virtual border nodes.

S-RGP, every border node and forbidden-region boundary node $b(O_i)$ evaluates two end extreme corner nodes $C_A(O_i)$ and $C_B(O_i)$, which are the two extreme corner nodes closest to the sink node. Upon receiving the data packets, the border node $b(O_i)$ will determine either $C_A(O_i)$ or $C_B(O_i)$ as the intermediate destination of the best route. As shown in label (1) of Fig. 10, node $b(O_i)$ selects extreme corner node $C_S(O_i) = C_B(O_i)$ as the intermediate destination of the best route. When the packet leaves the end extreme corner node $C_S(O_i)$, it applies the greedy forwarding routing protocol. Afterward, the packet will be routed from $C_S(O_i)$ to border node $b(O_{i+1})$, as shown in label (2) of Fig. 10. To cope with the multiobstacle problem, in the case where $C_B(O_i)$ is selected, $b(O_i)$ appends the location and ID of $C_B(O_i)$ information to the data packet, as shown in label (2) of Fig. 10. If the packet is delivered to the sink node without encountering any obstacle by using greedy forwarding routing protocol, the sink node will return no information to $C_B(O_i)$. On the contrary, as shown in Fig. 10, the packet encounters the border node $b(O_{i+1})$ of obstacle O_{i+1}. Node $b(O_{i+1})$ will not only guide the data packet to $C_S(O_{i+1}) = C_B(O_{i+1})$ by applying the S-RGP but will also prepare an *obstacle_notification packet* and send it to $C_B(O_i)$, as shown in the labels (3) and (4) of Fig. 10. This *obstacle_notification* packet includes the location and ID of receiver $C_B(O_i)$ and the location and role of the four extreme corner nodes $C_W(O_{i+1})$, $C_E(O_{i+1})$, $C_S(O_{i+1})$, and $C_N(O_{i+1})$. Upon receiving the *obstacle_notification* packet, node $C_B(O_i)$ will send the information about the extreme corner nodes of obstacle O_{i+1} to all the border nodes and boundary nodes of obstacle O_i. These nodes then apply the following approaches to derive the optimal route by integrating the information of obstacles O_i and O_{i+1}.

The policies designed for M-RGP intend to comprise the S-RGP. In general, the extreme corner nodes of obstacle O_i can be treated as the virtual border node of O_{i+1}. Then, these virtual border nodes and O_{i+1} can be considered a single obstacle, and the multiobstacle problem is therefore transformed to the single-obstacle problem.

Every border node $b(O_i)$ will execute the following operations to cope with the new obstacle O_{i+1}. First, node $b(O_i)$ will select two of the four extreme corner nodes $C_N(O_i)$, $C_E(O_i)$, $C_S(O_i)$, and $C_W(O_i)$ that are located at the left and right hand sides of $b(O_i)$ to serve as the virtual border nodes of O_{i+1}. The selection of the virtual border nodes is described as follows. Node $b(O_i)$ separately links the sink node and each of the four extreme corner nodes. If there is a link that connects one extreme node and the sink node such that the other three extreme corner nodes are located at the same side of the link, this extreme corner node is a virtual border node of obstacle O_{i+1}. Using this approach, two extreme corner nodes, denoted by V_{left} and V_{right}, that are located at the left- and right-hand sides of $b(O_i)$ can be determined, respectively. As shown in Fig. 11, the extreme corner nodes $C_S(O_i)$ and $C_W(O_i)$, which are, respectively, located at the left- and right-hand sides of obstacle O_i, play the virtual border node of obstacle O_{i+1}. Because the link connected by the sink and $C_S(O_i)$ shows that the other three extreme corner nodes are located at the left side of the link, $C_S(O_i)$ serves as the V_{right}. Similarly, $C_w(O_i)$ serves as the V_{left} by applying the same approach.

Then, $b(O_i)$ separately constructs the link between the sink node and each of V_{left} and V_{right}. For each of the two links, if $C_N(O_{i+1})$, $C_E(O_{i+1})$, $C_S(O_{i+1})$, and $C_W(O_{i+1})$ are located at the same side, the packets sent by the virtual border nodes to the sink node using greedy forwarding routing will not encounter the obstacle O_{i+1}. Therefore, the evaluation $dis(virtual\ border\ node, sink)/r$ is used to estimate the number of hops between the virtual border node and the sink node, and the cost of sending a packet from $b(O_i)$ to the sink through the virtual border node can be derived as follows:

$$hops\,(b(Q_i), V) + dis(V, sink)/r$$

where V represents the virtual border node.

On the contrary, if the four extreme corner nodes of O_{i+1} are not located at the same side of the established link, the S-RGP protocol will be applied to derive the best route between the virtual border nodes and the sink node. We evaluate the hop count from the virtual border node, e.g., V, to the sink node X as follows:

$$RGP\_Single\_Obstacle\_Cost\\ (V, sinkX, C_N(Q_{i+1}), C_E(Q_{i+1}), C_S(Q_{i+1}), C_W(Q_{i+1}))\,.$$

Finally, node $b(O_i)$ selects the virtual border node with the smaller cost from V_{left} and V_{right} as the intermediate node of the best route as follows:

$$hops\,(b(Q_i), V) + RGP\_Single\_Obstacle\_Cost\\ (V, sinkX, C_N(Q_{i+1}), C_E(Q_{i+1}), C_S(Q_{i+1}), C_W(Q_{i+1}))\,.$$

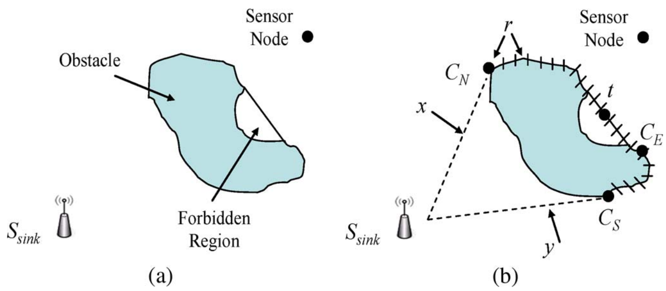


Fig. 12. Single-obstacle environment considered for analyzing the efficiency of S-RGP and PAGER.

V. Analysis on Routing Length

This section analyzes the performance efficiency of S-RGP and M-RGP and compares the two approaches with PAGER in terms of routing hops. To generalize the analysis, the shapes of the obstacles are assumed to be irregular, and the sizes and locations of the obstacles are unknown. First, a basic analysis for the single-obstacle environment is presented. Afterward, the analysis for the multiobstacle environment is proposed.

Fig. 12 shows an environment with a single irregular obstacle. Initially, both S-RGP and PAGER will identify the concave region of the obstacle and then mark it as a forbidden region, as shown in Fig. 12(a). To simplify the analysis, we assume that the area and shape of the forbidden regions created by S-RGP and PAGER are identical. By applying S-RGP, the east, north, and south extreme corner nodes of the obstacle can be determined. Let C_E, C_S, and C_N denote the east, north, and south extreme corner nodes, respectively, as shown in Fig. 12(b). Let r be the communication range. Let the boundary $C_N - C_E - C_S$ be partitioned into several segments, where the length of each segment is r. S-RGP will naturally identify the middle node t, which satisfies

$$Hops(S_{sink}, C_N) + Hops(C_N, t) = Hops(t, C_S) + Hops(C_S, S_{sink})$$

where the $Hops(a, b)$ denotes the number of routing hops from node a to node b. Let $Hops(S_{sink}, C_N)$ and $Hops(C_S, S_{sink})$ be xr and yr, respectively, and the routing hops of the obstacle boundary $C_N - C_E - C_S$ be kr. Furthermore, let $Hops(C_N, t)$ and $Hops(t, C_E)$ be αkr and βkr, respectively, where $0 \leq \alpha,\ \beta \leq 1$. The value α can be determined as follows:

$$xr + \alpha kr = (1-\alpha)kr + yr$$
$$\alpha = \frac{y - x + k}{2k}.$$

In S-RGP, border nodes that belong to curve (t, C_N) will guide the arrived packets to C_N. Then, the packet will be forwarded to the sink node by applying the greedy forwarding routing protocol. Similarly, border nodes that belong to curve (t, C_S) will guide the arrived packets to C_S. Assume that k packets arrive at the boundary and that they are normally distributed over the boundary $C_N - C_E - C_S$. Let the total routing hops of these k packets from nodes t to C_N and from nodes t to C_S be H_N and H_S, respectively. The values of H_N and H_S can be obtained by the following evaluations:

$$H_N = \sum_{i=1}^{u}(x+1) = \frac{u}{2}(u + 2x + 1), \quad \text{where } u = \lceil \alpha k \rceil$$
$$H_S = \sum_{i=1}^{v}(y+i) = \frac{v}{2}(v + 2y + 1), \quad \text{where } v = \lceil (1-\alpha)k \rceil .$$

When PAGER is applied, all border nodes that belong to curves (C_N, C_E) and (C_E, C_S) will guide the arrived packets to C_N and C_S, respectively. Afterward, the packets will be forwarded by applying the greedy forwarding routing protocol. Suppose that k packets arrive at the boundary and that they are normally distributed over the boundary. Let the total routing hops of packets around the obstacle boundaries from nodes C_N to C_E and from nodes C_E to C_S be H'_N and H'_S, respectively. The values of H'_N and H'_S can be evaluated as follows:

$$\begin{aligned} H'_N &= H_N + \sum_{i=1}^{w}(x + \alpha k + i), \quad \text{where } w = \lceil \beta k \rceil \\ &= H_N + w(x + \alpha k) + \sum_{i=1}^{w} i \\ &= H_N + \frac{w}{2}(w + 2x + 2\alpha k + 1) \\ H'_S &= H_S - \sum_{i=1}^{w}\left(y + i + (1-\alpha-\beta)k\right), \quad \text{where } w = \lceil \beta k \rceil \\ &= H_S - wy - (1-\alpha-\beta)wk - \frac{w(w+1)}{2}. \end{aligned}$$

When the k forwarded packets arrive at the obstacle boundary $C_N - C_E - C_S$, the average difference of routing hops between S-RGP and PAGER, which are called D_{avg}, can be evaluated as follows:

$$\begin{aligned} D_{avg} &= \frac{1}{k}\left((H'_N - H_N) + (H'_S - H_S)\right) \\ &= \frac{w}{2k}\left(2(x-y) + 4\alpha k + 2\beta k - 2k\right), \quad \text{where } w = \lceil \beta k \rceil \\ &= \frac{w}{k}\left((x-y) + 2\alpha k + \beta k - k\right), \quad \text{where } \alpha = \frac{y-x+k}{2k} \\ &= \beta \lceil \beta k \rceil > \beta^2 k. \end{aligned} \tag{4}$$

Expression (4) implies that increasing the length of the obstacle boundary $C_N - C_E - C_S$ or the distance between nodes t and C_E will increase the difference of routing hops between S-RGP and PAGER. Expression (4) also indicates that increasing the size of obstacle or the probability that packets will encounter an obstacle will increase the difference of routing hops between S-RGP and PAGER.

Based on the basic analysis, we can analyze the efficiency of M-RGP and PAGER for a multiobstacle environment. To generalize the analysis, the shape, size, and location of the obstacles are assumed to be unknown. Assume that there are m obstacles $O_1, O_2, \ldots, O_m$ in the monitoring area. Fig. 13 shows an example of a monitoring region with multiple obstacles. Initially,

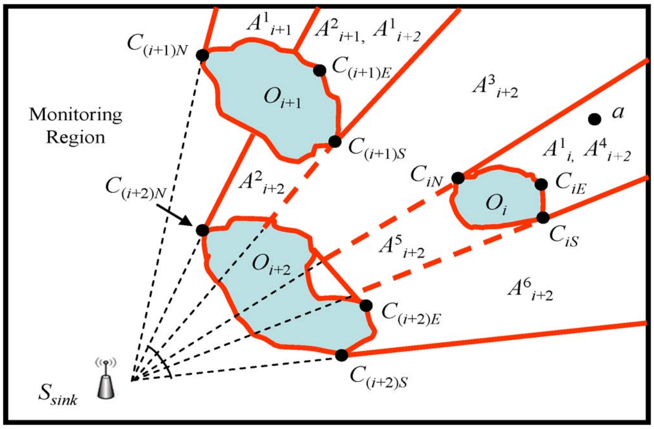


Fig. 13. Example of monitoring the region with multiple obstacles is considered to analyze the efficiency of M-RGP and PAGER.

both M-RGP and PAGER will construct forbidden regions for the concave regions of the obstacles. By applying the proposed *M-RGP*, the east, north, and south extreme corner nodes of each obstacle can be determined. Let C_{iE}, C_{iS}, and C_{iN} be the east, north, and south extreme corner nodes of obstacle O_i, respectively, as shown in Fig. 13.

Let T be the total number of forwarded packets that will encounter the obstacles and N_j be the number of the packets that will encounter obstacle O_j, where

$$T = \sum_{j=1}^{m} N_j.$$

In a monitoring region with multiple obstacles, a forwarded packet may encounter several obstacles. For example, as shown in Fig. 13, the packets sent by sensor a will encounter obstacles O_i and O_{i+2}. Therefore, the monitoring region can be partitioned into many subregions based on the probability that packets will encounter obstacles. The index i of subregion A^i_j will be assigned by the orders from top to bottom and from left to right. Let n^i_j be the number of packets initiated from subregion A^i_j and will encounter obstacle O_i by applying M-RGP or PAGER. The N_j can be represented by

$$N_j = \sum_i n^i_j.$$

Note that a region can have several labels, because a forwarded packet may encounter several obstacles. For example, subregions A^1_i and A^5_{i+2} are identical, and n^1_i is equal to n^5_{i+2}, because the packets initiated from subregions A^1_i or A^5_{i+2} will encounter obstacles O_i and O_{i+2}.

Let $S_{obs}(p_i)$ be the obstacle set encountered by a forwarded packet p_i, where $S_{obs}(p_i) \subseteq \{O_1, O_2, \ldots, O_m\}$. For example, as shown in Fig. 13, obstacles O_i and O_{i+2} belong to $S_{obs}(P)$ when P denotes the set of packets initiated from subregions A^1_i or A^5_{i+2}. Let the number of routing hops of obstacle O_i in boundary $C_N - C_E - C_S$ be $k_i r$. According to (4), let $D_{avg}(O_i)$ be the average difference of routing hops between S-RGP and PAGER for obstacle O_i, where $D_{avg}(O_i) > b_i^2 k_i$. Therefore, the total difference of routing hops between M-RGP and PAGER, which is called H_{diff}, can be estimated as follows:

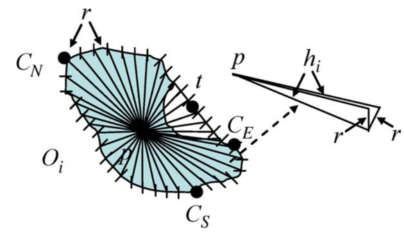


Fig. 14. Obstacle O_i can be partitioned into several fans with the same arc length r.

$$\begin{aligned} H_{diff} &= \sum_{j=1}^{T} \sum_{\forall v \in S_{obs}(p_j)} D_{avg}(v) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{N_j} D_{avg}(O_j), \quad \text{where } N_j = \sum_{\forall z} n_j^z \\ &= \sum_{j=1}^{m} D_{avg}(O_j) N_j > \sum_{j=1}^{m} \beta_j^2 k_j N_j. \end{aligned} \tag{5}$$

Expression (5) implies that the total difference of routing hops between M-RGP and PAGER will be increased with the number of obstacles encountered by a packet. Because the value of n^i_j can be estimated by the deployed density of sensors and the area size of subregion A^i_j, the proposed M-RGP will outperform PAGER in terms of routing hops when the area sizes of subregions are increased. Furthermore, the proposed M-RGP will significantly improve PAGER in terms of routing hops when a forwarded packet encounters different obstacles.

The following discussion extends our analysis by investigating the impact of the ratio of obstacle size to the size of monitoring region. Let the sizes of monitoring region and obstacle O_i be R_m and $\delta_i R_m$, respectively, where $0 \le \delta_i \le 1$. Let p be the central point of obstacle O_i. As shown in Fig. 14, an obstacle can be partitioned into several fans with arc length r, where r is the communication range. Let h_i be the maximum length from the border nodes to the central point p. Let $e_i k_i r$ be the border boundary length of obstacle O_i. That is, the obstacle can be partitioned into $e_i k_i$ fans. As shown in Fig. 14, the area size of each fan will be smaller than $r h_i / 2$. Therefore, the area size of obstacle O_i, which is a summation of the fans' area, will satisfy

$$\delta_i R_m < \frac{e_i k_i r h_i}{2}. \tag{6}$$

According to (6), the value of k_i can be presented as follows:

$$k_i > \frac{2\delta_i R_m}{e_i r h_i}. \tag{7}$$

The value of k_i in (7) can be applied to replace k_j in (5). Therefore, the total difference of routing hops between M-RGP and PAGER can be estimated as follows. The total difference of routing hops will increase with the ratio of the area size of obstacles to the boundary length of obstacles, i.e.,

$$H_{diff} > \sum_{j=1}^{m} \beta_j^2 k_j N_j > \sum_{j=1}^{m} \frac{2\beta_j^2 \delta_j R_m N_j}{e_j r h_j}.$$

VI. Simulation Study

This section examines the performance improvement of S-RGP and M-RGP compared with the existing protocols in terms of the number of border nodes, the control overhead, the average number of hops, and the power consumption. The simulator Tossim [13] is used to implement the compared protocols. Tossim is used for simulating protocols that run in a WSN at bit level and support nesC language, which has widely been used in programming real sensor nodes such as MICA2 and MICAz. With the support of nesC, Tossim can simulate the actuation of a sensor node, such as sensor reading, communication, information processing, and synchronization. It replaces hardware with software components, utilizes the TinyOS component-based architecture, and allows users to scale the WSN to thousands of nodes. Furthermore, TinyViz provides an extensible graphical user interface for debugging, visualizing, and interacting with the Tossim simulations of TinyOS applications. Tossim can be attached to a running simulation. It allows users to capture all of the events in a given simulation. Using nesC programming in the Tossim simulator, users can investigate their performance study for the new developed mechanisms or protocols such as network deployment, topology control, routing, and Media Access Control (MAC) protocols.

The single-obstacle and multiobstacle environments are individually considered in the simulation. For the single-obstacle environment, the following methods are applied to construct regular and irregular obstacles with concave regions. The U-shaped obstacle, which contains a concave region, is used to construct the regular obstacle in the simulation. To construct the irregular obstacle, a solid rectangle obstacle is first produced and then divided into $2m * 2n$ equal-sized grids. The successive odd and even rows are grouped into pairs. For each pair, a random number $0 \leq p \leq m/2$ is generated to shift the consecutive p grids between the odd and the even rows. As a result, an irregular obstacle with several concave regions can be constructed after executing the aforementioned operations on each pair.

In the single-obstacle and multiobstacle environments, the network sizes are set to 600 m ∗ 600 m and 2500 m ∗ 2500 m, respectively. The communication range of each sensor node is set to 40 m. We randomly deployed 500 and 2200 sensor nodes in the single-obstacle and multiobstacle environments, respectively. Fig. 15 shows the screenshot that applies the S-RGP on the environment with a single irregular obstacle. The border nodes, corner nodes, and extreme corner nodes are colored yellow, brown, and blue, respectively. Sensor nodes on the gray lines are the forbidden-region boundary nodes, which are responsible for preventing packets from visiting the concave regions.

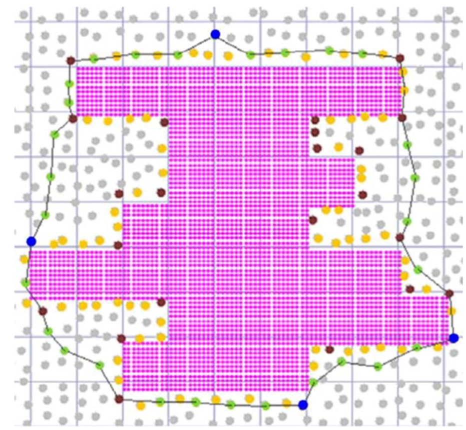


Fig. 15. Screenshot of S-RGP execution.

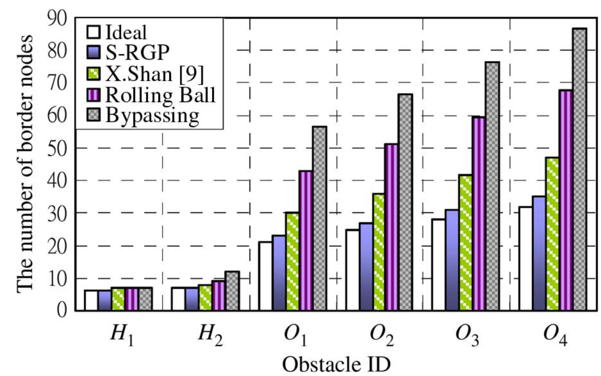


Fig. 16. Comparison of the four mechanisms in terms of the number of border nodes.

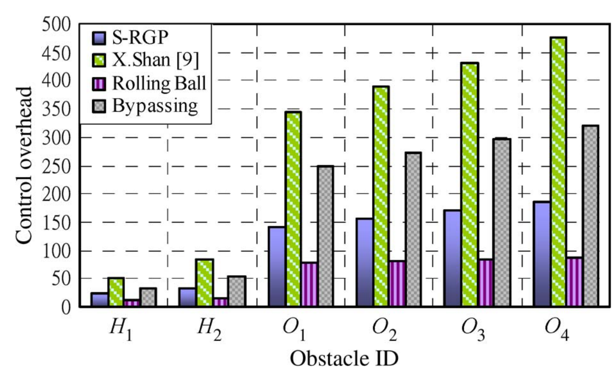


Fig. 17. Comparison of the four mechanisms in terms of the control overhead for identifying border nodes.

Several obstacle detection and hole detection algorithms [6]–[8] have been proposed in recent years. Figs. 16 and 17 investigate the improvement of S-RGP compared with the existing mechanisms in terms of the numbers of border nodes and control packets, respectively. The monitoring region consists of four physical obstacles O_1, O_2, O_3, and O_4 and of two small coverage holes H_1 and H_2. The sizes of physical obstacles O_1, O_2, O_3, and O_4 are set to 200 m ∗ 200 m, 250 m ∗ 250 m, 300 m ∗ 300 m, and 350 m ∗ 350 m, respectively. Fig. 16 compares the four mechanisms in terms of the number of identified border nodes. S-RGP outperforms the other three existing mechanisms in terms of the number of border nodes. Moreover,

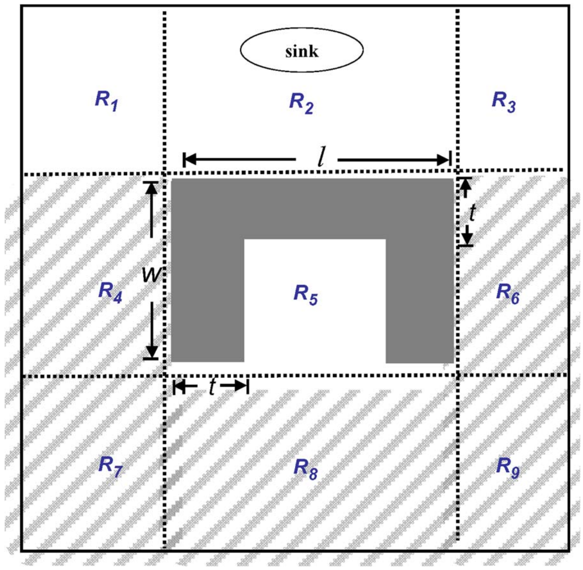


Fig. 18. WSN contains a single regular obstacle, and the source sensors are located at the shadow subregions.

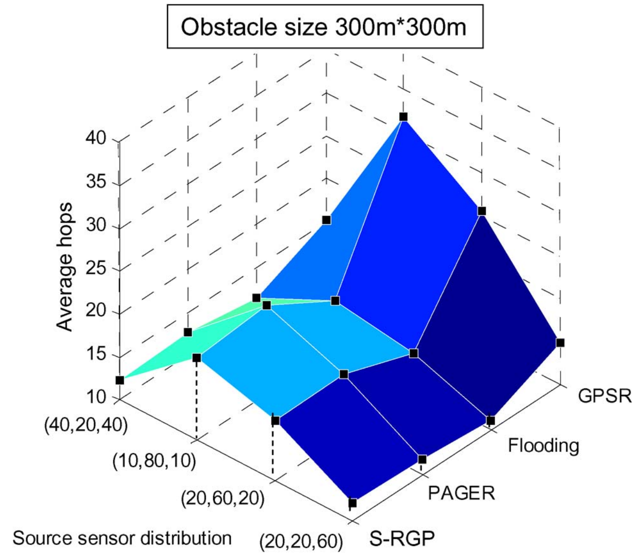


Fig. 19. Comparison of the S-RGP, flooding, GPSR, and PAGER mechanisms in terms of the average hop count with different distributions of source sensor nodes in regions R_4, R_6, R_7, R_8, and R_9.

S-RGP detects all physical obstacles and holes and thus results in a 100% obstacle detection ratio. Although the numbers of border nodes identified by the bypassing mechanism [6] is the largest, as shown in Fig. 16, it still fails to identify some border nodes due to the failure of the proposed hole detection rule. Therefore, the identified border nodes cannot surround the physical obstacles or holes. The mechanisms proposed in [7] and [8] identify some invalid border nodes, which are not near the physical obstacles or holes and fail to find some border nodes that surround the physical obstacles or holes.

Fig. 17 further compares the four algorithms in terms of the control overhead. The rolling ball creates the least number of control packets. However, it fails to find all border nodes. S-RGP outperforms the other two algorithms in terms of the control overhead. In one previous work [6], the stuck nodes collect the whole information of the hole, significantly increasing the control overhead. Compared with other protocols, the mechanism in [8] uses two-hop neighboring information to check the disconnection of adjacent neighbor pairs; therefore, its control overhead is the largest.

Next, we intend to compare the proposed S-RGP and existing obstacle-handling mechanisms [2], [9] in terms of routing performance. We expect that the performance of a routing protocol depends on the size and the shape of obstacles, the size of a concave region, and the relative positions of the sink node and the source sensor nodes. Therefore, the size of regular and irregular obstacles and the relative positions of the sink and sensor nodes are varied in the experiments. Fig. 18 depicts the scenario of a single regular obstacle. The whole network area is partitioned into nine subregions, i.e., R_1–R_9. The sink node is placed in subregion R_2, and the U-shaped obstacle is placed in subregion R_5. The size of subregion R_5 varies with the obstacle size. The size of concave regions can be derived by $(l - 2t) * (w - t)$, where l, w, and t are the length, width, and thickness of the obstacle, respectively. The size of the concave region can be controlled by setting the value of t to 50, 75, and 100 in the simulation.

Fig. 19 investigates the impact of the relative locations of the source sensor and the sink node on each compared routing protocol. The S-RGP, flooding, GPSR [2], and PAGER [9] mechanisms are compared in terms of the average hop count. The obstacle size is set to 300 m $*$ 300 m. As shown in Fig. 18, because sensor nodes located at regions R_1, R_2, and R_3 will not encounter any obstacle when they send the data packets to the sink node, 100 source sensor nodes are selected with different distributions from regions R_4, R_6, R_7, R_8, and R_9. In Fig. 19, (x, y, z) represents a distribution of 100 source sensors, where x, y, and z denote the number of sensor nodes located at regions $R_4 + R_7$, R_8, and $R_6 + R_9$, respectively. Because region R_8 faces the concave region, the average hop count increases with the number of source sensors in region R_8. This result is valid for the four routing protocols. The average hop count of S-RGP is similar to the flooding scheme but is much smaller than the other two protocols. Although flooding has the smallest average hop count, it generates considerable traffic and consumes significant energy of sensor nodes.

Fig. 20 compares the average hop count of the flooding, S-RGP, GPSR, and PAGER mechanisms by varying the size of a single regular obstacle. The distribution of source sensors is set to a constant of $(x, y, z) = (20, 60, 20)$, and the obstacle size is varied as 200 m $*$ 200 m, 250 m $*$ 250 m, 300 m $*$ 300 m, and 350 m $*$ 350 m. In general, the average hop count of all mechanisms increases with the size of the obstacle. The average hop count of S-RGP is similar to the flooding mechanism but is much smaller than GPSR and PAGER. Because GPSR leads the packets to the concave region of an obstacle, it has the largest average hop count.

Both PAGER and the proposed mechanism S-RGP belong to the active obstacle-handling category, which keeps the packets from visiting the concave region of an obstacle. However, the obstacle-preprocessing operations raise the control overhead, which is measured by the number of messages exchanged between neighboring nodes. Fig. 21 depicts the control overhead

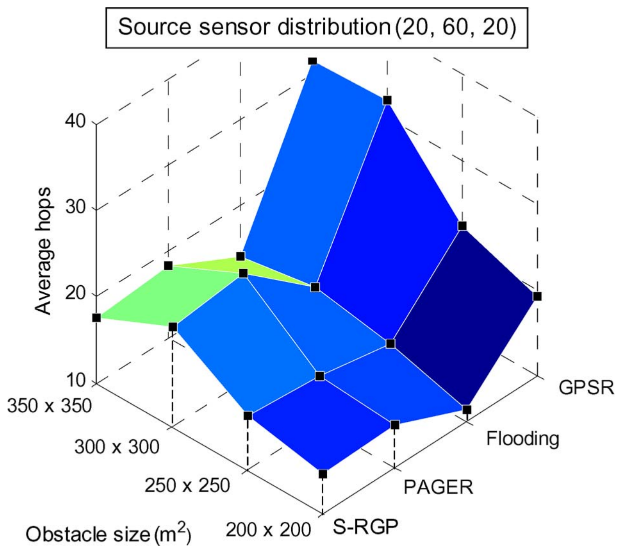


Fig. 20. Comparison of the average hop count of the flooding, S-RGP, GPSR, and PAGER mechanisms by varying the size of the single regular obstacle.

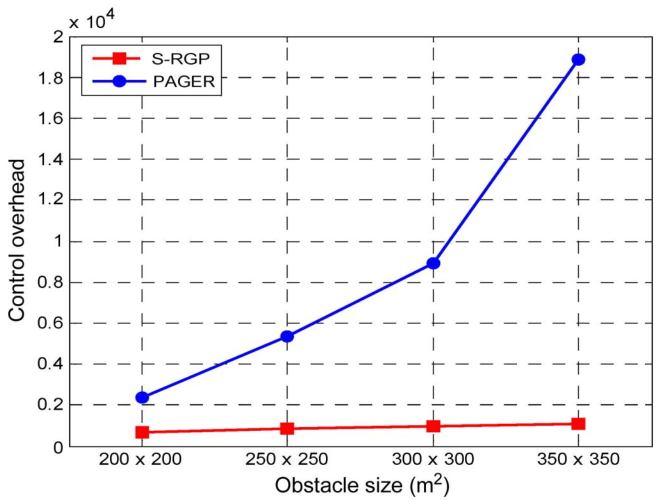


Fig. 21. Comparison of S-RGP and PAGER in terms of the control overhead by varying the size of the single regular obstacle.

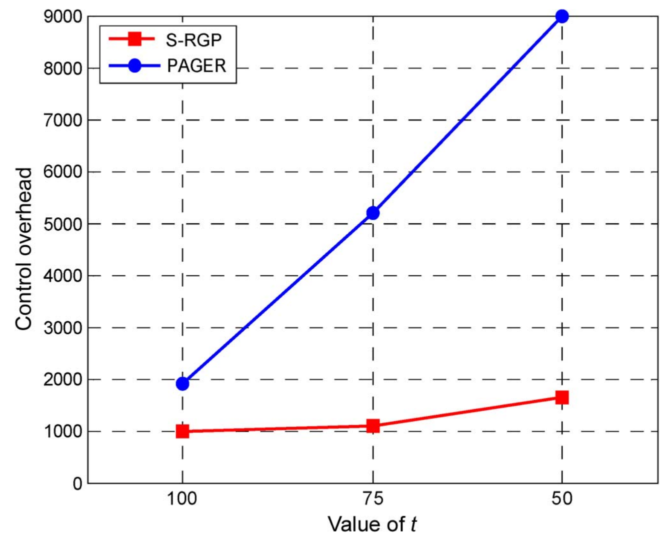


Fig. 22. Comparison of S-RGP and PAGER in terms of the control overhead by varying the size of the concave region in the environment with a single regular obstacle.

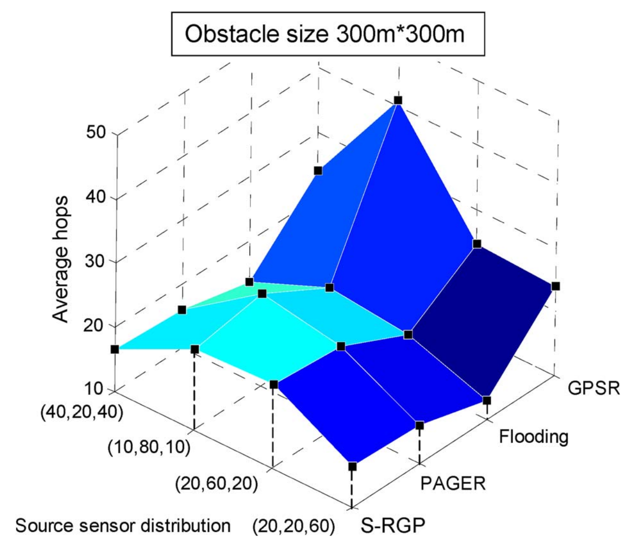


Fig. 23. Comparison of the flooding, S-RGP, GPSR, and PAGER mechanisms in terms of the average hop count in the environment with a single regular obstacle.

of PAGER and S-RGP by varying the size of a single obstacle. In PAGER, the sensor nodes located at the shadow region have to frequently exchange the weight value between neighboring nodes. Compared with PAGER, S-RGP significantly reduces the control overhead, regardless of the change in obstacle size.

Fig. 22 compares the control overhead of PAGER and S-RGP in an environment with a single regular obstacle. The size of the obstacle is set to 350 m $*$ 350 m, and the size of the concave region is decreased with the thickness of the obstacle, which is varied as 50, 75, and 100. As shown in Fig. 22, the control overhead of both mechanisms increases with the size of the concave region. Compared with PAGER, S-RGP significantly improves the control overhead in all cases.

Figs. 23–27 evaluate the performance of the flooding, S-RGP, GPSR, and PAGER mechanisms in the environment with a single irregular obstacle. Fig. 23 compares S-RGP with flooding, GPSR, and PAGER in terms of the average hot count. The network size is set to 300 m $*$ 300 m, and the distributions of the source sensor nodes are varied. The average hop count of S-RGP is similar to flooding and is much better than the other two mechanisms. S-RGP guides the data packet to the shorter routes and works well, although the irregular obstacle with different shapes is encountered.

Fig. 24 investigates the average hop count of the compared mechanisms at different obstacle sizes. The distribution of the source sensor nodes is fixed at $(x, y, z) = (20, 60, 20)$, and the obstacle size is varied as 200 m $*$ 200 m, 250 m $*$ 250 m, 300 m $*$ 300 m, and 350 m $*$ 350 m. In general, the number of hop count increases with the obstacle size for all mechanisms. S-RGP works better than PAGER and GPSR due to the valid guiding for the packet transmission.

Fig. 25 compares the S-RGP and PAGER mechanisms in terms of the control overhead. The distribution of source sensor nodes is set to a constant of $(x, y, z) = (20, 60, 20)$. The size

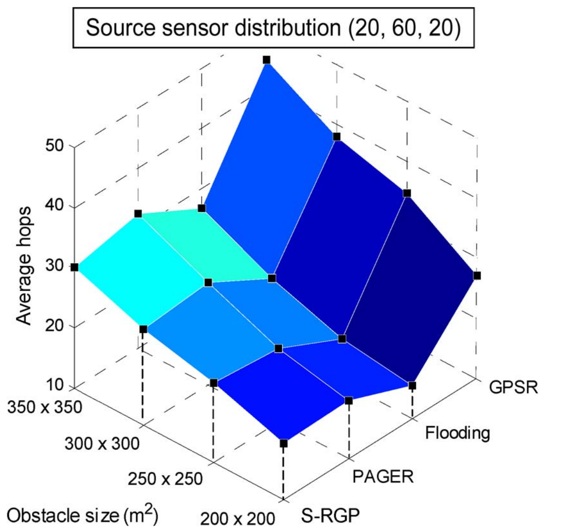


Fig. 24. Comparison of the flooding, S-RGP, GPSR, and PAGER mechanisms in terms of the average hop count by varying the size of the single irregular obstacle.

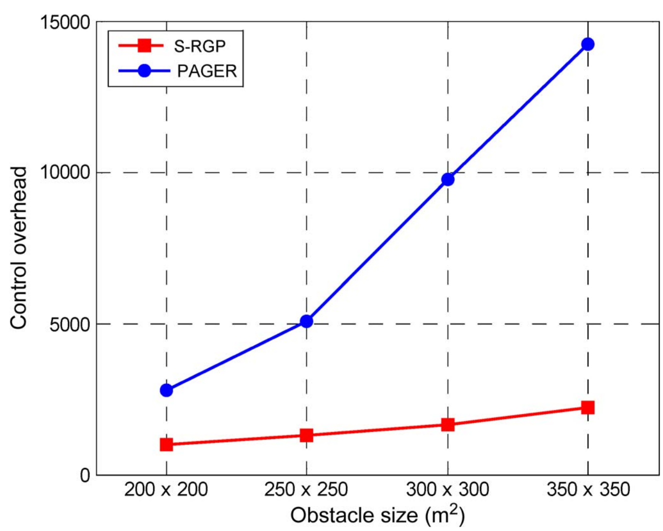


Fig. 25. Comparison of S-RGP and PAGER in terms of the control overhead by varying the size of the single irregular obstacle.

of the single irregular obstacle is varied as 200 m $*$ 200 m, 250 m $*$ 250 m, 300 m $*$ 300 m, and 350 m $*$ 350 m. In general, the control overhead of S-RGP and PAGER increases with the size of the obstacle. Nevertheless, S-RGP outperforms PAGER in terms of the control overhead, regardless of the change in obstacle size.

Fig. 26 compares the flooding, S-RGP, GPSR, and PAGER mechanisms in terms of the remaining energy in an environment with a single irregular obstacle. The obstacle size is set to 300 m $*$ 300 m. One sensor node is selected from region R_4, R_6, R_7, R_8, or R_9 to transmit its sensing data to the sink node at every time unit. The energy model used in this paper is given as follows. Each sensor node is assumed to have a radio range of 40 m and with an initial power of 15 W. The sensor characteristics are given in Table I. These values are taken from the specifications for the TR1000 radio from RF Monolithics [14].

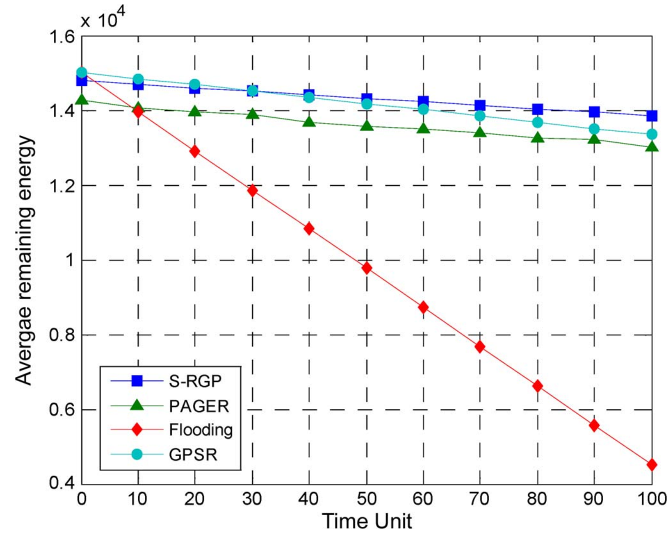


Fig. 26. Comparison of the flooding, S-RGP, GPSR, and PAGER mechanisms in terms of the remaining energy in the environment with a single regular obstacle.

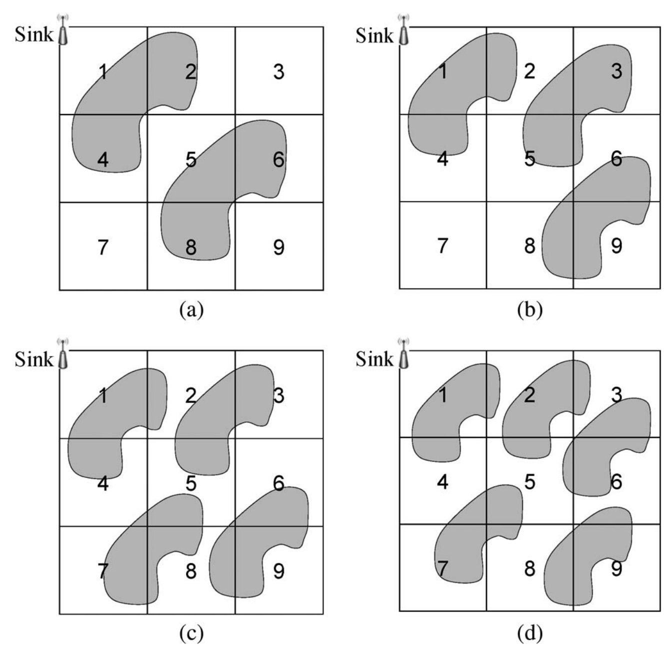


Fig. 27. Obstacle distributions with various numbers of irregular obstacles.

TABLE I
SENSOR CHARACTERISTICS

Radio Mode	Power Consumption (mW)
Transmit (T_x)	14.88
Receive (R_x)	12.50
Idle	12.36
Sleep	0.016

PAGER outperforms GPSR because it keeps the packets from visiting the concave regions. However, S-RGP outperforms PAGER, because PAGER does not always guide packets to the shorter route.

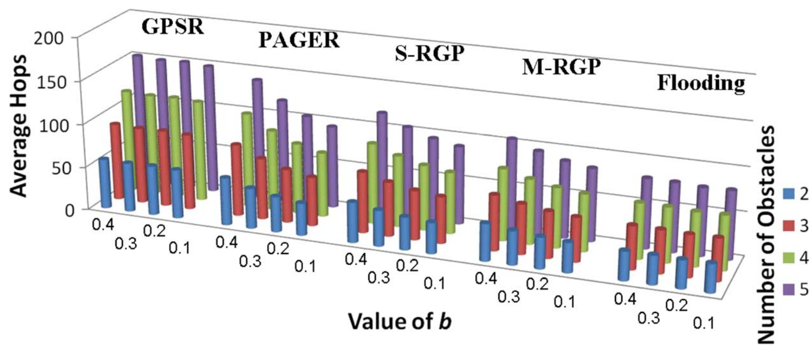


Fig. 28. Comparison of the S-RGP, M-RGP, flooding, GPSR, and PAGER mechanisms in terms of the average hop count by varying the number of multiple irregular obstacles and the value of b.

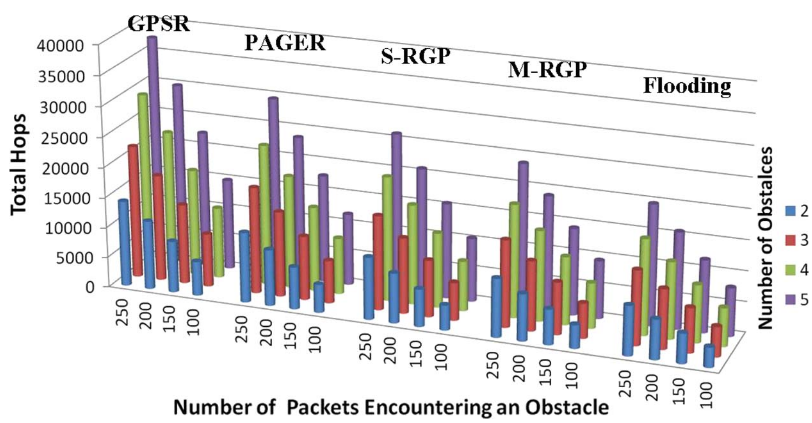


Fig. 29. Comparison of the S-RGP, M-RGP, flooding, GPSR, and PAGER mechanisms in terms of the total hop count by varying the number of multiple irregular obstacles and the packets that encounter an obstacle.

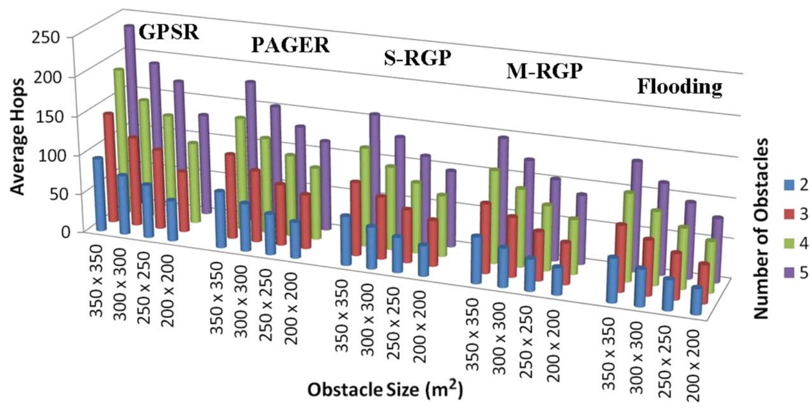


Fig. 30. Comparison of the S-RGP, M-RGP, flooding, GPSR, and PAGER mechanisms in terms of the average hop count by varying the number of multiple irregular obstacles and the obstacle size.

Figs. 28–31 examine the performance improvement of the proposed M-RGP in an environment with multiple irregular obstacles. The entire WSN is partitioned into $3 * 3$ subareas, as shown in Fig. 27. Increasing the number obstacles also increases the probability that packets will encounter obstacles. Fig. 27 depicts four distributions, with the number of irregular obstacles ranging from two to five.

Fig. 28 compares the average hop count of S-RGP, M-RGP, flooding, GPSR, and PAGER in the environment with multiple irregular obstacles, as shown in Fig. 27. According to the aforementioned analysis, increasing the distance between nodes t and C_E will increase the difference of routing hops between S-RGP and PAGER. Therefore, the value of b varies from 0.1 to 0.4. In general, the average hop count of the five mechanisms increases with the number of obstacles. Moreover, the average hop count of S-RGP, M-RGP, and PAGER also increases with the value of b. The result shows that the M-RGP and flooding mechanisms have similar hop counts because M-RGP successfully guides the data packets to the paths that are closed to the shorter paths and, therefore, outperforms S-RGP, GPSR, and PAGER in the average hop count. Because GPSR will guide the data

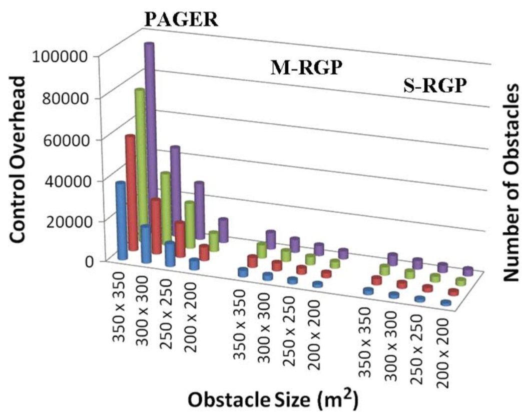


Fig. 31. Comparison of S-RGP, M-RGP, and PAGER in terms of the control overhead by varying the number of multiple irregular obstacles and the obstacle size.

packets into the concave region, it has the largest average hop count.

Fig. 29 depicts the total hop count by applying the S-RGP, M-RGP, flooding, GPSR, and PAGER mechanisms. Because the total difference of routing hops between M-RGP and PAGER is expected to be increased with the number of obstacles encountered by a packet, the number of packets that encounter an obstacle varies from 100 to 250 to observe the change in the total hop count. In general, the total hop count of the five mechanisms increases with the number of obstacles and the number of packets that encounter an obstacle. However, compared to the other four mechanisms, the total hop count of GPSR is significant. By applying GPSR, the packets will be guided into the concave regions. S-RGP and PAGER do not consider multiple obstacles and, thus, lead to a longer route. Therefore, the total hop count of S-RGP and PAGER are larger than M-RGP. Compared with GPSR, S-RGP, and PAGER, M-RGP looks ahead the other obstacles in a multiobstacle environment and guides the packet along the efficient route. Consequently, M-RGP outperforms the other three mechanisms in terms of the total hop count and has similar performance to flooding.

Fig. 30 compares the flooding, S-RGP, M-RGP, GPSR, and PAGER mechanisms in terms of the average hop count by varying the number of multiple irregular obstacles and the obstacle size. The obstacle size is varied as 200 m $*$ 200 m, 250 m $*$ 250 m, 300 m $*$ 300 m, and 350 m $*$ 350 m. In general, the average hop count of the five mechanisms that we compared increases with the obstacle size and the number of obstacles. M-RGP outperforms S-RGP, GPSR, and PAGER in terms of the average hop count and has a similar result to the flooding mechanism, because it can reduce the routing length by looking ahead obstacles that are closer to the sink node.

Fig. 31 investigates the control overhead of PAGER, S-RGP, and M-RGP in the WSN with multiple irregular obstacles, where the obstacle size is varied as 200 m $*$ 200 m, 250 m $*$ 250 m, 300 m $*$ 300 m, and 350 m $*$ 350 m. The control overhead of the three mechanisms increases with the size of obstacle and the number of obstacles. In PAGER, the sensor nodes located at the shadow region have to frequently exchange weight values with neighboring nodes, introducing a significant control overhead. Compared with PAGER, M-RGP reduces the control overhead, regardless of the change in obstacle size. Therefore, M-RGP has a better performance than PAGER in all cases of obstacle size. S-RGP has the least control overhead, because it does not execute the operations of looking ahead obstacles that are closer to the sink node.

Fig. 32 compares the routing performance of the existing flooding, PAGER, and GPSR mechanisms to the proposed S-RGP and M-RGP mechanisms in terms of the average hop count by varying the obstacle ratio and the number of irregular obstacles. In general, the average number of hop counts is increased with the value of k, which represents the fraction of obstacle area of the monitoring area, because a larger value of k causes more packets to visit the obstacles and thus detour a longer routing path. In the case where the obstacle size is fixed, the average routing lengths of the mechanisms that we compared are increased with the number of obstacles. The major reason is that a big obstacle partitioned into several small obstacles will increase the total length of obstacle boundaries and hence increase the average routing length. However, the proposed S-RGP and M-RGP mechanisms outperform the existing PAGER and GPSR mechanisms because the border nodes guide the visited packets to efficient routes.

We have further evaluated the performance result of M-RGP in terms of the average remaining energy. Fig. 33 compares the flooding, S-RGP, M-RGP, GPSR, and PAGER mechanisms in terms of the environment with multiple irregular obstacles. Three obstacles exist in the network, and the obstacle size is set to 300 m $*$ 300 m. One sensor node is selected from region R6, R8, or R9 to transmit its sensing data to the sink node every time unit, as shown in Fig. 27(b). PAGER outperforms GPSR because it prevents the packets from visiting the concave regions. The S-RGP outperforms PAGER because PAGER does not always guide packets to the shortest route. M-RGP outperforms the other mechanisms that we compared because it integrates the information of newly and previously formed obstacles and then dynamically guides the packets to the best route.

VII. Conclusion

In the WSN environment, the unknown obstacles might significantly drop the performance of existing location-aware routing protocols. Existing passive routing protocols can only lead the packets away from the obstacles but cannot guide packets to a better route. This paper has proposed a protocol, S-RGP, that actively constructs the forbidden region for the concave region as soon as the obstacle is formed. The selected border nodes and forbidden-region boundary nodes not only prevent the packets from visiting the concave regions but guide the packet to the best route as well. In addition, the proposed M-RGP copes with the route optimization problem due to the existence of multiple obstacles that are formed at different times. M-RGP integrates the information of newly and previously formed obstacles and dynamically guides the packets to the best route. Simulation results reveal that the both S-RGP and M-RGP have lower control overhead compared with PAGER and outperform GPSR and PAGER in terms of the average hop count and power consumption.

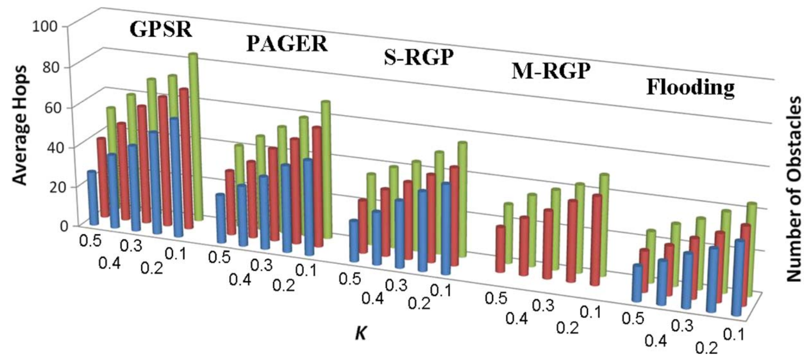


Fig. 32. Comparison of the S-RGP, M-RGP, flooding, GPSR, and PAGER mechanisms in terms of the average hop count by varying the obstacle ratio and the number of irregular obstacles.

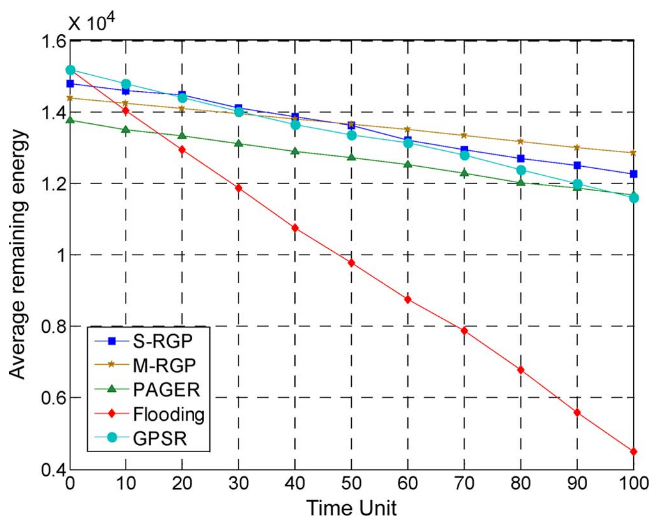


Fig. 33. Comparison of the flooding, S-RGP, M-RGP, GPSR, and PAGER mechanisms in terms of the remaining energy in the environment with multiple irregular obstacles.

REFERENCES

[1] I. Stojmenovic and X. Lin, "GEDIR: Loop-free location-based routing in wireless networks," in *Proc. 11th IASTED Int. Conf. PDCS*, Nov. 1999, pp. 1025–1028.

[2] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput.*, Aug. 2000, pp. 243–254.

[3] C. Y. Chang, C. T. Chang, and S. C. Tu, "Obstacle-free geocasting protocols for single/multidestination short message services in ad hoc networks," *Wireless Netw.*, vol. 9, no. 2, pp. 143–155, Mar. 2003.

[4] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An algorithmic approach to geographic routing in ad hoc and sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 51–62, Feb. 2008.

[5] X. Ma, M. T. Sun, G. Zhao, and X. Liu, "An efficient-path pruning algorithm for geographical routing in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2474–2488, Jul. 2008.

[6] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing holes in sensor networks," *Mobile Netw. Appl.*, vol. 11, no. 2, pp. 187–200, Apr. 2006.

[7] W. J. Liu and K. T. Feng, "Greedy routing with antivoid traversal for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 7, pp. 910–922, Jul. 2009.

[8] X. Shan and J. Tan, "Mobile sensor deployment for a dynamic cluster-based target-tracking sensor network," in *Proc. IEEE/RSJ Int. Conf. IROS*, Aug. 2005, pp. 1452–1457.

[9] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead-end problem of location-based routing in sensor networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 4, pp. 1509–1522, Jul. 2005.

[10] S. Chen, G. Fan, and J. Cui, "Avoid 'void' in geographic routing for data aggregation in sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.—Special Issue on Wireless Sensor Networks*, vol. 1, no. 4, pp. 169–178, Jul. 2006.

[11] V. Vivekanandan and V. W. S. Wong, "Concentric anchor beacon localization algorithm for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2733–2744, Sep. 2007.

[12] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Jun. 2002, pp. 1567–1576.

[13] [Online]. Available: http://www.tinyos.net/

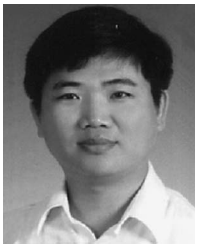
[14] *Ash Transceiver's Designers Guide*, 2004. [Online]. Available: http://www.rfm.com

[15] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

Chih-Yung Chang (M'01–A'01) received the Ph.D. degree in computer science and information engineering from the National Central University, Taoyuan, Taiwan, in 1995.

He joined the faculty of the Department of Computer and Information Science, Aletheia University, Danshui, Taiwan, as an Assistant Professor in 1997 and served as the Chair from August 2000 to July 2002. Since August 2002, he has been an Associate Professor with the Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan, where he is currently a Full Professor. He served as an Associate Guest Editor for the *IET Communications* (2011), *Telecommunication Systems* (2010), the *Journal of Information Science and Engineering* (2008), the *Journal of Internet Technology* (2004 and 2008), and the *Journal of Mobile Multimedia* (2005). He served as a member of the Editorial Boards of the *Tamsui Oxford Journal of Mathematical Sciences* (2001–2008) and the *Journal of Information Technology and Applications* (2008). His research interests include wireless sensor networks, Bluetooth radio networks, ad hoc wireless networks, and WiMAX broadband technologies.

Dr. Chang is a member of the IEEE Computer Society, the IEEE Communications Society, and the Institute of Electrical, Information, and Communication Engineers. He was an Area Chair of AINA 2005, the Vice Chair of IEEE WisCom 2005, and EUC 2005. He was the Track Chair for the Learning Technology in Education Track at ITRE 2005 and a Program Cochair of WASN 2007, MNSAT 2005, and UbiLearn 2006. He was a Workshop Cochair of ICS 2008, INA 2005, MSEAT 2003, and MSEAT 2004 and the Publication Chair of MSEAT 2005 and SCORM 2006.



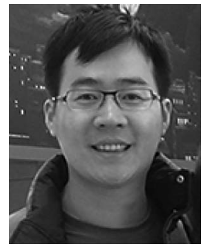
Chao-Tsun Chang was born in Taipei, Taiwan. He received the Ph.D. degree in computer science and information engineering from the National Central University, Taoyuan, Taiwan, in 2006.

Since 2006, he has been with the faculty of the Department of Information Management, Hsiuping Institute of Technology, Taichung, Taiwan, as an Assistant Professor. Over the last ten years, he has directed 11 research projects, including three national National Science Council projects and eight information system development programs. His research interests include wireless sensor networks, Bluetooth radio networks, ad hoc wireless networks, and mobile computing.

Dr. Chang is a member of the IEEE Computer and Communications Societies.

Yu-Chieh Chen received the B.S. degree in computer science and information engineering in 2005 from Ming Chuan University, Taipei, Taiwan, and the M.S. degree in computer science and information engineering in 2007 from Tamkang University, Tamsui, Taiwan, where he has been working toward the Ph.D. degree with the Department of Computer Science and Information Engineering since 2007. He received several scholarship grants in Taiwan.

He has participated in several wireless sensor network projects. His research interests include wireless sensor networks, ad hoc wireless networks, mobile/wireless computing, and WiMAX.



Shih-Chieh Lee received the B.S. and Ph.D. degrees in computer science and information engineering from Tamkang University, Tamsui, Taiwan, in 1997 and 2006, respectively.

He is currently with Altek Corporation, Hsinchu, Taiwan. His research interests include wireless networks, Bluetooth radio systems, sensor networks, mobile computing, and the design of network protocols.