Greedy Routing with Anti-Void Traversal for Wireless Sensor Networks

Wen-Jiunn Liu, *Student Member*, *IEEE*, and Kai-Ten Feng, *Member*, *IEEE*

Abstract—The unreachability problem (i.e., the so-called void problem) that exists in the greedy routing algorithms has been studied for the wireless sensor networks. Some of the current research work cannot fully resolve the void problem, while there exist other schemes that can guarantee the delivery of packets with the excessive consumption of control overheads. In this paper, a greedy anti-void routing (GAR) protocol is proposed to solve the void problem with increased routing efficiency by exploiting the boundary finding technique for the unit disk graph (UDG). The proposed rolling-ball UDG boundary traversal (RUT) is employed to completely guarantee the delivery of packets from the source to the destination node under the UDG network. The boundary map (BM) and the indirect map searching (IMS) scheme are proposed as efficient algorithms for the realization of the RUT technique. Moreover, the hop count reduction (HCR) scheme is utilized as a short-cutting technique to reduce the routing hops by listening to the neighbor's traffic, while the intersection navigation (IN) mechanism is proposed to obtain the best rolling direction for boundary traversal with the adoption of shortest path criterion. In order to maintain the network requirement of the proposed RUT scheme under the non-UDG networks, the partial UDG construction (PUC) mechanism is proposed to transform the non-UDG into UDG setting for a portion of nodes that facilitate boundary traversal. These three schemes are incorporated within the GAR protocol to further enhance the routing performance with reduced communication overhead. The proofs of correctness for the GAR scheme are also given in this paper. Comparing with the existing localized routing algorithms, the simulation results show that the proposed GAR-based protocols can provide better routing efficiency.

Index Terms—Greedy routing, void problem, unit disk graph, localized algorithm, wireless sensor network.

1 Introduction

A wireless sensor network (WSN) consists of sensor nodes (SNs) with wireless communication capabilities for specific sensing tasks. Due to the limited available resources, efficient design of localized multihop routing protocols [1] becomes a crucial subject within the WSNs. How to guarantee delivery of packets is considered an important issue for the localized routing algorithms. The well-known greedy forwarding (GF) algorithm [2] is considered a superior scheme with its low routing overheads. However, the void problem [3], which makes the GF technique unable to find its next closer hop to the destination, will cause the GF algorithm failing to guarantee the delivery of data packets.

Several routing algorithms are proposed to either resolve or reduce the void problem, which can be classified into non-graph-based and graph-based schemes. In the non-graph-based algorithms [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], the intuitive schemes as proposed in [4] construct a two-hop neighbor table for implementing the GF algorithm. The network flooding mechanism is adopted within the GRA [5] and PSR [6] schemes while the void problem occurs. There also exist routing protocols that adopt the backtracking method at the occurrence of the network holes (such as GEDIR, [4], DFS [7], and SPEED [8]). The routing schemes as proposed by ARP [9] and LFR [10] memorize the routing path after the void problem takes place. Moreover, other routing protocols (such as PAGER [11], NEAR [12], DUA [13], INF [14], and YAGR [15]) propagate and update the information of the observed void node in order to reduce the probability of encountering the void problem. By exploiting these routing algorithms, however, the void problem can only be either 1) partially alleviated or 2) resolved with considerable routing overheads and significant converging time.

On the other hand, there are research works on the design of graph-based routing algorithms [3], [16], [17], [18], [19], [20], [21], [22], [23] to deal with the void problem. Several routing schemes as surveyed in [16] adopt the planar graph [24] derived from the unit disk graph (UDG) as their network topologies, such as GPSR [3], GFG [17], Compass Routing II [18], AFR [19], GOAFR [20] GOAFR+ [21], GOAFR++ [16], and GPVFR [22]. For conducting the above planar graph-based algorithms, the planarization technique is required to transform the underlying network graph into the planar graph. The Gabriel graph (GG) [25] and the relative neighborhood graph (RNG) [26] are the two commonly used localized planarization techniques that abandon some communication links from the UDG for achieving the planar graph. Nevertheless, the usage of the GG and RNG graphs has significant pitfalls due to the removal of critical communication links, leading to longer routing paths to the destination. As shown in Fig. 1, the nodes (N_S, N_D) are considered the transmission pair, while N_V represents the node that the void problem occurs. The representative planar graph-based GPSR scheme can not forward the packets from N_V to N_A directly since both the GG and the RNG planarization rules abandon the communication link from N_V to N_A. Considering the GG planarization rule for example, the communication link from N_V

• *The authors are with the Department of Communication Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan, ROC.*
E-mail: jiunn.cm94g@nctu.edu.tw, ktfeng@mail.nctu.edu.tw.

Manuscript received 17 Nov. 2007; revised 1 July 2008; accepted 21 Oct. 2008; published online 6 Nov. 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2007-11-0346. Digital Object Identifier no. 10.1109/TMC.2008.162.

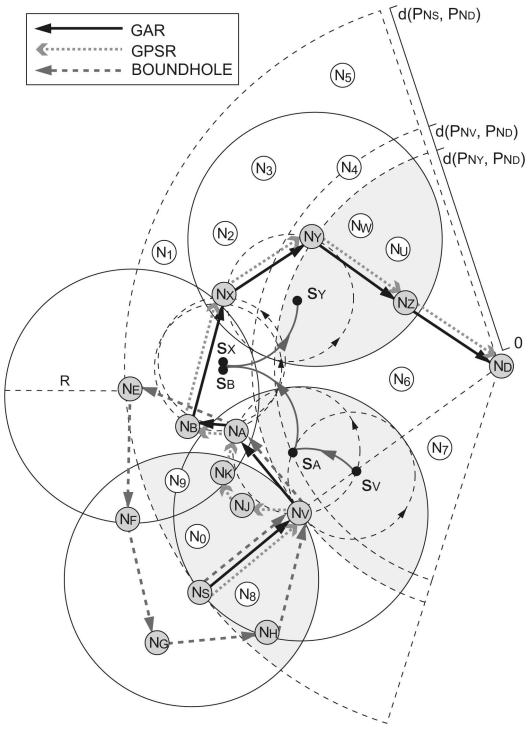

Fig. 1. Example routing paths constructed by using the GAR, the GPSR, and the BOUNDHOLE algorithms under the existence of the void problem.

to N_A is discarded since both N_J and N_K are located within the forbidden region, which is defined as the smallest disk passing through both N_V and N_A. Therefore, based on the right-hand rule, the resulting path by adopting the GPSR protocol can be obtained as $\{N_S, N_V, N_J, N_K, N_A, N_B, N_X, N_Y, N_Z, N_D\}$. The two unnecessary forwarding nodes N_J and N_K are observed, as in Fig. 1.

Furthermore, the planar graph-based schemes, e.g., the GPSR and GOAFR++ algorithms, will in general lose their properties of guaranteed packet delivery due to the unexpected network partition within the non-UDG networks. The reason is also attributed to the situations that critical communication links are removed by adopting the GG and RNG planarization techniques. In order to resolve the network partition problem, a cross-link detection protocol (CLDP) is therefore suggested in [27] for planarization of the underlying non-UDG networks. However, for the purposes of both detecting the cross links and planarizing the underlying network, the CLDP planarization will introduce excessive control overhead since all communication links are required to be probed and frequently traversed. Moreover, the problems of multiple cross links and concurrent probing can further enlarge the total number of communication overhead within the CLDP technique.

Due to the drawbacks of link removal from the planar graph-based algorithms, the adoption of UDG without planarization for the modeling of underlying network is suggested. A representative UDG-based routing scheme, i.e., the BOUNDHOLE algorithm [23], forwards the packets around the network holes by identifying the locations of the holes. However, due to the occurrence of routing loop, the delivery of packets can not be guaranteed in the BOUNDHOLE scheme even if a route exists from the source to the destination node. For example, as shown in Fig. 1, it is assumed that the node N_X is located within the transmission range of N_B, while it is considered out of the transmission ranges of nodes N_A and N_E. Based on the minimal sweeping angle criterion within the BOUNDHOLE algorithm, N_A will choose N_E as its next hop node since the counterclockwise sweeping from N_V to N_E (hinged at N_A) is smaller comparing with that from N_V to N_B. Therefore, the missing communication link from N_B to N_X can be observed, and the resulting path by adopting the BOUNDHOLE scheme becomes $\{N_S, N_V, N_A, N_E, N_F, N_G, N_H, N_V\}$. It is observed that the undeliverable routing path from the source node N_S is constructed even with unpartitioned network topology. Moreover, two cases of edge intersections within the BOUNDHOLE algorithm [23] result in high routing overhead in order to identify the network holes.

In this paper, a greedy anti-void routing (GAR) protocol is proposed to guarantee packet delivery with increased routing efficiency by completely resolving the void problem based on the UDG setting. The GAR protocol is designed to be a combination of both the conventional GF algorithm and the proposed rolling-ball UDG boundary traversal (RUT) scheme. The GF scheme is executed by the GAR algorithm without the occurrence of the void problem, while the RUT scheme is served as the remedy for resolving the void problem, leading to the assurance for packet delivery. Moreover, the correctness of the proposed GAR protocol is validated via the given proofs. The implementation of the GAR protocol is also explained, including that for the proposed boundary map (BM) and the indirect map searching (IMS) algorithm for the BM construction.

Furthermore, the associated three additional enhanced mechanisms are also exploited, including the hop count reduction (HCR), the intersection navigation (IN), and the partial UDG construction (PUC) schemes. The HCR scheme is a short-cutting technique that acquires information by listening to one-hop neighbor's packet forwarding, while the other short-cutting method, as proposed in [28], requires information from two-hop neighbors that can result in excessive control packet exchanges. With the occurrence of the void node, the IN mechanism determines its rolling direction based on the criterion of smallest hop counts (HCs) for boundary traversal. Similar to the CLDP method [27], the IN scheme acquires information over multiple hops in order to process its algorithm. However, it is required for the CLDP technique to traverse all the communication links in the networks, while the IN scheme only exploits a small portion of network links for conducting the boundary traversal. Moreover, in order to meet the network requirement for the RUT scheme under non-UDG networks, the PUC mechanism is utilized to transform the non-UDG into the UDG setting for the nodes that are adopted for boundary traversal.

By adopting these three enhanced schemes, both the routing efficiency and the communication overhead of the original GAR algorithm can further be improved. The performance of the proposed GAR protocol and the version with the enhanced mechanisms (denoted as the GAR-E algorithm) is evaluated via simulations under both the UDG network for the ideal case and the non-UDG setting for realistic scenario. The simulation results show that the GAR-based schemes can both guarantee the delivery of data packets and pertain better routing performance under the UDG network. On the other hand, comparing with the other

existing schemes, feasible routing performance with reduced communication overhead can be provided by the GAR-based algorithms within the non-UDG network environment.

The remainder of this paper is organized as follows: Section 2 formulates the problem of interest by the underlying network model. The proposed GAR protocol is explained in Section 3, while Section 4 provides the practical realization of the GAR algorithm. Section 5 exploits the three enhanced mechanisms, including the HCR, the IN, and the PUC mechanisms. The performance of the GAR-based protocols is evaluated and compared in Section 6. Section 7 draws the conclusions.

2 Problem Formulation

Considering a set of SNs $\mathbf{N} = \{N_i \mid \forall\, i\}$ within a two-dimensional (2D) euclidean plane, the locations of the set $\mathbf{N}$, which can be acquired by their own positioning systems, are represented by the set $\mathbf{P} = \{P_{N_i} \mid P_{N_i} = (x_{N_i}, y_{N_i}), \forall\, i\}$. It is assumed that all the SNs are homogeneous and equipped with omnidirectional antennas. The set of closed disks defining the transmission ranges of $\mathbf{N}$ is denoted as $\overline{\mathbf{D}} = \{\overline{D}(P_{N_i}, R) \mid \forall\ i\}$, where $\overline{D}(P_{N_i}, R) = \{x \mid \|x - P_{N_i}\| \leq R, \forall\, x \in \mathbb{R}^2\}$. It is noted that P_{N_i} is the center of the closed disk with R denoted as the radius of the transmission range for each N_i. Therefore, the network model for the WSNs can be represented by a UDG as $G(\mathbf{P}, \mathbf{E})$ with the edge set $\mathbf{E} = \{E_{ij} \mid E_{ij} = (P_{N_i}, P_{N_j}), P_{N_i} \in \overline{D}(P_{N_j}, R), \forall\, i \neq j\}$. The edge E_{ij} indicates the unidirectional link from P_{N_i} to P_{N_j} whenever the position P_{N_i} is within the closed disk region $\overline{D}(P_{N_j}, R)$. Moreover, the one-hop neighbor table for each N_i is defined as

$$\mathbf{T}_{N_i} = \{[ID_{N_k}, P_{N_k}] \mid P_{N_k} \in \overline{D}(P_{N_i}, R), \forall\, k \neq i\}, \quad (1)$$

where ID_{N_k} represents the designated identification number for N_k. In the GF algorithm, it is assumed that the source node N_S is aware of the location of the destination node N_D. If N_S wants to transmit packets to N_D, it will choose the next hop node from its $\mathbf{T}_{N_S}$, which 1) has the shortest euclidean distance to N_D among all the SNs in $\mathbf{T}_{N_S}$ and 2) is located closer to N_D compared to the distance between N_S and N_D (e.g., N_V, as in Fig. 1). The same procedure will be performed by the intermediate nodes (such as N_V) until N_D is reached. However, the GF algorithm will be inclined to fail due to the occurrences of voids even though some routing paths exist from N_S to N_D. The void problem is defined as follows:

Problem 1 (void problem). *The GF algorithm is exploited for packet delivery from N_S to N_D. The void problem occurs while there exists a void node (N_V) in the network such that no neighbor of N_V is closer to the destination as*

$$\{P_{N_k} \mid d(P_{N_k}, P_{N_D}) < d(P_{N_V}, P_{N_D}), \forall\, P_{N_k} \in \mathbf{T}_{N_V}\} = \emptyset, \quad (2)$$

where $d(x, y)$ represents the euclidean distance between x and y. $\mathbf{T}_{N_V}$ is the one-hop neighbor table of N_V.

3 Proposed Greedy Anti-Void Routing (GAR) Protocol

The objective of the GAR protocol is to resolve the void problem such that the packet delivery from N_S to N_D can be guaranteed. Before diving into the detail formulation of the proposed GAR algorithm, an introductory example is described in order to facilitate the understanding of the GAR protocol. As shown in Fig. 1, the data packets initiated from the source node N_S to the destination node N_D will arrive in N_V based on the GF algorithm. The void problem occurs as N_V receives the packets, which leads to the adoption of the RUT scheme as the forwarding strategy of the GAR protocol. A circle is formed by centering at $\mathbf{s}_V$ with its radius being equal to half of the transmission range $R/2$. The circle is hinged at N_V and starts to conduct counterclockwise rolling until an SN has been encountered by the boundary of the circle, i.e., N_A, as in Fig. 1. Consequently, the data packets in N_V will be forwarded to the encountered node N_A.

Subsequently, a new equal-sized circle will be formed, which is centered at $\mathbf{s}_A$ and hinged at node N_A. The counterclockwise rolling procedure will be proceeded in order to select the next hop node, i.e., N_B in this case. Similarly, same process will be performed by other intermediate nodes (such as N_B and N_X) until the node N_Y is reached, which is considered to have a smaller distance to N_D than that of N_V to N_D. The conventional GF scheme will be resumed at N_Y for delivering data packets to the destination node N_D. As a consequence, the resulting path by adopting the GAR protocol becomes $\{N_S, N_V, N_A, N_B, N_X, N_Y, N_Z, N_D\}$. In the following sections, the formal description of the RUT scheme will be described in Section 3.1, while the detail of the GAR algorithm is explained in Section 3.2. The proofs of correctness of the GAR protocol are given in Section 3.3.

3.1 Proposed Rolling-Ball UDG Boundary Traversal (RUT) Scheme

The RUT scheme is adopted to solve the boundary finding problem, and the combination of the GF and the RUT scheme (i.e., the GAR protocol) can resolve the void problem, leading to the guaranteed packet delivery. The definition of boundary and the problem statement are described as follows:

Definition 1 (boundary). *If there exists a set $\mathbf{B} \subseteq \mathbf{N}$ such that 1) the nodes in $\mathbf{B}$ form a simple unidirectional ring and 2) the nodes located on and inside the ring are disconnected with those outside of the ring, $\mathbf{B}$ is denoted as the boundary set and the unidirectional ring is called a boundary.*

Problem 2 (boundary finding problem). *Given a UDG $G(\mathbf{P}, \mathbf{E})$ and the one-hop neighbor tables $\mathbf{T} = \{\mathbf{T}_{N_i} \mid \forall\ N_i \in \mathbf{N}\}$, how can a boundary be obtained by exploiting the distributed computing techniques?*

There are three phases within the RUT scheme, including the initialization, the boundary traversal, and the termination phases.

3.1.1 Initialization Phase

No algorithm can be executed without the algorithm-specific trigger event. The trigger event within the RUT scheme is called the starting point (SP). The RUT scheme can be initialized from any SP, which is defined as follows:

Definition 2 (rolling ball). *Given $N_i \in \mathbf{N}$, a rolling ball $RB_{N_i}(\mathbf{s}_i, R/2)$ is defined by 1) a rolling circle hinged at P_{N_i} with its center point at $\mathbf{s}_i \in \mathbb{R}^2$ and the radius equal to $R/2$, and 2) there does not exist any $N_k \in \mathbf{N}$ located inside the rolling*

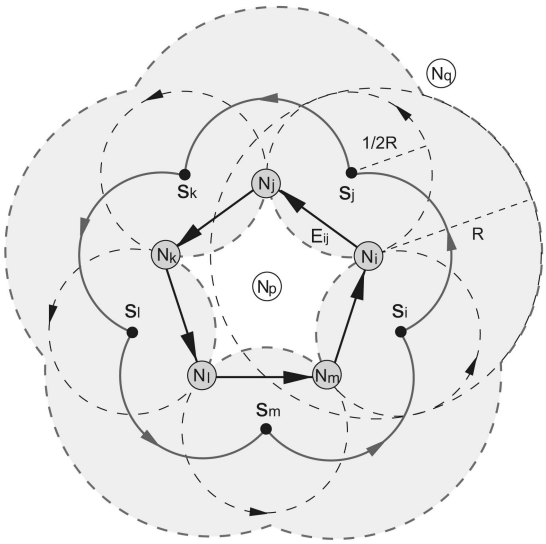

Fig. 2. The proposed RUT scheme.

ball as $\{RB^{\sim}_{N_i}(\mathbf{s}_i, R/2) \cap \mathbf{N}\} = \emptyset$, *where* $RB^{\sim}_{N_i}(\mathbf{s}_i, R/2)$ *denotes the open disk within the rolling ball.*

Definition 3 (starting point). *The SP of* N_i *within the RUT scheme is defined as the center point* $\mathbf{s}_i \in \mathbb{R}^2$ *of* $RB_{N_i}(\mathbf{s}_i, R/2)$.

As shown in Fig. 2, each node N_i can verify if there exists an SP since the rolling ball $RB_{N_i}(\mathbf{s}_i, R/2)$ is bounded by the transmission range of N_i. According to Definition 3, the SPs should be located on the circle centered at P_{N_i} with a radius of $R/2$. As will be proven in Lemmas 1 and 2, all the SPs will result in the red solid flower-shaped arcs, as in Fig. 2. It is noticed that there should always exist an SP, while the void problem occurs within the network, which will be explained in Section 3.2. At this initial phase, the location $\mathbf{s}_i$ can be selected as the SP for the RUT scheme.

3.1.2 Boundary Traversal Phase

Given $\mathbf{s}_i$ as the SP associated with its $RB_{N_i}(\mathbf{s}_i, R/2)$ hinged at N_i, either the counterclockwise or clockwise rolling direction can be utilized. As shown in Fig. 2, $RB_{N_i}(\mathbf{s}_i, R/2)$ is rolled counterclockwise until the next SN is reached (i.e., N_j in Fig. 2). The unidirectional edge $E_{ij} = (P_{N_i}, P_{N_j})$ can therefore be constructed. A new SP and the corresponding rolling ball hinged at N_j (i.e., $\mathbf{s}_j$ and $RB_{N_j}(\mathbf{s}_j, R/2)$) will be assigned, and consequently, the same procedure can be conducted continuously.

3.1.3 Termination Phase

The termination condition for the RUT scheme happens while the first unidirectional edge is revisited. As shown in Fig. 2, the RUT scheme will be terminated if the edge E_{ij} is visited again after the edges E_{ij}, E_{jk}, E_{kl}, E_{lm}, and E_{mi} are traversed. The boundary set initiated from N_i can therefore be obtained as $\mathbf{B} = \{N_i, N_j, N_k, N_l, N_m\}$.

3.2 Detail Description of Proposed GAR Protocol

As shown in Fig. 1, the packets are intended to be delivered from N_S to N_D. N_S will select N_V as the next hop node by adopting the GF algorithm. However, the void problem prohibits N_V to continue utilizing the same GF algorithm for packet forwarding. The RUT scheme is therefore employed by assigning an SP (i.e., $\mathbf{s}_V$) associated with the rolling ball $RB_{N_V}(\mathbf{s}_V, R/2)$ hinged at N_V. As illustrated in Fig. 1, $\mathbf{s}_V$ can be chosen to locate on the connecting line between N_V and N_D with $R/2$ away from N_V. It is noticed that there always exists an SP for the void node (N_V) since there is not supposed to have any SN located within the blue-shaded region (as in Fig. 1), which is large enough to satisfy the requirements, as in Definitions 2 and 3. The RUT scheme is utilized until N_Y is reached (after traversing N_A, N_B, and N_X). Since $d(P_{N_Y}, P_{N_D}) < d(P_{N_V}, P_{N_D})$, the GF algorithm is resumed at N_Y, and the next hop node will be selected as N_Z. The route from N_S to N_D can therefore be constructed for packet delivery. Moreover, if there does not exist a node N_Y such that $d(P_{N_Y}, P_{N_D}) < d(P_{N_V}, P_{N_D})$ within the boundary traversal phase, the RUT scheme will be terminated after revisiting the edge E_{VA}. The result indicates that there does not exist a routing path between N_S and N_D.

3.3 Proof of Correctness

In this section, the correctness of the RUT scheme is proven in order to solve Problem 2, while the GAR protocol is also proven for resolving the void problem (i.e., Problem 1) in order to guarantee packet delivery.

Fact 1. *A simple closed curve is formed by traversing a point on the border of a closed filled 2D geometry with fixed orientation.*

Lemma 1. *All the SPs within the RUT scheme form the border of a shape that results from overlapping the closed disks* $\overline{D}(P_{N_i}, R/2)$ *for all* $N_i \in \mathbf{N}$, *and vice versa.*

Proof. Based on Definitions 2 and 3, the set of SPs can be obtained as $\mathbf{S} = \mathbf{R_1} \cap \mathbf{R_2} = \{\mathbf{s}_i \mid \|\mathbf{s}_i - P_{N_i}\| = R/2, \exists N_i \in \mathbf{N}, \mathbf{s}_i \in \mathbb{R}^2\} \cap \{\mathbf{s}_j \mid \|\mathbf{s}_j - P_{N_j}\| \geq R/2, \forall\, N_j \in \mathbf{N}, \mathbf{s}_j \in \mathbb{R}^2\}$ by adopting the 1) and 2) rules within Definition 2. On the other hand, the border of the resulting shape from the overlapped closed disks $\overline{D}(P_{N_i}, R/2)$ for all $N_i \in \mathbf{N}$ can be denoted as $\mathbf{\Omega} = \mathbf{Q_1} - \mathbf{Q_2} = \bigcup_{N_i \in \mathbf{N}} C(P_{N_i}, R/2) - \bigcup_{N_i \in \mathbf{N}} D(P_{N_i}, R/2)$, where $C(P_{N_i}, R/2)$ and $D(P_{N_i}, R/2)$ represent the circle and the open disk centered at P_{N_i} with a radius of $R/2$, respectively. It is obvious to notice that $\mathbf{R_1} = \mathbf{Q_1}$ and $\mathbf{R_2} = \mathbf{Q_2'}$, which result in $\mathbf{S} = \mathbf{\Omega}$. It completes the proof. □

Lemma 2. *A simple closed curve is formed by the trajectory of the SPs.*

Proof. Based on Lemma 1, the trajectory of the SPs forms the border of the overlapped closed disks $\overline{D}(P_{N_i}, R/2)$ for all $N_i \in \mathbf{N}$. Moreover, the border of a closed filled 2D geometry is a simple closed curve according to Fact 1. Therefore, a simple closed curve is constructed by the trajectory of the SPs, e.g., the solid flower-shaped closed curve, as in Fig. 2. It completes the proof. □

Theorem 1. *The boundary finding problem (Problem 2) is resolved by the RUT scheme.*

Proof. Based on Lemma 2, the RUT scheme can draw a simple closed curve by rotating the rolling balls $RB_{N_i}(\mathbf{s}_i, R/2)$ hinged at P_{N_i} for all $N_i \in \mathbf{N}$. The closed curve can be divided into arc segments $S(\mathbf{s}_i, \mathbf{s}_j)$, where $\mathbf{s}_i$ is the starting SP associated with N_i, and $\mathbf{s}_j$ is the anchor point while rotating the $RB_{N_i}(\mathbf{s}_i, R/2)$ hinged at P_{N_i}. The arc segments $S(\mathbf{s}_i, \mathbf{s}_j)$ can be mapped into the unidirectional

edges $E_{ij} = (P_{N_i}, P_{N_j})$ for all N_i, $N_j \in \mathbf{U}$, where $\mathbf{U} \subseteq \mathbf{N}$. Due to the one-to-one mapping between $S(\mathbf{s}_i, \mathbf{s}_j)$ and E_{ij}, a simple unidirectional ring is constructed by E_{ij} for all N_i, $N_j \in \mathbf{U}$.

According to the RUT scheme, there does not exist any $N_i \in \mathbf{N}$ within the area traversed by the rolling balls, i.e., inside the light blue region, as in Fig. 2. For all $N_p \in \mathbf{N}$ located inside the simple unidirectional ring, the smallest distance from N_p to N_q, which is located outside of the ring, is greater than the SN's transmission range R. Therefore, there does not exist any $N_p \in \mathbf{N}$ inside the simple unidirectional ring that can communicate with $N_q \in \mathbf{N}$ located outside of the ring. Based on Definition 1, the set $\mathbf{U}$ is identical to the boundary set, i.e., $\mathbf{U} = \mathbf{B}$. It completes the proof. □

Theorem 2. *The void problem (Problem 1) in UDGs is solved by the GAR protocol with guaranteed packet delivery.*

Proof. With the existence of the void problem occurred at the void node N_V, the RUT scheme is utilized by initiating an SP ($\mathbf{s}_V$) with the rolling ball $RB_{N_V}(\mathbf{s}_V, R/2)$ hinged at N_V. The RUT scheme within the GAR protocol will conduct boundary (i.e., the set $\mathbf{B}$) traversal under the condition that $d(P_{N_i}, P_{N_D}) \geq d(P_{N_V}, P_{N_D})$ for all $N_i \in \mathbf{B}$. If the boundary within the underlying network is completely traveled based on Theorem 1, it indicates that the SNs inside the boundary (e.g., N_V) are not capable of communicating with those located outside of the boundary (e.g., N_D). The result shows that there does not exist a route from the void node (N_V) to the destination node (N_D), i.e., the existence of network partition. On the other hand, if there exists a node N_Y such that $d(P_{N_Y}, P_{N_D}) < d(P_{N_V}, P_{N_D})$ (as shown in Fig. 1), the GF algorithm will be adopted within the GAR protocol to conduct data delivery toward the destination node N_D. Therefore, the GAR protocol solves the void problem with guaranteed packet delivery, which completes the proof. □

4 Proposed GAR Protocol Realization

The implementation of the proposed GAR protocol is explained in this section. The format of the one-hop neighbor table $\mathbf{T}_{N_i}$, as defined in (1), is realized for the implementation purpose. $\mathbf{T}_{N_i}$ is considered a major information source in the localized routing protocols, which can be obtained via the neighbor information acquisition [29]. It is noticed that the one-hop neighbor table $\mathbf{T}_{N_i}$ is considered stable, while N_i is making its next-hop decision, i.e., $\mathbf{T}_{N_i}$ remains unchanged while N_i is determining the next-hop SN for packet transmission. Sections 4.1 and 4.2 describe the implementation aspect of the GAR algorithm, which consists of the GF and the RUT schemes. The proofs of correctness are illustrated in Section 4.3.

4.1 Implementation of GF Scheme

As described in Section 2, the GF scheme is considered a straightforward algorithm that only requires the implementation of the one-hop neighbor table $\mathbf{T}_{N_i}$. The next hop node can be found by the linear search of $\mathbf{T}_{N_i}$ if the void problem does not occur; otherwise, the RUT scheme will be adopted based on the proposed GAR protocol.

4.2 Implementation of RUT Scheme

4.2.1 Problem Statement for Implementation

As mentioned in Section 3.2, the GAR protocol changes its routing mode into the RUT scheme while the void problem occurs at N_V. The boundary traversal phase is conducted by assigning an SP (i.e., $\mathbf{s}_V$ as shown in Fig. 1) associated with the rolling ball $RB_{N_V}(\mathbf{s}_V, R/2)$ hinged at N_V. While there is no doubt regarding the description of boundary traversal, there can be considerable efforts required in order to realize the continuous rolling ball mechanism. A brute-force method can be adopted as a potential solution by rotating the rolling ball incrementally and verifying if a new SN has been encountered at each computing step. However, an infinite number of computational runs are required by adopting the brute-force method, which is considered impractical for realistic computing machines. Therefore, a feasible and efficient mechanism for the boundary traversal should be obtained in order to overcome the computational limitation.

4.2.2 Concept of Boundary Map

In order to resolve the implementation issue of the boundary traversal as mentioned above, a new parameter called BM (denoted as $\mathbf{M}_{N_i}$ for each N_i) is introduced in this section. Moreover, the BM $\mathbf{M}_{N_i}$ is mainly derived from the one-hop neighbor table $\mathbf{T}_{N_i}$ via the IMS method, as shown in Algorithm 1. Instead of diving into the IMS algorithm, the functionality of $\mathbf{M}_{N_i}$ is first explained.

Algorithm 1: *Indirect Map Searching Algorithm*

```
Data: R, P_{N_i}, T_{N_i}
Result: M_{N_i}, L_{N_i}
1  begin
2      M_{N_i} ← null
3      L_{N_i} ← null
4      if T_{N_i} ≠ ∅ then
5          foreach (id_{N_j}, P_{N_j}) ∈ T_{N_i} do
6              compute S^{SP}_{N_i ∘ N_j}(P_A, P_B) by R, P_{N_i}, and P_{N_j}
7              Ψ(P_A) ←
               [id_{N_j}, RIGHT, angle(P_A, P_{N_i}), FALSE, Ψ(P_B)]
8              Ψ(P_B) ←
               [id_{N_j}, LEFT, angle(P_B, P_{N_i}), FALSE, Ψ(P_A)]
9              wrap and insert Ψ(P_A) and Ψ(P_B) into L_{N_i}
10         end
11         sort(L_{N_i})
12         foreach ℓ_A ∈ L_{N_i} and ℓ_A.flag() = RIGHT do
13             ℓ_B ← ℓ_A.counterpart()
14             foreach ℓ_C ∈ L_{N_i} located between [ℓ_A, ℓ_B] do
15                 set Color for ℓ_C
16             end
17         end
18         foreach ℓ_B ∈ L_{N_i} and ℓ_B.flag() = LEFT do
19             if ℓ_B.color() = FALSE then
20                 ℓ_C ← ℓ_B.next()
21                 while ℓ_C.flag() = LEFT do
22                     ℓ_C ← ℓ_C.next()
23                 end
24                 get the SNs N_B and N_C from ℓ_B and ℓ_C
25                 create the direct mapping from N_B to N_C
26                 insert the mapping N_B → N_C into M_{N_i}
27             end
28         end
29     end
30 end
```

The purpose of the BM $\mathbf{M}_{N_i}$ is to provide a set of direct mappings between the input SNs and their corresponding

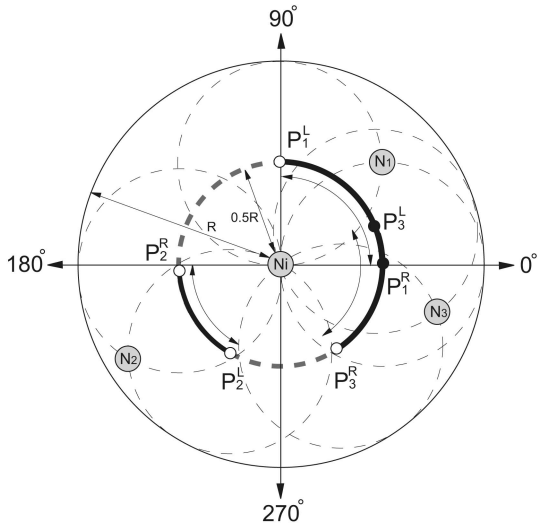

Fig. 3. The converged SP and non-SP arc segments with respect to N_i and the resulting BM.

output SNs with respect to N_i. Based on Theorem 1, the two adjacent communication links formed by the input node, the node N_i, and the corresponding output node within the RUT scheme consist part of the network boundary. Therefore, the direct mappings between the input SNs and their corresponding output SNs with respect to N_i lead to the so-called BM. An example is shown in Fig. 3 to illustrate the functionality of $\mathbf{M}_{N_i}$. Based on Definition 2, the rolling balls hinged at N_i can be constructed by rotating the dashed circle counterclockwise from N_1 to N_2. The SPs associated with the rolling balls (from Definition 3) result in the arc segment $S^{SP}_{N_i}(P_1^L, P_2^R)$ between the endpoints P_1^L and P_2^R, i.e., the dashed arc segment, as in Fig. 3. Similarly, the arc segment $S^{SP}_{N_i}(P_2^L, P_3^R)$ can be constructed by rotating the rolling balls (hinged at N_i) counterclockwise from N_2 to N_3. Based on the description as above, the following definitions are introduced.

Definition 4 (SP and non-SP arc segments). *Given an SN $N_i \in \mathbf{N}$ and a pair of points (P_A, P_B) on the circle (centered at P_{N_i} with a radius of $R/2$), an SP arc segment $S^{SP}_{N_i}(P_A, P_B)$ of N_i is defined by the arc from P_A to P_B counterclockwise where all points on this arc segment are SPs. Likewise, a non-SP arc segment $S^{\overline{SP}}_{N_i}(P_A, P_B)$ of N_i is defined by the endpoint-excluding arc from P_A to P_B counterclockwise, where all points on this arc segment are not SPs.*

Definition 5 (converged SP and non-SP arc segments). *Given an SP arc segment $S^{SP}_{N_i}(P_A, P_B)$, it is regarded as a converged SP arc segment if there does not exist any SP arc segment $S^{SP}_{N_i}(P_J, P_K)$ such that $S^{SP}_{N_i}(P_A, P_B) \subset S^{SP}_{N_i}(P_J, P_K)$. Similarly, a non-SP arc segment $S^{\overline{SP}}_{N_i}(P_A, P_B)$ is considered as a converged non-SP arc segment if there exists no other non-SP arc segment $S^{\overline{SP}}_{N_i}(P_J, P_K)$ such that $S^{\overline{SP}}_{N_i}(P_A, P_B) \subset S^{\overline{SP}}_{N_i}(P_J, P_K)$.*

It is noticed that the converged arc segments are defined to represent the combined arc segments, e.g., the converged non-SP arc segment $S^{\overline{SP}}_{N_i}(P_3^R, P_1^L)$ is formed by overlapping the non-SP segments $S^{\overline{SP}}_{N_i}(P_3^R, P_3^L)$ and $S^{\overline{SP}}_{N_i}(P_1^R, P_1^L)$, as shown in Fig. 3. As will be proven in Theorem 3, all incoming packets to N_i that are acquired from its neighbor node N_1 (which induces the rightmost endpoint P_1^L of the converged SP arc segment $S^{SP}_{N_i}(P_1^L, P_2^R)$) will be forwarded to its neighbor node N_2 (which results in the leftmost endpoint P_2^R of the same converged SP arc segment) under the counterclockwise rolling direction. Consequently, if all the converged SP arc segments of N_i can be obtained, the direct mappings between the input SNs and their corresponding output SNs with respect to N_i can also be constructed. As shown in Fig. 3, there exist two converged SP arc segments $S^{SP}_{N_i}(P_1^L, P_2^R)$ and $S^{SP}_{N_i}(P_2^L, P_3^R)$, where $S^{SP}_{N_i}(P_1^L, P_2^R)$ is constructed by the input SN N_1 and the corresponding output SN N_2 and $S^{SP}_{N_i}(P_2^L, P_3^R)$ is established by the input N_2 and the output N_3. As a result, the BM with respect to N_i can be obtained as $\mathbf{M}_{N_i} = \{(N_1 \rightarrow N_2), (N_2 \rightarrow N_3)\}$. Therefore, all packets from N_1 will be forwarded to N_2, while those from N_2 will be relayed to N_3 according to the BM.

4.2.3 Construction of Boundary Map

As mentioned above, the BM $\mathbf{M}_{N_i}$ can be constructed via the converged SP arc segments with respect to N_i. However, it is observed to be a difficult task for obtaining the converged SP arc segments directly in realization. An IMS algorithm is proposed in this section in order to acquire the BM for implementation. The definition of the neighbor-related non-SP arc segment and two associated properties are first introduced.

Definition 6 (neighbor-related non-SP arc segment). *A non-SP arc segment $S^{\overline{SP}}_{N_i}(P_A, P_B)$ of N_i is given. If there exists $N_j \in \mathbf{N}$ as a neighbor node of N_i such that an arc segment of $C(P_{N_i}, R/2)$ that lies inside the closed disk $\overline{D}(P_{N_j}, R/2)$ is identical to $S^{\overline{SP}}_{N_i}(P_A, P_B)$, this segment $S^{\overline{SP}}_{N_i}(P_A, P_B)$ is called a neighbor-related non-SP arc segment $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$, distinguished by N_j.*

Two properties that are related to the SP and non-SP arc segments are described as follows:

Property 1. *The circle $C(P_{N_i}, R/2)$ centered at P_{N_i} with a radius of $R/2$ is entirely composed by all the converged SP and non-SP arc segments of N_i.*

Proof. Based on Definitions 2 and 3, it can be observed that each point on the circle $C(P_{N_i}, R/2)$ must either be an SP or a non-SP. A number of adjacent SPs on $C(P_{N_i}, R/2)$ will establish an SP arc segment with respect to N_i; while there must exist the largest number of adjacent SPs such that the underlying SP arc segment is a converged SP arc segments with respect to N_i. Therefore, all the adjacent SPs on $C(P_{N_i}, R/2)$ will result in converged SP arc segments with respect to N_i. Similarly, all the adjacent non-SPs on $C(P_{N_i}, R/2)$ must be aggregated into converged non-SP arc segments with respect to N_i. On the other hand, the circle $C(P_{N_i}, R/2)$ is entirely composed by the SPs and non-SPs corresponding to N_i. Consequently, all the converged SP and non-SP arc segments of N_i will construct the entire circle $C(P_{N_i}, R/2)$. It completes the proof. □

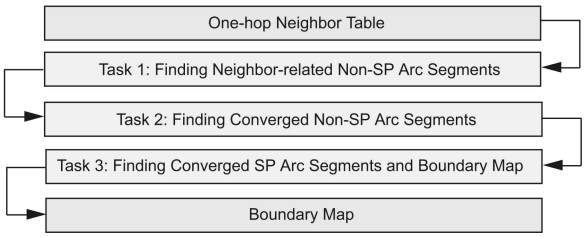

Fig. 4. The process flow of the IMS algorithm.

Property 2. *The union of all the neighbor-related non-SP arc segments with respect to N_i is equivalent to the union of all the converged non-SP arc segments with respect to N_i.*

Proof. This property will be proven by contradiction as follows: It is assumed that the union of all the neighbor-related non-SP arc segments corresponding to N_i is not equivalent to the union of all the converged non-SP arc segments with respect to N_i. Based on Definitions 4 and 5, and Property 1, it is stated that all the converged non-SP arc segments with respect to N_i result in the union of all the non-SPs on $C(P_{N_i}, R/2)$. Therefore, there must exist a non-SP P_J located on $C(P_{N_i}, R/2)$ such that it does not relate to any neighbor-related non-SP arc segments with respect to N_i, i.e., there does not exist any $N_k \in \mathbf{T}_{N_i}$ that lies inside the rolling ball $RB_{N_i}(P_J, R/2)$. However, based on Definitions 2 and 3, there should exist at least a node N_k within the rolling ball $RB_{N_i}(P_J, R/2)$ since P_J is a non-SP on $C(P_{N_i}, R/2)$, which contradicts with the previous statement. It completes the proof. □

The concept of the proposed IMS algorithm is described as follows: Based on Property 1, the converged SP arc segments for each N_i can be obtained by acquiring its corresponding converged non-SP arc segments, i.e., the complement arc segments on the circle $C(N_i, R/2)$. Moreover, according to Property 2, the converged non-SP arc segments of N_i can be acquired via the neighbor-related non-SP arc segments. Consequently, the problem of finding the converged SP arc segments with respect to N_i is transformed into the problem of obtaining the converged non-SP arc segments with respect to N_i, which can be acquired via merging the corresponding neighbor-related non-SP arc segments. The BM $\mathbf{M}_{N_i}$ can therefore be indirectly established by adopting the IMS method. The process flow of the IMS algorithm is summarized, as shown in Fig. 4.

In order to acquire and construct the BM $\mathbf{M}_{N_i}$, as shown in Algorithm 1, the IMS scheme is proposed. It is considered a localized algorithm where only three parameters are required within Algorithm 1, i.e., the maximum communication distance R, the position of N_i (P_{N_i}), and the one-hop neighbor table $\mathbf{T}_{N_i}$.

Table 1 summarizes the notations in the IMS algorithm, and the pseudocode of the IMS method, as shown in Algorithm 1, is explained as follows: Based on that in Fig. 4, the first task within the IMS algorithm is to identify each neighbor-related non-SP arc segment $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$ with respect to N_i that is distinguished by its neighbor N_j. Intuitively, it is feasible to utilize the two endpoints P_A and P_B to represent $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$, where each endpoint P_ζ (for $\zeta \in \{A, B\}$) can be characterized by an endpoint entry defined as

TABLE 1
Notations for IMS Algorithm

Notation	Description
R	Maximum Communication Distance
P_{N_i}	Position of N_i
$\mathbf{T}_{N_i}$	One-hop Neighbor Table of N_i
$\mathbf{M}_{N_i}$	Boundary Map of N_i
$\mathbf{L}_{N_i}$	Circular Doubly-linked List of N_i
(id_{N_j}, P_{N_j})	ID and Position of a Neighbor N_j
$S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$	Neighbor-related Non-SP Arc Segment
$\Psi(P_A), \Psi(P_B)$	Endpoint Entries of $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$
ℓ_A, ℓ_B, ℓ_C	List Items of $\mathbf{L}_{N_i}$

$$\Psi_{N_i}(P_\zeta) = [Id, Flag, Angle, Color, Counterpart]. \quad (3)$$

The parameter Id is utilized as the identification number of the corresponding neighbor SN for this entry. $Flag$ represents the endpoint type of this entry, which is denoted as either $RIGHT$ or $LEFT$ (e.g., the $Flag$ field of the right endpoint P_A is denoted as $RIGHT$, while that of the left endpoint P_B is indicated as $LEFT$). The $Angle$ field is adopted to represent the polar angle with respect to N_i by rotating counterclockwise from the x-axis. The $Color$ field is employed to indicate whether the endpoint P_ζ is a non-SP or not (i.e., $Color = TRUE$ denotes that P_ζ is a non-SP). The $Counterpart$ field provides the linkage to the counterpart endpoint entry that possesses the opposite $Flag$ value (e.g., the counterpart of $\Psi_{N_i}(P_A)$ is $\Psi_{N_i}(P_B)$, and vice versa). Therefore, the neighbor-related non-SP arc segment $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$ can be denoted by a pair of the endpoint entries as $[\Psi_{N_i}(P_A), \Psi_{N_i}(P_B)]$.

In order to store and to maintain the relative locations among the entire set of endpoints $\mathbb{P}$ for all the neighbor-related non-SP arc segments with respect to N_i, a circular doubly linked list [30] sorted by the polar angle is employed as

$$\mathbf{L}_{N_i} = \left\{ \ell_{N_i}(P_\zeta) \,|\, \ell_{N_i}(P_\zeta) = \begin{bmatrix} \Psi_{N_i}(P_\zeta) \\ Next \\ Prev \end{bmatrix}, \forall P_\zeta \in \mathbb{P} \right\}, \quad (4)$$

where the list item $\ell_{N_i}(P_\zeta)$ in $\mathbf{L}_{N_i}$ is composed of an endpoint entry $\Psi_{N_i}(P_\zeta)$ associated with two fields, $Next$ and $Prev$. The fields $Next$ and $Prev$ provide the addresses of the next and the previous entries of $\ell_{N_i}(P_\zeta)$ within $\mathbf{L}_{N_i}$. Considering the example, as shown in Fig. 3, there exist three neighbor-related non-SP arc segments with respect to N_i as $S^{\overline{SP}}_{N_i \circ N_1}(P_1^R, P_1^L)$, $S^{\overline{SP}}_{N_i \circ N_2}(P_2^R, P_2^L)$, and $S^{\overline{SP}}_{N_i \circ N_3}(P_3^R, P_3^L)$, which result in three pairs of endpoint entries as $[\Psi_{N_i}(P_1^R), \Psi_{N_i}(P_1^L)]$, $[\Psi_{N_i}(P_2^R), \Psi_{N_i}(P_2^L)]$, and $[\Psi_{N_i}(P_3^R), \Psi_{N_i}(P_3^L)]$, respectively. Consequently, the linked list $\mathbf{L}_{N_i}$ will contain these endpoint entries as $\mathbf{L}_{N_i} = [\ell_{N_i}(P_1^R), \ell_{N_i}(P_3^L), \ell_{N_i}(P_1^L), \ell_{N_i}(P_2^R), \ell_{N_i}(P_2^L), \ell_{N_i}(P_3^R)]$. It is noticed that the endpoint entries are connected by their $Next$ and $Prev$ fields, while these entries are sorted by the corresponding $Angle$ field in the ascending order. By taking the entry $\ell_{N_i}(P_3^L)$ as an example (as in Fig. 3), the parameter $Next$ refers to the

address of $\ell_{N_i}(P_1^L)$, while the $Prev$ field is denoted as the address of $\ell_{N_i}(P_1^R)$. Since $\mathbf{L}_{N_i}$ is a circular doubly linked list, the parameter $Prev$ for $\ell_{N_i}(P_1^R)$ points to the address of $\ell_{N_i}(P_3^R)$, while the $Next$ field of $\ell_{N_i}(P_3^R)$ refers to $\ell_{N_i}(P_1^R)$. In the case that two endpoint entries share the same $Angle$ value, the $Flag$ field will be utilized to provide the order of the entries, i.e., by taking the entry with $Flag = LEFT$ first and $Flag = RIGHT$ as the next entry in $\mathbf{L}_{N_i}$. It is noticed that the order of the endpoint entries within $\mathbf{L}_{N_i}$ is crucial for the construction of $\mathbf{M}_{N_i}$. The construction and sorting mechanisms of $\mathbf{L}_{N_i}$, as described above, are summarized at Lines 1-11 in Algorithm 1, which completes the first task in Fig. 4.

The next task within the IMS algorithm, as in Fig. 4, is to merge all the neighbor-related non-SP arc segments into the converged non-SP arc segments with respect to N_i. Given a neighbor-related non-SP arc segment $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$ represented by $[\Psi_{N_i}(P_A), \Psi_{N_i}(P_B)]$, the combining process is to assign the $Color$ field of each endpoint entry $\Psi_{N_i}(P_C)$, which is located within $[\Psi_{N_i}(P_A), \Psi_{N_i}(P_B)]$ to become colored, i.e., with the $TRUE$ value. This indicates that the endpoint P_C is merged into the neighbor-related non-SP arc segment $S^{\overline{SP}}_{N_i \circ N_j}(P_A, P_B)$. Based on Property 2, all the remaining uncolored endpoint entries consequently become the endpoints of the converged non-SP arc segments with respect to N_i. As shown in Fig. 3, $\Psi_{N_i}(P_1^R)$ and $\Psi_{N_i}(P_3^L)$ are the colored endpoint entries, while the uncolored ones are $\Psi_{N_i}(P_1^L)$, $\Psi_{N_i}(P_2^R)$, $\Psi_{N_i}(P_2^L)$, and $\Psi_{N_i}(P_3^R)$. It is noticed that the uncolored endpoint entries will establish the converged non-SP arc segments with respect to N_i, which are denoted as $[\Psi_{N_i}(P_3^R), \Psi_{N_i}(P_1^L)]$ and $[\Psi_{N_i}(P_2^R), \Psi_{N_i}(P_2^L)]$. The combining process for the neighbor-related non-SP arc segments is summarized at Lines 12-17 in Algorithm 1, completing the second task in Fig. 4.

Within the last task listed in Fig. 4, based on Property 1, all the converged SP arc segments with respect to N_i can be obtained by excluding all the converged non-SP arc segments with respect to N_i on the circle $C(P_{N_i}, R/2)$. For each two sequential converged non-SP arc segments $S^{\overline{SP}}_{N_i}(P_A, P_B)$ and $S^{\overline{SP}}_{N_i}(P_C, P_D)$ (denoted as $[\Psi_{N_i}(P_A), \Psi_{N_i}(P_B)]$ and $[\Psi_{N_i}(P_C), \Psi_{N_i}(P_D)]$), the resulting converged SP arc segment $S^{SP}_{N_i}(P_B, P_C)$ can be obtained as $[\Psi_{N_i}(P_B), \Psi_{N_i}(P_C)]$ in view of endpoint entries. Consequently, the converged SP arc segment $S^{SP}_{N_i}(P_B, P_C)$ with respect to N_i can be acquired by taking each uncolored endpoint entry $\Psi_{N_i}(P_B)$ with $Flag = LEFT$ associated with the next endpoint entry $\Psi_{N_i}(P_C)$ with $Flag = RIGHT$. As shown in Fig. 3, the converged SP arc segments with respect to N_i are obtained as $S^{SP}_{N_i}(P_1^L, P_2^R)$ and $S^{SP}_{N_i}(P_2^L, P_3^R)$, which are denoted as $[\Psi_{N_i}(P_1^L), \Psi_{N_i}(P_2^R)]$ and $[\Psi_{N_i}(P_2^L), \Psi_{N_i}(P_3^R)]$. For each converged SP arc segment $S^{SP}_{N_i}(P_B, P_C)$ denoted as $[\Psi_{N_i}(P_B), \Psi_{N_i}(P_C)]$, the direct mapping (i.e., $N_B \rightarrow N_C$) between input/output SNs (as mentioned in Section 4.2.2) can therefore be obtained by mapping N_B specified in $\Psi_{N_i}(P_B)$ to N_C denoted in $\Psi_{N_i}(P_C)$. The acquisition of the converged SP arc segments and the construction of the BM $\mathbf{M}_{N_i}$ are summarized at Lines 18-30 in Algorithm 1.

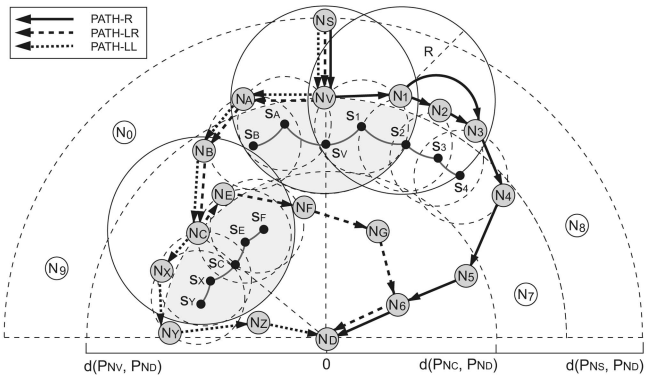

Fig. 5. The HCR and the IN mechanisms.

4.3 Proof of Correctness

Theorem 3. *Given a converged SP arc segment $S^{SP}_{N_i}(P_S, P_T)$ with respect to N_i, where 1) the rightmost endpoint P_S is an SP for both N_i and its neighbor N_S, and 2) the leftmost endpoint P_T is an SP for both N_i and its neighbor N_T, respectively. All incoming packets to N_i that are acquired from its neighbor node N_S will be forwarded to N_T.*

Proof. Based on Definitions 4 and 5, a converged SP arc segment $S^{SP}_{N_i}(P_S, P_T)$ is an arc segment composed by some of the SPs with respect to N_i. According to Lemma 2, a simple closed curve is constructed by the trajectory of the SPs. In order to form the closed curve, there must exist other converged SP arc segments contributed by other SNs that are connected to the endpoints P_S and P_T. In other words, the endpoints P_S and P_T must also be owned by one of N_i's neighbor, respectively. On the contrary, the other points on this converged SP arc segment $S^{SP}_{N_i}(P_S, P_T)$ should only be contributed by N_i based on Definitions 2 and 3. Moreover, it is intuitive to observe (from Definition 3) that the distances between the SNs related to the same endpoint SP (i.e., either P_S or P_T) must be located in their transmission ranges. By adopting the RUT scheme (as stated in Theorems 1 and 2) starting from N_S, the rolling ball will be traversed counterclockwise via N_i to N_T. This corresponds to the situation that all the packets coming from N_S to N_i will be forwarded to N_T. It completes the proof. □

5 Enhanced Mechanisms for Proposed GAR Protocol

In order to enhance the routing efficiency of the proposed GAR protocol, three mechanisms are proposed in this section, i.e., the HCR, the IN, and the PUC schemes. These three mechanisms are described as follows:

5.1 Hop Count Reduction (HCR) Mechanism

Based on the rolling-ball traversal within the RUT scheme, the selected next-hop nodes may not be optimal by considering the minimal HC criterion. Excessive routing delay associated with power consumption can occur if additional hop nodes are traversed by adopting the RUT scheme. As shown in Fig. 5, the void node N_V starts the RUT scheme by selecting N_1 as its next hop node with the

counterclockwise rolling direction, while N_2 and N_3 are continuously chosen as the next hop nodes. Considering the case that N_3 is located within the same transmission range of N_1, it is apparently to observe that the packets can directly be transmitted from N_1 to N_3. Excessive communication waste can be preserved without conducting the rerouting process to N_2. Moreover, the boundary set $\mathbf{B}$ forms a simple unidirectional ring based on Theorem 1, which indicates that a node's next-hop SN can be uniquely determined if its previous hop SN is already specified. For instance (as in Fig. 5), if N_V is the previous node of N_1, N_1's next hop node N_2 is uniquely determined, i.e., the transmission sequences of every three nodes (e.g., $\{N_V \rightarrow N_1 \rightarrow N_2\}$ or $\{N_1 \rightarrow N_2 \rightarrow N_3\}$) can be uniquely defined.

According to the concept as stated above, the HCR mechanism is to acquire the information of the next few hops of neighbors under the RUT scheme by listening to the same forwarded packet. It is also worthwhile to notice that the listening process does not incur additional transmission of control packets. As shown in Fig. 5, N_1 chooses N_2 as its next-hop node for packet forwarding. while N_2 selects N_3 as the next hop node in the same manner. Under the broadcast nature, N_1 will listen to the same packets in the forwarding process from N_2 to N_3. By adopting the HCR mechanism, N_1 will therefore select N_3 as its next hop node instead of choosing N_2 while adopting the original RUT scheme. Consequently, N_1 will initiate its packet forwarding process to N_3 directly by informing the RUT scheme that the rerouting via N_2 can be skipped.

5.2 Intersection Navigation (IN) Mechanism

The IN mechanism is utilized to determine the rolling direction in the RUT scheme while the void problem occurs. It is noticed that the selection of rolling direction (i.e., either counterclockwise or clockwise) does not influence the correctness of the proposed RUT scheme to solve Problem 2, as in Theorem 1. However, the routing efficiency may be severely degraded if a comparably longer routing path is selected at the occurrence of a void node. The primary benefit of the IN scheme is to choose a feasible rolling direction while a void node is encountered. Consequently, smaller rerouting HCs and packet transmission delay can be achieved.

Based on the transmission pair (N_S, N_D), as shown in Fig. 5, N_V and N_C become the void nodes within the network topology. There exist three potential paths from N_S to N_D by adopting the RUT scheme, i.e., PATH-R, PATH-LR, and PATH-LL. The suffixes R, LR, and LL represent the sequences of the adopted rolling direction at each encountered void node, where the symbol R is denoted as counterclockwise rolling direction, and L represents clockwise direction. It is noted that the suffix with two symbols indicates that two void nodes are encountered within the path. The entire node traversal for each path is as follows: $\text{PATH-R} = \{N_S, N_V, N_1, N_3, N_4, N_5, N_6, N_D\}$, $\text{PATH-LR} = \{N_S, N_V, N_A, N_B, N_C, N_E, N_F, N_G, N_6, N_D\}$, and $\text{PATH-LL} = \{N_S, N_V, N_A, N_B, N_C, N_X, N_Y, N_Z, N_D\}$. Different HCs are observed with each path as $\text{HC(PATH-R)} = 7$, $\text{HC(PATH-LR)} = 9$, and $\text{HC(PATH-LL)} = 8$.

The main objective of the IN scheme is to monitor the number of HC such that the path with the shortest HC can be selected, i.e., PATH-R in this case. A navigation map control packet (NAV_MAP) defined in the IN scheme is utilized to indicate the rolling direction while the void node is encountered. For example, two NAV_MAP packets are initiated after N_V is encountered, where $\text{NAV\_MAP} = \{\text{R}\}$ is delivered via the counterclockwise direction to N_D, and $\text{NAV\_MAP} = \{\text{L}\}$ is carried with the clockwise direction. It is noticed that the HC associated with each navigation path is also recorded within the NAV_MAP packets. As the second void node N_C is observed, the control message $\text{NAV\_MAP} = \{\text{L}\}$ is transformed into two different navigation packets (i.e., $\text{NAV\_MAP} = \{\text{LR}\}$ and $\text{NAV\_MAP} = \{\text{LL}\}$), which traverse the two different rolling directions toward N_D. As a result, the destination node N_D will receive several NAV_MAP packets at different time instants associated with the on-going transmission of the data packets. The NAV_MAP packet with the shortest HC value (i.e., $\text{NAV\_MAP} = \{\text{R}\}$ in this case) will be selected as the targeting path. Therefore, the control packet with $\text{NAV\_MAP} = \{\text{R}\}$ will be traversed from N_D back to N_S in order to notify the source node N_S with the shortest path for packet transmission. After acquiring the NAV_MAP information, N_S will conduct its remaining packet delivery based on the corresponding rolling direction. Considerable routing efficiency can be preserved as a shorter routing path is selected by adopting the IN mechanism.

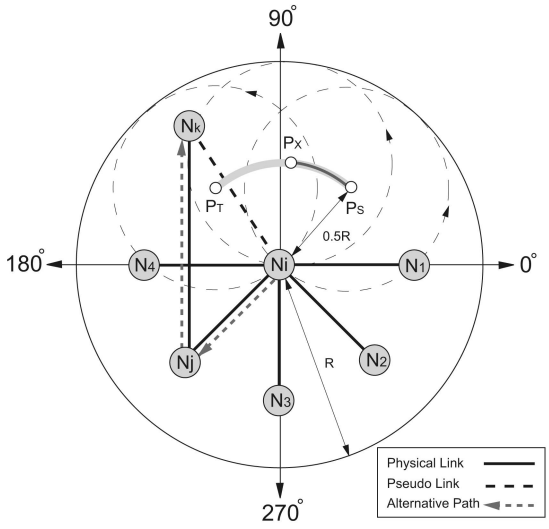

Fig. 6. The PUC mechanism.

5.3 Partial UDG Construction (PUC) Mechanism

The PUC mechanism is targeted to recover the UDG linkage of the boundary node N_i within a non-UDG network. The boundary nodes within the proposed GAR protocol are defined as the SNs that are utilized to handle the packet delivery after encountering the void problem. As shown in Fig. 6, node N_i is considered a boundary node since the converged SP arc segment $S_{N_i}^{SP}(P_S, P_T)$ exists after N_i conducts the proposed IMS algorithm by the input of the current one-hop neighbors $\{N_1, N_2, N_3, N_4, N_j\}$. It is noted that the boundary nodes consist of a portion of the network SNs. Therefore, conducting the PUC mechanism only by the boundary nodes can conserve network resources than most

of the existing flooding-based schemes that require information from all the network nodes.

The physical links of an exemplified topology are identified by the black solid lines, as shown in Fig. 6. It is considered that the boundary node N_i does not possess full UDG linkages since a node N_k within N_i's transmission range can not directly communicate with N_i. The proposed PUC mechanism will be initiated at the boundary node N_i under the non-UDG networks as follows: Initially, N_i broadcasts the PUC_REQ control packet containing its neighbor list for requesting the recovery of UDG linkages. After the neighbor N_j receives the PUC_REQ packet, N_j's neighbor table will be examined to verify if there exists any neighbor node N_k that is not in the neighbor list of the PUC_REQ packet but is actually located within the transmission range of N_i. In the case that such node N_k is observed, N_j will initiate a feedback message, i.e., the PUC_REP control packet, in order to inform N_i that a pseudo link from N_i to N_k should be constructed via the alternative paths of the two physical links from N_i to N_j and from N_j to N_k. Therefore, the UDG linkage of N_i can be recovered, which results in the current one-hop neighbors of N_i as $\{N_1, N_2, N_3, N_4, N_j, N_k\}$, while the converged SP arc segment $S^{SP}_{N_i}(P_S, P_T)$ will be changed into $S^{SP}_{N_i}(P_S, P_X)$.

6 Performance Evaluation

The performance of the proposed GAR algorithm is evaluated and compared with other existing localized schemes via simulations, including the reference GF algorithm, the planar graph-based GPSR and GOAFR++ schemes, and the UDG-based BOUNDHOLE algorithm. It is noted that the GPSR and GOAFR++ schemes that adopt the GG planarization technique to planarize the network graph are represented as the GPSR(GG) and GOAFR++(GG) algorithms, while the variants of these two schemes with the CLDP planarization algorithm are denoted as the GPSR(CLDP) and GOAFR++(CLDP) protocols. The random topology is considered in both two different types of network simulations as follows: 1) the pure UDG network as the ideal case, and 2) the non-UDG network for realistic network environment. Furthermore, the GAR protocol with the enhanced mechanisms (i.e., the HCR, the IN, and the PUC schemes) is also implemented, which is denoted as the GAR-E algorithm. The simulations are conducted in the network simulator (NS-2, [31]) with wireless extension, using the IEEE 802.11 DCF as the MAC protocol. The parameters utilized in the simulations are listed, as shown in Table 2, and the following five performance metrics are utilized in the simulations for performance comparison:

1. *Packet arrival rate.* The ratio of the number of received data packets to the number of total data packets sent by the source.
2. *Average end-to-end delay.* The average time elapsed for delivering a data packet within a successful transmission.
3. *Path efficiency.* The ratio of the number of total HCs within the entire routing path over the number of HCs for the shortest path.

TABLE 2
Simulation Parameters

Parameter Type	Parameter Value
Network Area	1000 x 800 m^2
Simulation Time	150 sec
Transmission Range	250 m
Traffic Type	Constant Bit Rate (CBR)
Data Rate	12 Kbps
Size of Data Packets	512 Bytes
Node Degree	17.5
Communication Pairs	3
Number of Void Blocks	3
Void Width	300 m
Void Height	150, 225, 300, 375, 450 m

4. *Communication overhead.* The average number of transmitted control bytes per second, including both the data packet header and the control packets.
5. *Energy consumption.* The energy consumption for the entire network, including transmission energy consumption for both the data and control packets under the bit rate of 11 megabits per second (Mbps) and the transmitting power of 15 dBm for each SN.

The simulation scenario is explained as follows: The SNs are randomly deployed with the node degree of 17.5 in the network, where the node degree is defined as the average number of nodes within a transmission range. Three pairs of source and destination nodes are respectively located around the left and the right boundaries of the network area. There also exist three equal-height void blocks of width 300 meters that are randomly placed in the network in order to simulate the occurrence of void problems. In other words, there are SNs around the peripheral of the void blocks, while none of the nodes is situated inside the void blocks. The simulations of the performance metrics versus the void height, i.e., the height of each void block, are conducted and compared with other baseline protocols under the UDG and the non-UDG networks. The non-UDG network is obtained by randomly removing some of the communication links within the original UDG network for violating the properties of the UDG setting.

6.1 Simulation Results for UDG Networks

Figs. 7a, 7b, 7c, 7d, and 7e present the performance comparison between these six algorithms with different void heights under the UDG network. As shown in Fig. 7a, both the GAR-based algorithms and the planar graph-based GPSR(GG) and GOAFR++(GG) protocols can achieve 100 percent delivery rate owing to their design nature with guaranteed packet delivery. The BOUNDHOLE and GF algorithms result in lowered delivery ratio due to the occurrence of routing loop and the ignorance of the void problem, respectively. Furthermore, with the augmentation of void height, decreased packet delivery rate can be observed from both the BOUNDHOLE and GF schemes since the probability of encountering the void problem is enlarged.

The performance of the average end-to-end delay versus the void height is shown in Fig. 7b. The smallest end-to-end delay can be found in the GF algorithm,

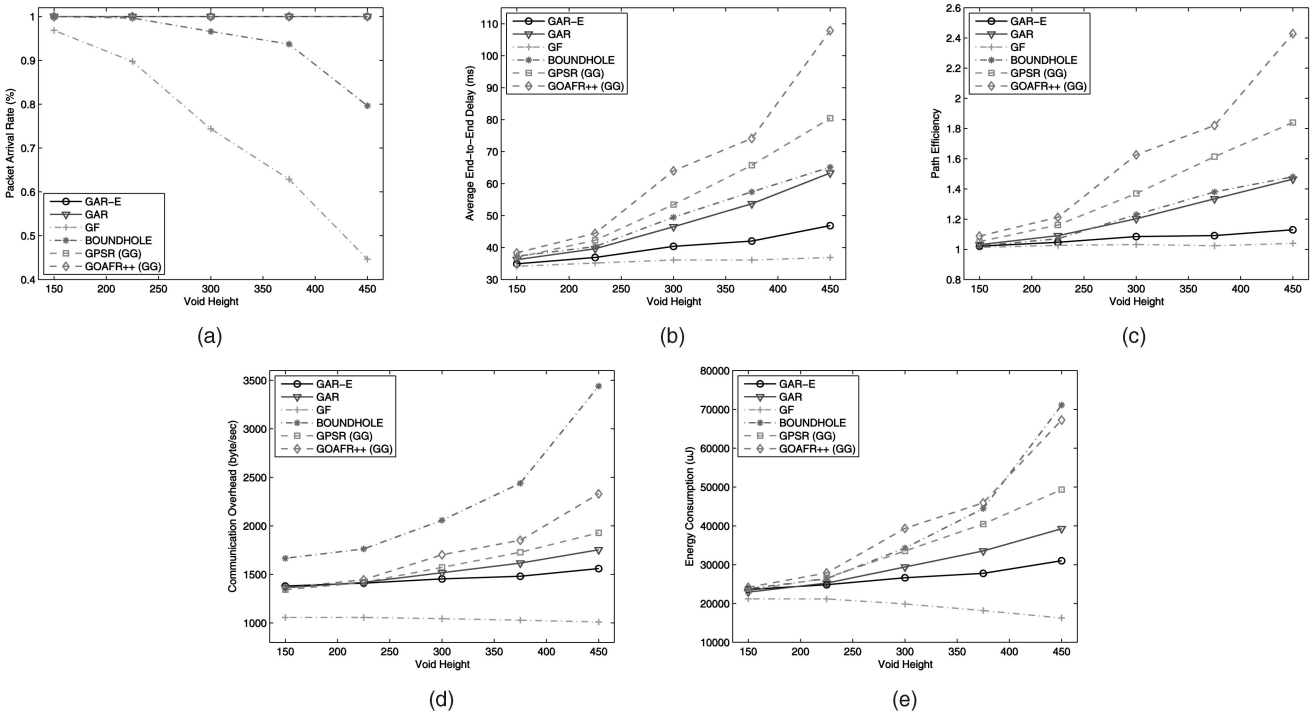

Fig. 7. Evaluation of performance metrics versus the void height for random UDG networks. (a) Packet arrival rate (percent). (b) Average end-to-end delay (ms). (c) Path efficiency. (d) Communication overhead (byte/sec). (e) Energy consumption (uJ).

owing to the negligence of the void problem, while the GOAFR++(GG) scheme results in the largest delay value due to its bounding techniques [16], [21] that in general cause the back-and-forth forwarding attempts around the large void block. The planar graph-based GPSR(GG) and GOAFR++(GG) schemes possess additional delay in comparison with the proposed UDG-based GAR and GAR-E protocols, owing to the required unnecessary forwarding nodes, as illustrated in Fig. 1. With the adoption of both the HCR and IN mechanisms, the most feasible end-to-end delay performance can be observed from the proposed GAR-E protocol in comparison with the other schemes. It is also noted that the end-to-end delays from all the algorithms will be increased with the augmentation of the void height, which can be attributed to the enlarged number of forwarding hops for boundary traversal. Owing to the closely related characteristics with the end-to-end delay performance, the path efficiency obtained from these schemes follows similar trends, as can be observed in Fig. 7c.

Figs. 7d and 7e illustrate the performance comparisons for communication overhead and energy consumption versus the void height. Except for the BOUNDHOLE scheme, the performance trends from all the other protocols can be observed to be similar with those from the path efficiency in Fig. 7c due to the elongated routing path. Excessive communication overhead associated with more energy consumption will be produced from the BOUNDHOLE scheme comparing with the other algorithm, which can be attributed to its usage of excessive header bytes for preventing the routing loops. It is noted that the decreasing trend within the GF method is primarily due to its relatively low packet delivery ratio, which results in less communication overhead and energy consumption. It is also noticed that even though the GAR-E scheme requires additional NAV_MAP control packets for achieving the IN mechanism, the total required communication overhead and energy consumption are smaller than those from the GAR method due to its comparably smaller rerouting number of HCs. Therefore, except for the reference GF scheme, it can be expected that the GAR-E algorithm possesses the lowest communication overhead and the energy consumption, which support the merits of the protocol design.

6.2 Simulation Results for Non-UDG Networks

Fig. 8a shows the performance comparison for packet arrival rate versus the void height under the non-UDG network. With the adoption of the PUC mechanism, 100 percent of packet arrival rate can be achieved by exploiting the proposed GAR-E protocol. Moreover, both the GPSR(CLDP) and GOAFR++(CLDP) schemes can also attain the same delivery rate. Nevertheless, these CLDP-enabled schemes will introduce extremely high communication overhead, as illustrated in Fig. 8d. With the augmentation of the void height, it is intuitive to observe that the packet arrival rate obtained from the remaining algorithms will be decreased owing to the increasing severity of the void problem.

The performance comparisons for the average end-to-end delay and the path efficiency are shown in Figs. 8b and 8c. Owing to the guaranteed packet delivery rate, the GPSR(CLDP) and GOAFR++(CLDP) schemes will result in larger delay and worse path efficiency compared to their counterparts, i.e., the GPSR(GG) and GOAFR++(GG) protocols. Other similar results can be found in Figs. 7b and 7c, respectively. It is observed that the proposed GAR-E

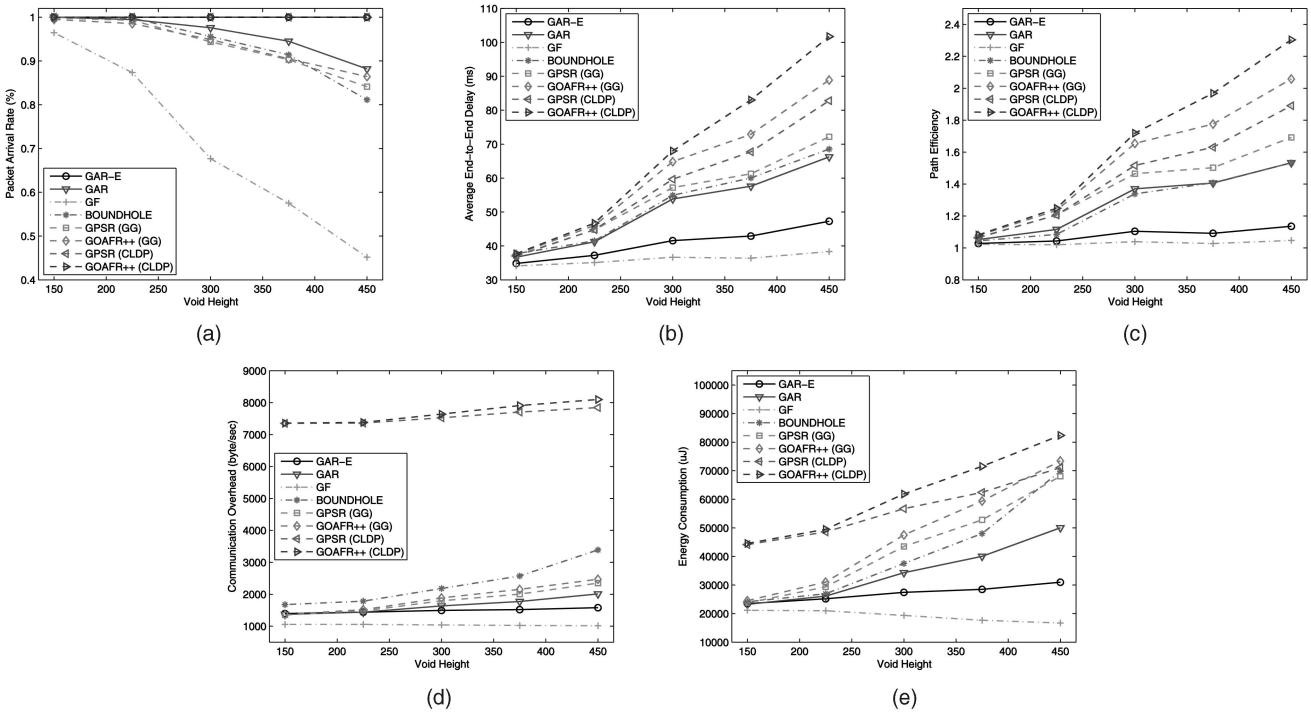

Fig. 8. Evaluation of performance metrics versus the void height for random non-UDG networks. (a) Packet arrival rate (percent). (b) Average end-to-end delay (ms). (c) Path efficiency. (d) Communication overhead (byte/sec). (e) Energy consumption (uJ).

scheme can still provide better routing efficiency comparing with other algorithms under the non-UDG networks.

Figs. 8d and 8e present the performance comparisons for the communication overhead and the energy consumption versus the void height. It is especially noticed that extremely high-communication overheads are observed within the GPSR(CLDP) and GOAFR++(CLDP) schemes in comparison with the other six protocols. The main reason is that according to the CLDP algorithm, all communication links will be probed and traversed via additional control packets in order to fulfill the required tasks for planarization. Other similar results can be found in Figs. 7d and 7e, respectively. As can be expected, except for the GF scheme, lowered communication overhead and energy consumption are acquired by the GAR-based algorithms in comparison with the other methods. The merits of the proposed GAR-E scheme can therefore be observed under the non-UDG networks.

7 Conclusion

In this paper, a UDG-based GAR protocol is proposed to resolve the void problem incurred by the conventional GF algorithm. The RUT scheme is adopted within the GAR protocol to solve the boundary finding problem, which results in guaranteed delivery of data packets under the UDG networks. The BM and the IMS are also proposed to conquer the computational problem of the rolling mechanism in the RUT scheme, forming the direct mappings between the input/output nodes. The correctness of the RUT scheme and the GAR algorithm is properly proven. The HCR and the IN mechanisms are proposed as the delay-reducing schemes for the GAR algorithm, while the PUC mechanism is utilized to generate the required topology for the RUT scheme under the non-UDG networks. All these enhanced mechanisms associated with the GAR protocol are proposed as the enhanced GAR (GAR-E) algorithm that inherits the merit of guaranteed delivery. The performance of both the GAR and GAR-E protocols is evaluated and compared with existing localized routing algorithms via simulations. The simulation study shows that the proposed GAR and GAR-E algorithms can guarantee the delivery of data packets under the UDG network, while the GAR-E scheme further improves the routing performance with reduced communication overhead under different network scenarios.

Acknowledgments

This work was in part funded by the MOE ATU Program 95W803C, NSC 96-2221-E-009-016, MOEA 96-EC-17-A-01-S1-048, the MediaTek research center at National Chiao Tung University, and the Universal Scientific Industrial (USI), Taiwan.

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM MobiCom*, pp. 263-270, Aug. 1999.

[2] G.G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Technical Report ISI/RR-87-180, Information Sciences Inst., Mar. 1987.

[3] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom*, pp. 243-254, Aug. 2000.

[4] I. Stojmenović and X. Lin, "Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023-1032, Oct. 2001.

[5] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *IEEE Personal Comm. Magazine,* vol. 8, no. 1, pp. 48-57, Feb. 2001.
[6] D. Chen and P.K. Varshney, "On-Demand Geographic Forwarding for Data Delivery in Wireless Sensor Networks," *Elsevier Computer Comm.,* vol. 30, no. 14-15, pp. 2954-2967, Oct. 2007.
[7] I. Stojmenović, M. Russell, and B. Vukojevic, "Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks," *Proc. IEEE Int'l Conf. Parallel Processing (ICPP '00),* pp. 173-180, Aug. 2000.
[8] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS '03),* pp. 46-55, May 2003.
[9] V.C. Giruka and M. Singhal, "Angular Routing Protocol for Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems Workshops (ICDCSW '05),* pp. 551-557, June 2005.
[10] W.J. Liu and K.T. Feng, "Largest Forwarding Region Routing Protocol for Mobile Ad Hoc Networks," *Proc. IEEE Global Comm. Conf. (GLOBECOM '06),* pp. 1-5, Nov. 2006.
[11] L. Zou, M. Lu, and Z. Xiong, "A Distributed Algorithm for the Dead End Problem of Location Based Routing in Sensor Networks," *IEEE Trans. Vehicular Technology,* vol. 54, no. 4, pp. 1509-1522, July 2005.
[12] N. Arad and Y. Shavitt, "Minimizing Recovery State in Geographic Ad-Hoc Routing," *Proc. ACM MobiHoc '06,* pp. 13-24, May 2006.
[13] S. Chen, G. Fan, and J.H. Cui, "Avoid "Void" in Geographic Routing for Data Aggregation in Sensor Networks," *Int'l J. Ad Hoc and Ubiquitous Computing,* vol. 1, no. 4, pp. 169-178, 2006.
[14] D.D. Couto and R. Morris, "Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding," Technical Report MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.
[15] J. Na, D. Soroker, and C.K. Kim, "Greedy Geographic Routing Using Dynamic Potential Field for Wireless Ad Hoc Networks," *IEEE Comm. Letters,* vol. 11, no. 3, pp. 243-245, Mar. 2007.
[16] H. Frey and I. Stojmenović, "On Delivery Guarantees of Face and Combined Greedy Face Routing in Ad Hoc and Sensor Networks," *Proc. ACM MobiCom '06,* pp. 390-401, Sept. 2006.
[17] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *ACM/Kluwer Wireless Networks,* vol. 7, no. 6, pp. 609-616, Nov. 2001.
[18] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. Canadian Conf. Computational Geometry (CCCG '99),* pp. 51-54, Aug. 1999.
[19] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically Optimal Geometric Mobile Ad-Hoc Routing," *Proc. Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (Dial-M '02),* pp. 24-33, Sept. 2002.
[20] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," *Proc. ACM MobiHoc '03,* pp. 267-278, June 2003.
[21] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad-Hoc Routing: Of Theory and Practice," *Proc. ACM Symp. Principles of Distributed Computing (PODC '03),* pp. 63-72, July 2003.
[22] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '05),* pp. 147-158, Nov. 2005.
[23] Q. Fang, J. Gao, and L. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," *Proc. IEEE INFOCOM '04,* pp. 2458-2468, Mar. 2004.
[24] D.B. West, *Introduction to Graph Theory,* second ed. Prentice Hall, 2000.
[25] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology,* vol. 18, no. 3, pp. 259-278, Sept. 1969.
[26] G.T. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition,* vol. 12, no. 4, pp. 261-268, 1980.
[27] Y.J. Kim, R. Govindan, B. Karp, and S. Shenker, "On the Pitfalls of Geographic Face Routing," *Proc. ACM/SIGMOBILE Joint Workshop Foundations of Mobile Computing (DIALM-POMC '05),* pp. 34-43, Sept. 2005.
[28] S. Datta, I. Stojmenović, and J. Wu, "Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Kluwer Cluster Computing,* vol. 5, no. 2, pp. 169-178, 2002.
[29] V.C. Giruka and M. Singhal, "Hello Protocols for Ad-Hoc Networks: Overhead and Accuracy Tradeoffs," *Proc. IEEE Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '05),* pp. 354-361, June 2005.
[30] E. Horowitz, S. Sahni, and D. Mehta, *Fundamentals of Data Structures in C++,* second ed. Silicon Press, 2006.
[31] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwak, K. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of Detail in Wireless Network Simulation," *Proc. SCS Multiconf. Distributed Simulation,* pp. 3-11, Jan. 2001.

Wen-Jiunn Liu received the BS degree from National Chiao Tung University, Hsinchu, Taiwan, ROC, in June 2005. Since September 2005, he has been a PhD candidate in the Department of Communication Engineering, National Chiao Tung University. His current research interests include the MAC and network protocol design for mobile ad hoc networks, wireless sensor networks, and broadband wireless networks. He is a student member of the IEEE.

Kai-Ten Feng received the BS degree from National Taiwan University, Taipei, in 1992, the MS degree from the University of Michigan, Ann Arbor, in 1996, and the PhD degree from the University of California, Berkeley, in 2000. Since August 2007, he has been with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, as an associate professor. He was an assistant professor with the same department between February 2003 and July 2007. He was with the OnStar Corp., a subsidiary of General Motors Corporation, as an in-vehicle development manager/senior technologist between 2000 and 2003, working on the design of future Telematics platforms and the in-vehicle networks. His current research interests include cooperative and cognitive networks, mobile ad hoc and sensor networks, embedded system design, wireless location technologies, and Intelligent Transportation Systems (ITSs). He received the Best Paper Award from the IEEE Vehicular Technology Conference Spring 2006, which ranked his paper first among the 615 accepted papers. He is also the recipient of the Outstanding Young Electrical Engineer Award in 2007 from the Chinese Institute of Electrical Engineering (CIEE). He has served on the technical program committees of VTC, ICC, and APWCS. He is a member of the IEEE.