15

Greedy Anti-Void Forwarding Strategies for Wireless Sensor Networks

Wen-Jiunn Liu and Kai-Ten Feng
Department of Communication Engineering, National Chiao Tung University
Taiwan, R.O.C.

1. Introduction

A wireless sensor network (WSN) consists of sensor nodes (SNs) with wireless communication capabilities for specific sensing tasks. Each SN maintains connectivity and exchanges messages between the decentralized nodes in the multi-hop manners. A source node can communicate with its destination via a certain number of relaying nodes, which consequently enlarges the wireless coverage of the source node. In conventional multi-hop routing algorithms, either the proactive or reactive schemes, significant amounts of routing tables and control packets are required for the construction of routing paths. Due to the limited available resources, efficient design of localized multi-hop routing protocols (Estrin et al., 1999) 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 (Finn, 1987) is considered a superior localized scheme with its low routing overheads, which is fit for conducting the routing task of WSNs. However, the void problem (Karp & Kung, 2000) that occurs within the GF technique will fail 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, the intuitive schemes as proposed in the research work (Stojmenovi´c & Lin, 2001) construct a two-hop neighbor table for implementing the GF algorithm. The network flooding mechanism is adopted 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 (Stojmenović & Lin, 2001), DFS (Stojmenović et al., 2000), and SPEED (He et al., 2003). The routing schemes as proposed by ARP (Giruka & Singhal, 2005) and LFR (Liu & Feng, 2006) memorize the routing path after the void problem takes place. Moreover, other routing protocols, such as PAGER (Zou & Xiong, 2005), NEAR (Arad & Shavitt, 2006), DUA (Chen et al., 2006), and YAGR (Na et al., 2007), 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 (i) partially alleviated or (ii) 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 to deal with the void problem. Several routing schemes as surveyed in the literature (Frey & Stojmenović, 2006) adopt the planar graph (West, 2000) as their network

Source: Advances in Greedy Algorithms, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

topologies, such as GPSR (Karp & Kung, 2000), GFG (Bose et al., 2001), Compass Routing II (Kranakis at al., 1999), GOAFR+ (Kuhn at al., 2003), GOAFR++ (Kuhn at al., 2003), and GPVFR (Leong at al., 2005). Nevertheless, the usage of the planar graphs has significant pitfalls due to the removal of communication links leading to the sparse network link distribution; while the adoption of the unit disk graph (UDG) for modeling the underlying network is suggested. A representative UDG-based greedy routing scheme, i.e. the BOUNDHOLE algorithm (Fang at al., 2004), forwards the packets around the network holes by identifying the locations of the holes. However, the delivery of packets cannot be guaranteed in the BOUNDHOLE scheme even if a route exists from the source to the destination node.

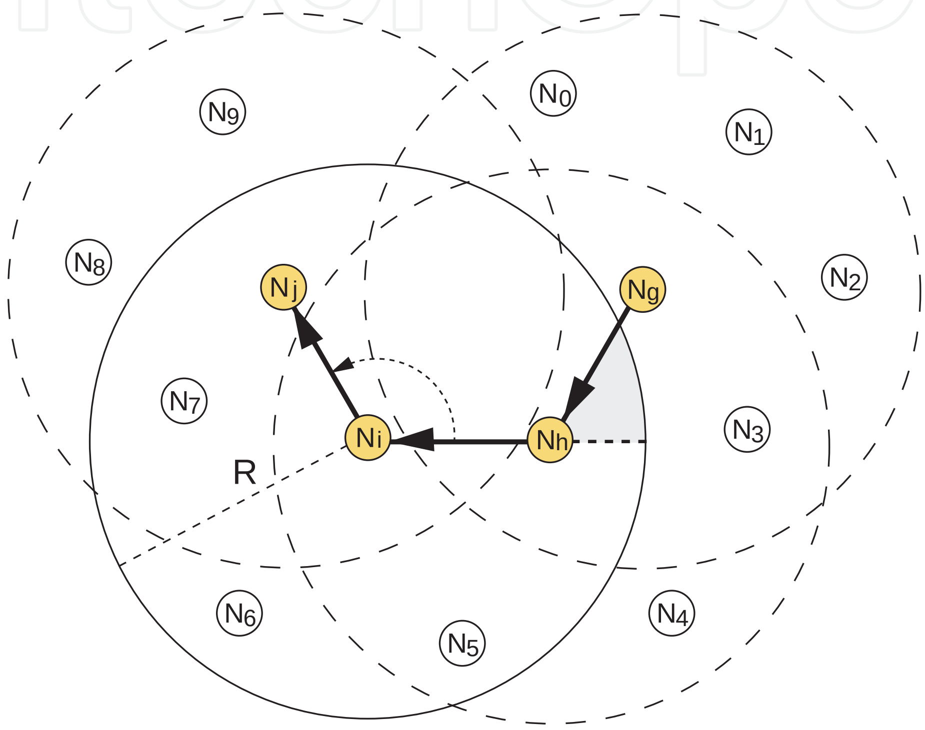


Fig. 1. The forbidden region and the minimal sweeping angle criterion of the BOUNDHOLE algorithm: The node N_i determines the next-hop node of the packets based on the previous two hops N_h and N_g. The forbidden region is defined as the area bounded by (i) the two backward-extended edges of E_{gh} and E_{hi} and (ii) the transmission range border, i.e. the grey region. The node N_j is selected as the next-hop node of N_i since it has the minimal sweeping angle from the previous hop N_h.

In the beginning, the principle of the BOUNDHOLE routing algorithm is briefly described. As shown in Fig. 1, the node N_i is conducting the routing tasks of the packets based on the previous two hops N_h and N_g. The BOUNDHOLE algorithm adopts the forbidden region and the minimal sweeping angle criterion within its formulation. The forbidden region is defined as the area bounded by (i) the backward-extended edges of E_{gh} and E_{hi} and (ii) the transmission range border, i.e. the grey region as in Fig. 1. All nodes in the forbidden region are not considered as the next-hop of N_i. The criterion of the minimal sweeping angle from the previous hop is utilized in the determination of the next-hops within the BOUNDHOLE algorithm. For example, as shown in Fig. 1, the node N_j, which is not in the forbidden region, has the minimal sweeping angle from the previous node N_h. The node N_j is therefore selected as the next-hop node of N_i.

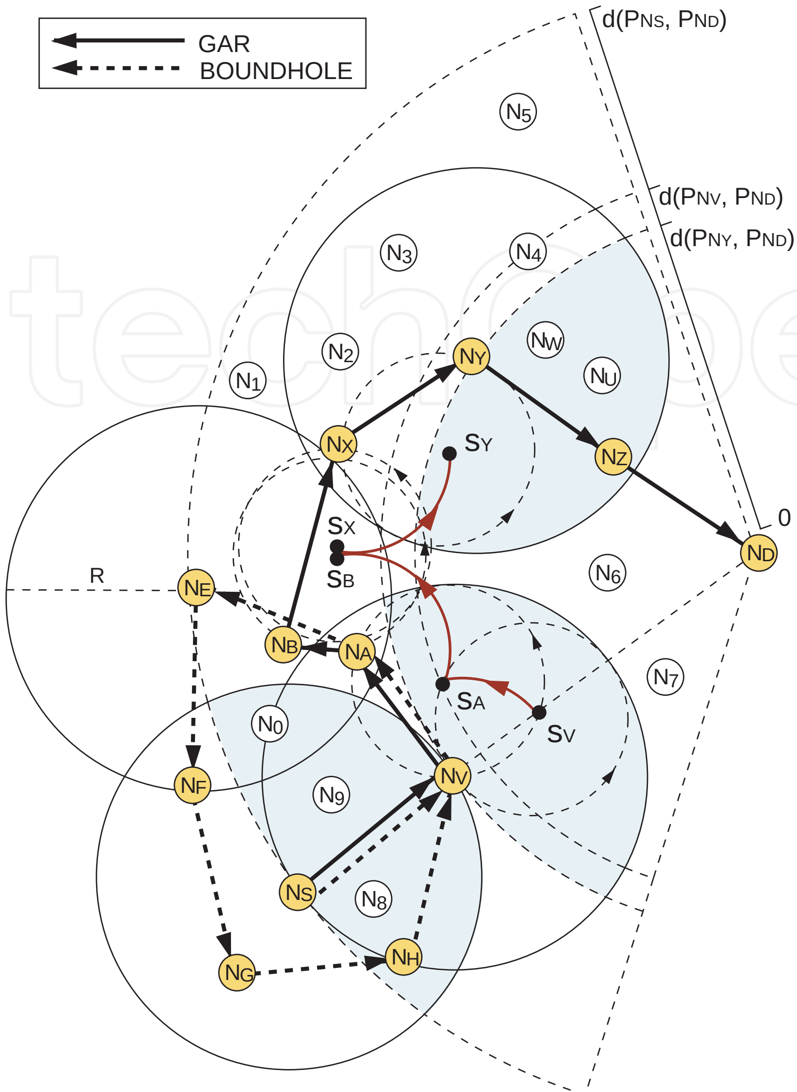


Fig. 2. The example routing paths constructed by using the GAR and the BOUNDHOLE algorithms under the existence of the void problem: (N_S, N_D) is the transmission pair and N_V is the void node. N_X is within the transmission range of N_B; while it is out of the range of N_A and N_E. The GAR protocol utilizes the RUT scheme (with red solid arcs denoted as the trajectory of the SPs); while the minimal angle criterion is employed by the BOUNDHOLE algorithm. The resulting paths obtained from these two schemes are {N_S, N_V, N_A, N_B, N_X, N_Y, N_Z, N_D} using the GAR protocol and {N_S, N_V, N_A, N_E, N_F, N_G, N_H, N_V} by adopting the BOUNDHOLE algorithm, which is observed to be undeliverable. The blue-shaded region associated with each SN is utilized to determine if the SN is a void node or not.

For the comparison purposes, the BOUNDHOLE algorithm is further investigated via an illustrative example. As shown in Fig. 2, the nodes (N_S, N_D) are considered the transmission pair; while N_V represents the node that the void problem occurs. In this example, 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 hopping node since the counter-clockwise sweeping from N_V to N_E (hinged at N_A) is smaller comparing with that from N_V to N_B. Therefore, 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 un-partitioned network topology. Moreover, two cases of edge intersections within the BOUNDHOLE algorithm result in high routing overhead in order to identify the network holes.

In this book chapter, the greedy anti-void routing (GAR) protocol is proposed to resolve the void problem by exploiting the boundary finding technique under the UDG-based network topology. The proposed rolling-ball UDG boundary traversal (RUT) scheme is also employed to completely guarantee the delivery of packets from the source to the destination nodes. Moreover, the hop count reduction (HCR) and the intersection navigation (IN) mechanisms are incorporated within the GAR protocol (denoted as the GAR-E algorithm) to further improve the routing efficiency and the communication overhead. The proofs of correctness for the GAR scheme are also given in this book chapter. Comparing with the existing anti-void routing algorithms, the simulation results show that the proposed GAR-based protocols can provide better routing efficiency with guaranteed packet delivery.

The remainder of this book chapter is organized as follows. Section 2 describes the network model and the problem statement. The proposed GAR protocol is explained in Section 3; while Section 4 exploits the two enhanced mechanisms, i.e. the hop count reduction (HCR) and the intersection navigation (IN) schemes. The performance of the GAR-based protocols is evaluated and compared in Section 5. Section 6 draws the conclusions.

2. Network model and problem statement

Considering a set of SNs $\mathbf{N} = \{N_i \mid \forall i\}$ within a two-dimensional Euclidean plane, the locations of the set **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 omni-directional antennas. The set of closed disks defining the transmission ranges of **N** is denoted as $\overline{\mathbf{D}} = \{\overline{D}(P_{N_i}, R) \mid \forall i\}$, where $\overline{D}(P_{N_i}, R) = \{x \mid \left\| x - P_{N_i} \right\| \le R, \forall x \in \mathbf{R}^2\}$. It is noted that $\mathbf{R}^2$ presents the two-dimensional real vector space and 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 underlying network model for the WSNs can be represented by a unit disk graph (UDG) as $G(\mathbf{P}, \mathbf{E})$ with the edge set $\mathrm{E} = \{E_{ij} \mid E_{ij} = (P_{N_i}, P_{N_j}), P_{N_i} \in \overline{D}(P_{N_j}, R), \forall i \ne j\}$. 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

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

where ID_{N_k} represents the designated identification number for N_k. In the greedy forwarding (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 hopping node from its $\mathbf{T}_{N_S}$ which (i) has the shortest Euclidean distance to N_D among all the SNs in $\mathbf{T}_{N_S}$ and (ii) is located closer to N_D compared to the distance between N_S and N_D (e.g. N_V as in Fig. 2). 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 greedy forwarding (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

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

where $d(x, y)$ represents the Euclidean distance between x and y. $\mathbf{T}_{N_V}$ is the 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. 2, 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 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. 2. 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 s_A and hinged at node N_A. The counter-clockwise rolling procedure will be proceeded in order to select the next hopping node, i.e. N_B in this case. Similarly, the 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 subsections, the formal description of the RUT scheme will be described in Subsection 3.1; while the detail of the GAR algorithm is explained in Subsection 3.2. The proofs of correctness of the GAR protocol are given in Subsection 3.3.

3.1 Rolling-ball UDG boundary Traversal (RUT) scheme

The RUT scheme is adopted to solve the boundary finding problem. 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 (i) the nodes in **B** form a simple unidirectional ring and (ii) the nodes located on and inside the ring are disconnected with those outside of the ring, **B** is denoted as the boundary set and the unidirectional ring is called a boundary.

Problem 2 (Boundary Finding Problem). Given a UDG G(**P**, **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?

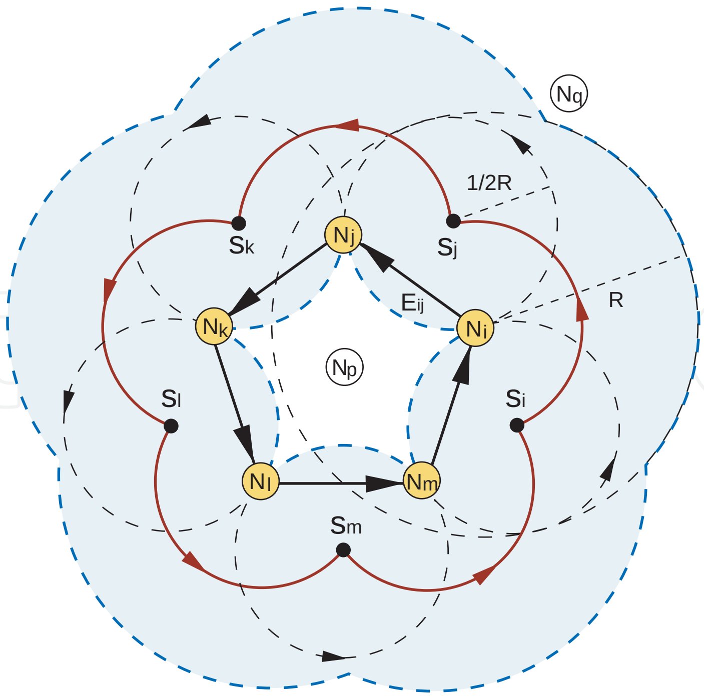


Fig. 3. The rolling-ball UDG boundary traversal (RUT) scheme: Given s_i and N_i, the RUT scheme rotates the rolling ball $RB_{N_i}(s_i, R/2)$ counter-clockwise and constructs the simple closed curve (i.e. the flower-like red solid curve). The boundary set $\mathbf{B} = \{N_i, N_j, N_k, N_l, N_m\}$ is established as a simple unidirectional ring by using the RUT scheme.

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 $RB_{N_i}(s_i, R/2)$ is defined by (i) a rolling circle hinged at P_{N_i} with its center point at $s_i \in \mathbf{R}^2$ and the radius equal to $R/2$; and (ii) there does not exist any $N_k \in \mathbf{N}$ located inside the rolling ball as $\{RB_{\tilde{N}_i}(s_i, R/2) \cap \mathbf{N}\} = \phi$, where $RB_{\tilde{N}_i}(s_i, R/2)$ denotes the open disk within the rolling ball.

Definition 3 (Starting Point). The starting point of N_i within the RUT scheme is defined as the center point $s_i \in \mathbf{R}^2$ of $RB_{N_i}(s_i, R/2)$.

As shown in Fig. 3, each node N_i can verify if there exists an SP since the rolling ball $RB_{N_i}(s_i, R/2)$ 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. 3. It is noticed that there should always exist an SP while the void problem occurs within the network, which will be explained in Subsection 3.2. At this initial phase, the location s_i can be selected as the SP for the RUT scheme.

3.1.2 Boundary traversal phase

Given s_i as the SP associated with its $RB_{N_i}(s_i, R/2)$ hinged at N_i, either the counter-clockwise or clockwise rolling direction can be utilized. As shown in Fig. 3, $RB_{N_i}(s_i, R/2)$ is rolled counter-clockwise until the next SN is reached (i.e. N_j in Fig. 3). 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. s_j and $RB_{N_j}(s_j, R/2)$, 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. 3, 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. 2, the packets are intended to be delivered from N_S to N_D. N_S will select N_V as the next hopping 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. s_V) associated with the rolling ball $RB_{N_V}(s_V, R/2)$ hinged at N_V. As illustrated in Fig. 2, 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. 2), 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 hopping 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 subsection, 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 two-dimensional geometry.

Lemma 1. All the SPs within the RUT scheme form the border of the resulting shape by 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 $\mathbf{S} = \mathbf{R_1} \cap \mathbf{R_2}$ = $\{s_i \mid \|s_i - P_{N_i}\| = R/2, \exists N_i \in \mathbf{N}, s_i \in \mathbf{R}^2\} \cap \{s_j \mid \|s_j - P_{N_j}\| \geq R/2, \forall N_j \in \mathbf{N}, s_j \in \mathbf{R}^2\}$ by adopting the (i) and (ii) 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 two-dimensional 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. 3. 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 $RB_{N_i}(s_i, R/2)$ hinged at P_{N_i} for all $N_i \in \mathbf{N}$. The closed curve can be divided into arc segments $S(s_i, s_j)$, where s_i is the starting SP associated with N_i; and s_j is the anchor point while rotating the $RB_{N_i}(s_i, R/2)$ hinged at P_{N_i}. The arc segments $S(s_i, s_j)$ 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 $S(s_i, s_j)$ 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. 3. 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) is solved by the GAR protocol with guaranteed packet delivery.

Proof: With the existence of the void problem occurring at the void node N_V, the RUT scheme is utilized by initiating an SP (s_V) with the rolling ball $RB_{N_V}(s_V, R/2)$ 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 travelled 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. 2), 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. Enhanced mechanisms for proposed GAR protocol

In order to enhance the routing efficiency of the proposed GAR protocol, two mechanisms are proposed in this section, i.e. the hop count reduction (HCR) and the intersection navigation (IN) schemes. These two mechanisms are described as follows.

4.1 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 hop count criterion. Excessive routing delay associated with power consumption can occur if additional hopping nodes are traversed by adopting the RUT scheme. As shown in Fig. 4, the void node N_V starts the RUT scheme by selecting N_1 as its next hop node with the counter-clockwise rolling direction; while N_2 and N_3 are continuously chosen as the next hopping 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 **B** forms a simple unidirectional ring based on Theorem 1, which indicates that the next-hop SN of a node can be uniquely determined if its previous hopping SN is already specified. For instance (as in Fig. 4), if N_V is the previous node of N_1, N_1's next hopping 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.

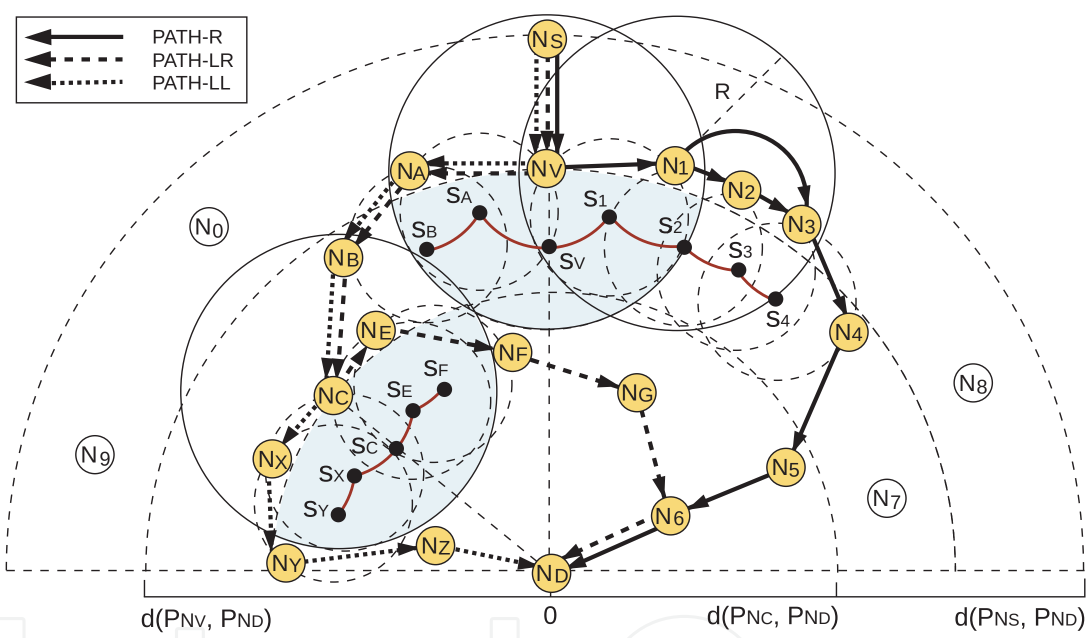


Fig. 4. The hop count reduction (HCR) and the intersection navigation (IN) mechanisms: (N_S, N_D) is the transmission pair, and N_V and N_C are the void nodes. The HCR mechanism: The SN N_1 can create a short cut to its neighbor N_3 by listening to the packet forwarding since the path $\{N_1 \rightarrow N_2 \rightarrow N_3\}$ is uniquely determined. The IN mechanism: Counter-clockwise and clockwise rolling directions (denoted as the symbols of R and L) can be adopted in the RUT scheme. By flooding the navigation map (NAV_MAP) control packets, the shortest path can be acquired as PATH-R $\{N_S, N_V, N_1, N_3, N_4, N_5, N_6, N_D\}$ with 7 hops. Consequently, all packets will contain the sequence {R} to choose the counter-clockwise rolling direction at the first void node N_V. In the case that the path were selected as PATH-LR, the sequence would change to {LR} by choosing the clockwise rolling direction at the first void node N_V and counter-clockwise at the second void node N_C.

According to the concept as stated above, the hop count reduction (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. 4, N_1 chooses N_2 as its next-hop node for packet forwarding; while N_2 selects N_3 as the next hopping 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 hopping 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.

4.2 Intersection Navigation (IN) mechanism

The intersection navigation (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 counter-clockwise 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 hop counts (HC) and packet transmission delay can be achieved.

Based on the transmission pair (N_S, N_D) as shown in Fig. 4, 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 counter-clockwise 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: PATH-R = {N_S, N_V, N_1, N_3, N_4, N_5, N_6, N_D}, PATH-LR = {N_S, N_V, N_A, N_B, N_C, N_E, N_F, N_G, N_6, N_D}, 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 HC(PATH-R) = 7, HC(PATH-LR) = 9, and 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 NAV_MAP = {R} is delivered via the counter-clockwise direction to N_D and NAV_MAP = {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 NAV_MAP = {L} is transformed into two different navigation packets (i.e. NAV_MAP = {LR} and NAV_MAP = {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. NAV_MAP = {R} in this case) will be selected as the targeting path. Therefore, the control packet with NAV_MAP = {R} will be traversed from N_D back to the 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.

5. Performance evaluation

The performance of the proposed GAR algorithm is evaluated and compared with the existing localized schemes (i.e. the GF and the BOUNDHOLE algorithms) via simulations. Furthermore, the GAR protocol with the enhanced mechanisms (i.e. the HCR and the IN schemes) is also implemented, which is denoted as the GAR-E algorithm. The simulations are conducted in the NS-2 network simulator (Heidemann at al., 2001) 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 1.

Parameter Type	**Parameter Value**
Grid Area	1000 x 800 m^2
Void Block	500 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
Number of Nodes	41, 51, 61, 71, 81

Table 1. Simulation Parameters

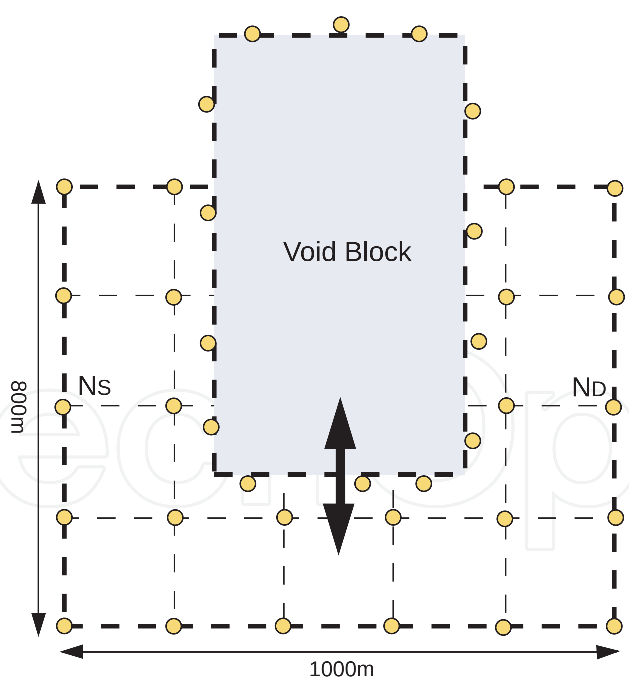


Fig. 5. The simulation scenario: The transmission pair (N_S, N_D) is located at the center of the left and right boundaries of the grid topology. Moreover, there exists a void block with SNs located around the peripheral of the block; while none of the SNs is situated inside the block. The void block is randomly moved in the vertical direction in order to simulate the existence of a void problem within the network.

The simulation scenario is described as follows. As shown in Fig. 5, the grid topology with the existence of a void block is considered in the simulation. It is noted that there are SNs located around the peripheral of the void block; while none of the SNs is situated inside the block. The source and destination nodes N_S and N_D are located at the center of the left and right boundaries as shown in Fig. 5. The data packets are transmitted from N_S to N_D with the void block that is randomly moved with vertical direction in order to simulate the existence of a void problem within the network. It is noted that network partition between N_S and N_D is not considered to exist in the simulation. One hundred simulation runs are conducted for each randomly moved void block case. The following five 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 hop counts within the entire routing path over the number of hop counts for the shortest path.
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 Mbps and the transmitting power of 15 dBm for each SN.

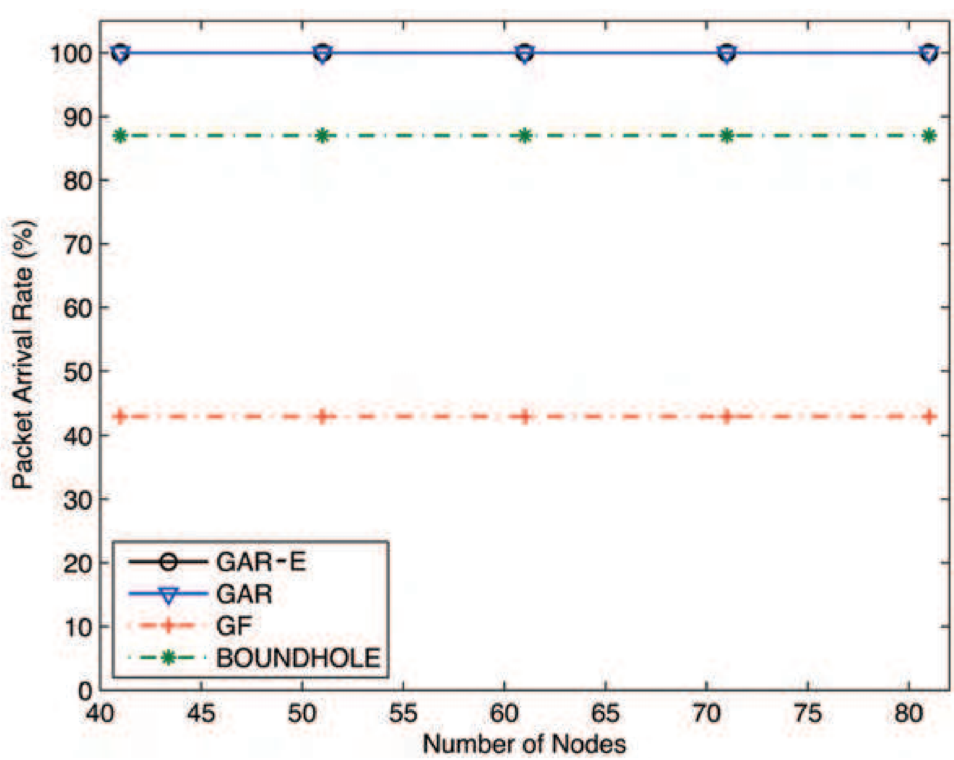


Fig. 6. Packet Arrival Rate (%) vs. Number of Nodes

Figs. 6 to 10 show the performance comparison between these four algorithms under different number of nodes within the UDG network. As can be seen from Fig. 6, the packet arrival rates obtained from these four algorithms are independent to the number of nodes within the network, which is attributed to the design nature of these four schemes. In both of the proposed GAR and GAR-E protocols, 100% of packet arrival rate is guaranteed under different number of nodes. These results are consistent with the protocol design that is proven to ensure 100% of packet arrival rate as long as the network is not partitioned between N_S and N_D. It can also be observed in Fig. 6 that the BOUNDHOLE algorithm can achieve around 88% of packet arrival rate due to the occurrence of routing loop; while the GF scheme can only attain around 45% since the void problem is not considered within its protocol design.

Fig. 7 shows the average end-to-end delay for successful packet delivery by adopting these four algorithms. The conventional GF protocol possesses the smallest end-to-end delay due to its negligence of the void problem, which leads to less than 50% of packet arrival rate as shown in Fig. 6. On the other hand, the BOUNDHOLE algorithm results in the largest end-to-end delay owing to its potential rerouting and looping under certain cases, e.g. as in Fig. 2. The proposed GAR and GAR-E protocols can achieve comparably less routing delay comparing with the BOUNDHOLE scheme, i.e. around 15 to 25 ms less in end-to-end delay. Moreover, the GAR-E algorithm can provide additional 8 to 15 ms less delay comparing with the original GAR protocol due to the enhanced HCR and IN mechanisms. It is also noteworthy to observe the *M*-shape curves resulted within these four schemes. The primary reason can be attributed to the different hop counts between the source/destination pair generated by the GF algorithm. It is noted that the GAR, GAR-E, and BOUNDHOLE schemes implement the GF algorithm without the occurrence of the void problem. The hop counts under the cases of the five different numbers of SNs are computed as 5, 7, 6, 6, and 5. It can be apparently translated into the *M*-shape curves of the end-to-end delay performance as shown in Fig. 7.

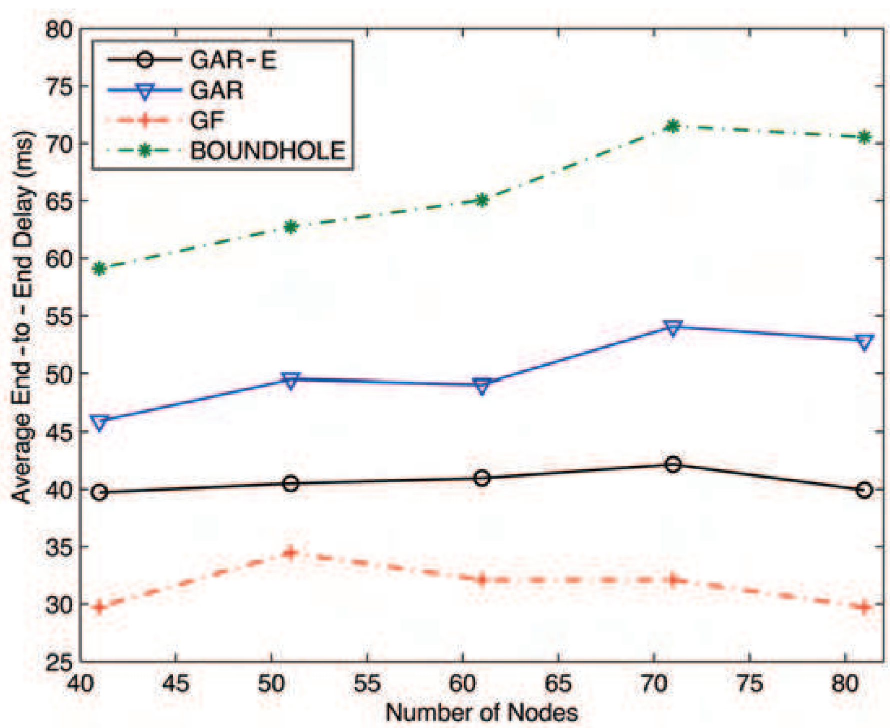


Fig. 7. Average End-to-End Delay (ms) vs. Number of Nodes

As shown in Fig. 8, the path efficiency acquired from these four schemes follows the similar trend as that from the average end-to-end delay. Due to the greedy nature and the negligence of the void problem, the path efficiency of the conventional GF scheme can achieve almost one in the simulations, i.e. the total number of hop counts is almost equal to that of the shortest path. The proposed GAR algorithm possesses the path efficiency of around 1.3 to 1.5. Furthermore, the GAR-E protocol further enhances the path efficiency to around the value of 1.1, which greatly outperforms the BOUNDHOLE schemes.

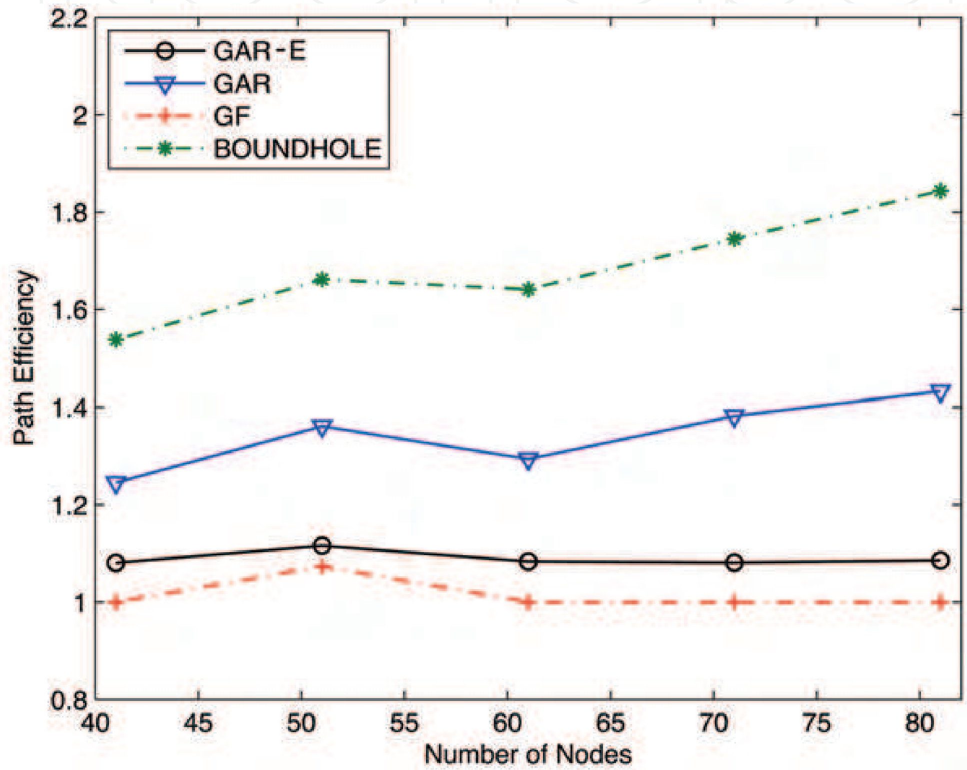


Fig. 8. Path Efficiency vs. Number of Nodes

Fig. 9 shows the communication overheads resulting from these four schemes, which are observed to increase as the increment of the number of nodes. The reason is attributed to the excessive control packets that are required for obtaining the neighbor's locations while the number of nodes is augmented. It is noted that the GF algorithm possesses the smallest communication overheads owing to its ignorance of the void problem. The BOUNDHOLE algorithm results in the largest communication overhead among all the schemes due to its usage of excessive header bytes for preventing the routing loops. It is noticed that even though the GAR-E scheme requires additional NAV_MAP control packets for achieving the IN mechanism, the total required communication overhead is smaller than that from the GAR protocol due to its comparably smaller rerouting number of hop counts.

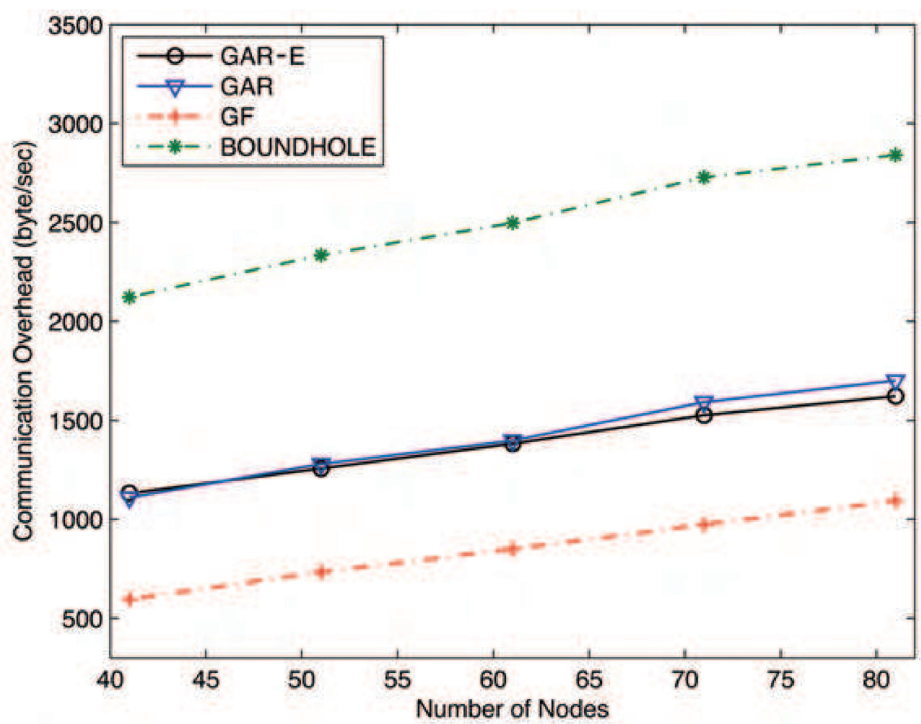


Fig. 9. Communication Overhead (byte/sec) vs. Number of Nodes

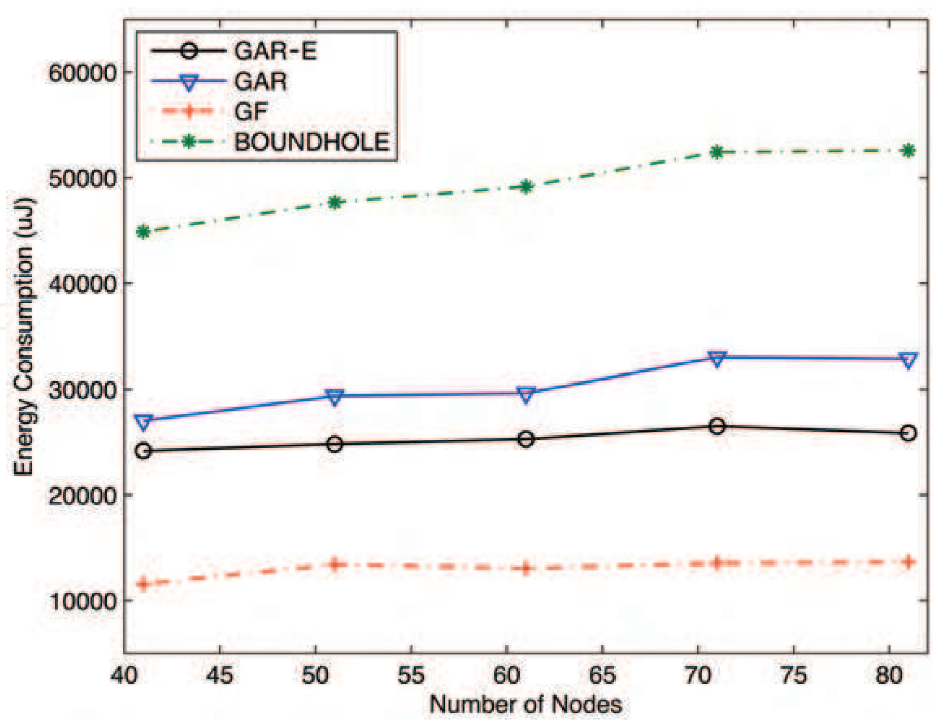


Fig. 10. Energy Consumption (uJ) vs. Number of Nodes

The comparison for energy consumption between these algorithms is presented in Fig. 10. Similar performance trend can be observed between the energy consumption and the communication overhead as shown in Fig. 9. Except for the reference GF protocol, the proposed GAR and GAR-E algorithms can effectively reduce the energy consumption in comparison with the baseline BOUNDHOLE scheme. The merits of the proposed GAR and GAR-E algorithms are observed and validated via the simulation results.

6. Conclusion

In this book chapter, a greedy anti-void routing (GAR) protocol is proposed to completely resolve the void problem incurred by the conventional greedy forwarding algorithm. The rolling-ball UDG boundary traversal (RUT) scheme is adopted within the GAR protocol to solve the boundary finding problem, which results in guaranteed delivery of data packets. The correctness of the RUT scheme and the GAR algorithm are properly proven. The GAR protocol with two delay-reducing schemes, the hop count reduction (HCR) and the intersection navigation (IN) mechanisms, is 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 via simulations and is compared with existing localized routing algorithms. The simulation study shows that the proposed GAR and GAR-E algorithms can guarantee the delivery of data packets; while the GAR-E scheme further improves the routing efficiency and the communication overhead. Feasible routing performance can therefore be achieved.

7. 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) Co., Taiwan.

8. References

Arad, N. & Shavitt, Y. (2006). Minimizing Recovery State in Geographic Ad-Hoc Routing, *Proceedings of ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pp. 13-24, May 2006

Bose, P.; Morin, P.; Stojmenovi´c, I. & Urrutia, J. (2001). Routing with Guaranteed Delivery in Ad Hoc Wireless Networks, *ACM/Kluwer Wireless Networks,* vol. 7, no. 6, Nov. 2001, pp. 609-616

Chen, S.; FAN, G. & Cui, J.H. (2006). Avoid "Void" in Geographic Routing for Data Aggregation in Sensor Networks, *Int. Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 1, no. 4, 2006, pp. 169-178

Estrin, D.; Govindan, R.; Heidemann, J. & Kumar, S. (1999). Next Century Challenges: Scalable Coordination in Sensor Networks, *Proceedings of ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom'99)*, pp. 263-270, Aug. 1999

Fang, Q.; Gao, J. & Guibas, L. (2004). Locating and Bypassing Routing Holes in Sensor Networks, *Proceedings of IEEE Int. Conf. Computer Communications (INFOCOM'04)*, pp. 2458-2468, Mar. 2004

Finn, G.G. (1987). Routing and Addressing Problems in Large Metropolitan-Scale Internetworks, *Info. Sci. Inst. (ISI), Tech. Rep. ISI/RR-87-180*, Mar. 1987

Frey, H. & Stojmenović, I. (2006). On Delivery Guarantees of Face and Combined Greedy Face Routing in Ad Hoc and Sensor Networks, *Proceedings of ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom'06)*, pp. 390-401, Sept. 2006

Giruka, V.C. & Singhal, M. (2005). Angular Routing Protocol for Mobile Ad Hoc Networks, *Proceedings of IEEE Int. Conf. Distributed Computing Systems Workshops (ICDCSW'05)*, pp. 551-557, Jun. 2005

He, T.; Stanković, J.A.; Lu, C. & Abdelzaher, T. (2003). SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks, *Proceedings of Int. Conf. Distributed Computing Systems (ICDCS'03)*, pp. 46-55, May 2003

Heidemann, J.; Bulusu, N.; Elson, J.; Intanagonwiwak, C.; Lan, K.; Xu, Y.; Ye, W.; Estrin, D. & Govindan, R. (2001). Effects of Detail in Wireless Network Simulation, *Proceedings of SCS Multiconference on Distributed Simulation*, pp. 3-11, Jan. 2001

Karp, B. & Kung, H.T. (2000). GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, *Proceedings of ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom'00)*, pp. 243-254, Aug. 2000

Kranakis, E.; Singh, H. & Urrutia, J. (1999). Compass Routing on Geometric Networks, *Proceedings of Canadian Conf. Computational Geometry (CCCG'99)*, pp. 51-54, Aug. 1999

Kuhn, F.; Wattenhofer, R.; Zhang, Y. & Zollinger, A. (2003). Geometric Ad-hoc Routing: Of Theory and Practice, *Proceedings of ACM Symp. Principles of Distributed Computing (PODC)*, pp. 63-72, Jul. 2003

Kuhn, F.; Wattenhofer, R. & Zollinger, A. (2003). Worst-case Optimal and Average-case Efficient Geometric Ad-hoc Routing, *Proceedings of ACM Int. Symp. Mobile Computing and Networking (MobiHoc'03)*, pp. 267-278, Jun. 2003

Leong, B.; Mitra, S. & Liskov, B. (2005). Path Vector Face Routing: Geographic Routing with Local Face Information, *Proceedings of IEEE Int. Conf. Network Protocols (ICNP'05)*, pp. 147-158, Nov. 2005

Liu, W.J. & Feng, K.T. (2006). Largest Forwarding Region Routing Protocol for Mobile Ad Hoc Networks, *Proceedings of IEEE Global Communications Conference (GLOBECOM'06)*, pp. 1-5, Nov. 2006

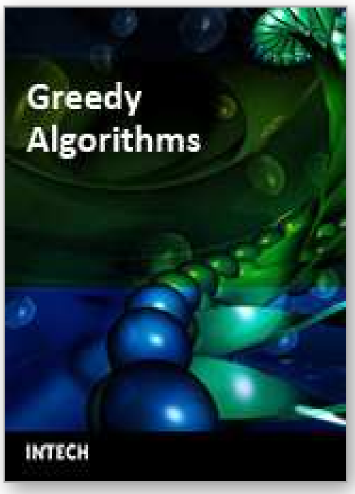
Na, J.; Soroker, D. & Kim, C.K. (2007). Greedy Geographic Routing Using Dynamic Potential Field for Wireless Ad Hoc Networks, *IEEE Commun. Lett.*, vol. 11, no. 3, Mar. 2007, pp. 243-245

Stojmenović, I. & Lin, X. (2001). Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks, *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, Oct. 2001, pp. 1023-1032

Stojmenović, I.; Russell, M. & Vukojevic, B. (2000). Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks, *Proceedings of IEEE Int. Conf. Parallel Processing (ICPP'00)*, pp. 173-180, Aug. 2000

West, D.B. (2000). *Introduction to Graph Theory*, 2nd ed., Prentice Hall

Zou, L.; Lu, M. & Xiong, Z. (2005). A Distributed Algorithm for the Dead End Problem of Location Based Routing in Sensor Networks, *IEEE Trans. Veh. Technol.*, vol. 54, no. 4, Jul. 2005, pp. 1509-1522



Greedy Algorithms
Edited by Witold Bednorz

ISBN 978-953-7619-27-5
Hard cover, 586 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821