科技部補助專題研究計畫成果報告
期末報告

應用於實際無線網路幾何繞徑的調變式合作循環橫掃方法

計 畫 類 別 ： 個別型計畫
計 畫 編 號 ： MOST 106-2221-E-003-005-
執 行 期 間 ： 106年08月01日至107年07月31日
執 行 單 位 ： 國立臺灣師範大學資訊工程系（所）

計畫主持人： 蔡榮宗

計畫參與人員： 碩士班研究生-兼任助理：陳品安
碩士班研究生-兼任助理：薩愛德
大專生-兼任助理：黃建堯
大專生-兼任助理：劉宇翔
大專生-兼任助理：蔡孟庭

中 華 民 國 107 年 10 月 23 日

中文摘要：有效的補救路由繞徑方法，用來克服貪婪式封包傳遞決策無法橫跨網路拓樸空洞屏障的缺失，是幾何繞徑在無線網路中可用性的關鍵問題。大部分學術研究界已提出的依賴區域性環境變數決策方法，在單位圓內連通性的假設下可達成成功傳遞的保證。尤其是著名的曲線循環橫掃運算法則更可以具有提供較低繞徑長度擴增的優勢。然而將這些補救方法使用在非單位圓內連通性的實際無線網路中，結果不僅無法保證成功傳遞，甚至僅能提供極低的封包傳送成功率，實際上就是不適用。針對此項重大缺失，我們將實際無線不完美連通性因素納入考慮，提出一個調變暨合作式曲線循環橫掃運算法則，可成功做為實際無線連結網路環境下的補救路由繞徑方法。此運算法則結合一個典型的曲線循環橫掃程序與一個合作式橫掃程序，兩者皆採用調變式曲線做遞回式循環橫掃。此運算法則運作的基本想法是合作式橫掃程序可解決繞徑路徑上節點隱藏的問題，遞回式循環橫掃運算可降低不能被圓曲線打到而被忽略掉重要節點的機率。而且, 為了降低在網路繞徑無窮迴圈上做無益的傳輸, 我們在補救路由繞徑過程中導入可持續繞徑運作的二維空間，路徑環繞總角度限制及補救繞徑節點總數限制所界定。電腦模擬實驗結果顯示此調變暨合作式曲線循環橫掃運算法則在實際無線環境中運作可以達成此目標: 犧牲封包表頭提供記憶座標空間大小換取極高的起始點到終點封包成功傳送率。

中文關鍵詞：幾何座標繞徑、單位圓內連通性網圖、網路內幾何繞徑空洞、 曲線循環橫掃運算法則、圓弧線段。

英文摘要：An effective approach to bypassing holes and recovering from greedy forwarding failure is vital for geographic routing in wireless networks. Existing localized approaches can guarantee delivery generally under the assumption of unit disk network graphs. Rotational sweep algorithms based on a circular arc, the state-of-the-art recovery method, can be employed to achieve low routing path stretch as well. However, the assumption is inappropriate for realistic wireless channels. Those approaches are thus impractical. Instead, we propose an adaptive and cooperative rotational sweep algorithm taking into account practically imperfect wireless connections. The algorithm involves a regular rotational sweep procedure and a cooperative one both making use of iterative sweepings with adaptive circular arcs. Essentially, the cooperative rotational sweep procedure resolves a potentially hidden node issue while iterative sweeping reduces the possibility of missing pivotal relays. Furthermore, to reduce futile transmissions on a routing loop, we introduce a two-dimensional routing lifetime region delimited by a total angle movement entailed by hole boundary traversal and by a maximum hop count used traditionally. Simulation results show that an extremely high end-to-end routing success probability with ingenious loop stopping can be achieved at

the cost of packet header overheads for memory.

英文關鍵詞： Geographic routing, unit disk graph, network hole, rotational sweep algorithm, circular arc.

Adaptive Cooperative-Rotational Sweep Schemes for Geographic Routing in Realistic Wireless Networks

A report for the MOST project of Taiwan under grant No MOST 106-2221-E-003 -005 -

Jung-Tsung Tsai

Dept. of Computer Science and Information Engineering

National Taiwan Normal University

email: jutsai@csie.ntnu.edu.tw

Abstract

An effective approach to bypassing holes and recovering from greedy forwarding failure is vital for geographic routing in wireless networks. Existing localized approaches can guarantee delivery generally under the assumption of unit disk network graphs. Rotational sweep algorithms based on a circular arc, the state-of-the-art recovery method, can be employed to achieve low routing path stretch as well. However, the assumption is inappropriate for realistic wireless channels. Those approaches are thus impractical. Instead, we propose an adaptive and cooperative rotational sweep algorithm taking into account practically imperfect wireless connections. The algorithm involves a regular rotational sweep procedure and a cooperative one both making use of iterative sweepings with adaptive circular arcs. Essentially, the cooperative rotational sweep procedure resolves a potentially hidden node issue while iterative sweeping reduces the possibility of missing pivotal relays. Furthermore, to reduce futile transmissions on a routing loop, we introduce a two-dimensional routing lifetime region delimited by a total angle movement entailed by hole boundary traversal and by a maximum hop count used traditionally. Simulation results show that an extremely high end-to-end routing success probability with ingenious loop stopping can be achieved at the cost of packet header overheads for memory.

The report is a working manuscript in modification and preparation for submission to IEEE Trans. Wireless Communications or other IEEE journals by Dec. 2018.

Index Terms

Geographic routing, unit disk graph, network hole, rotational sweep algorithm, circular arc.

I. INTRODUCTION

Geographic routing supports the scalability of wireless ad hoc or sensor networks through routing decisions made from minimal local state information on neighborhood node position and packet destination. For such information, it suffices that each node knows its own position and exchanges it with neighbors and that the source of a packet is aware of the packet destination position [1], [2], [3]. Location services or GPS devices can be used to obtain position information, exchanged with neighbors through, for example, IEEE 802.11 beacon messages. The overheads of routing table maintenance or performing route discovery procedures in conventional address-based routing are thus unnecessary.

The most popular and simple decision approach of geographic routing is greedy forwarding [4] by which a node selects as the next relay a neighbor closer to destination than the node itself. Greedy forwarding is loop-free but will stop at a node when such a neighbor does not exist. The forwarding procedure then fails and the packet is *stuck* at the node. This is known as a *local minimum* issue that generally happens at a node on the boundary of a network void or hole [5]. The issue is inevitable in wireless networks, considering that some voids may be created due to topography obstructors, node mobility, or defunct devices from power loss. To succeed in delivering a packet, it is thus imperative to have an auxiliary routing approach to bypassing a routing hole and recover from greedy forwarding failure.

There are several recovery techniques developed to strongly rely on the unit disk assumption (UDA) in that a wireless link exists between two nodes if and only if their distance is less than one unit. They can be divided into two categories, planar graph based face routing [2], [3], [6], [7], [8] and rotational sweep algorithms tracking nodes on a void boundary [9], [10], [11]. Each of them can be combined with greedy forwarding to achieve packet delivery guarantee. In particular, the rotational sweep algorithm based on curved stick [11] or twisting triangle (TT)[10] uses a sweep curve (SC), which is a circular arc of the transmission radius or one side of a Reuleaux triangle [10] (see Fig. 2), to sweep the coverage area of a wireless node and locate the next relay. With this recovery method, low routing path stretch, in terms of hop counts, and delivery guarantee can be both achieved. Similarly, routing decisions in [12] further take into account information on the representative angle of a neighbor node, which significantly cuts down routing path stretch.

However, UDA is fundamentally inappropriate for realistic wireless channels since there is no fixed boundary of transmission range due to signal path loss, interference and fading. Previous recovery approaches are thus impractical. In this work, we consider non-UDA of realistic wireless links that the existence of a bidirectional link between a pair of nodes is completely determined by the received signal power above some threshold at both ends. We propose a recovery method based on the rotational sweep algorithm [11](referred to as TT [10] here) for practical use.

Under non-UDA, a number of issues arise from performing TT to locate the next relay besides the well-known edge intersection [9], [11], [13] or crossing link problem [3], [10], [14]. The first is what size of an SC should be used since there is no magic distance of one unit fitted for all scenarios. The second is the hidden node issue [15], occurring when some node close by should have been hit and selected as the next relay but has no direct link to the node performing TT. The third is the problem of a missing neighbor which was not selected due to the size of SC either too small or too large. The fourth is how to stop routing on a loop earlier since the general way to detect a routing loop under UDA no longer applies.

The main contribution of this work is an adaptive and cooperative rotational sweep scheme (α-CO_mTT) to bypass a routing hole and recover from greedy forwarding failure; this is particularly designed for localized routing in realistic wireless networks. Specifically, it can be stated in four parts:

1) Propose an adaptive rotational sweep algorithm (α-TT) for a progressive search. It resolves SC size and missing node issues by utilizing iterative sweeping with SCs of diminishing sizes determined by locally available link distances and a minimum size constraint.
2) Develop an adaptive cooperative rotational sweep algorithm (α-CO_m) for a retracted search. It resolves the hidden node issue or some edge intersection problem through semi-replicating previous α-TT procedures. This is enabled by using an extra packet header field to carry a list of some visited node positions.
3) Through judiciously combining the two procedures α-TT and α-CO_m, we establish a robust α-CO_mTT which can smoothly work with a number of greedy forwarding strategies.
4) Define a feasible two-dimensional time-to-life region within which α-CO_mTT is allowed

to operate. This entails an early stop of routing over a loop and thus reduces energy wasted in futile relays.

A. Related work

To bypass a routing hole and resolve the local minimum issue, a number of hole boundary traversal schemes have been described notionally like rotating a curve of some geometry. In the BoundHole algorithm [13] a straight line of one unit, the transmission radius under UDA, is utilized to sweep a hole area according to the so-called righthand rule. The first node hit by the line is chosen as the next relay. Used alone, the method is unable to resolve crossing-link issues. Instead, greedy anti-void routing (GAR) [9] was proposed to solve the boundary finding problem by implementing a Rolling ball with a radius of *half* the radio transmission range and determine the next relay. Unfortunately, it guarantees delivery at the expense of increased mean hop counts.

One of the rotational sweep routing algorithms in [10] rotates a twisting triangle, which is a Reuleaux triangle formed by the intersection of three circles placed on the corners of an equilateral triangle with each side of one transmission radius. The rotation angle required for the circular arc side of a twisting triangle to hit a node is used to determine MAC contention latency, designed with the first hit node having the shortest wait after a greedy contention interval. It is a beaconless or contention-based routing [14], [16], [17] involving both MAC contention deference and routing decisions, which requires the support of RTS-CTS messaging. Instead, localized routing [11] utilizes a Curved Stick (CS), a circular arc with chord length and radius of one transmission range, as the sweep curve. Since a CS is exactly one side of Reuleaux triangle, this scheme is similar to the TT algorithm except for the start-sweep point. Both use an SC with a smaller curvature than the rolling ball and a chord length of the transmission radius to achieve lower mean path hop counts and delivery guarantee under UDA.

By defining the largest angle of consecutive edges incident at a node as the representative angle of the node, the routing decisions in [12] further exploit the angle that economically characterizes local topology and is also exchanged with neighbors besides using position information. For greedy forwarding, such neighbor's maximum angle information allows a partial vision about next two hops deterministically, in that a neighbor with the angle larger than or equal to $5\pi/3$ is definitely a candidate of local minimum that should be excluded and that a neighbor with a small representative angle, for example a surely progressive node in [12], definitely entails further advancement. To bypass a routing hole, a greedy network boundary traversal scheme which overlays the TT algorithm was proposed to skip unnecessary hops on a network void boundary. Consequently, the mean routing path hop count is remarkably reduced.

Our preliminary study [15] considers an incomplete non-UDA in that no wireless bidirectional link exists between two nodes absolutely if their distance is larger than one unit, that a link exists between them if their distance is less than a given threshold $r_c \in [0, 1]$, a constant modeling *connection regularity*, and that a link may exist between them depending on received signal power level if their distance is in the range $(r_c, 1]$. Under such a relaxed UDA, it was found that applied directly, even very efficient recovery schemes [11], [12] developed with UDA are hardly to achieve a sensibly routing success probability. A cooperative rotational sweep method, also based on TT, was then proposed to jointly operate with original TT as a recovery scheme. The cooperative approach relies on duplicated sweeps but performed by the current packet holding node in lieu of previously visited nodes whose positions are carried with the packet. It was shown that with the approach, a high routing success probability is achievable at the cost of packet header overheads, and that a given network with the assumption of more regular connection $r_c = 0.7$ performs much better than that with $r_c = 0$.

B. Outline

The rest of the paper is organized as follows. In Section II we describe system models and basic components of geographic routing related to this work. In Section III we propose rotational sweep schemes α-TT and α-CO$_m$, two key procedures involved in recovery mode operation. In Section IV, we address system operations and some issues raised from non-UDA that have been resolved or remain unresolved. Then a two dimensional routing stop region is defined. Section V presents simulation results with discussions. We conclude in Section VI with a brief remark.

II. System model

We use $G(V, E)$ to represent a wireless ad hoc network graph where V is the set of all nodes and E is the set of all bidirectional links. For node $v_i, v_j \in V$, edge $v_i v_j \in E$ if and only if v_i and v_j can communicate with each other directly. The set of neighbors of node v_i is represented by $N_i = \{v_j \in V; v_i v_j \in E, j \neq i\}$. Let $\overline{N}_i(\gamma) = \{v_j \in V; |v_i v_j| \leq \gamma, j \neq i\} - N_i$ denote a set of nodes which are within the distance γ but not connected to node v_i, i.e. the set of nodes in a nominal circle coverage area of radius γ, denoted by $C_i(\gamma)$, that are hidden to node v_i.

Let $A_f(v_i) = \{v_j \in N_i; |v_jD| < |v_iD|\}$, a set consisting of the neighbors of node v_i which have shorter distance to D than node v_i.

Consider that node v_i creates or receives a packet destined to D in greedy mode operation. If $A_f(v_i) \neq \emptyset$, the packet will advance to a selected node in $A_f(v_i)$. Otherwise the event of local minimum occurs at the so-called stuck node v_i, denoted by v_A, for $A_f(v_i) = \emptyset$. To lead the packet away from stuck node v_i, a recovery process is then invoked. Essentially, the routing process alternates *greedy mode* and *recovery mode* operations in a full course of delivering a packet. It switches to recovery mode operation whenever the greedy one fails.

In this work, we loosely define part of a hole (or void) boundary as a sequence of nodes $\{v_i, v_{i+1}, \cdots, v_{i+j}\}$ so that the righthand side of the ordered sequence contains no node having a link incident at any of those nodes, node v_{k+1} is within the sweep distance of some chosen SC at v_k, $k \in [i, i + j - 1]$, and a path exists between nodes v_k and v_{k+1} which may not connect directly. The boundary may change with SC size employed. Whenever greedy mode operation fails, we say that routing operation faces a routing hole or network void.

A. Greedy mode operation

We consider three different forwarding schemes for greedy mode operation. Greedy-Forwarding (GF) [4] is a popular scheme where node v_i selects as the next relay the neighbor in $A_f(v_i)$ with the least distance to D. The other twos are modified quasi-greedy forwarding (MQGF) and modified enhanced-greedy forwarding (MeGF) schemes adapted from [12] for use under non-UDA here. Both further involve representative angle information (RAI) besides position. Taking account of them, we will reach a more robust recovery scheme, developed in Sec. II-B, to successfully work with different greedy mode methods.

Denoted by θ_i^*, the representative angle of a node, say v_i, is the maximum angle between consecutive edges incident at v_i. For $N_i = \emptyset$, $\theta_i^* = \infty$ and for $|N_i| = 1$, $\theta_i^* = 2\pi$. Node v_i also communicates RAI θ_i^* to its connected neighbors in addition to position v_i required normally. Under non-UDA, a nominal transmission radius, η_L, is still required for the following adaptation. It is considered protocol-wise available, defined in Sec III-A.

By MeGF, a node, say v_i, in greedy mode first chooses next relay v^* for a packet destined to D according to

$$v^* = \arg \min_{\{v_j \in A_f(v_i), \theta^*_{v_j} < 5\pi/3\}} |v_j D| + \omega \frac{\theta^*_{v_j} - \theta_t(|v_j D|)}{5\pi/3 - \theta_t(|v_j D|)} \eta_L \tag{1}$$

where $\omega \geq 0$ is an enhance factor accounting for the impact of representative angles and

$$\theta_t(|v_j D|) = \begin{cases} 2\cos^{-1}(\frac{\eta_L}{2|v_j D|}), & |v_j D| \geq \eta_L \\ 2\pi/3, & |v_j D| < \eta_L. \end{cases} \tag{2}$$

If v^* is empty, node v_i then performs GF to choose from $A_f(v_i)$ a neighbor with $\theta^*_j \geq 5\pi/3$. If none can be found, the packet is stuck at node v_i.

To account for imperfect connections under non-UDA, we have modified the definition of $\theta_t(|v_j D|)$ [15], called SPN angle threshold in [12], by further defining $\theta_t(|v_j D|) = 2\pi/3$ for neighbor v_j with $|v_j D| < \eta_L$ in (2).

By MQGF, node v_i in greedy mode operation divides its neighbors in forwarding set $A_f(v_i)$ into three subsets. Subsets 1, 2, and 3 contain those neighbors with RAI $\theta^*_j < \theta_t(|v_j D|)$ defined in (2), $\theta_t(|v_j D|) \leq \theta^*_j < 5\pi/3$, and $\theta^*_j \geq 5\pi/3$, respectively. It chooses the next relay by GF first from subset 1, otherwise from subset 2, and finally from subset 3 if subsets 1 and 2 are both empty. MQGF doesn't require enhance factor ω used in (1) but loses some performance.

B. Recovery mode operation

The proposed scheme for recovery mode operation is depicted in the flowchart of Fig. 1. It includes two key procedures, α-TT for progressive search and α-CO$_m$ for retracted search, and two condition checks for returning to greedy mode operation. Their functions are separately described in the following.

1) α-TT: This is an iterative rotational sweep scheme based on circular arcs of varying size, for selecting as the next relay a connected neighbor supposed to be on a network void boundary. It replaces the role of the rotational sweep algorithm [10], [11], [12], [15] but adopts adaptive SC sizes to cope with non-circular and non-fixed wireless coverage area issues. Because of iterations, the chosen relay is now the first hit by the last sweep instead.

However, the next relay selected by α-TT, to be presented in Sec III-A, is no more certainly on a network void boundary under realistic wireless link connection assumption, due to hidden node issues. That is, node v_i selects a node in N_i as the relay while the boundary node v_{i+1} next to v_i may not be in N_i. This fundamentally breaks the design philosophy of TT. To raise this certainty, we thus adopt the following cooperative rotational sweep procedure adapted from [15].

2) α-CO_m: A semi-replicate of α-TTs performed previously at m visited nodes at most is employed to resolve possible hidden node issues. Its detail will be presented in Sec III-B.

Like CO_m proposed in [15], a packet in recovery mode operation is assumed to include a field capable of carrying the position information of visited nodes on a routing trajectory by the packet for m entries at most. When a node, say v_i, receives a packet with the recovery mode bit on, it first performs α-CO_m on behalf of those recent visited nodes with position information available from the m entries to check whether there are neighbors that should have been selected earlier but hidden to some visited nodes logged in the m entries. If so, node v_i sends the packet with updated history information of visited nodes to the neighbor newly selected by the cooperative procedures. If not, node v_i is supposed to be on a network void boundary, updates position information in m entries if necessary, and then performs a regular α-TT to locate the next relay.

3) $|v_{i+1}D| < |v_A D|$: Immediately after procedure α-TT, the condition is checked for a change to greedy mode GF, MQGF, or MeGF depending on design. Specifically, it is that next relay v_{i+1} selected by α-TT at node v_i has shorter distance to D than the stuck node v_A has. This is a more constrained condition because only node v_{i+1} selected by α-TT is involved, resulting in a more conservative way for returning to greedy mode.

We overlay this popular condition for operation mode change with a more aggressive one that there is some $v_j \in N_i \cup \{v_i\}$ such that $|v_j D| < |v_A D|$. This condition involves node v_i itself and all of its neighbors for test but is limited to apply in some network areas, as described next.

4) d_t: It is a threshold of distance to destination, within which recovery mode operation tends to switch to the greedy one in a more aggressive manner.

Routing in recovery mode has the benefit of memory effects for resolving routing problems, through exploiting m history information in a packet. On the other hand, routing in greedy mode has no such benefit. Whenever operation mode changes from recovery to greedy, routing operation thereafter loses the benefit of memory. This demands not to make a hasty change to greedy mode. However, it is intuitive that a packet should be sent to reach the destination as fast as possible instead of letting it hang around in recovery mode and face potentially more routing problems. This demands to make a change as early as possible to greedy mode operation which is generally much more efficient in delivering a packet. To make a compromise between these mentioned opposite facts, we thus utilize the distance threshold d_t, a design parameter, to determine where the aggressive approach for a change to greedy mode should be turned on.

Specifically, when receiving a packet with the recovery mode bit on, node v_i checks the following condition

$$\begin{aligned} &|v_i D| < d_t \text{ and there is some node} \\ &v_j \in N_i \cup \{v_i\} \text{ such that } |v_j D| < |v_A D|. \end{aligned} \tag{3}$$

If so, node v_i changes to greedy mode operation and then sends the packet directly to the neighbor $v_j \in N_i$ with the least distance to D if $v_j \neq v_i$. Notice that this is a direct use of GF even if MQGF or MeGF is chosen as the greedy mode procedure in design. This essentially resolves the issue of a routing loop caused by the effect of ping-pong operations by switching to greedy mode MQGF or MeGF in one node and switching to recovery mode in the other node [15].

Fig. 1 shows the flow chart of function blocks in recovery mode operation. This mode of operation begins with the righthand branch of the flow chart, α-TT, when routing at a node, say node v_i, encounters a greedy mode failure and switches to recovery mode operation. Thereafter a node, v_i, receiving a packet with recovery information will start with the lefthand branch, first checking whether condition (3) is true. If so, it changes operation mode as stated previously. If not, it proceeds to perform cooperative rotational sweep α-CO$_m$.

III. Rotational sweep with sweep curves of varying sizes

Besides destination D and stuck node position v_A, a packet in recovery mode includes subfields, denoted by array T_r with size $|T_r| \leq m$, sufficient to carry and track latest visited node positions on a routing trajectory for at most $m \geq 2$ entries and a bit $B_I \in \{\text{TRUE, FALSE}\}$ indicating whether buffer T_r is ever filled to overflowing. Additionally, suppose that there is a constraint on the minimum size of SC, with chord length η_L and hence curvature $\frac{1}{\eta_L}$. The threshold η_L is a design parameter that can be optimized for maximum routing success performance. It is considered protocol-wise available at each node.

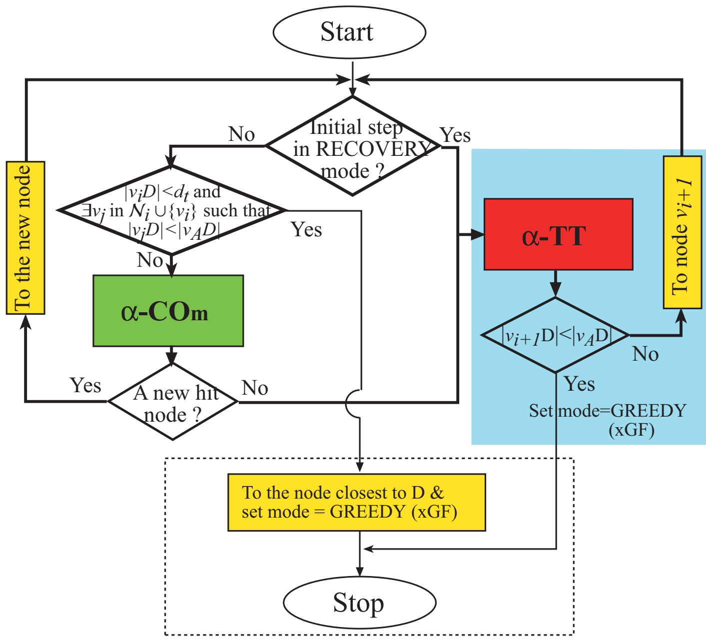


Fig. 1. The flow chart of adaptive and cooperative rotational sweep scheme α-CO$_m$TT. The scheme is invoked whenever routing in greedy forwarding fails. The way to terminate routing is not shown.

To perform a rotational sweep with an SC of some size hinged at node v_i we need to first resolve the problem of where to start sweeping. Our solution is to align the chord of an SC of any size $\eta \geq \eta_L$ on top of $\overrightarrow{v_i h_i}$ with the curvature center trailing behind and always sweep the coverage area $C_i(\eta)$ counterclockwise according to the righthand rule. Thus $v_i h_i$ is viewed as the start sweep line for an SC hinged node v_i. Point h_i is defined next.

If node v_i just changes operation mode from greedy to recovery or was exactly the node on a routing trajectory that started current recovery mode operation, then h_i is the intercept of circle $C_i(\eta_L)$ and $\overrightarrow{v_i D}$ or simply $h_i = D$. Otherwise, node v_i was definitely selected by a prior node, say v_{i-1} in T_r, on the routing trajectory of the current recovery mode. Then, h_i is the righthand intercept of circles $C_{i-1}(\max\{\eta_L, |v_i v_{i-1}|\})$ and $C_i(\max\{\eta_L, |v_i v_{i-1}|\})$ from v_{i-1} to v_i. The above for computing $h_i = h(v_{i-1}, v_i)$ is summarized in Algorithm 1.

A. α-TT

Besides determining a start-sweep line, the second issue is what size of an SC to be used for sweeping a wireless coverage and when to stop recursively sweeping. We sketch the procedures of iterative sweeping with reduced SC sizes adapted to local wireless environments though the

Algorithm 1: $h_k = h(v_{k-1}, v_k)$ on start-sweep line $v_k h_k$

Require: Node v_i has a packet with state information $\{D, \text{mode=Recovery}, v_A, T_r, B_I\}$. The minimum SC size is η_L.

Ensure: Set h_k, to which the start sweep line is from $v_k \in T_r \cup \{v_i\}$.

```
if v_{k-1} == D then
    h_k is the intercept of line v_k D and circle C_k(η_L).
    {Or, simply h_k ← D.}
else
    h_k is the intercept of C_{k-1}(max{η_L, |v_k v_{k-1}|}) and C_k(max{η_L, |v_k v_{k-1}|}) on the
    righthand side of
    v_{k-1} to v_k.
end if
return h_k
```

example in Fig.2.

In the figure, a packet destined to D has been sent from S to node v_0 after 4 hops in greedy mode. The packet is stuck at node v_0 where greedy mode operation fails and hence an adopted recovery method is invoked.

Let $\gamma_0 = \max\{|v_0 v_k|; v_k \in N_0\}$, the largest of distance from v_0 to its connected neighbor nodes. **(i)**: The size of the first SC chosen at node v_0 has chord length $\eta_0 = \max\{\gamma_0, \eta_L\}$. The essential idea for the choice is to make use of state information available at node v_0 without missing any neighbor that can potentially be hit by sweeping a given size of SC. Suppose that $|v_0 v_{15}| > \eta_L$. Then the size of the first SC chosen by v_0 is $\eta_0 = |v_0 v_{15}|$. With such an SC swept from start-sweep line $v_0 h_0$ or $v_0 D$ as determined by Algorithm 1 the sweep procedure won't miss node v_{15} and cause a routing failure if the connection $v_0 v_1$ doesn't exist. **(ii)**: Thus, the first node hit by the SC is the latest candidate for the next relay; If this node happens to be exactly the node whose distance to the SC-hinged node determines the current size of SC, the iteration of sweeping stops and the node is elected to be the next relay. This leads to one condition for stopping iteration of sweeps. Another condition is that if the current SC has the minimum chord length η_L, then the first node hit by this size of SC is the next relay and the process of iterative sweeping stops immediately.

As shown in the figure, node v_1 is first hit by the SC of size $\eta_0 = |v_0 v_{15}|$. Since $\eta_0 = |v_0 v_{15}|$ can be very large, the SC of this size is possibly unable to hit some node, on a convex boundary [12], behind $v_0 v_1$ (not drawn in the plot) that is a potential relay leading to destination [11]. Our approach to the issue is to apply recursive sweepings with reduced SC sizes. **(iii)**: If neither of the two stop conditions described previously in (ii) is satisfied, then the latest hit node v_1 determines a new SC of size $\eta_1 = \max\{|v_0 v_1|, \eta_L\}$ and procedure (ii) is repeated for the latest choice of node v_1 and SC size η_1.

In the example, $\eta_1 > \eta_L$ is assumed. The SC of size η_1 is then swept to first hit node v_1 again, which satisfies the first stop condition of iteration in (ii). This session of iterative sweeping thus determines node v_1 to be the next relay and ends.

The main procedure of α-TT has been described previously in (i), (ii), and (iii) on initial value, stop conditions and selection, and iterative functions, respectively. Detailed algorithmic procedures are presented in Algorithm 2 which includes a test for change to greedy mode operation at the next relay (lines 14 to 15) and otherwise a final push of current position v_i from the tail into buffer T_r of size m (lines 17 to 21), upon having determined the next relay v_{i+1} (line 13) where procedure α-TT is almost done except updating T_r when necessary.

B. α-CO_m

Receiving a packet in recovery mode, a node, say v_i, has additional information $\{T_r, B_I, v_A\}$ besides local state $N_i \cup \{v_i\}$. It first runs procedure α-CO$_m$ to check whether the ordered sequence of previously selected nodes in $T_r \cup \{v_i\}$ is exactly followed by performing a sequence of SC sweeps over nodes in $T_r \cup \{v_i\} \cup N_i$. The check starts with the SC hinged at $T_r[0]$ if buffer overflow status B_I is FALSE, because the information available is sufficient to determine the start sweep line $h(D, T_r[0])$ from $T_r[0]$ which is v_A; Otherwise, it starts with the SC hinged at $T_r[1]$, swept along line $h(T_r[0], T_r[1])$. The latter for B_I =TRUE is because there is no way to find the start-sweep line and replicate the previous sweep operation of SC hinged at $T_r[0]$ for a better utilization of available information, even if $T_r[0]$ is exactly position v_A due to recurrent visiting [11]. The above is implemented by lines 2 and 4 in Algorithm 3.

The initial size of the SC hinged at $T_r[j]$ is defined to have chord length $\eta = \max\{|T_r[j]T_r[j+1]|, \eta_L\}$ for $j < |T_r| - 1$ or $\eta = \max\{|T_r[j]v_i|, \eta_L\}$ for $j = |T_r| - 1$ (line 5 in Algorithm 3), unlike

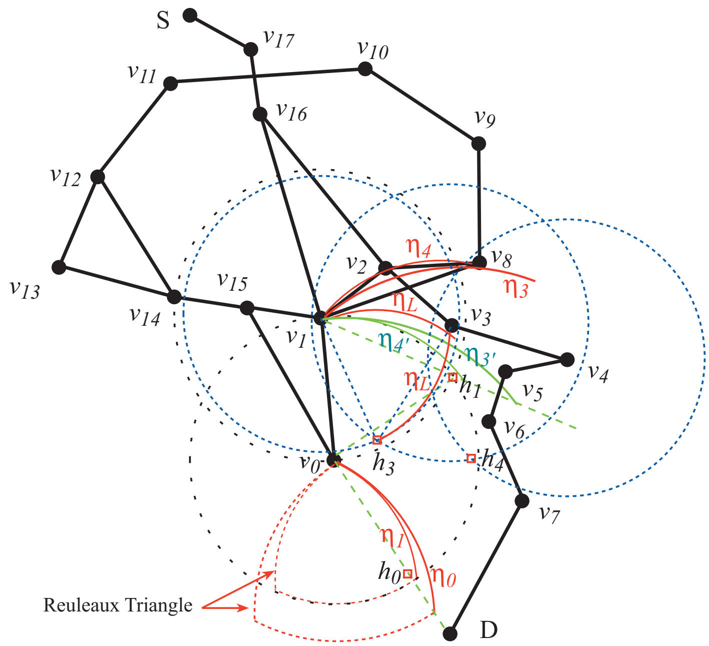


Fig. 2. A network graph of 20 nodes and 24 bidirectional links under non-UDA, with some sweep curves set at start sweep positions. The sweep curve, called a curve stick [11] or one side of a Reuleaux triangle [10], is now a circular arc of varying sizes. A packet originating at S and destined to D is first stuck at node v_0, where the proposed recovery mode procedure α-CO$_m$TT is invoked.

the one used in α-TT (line 2 in Algorithm 2). Upon determining a start-sweep line and the initial chord length for SC hinged at $T_r[j]$, the procedure of iterative sweepings with reduced SC sizes, lines 6 to 15 in Algorithm 3 like lines 3 to 12 in Algorithm 2 for α-TT but over known node positions $T_r \cup \{v_i\} \cup N_i$, then follows. The procedure, beginning with $j = 0$ for B_I =FALSE or $j = 1$ for B_I =TRUE, is sequentially performed to check whether $T_r[j+1]$, or v_i when $j = |T_r| - 1$, is again selected by the SC hinged at $T_r[j]$ (lines 2 to 21 in Algorithm 3). Notionally it partially replicates previous sweeps, viewed as requested sweeps for collaboration [15], to look for new witnessed nodes in N_i by v_i that may have been missed previously. This check for hidden nodes results in either of the following:

- If the sequence $T_r, \{v_i\}$ is exactly followed for no hidden node, then procedure α-TT, Algorithm 2, is called to select the next relay, say $v_{i+1} \in N_i$ (line 28 in Algorithm 3).
- Otherwise, there exists some smallest $k \in [0, |T_r| - 1]$ such that the latest selected node, say $v_n \in N_i$, by SC hinged node $T_r[k]$ differs from previous visited node (i) $T_r[k+1]$ for $k < |T_r| - 1$ or (ii) v_i for $k = |T_r| - 1$, due to hidden node problems.

In the case of (i) above, the path segment $T_r[j]$, $j \geq k+1$, is no longer all exact on a routing trajectory of a void boundary. The memory of routing path is then updated by flushing out all recorded node positions $T_r[j] \leftarrow \emptyset$ for $j \geq k+1$ (line 23 in Algorithm 3). If $v_n \neq v_i$, select the next relay $v_{i+1} \leftarrow v_n$ (lines 25,26 in Algorithm 3) as a potential follow-up of $T_r[k]$. Otherwise ($v_n == v_i$), the next relay v_n is the current node v_i itself, and therefore it is not necessary to perform one more cooperative sweep operation and get the same result; Instead, call α-TT to get the next relay v_{i+1} (line 28 in Algorithm 3).

In the case of (ii), no update on T_r is necessary and the discovered hidden node v_n is selected as the next relay, $v_{i+1} \leftarrow v_n$ (line 26 in Algorithm 3). Thereby, current position v_i is left off history list T_r.

Notice that it is not necessary to perform cooperative sweeps always from $T_r[0]$ or $T_r[1]$ in particular when the number $|T_r|$ is large. It can begin with the smallest j when condition $|T_r[j]v_i| < \max\{|T_r[j]T_r[j+1]|, \eta_L\} + \max_{v_\ell \in N_i} |v_i v_\ell|$ for a nonzero possibility of sweeping some node in N_i is satisfied. This speeds up α-CO$_m$ computations. Also notice that if the sequence $T_r, \{v_i\}$ is exactly followed for no hidden node after procedure α-CO$_m$, the position v_i should be pushed from the tail into T_r. This update of T_r has been moved into α-TT (lines 17 to 21 in Algorithm 2).

IV. α-CO$_m$TT OPERATION

To see how scheme α-CO$_m$TT works to recover from a greedy mode failure, we consider the network graph in Fig. 2 where a packet destined to D has been forwarded from its source S through nodes v_{17}, v_{16}, v_1 by GF and stuck at node v_0. Suppose that α-CO$_2$TT is employed to resolve the failure of GF at node v_0. Then, its timing sequence of operation is row-wise listed in Table I. The node where the packet is currently located is shown in column 1. The method in operation is listed in column 2. Column 3 first gives the chord length of an SC employed, followed by support and SC hinged node @ recorded position if shown. The first hit node by a chosen SC at the associated SC hinged node is listed in column 4. Column 5 shows history state T_r and $B_I \in \{False(F), True(T)\}$ when they are updated. The start sweep line for each sweep is supposed to be clear and thus not shown.

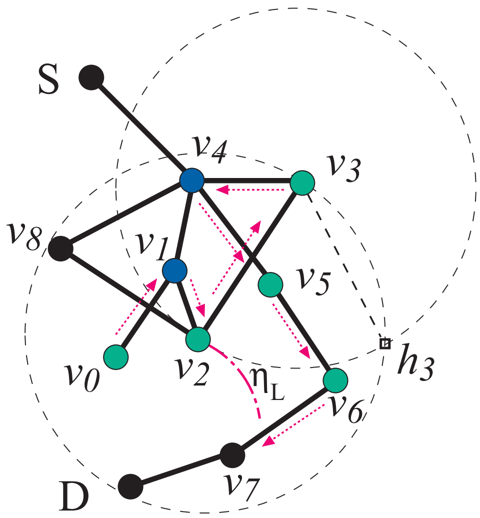
Recovery mode α-CO$_2$TT is operated from node v_0 to node v_6. It can be seen from row v_i in Table I and in Fig. 2 that method α-TT at node v_1 is able to select v_2 rather than v_8 as the next relay by iterative sweeping with reduced size of SC down to the minimum threshold η_L. This solves a potential edge intersection problem caused by edges v_1v_8 and v_2v_3. We also see that method α-TT at node v_3 will select node v_4 but miss v_5 as the next relay since node v_5 is hidden to v_3. Next, cooperative method α-CO_2 performed at node v_4 exactly solves the hidden node problem by sending the packet to node v_5 for a first correction. Method α-CO_2 performed at node v_5 then makes another correction for the SC size $\max\{\eta_L, |v_3v_5|\}$ hinged at $T_r[1]$(i.e. v_3) is greater than $|v_3v_6|$.

Without α-CO_m, α-TT can be used alone for recovery with the position of the previous node from which it received a packet, or D when GF failure occurs, and the current position to determine a start-sweep line. This way, node v_4 after receiving the packet from v_3 in the previous example will select v_3 as the next relay because the SC hinged at v_4 and swept from start-sweep line $h(v_3, v_4)$ is unable to hit and select v_5. Thereafter, the packet will visit nodes $v_3, v_2, v_{16}, v_{17}, \cdots$, falling into a routing loop. Certainly the main cause is the hidden node issue, resolved previously by the scheme further involving cooperative procedure α-CO_m.

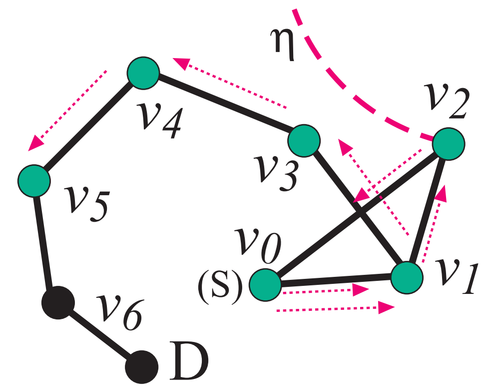
A. *Routing stop rules*

Under UDA, the routing stop rule [11] states that a routing loop event has occurred when the SC of one unit size is hinged at stuck node v_A again due to the return of a packet in recovery mode TT and swept to first hit the initial sweep position at the intercept of unit circle and v_AD, i.e. $h(D, v_A)$. This presents a beauty of design for routing-loop detection, not only to stop a truly infinite routing loop but also to imply that destination D is unreachable. These two inferences thus entail a guarantee of packet delivery as long as a path exists between S and D.

Both above features no longer apply to networks under non-UDA, besides that looping may occur somewhere beyond stuck node v_A, even by α-CO_mTT. To see these, we first show one example in the network illustrated in Fig. 3(a), where routing operation in greedy mode encounters a local minimum event at node v_0 and immediately switches to α-CO_mTT, $m \geq 3$, for recovery. Thereafter, the packet will be routed through nodes v_1, v_2, v_3, v_4, v_5, v_6, with carried history list $T_r = \{v_0\}$, $\{v_0\}$, $\{v_0, v_2\}$, $\{v_0, v_2, v_3\}$, $\{v_0, v_2\}$, and $\{v_0, v_2, v_5\}$, respectively. Note that the edge intersection problem due to edges v_2v_3 and v_4v_5 has been resolved. On the other hand, if $m = 2$ the packet will be routed through $v_1, v_2, v_3, v_4, S, v_4, v_8, v_2, v_3, \cdots$, because the memory size $m = 2$ is not sufficient for resolving the above cross-link problem. Since this infinite loop



(a)



(b)

Fig. 3. A packet to D is stuck at node v_0 where α-CO$_m$TT is then invoked. (a) A simple edge intersection problem caused by the crossing links v_2v_3 and v_4v_5 can be resolved if $m \geq 3$. For $m = 2$, the packet loops around and never returns to stuck node v_0. (b) Another edge intersection problem caused by crossing links v_0v_2 and v_1v_3 can be resolved if $m \geq 3$ and some traditional loop-routing detection rule is not employed for terminating routing. One traditional loop-routing detection is based on monitoring the occurrence that a sweep curve is hinged at v_A (i.e. v_0) again and swept to fully cross line v_AD before hitting any other node.

does not involve the stuck node v_0, it can't be detected and thus stopped by the conventional way.

Consider another example shown in in the network of Fig. 3(b) for a packet originated at v_0, or S, and immediately stuck there. For recovery mode α-CO$_m$TT, $m \geq 3$, the packet will be routed from v_0 through nodes $v_1, v_2, v_0, v_1, v_3, v_4, v_5$. Thereafter, the packet will be sent to v_6 and then to D in greedy mode. However, the conventional rule dictates that the packet is discarded for destination unreachable when it returns to stuck node v_0 and the SC swept from $(v_0, h(v_2, v_0))$ first hits $h(D, v_0)$ before v_1. By this rule, the packet is then dropped when returning to v_0 the first time, leading to a routing failure instead. Therefore it is no more sensible to borrow the conventional rule to give up relaying a packet here.

1) 1-d stop rule: Without an absolute way of certainty to determine and stop a routing loop, we resort to a familiar approach via a hop counter x_{HC}, but used in a sense of counting-up here. Furthermore, x_{HC} is used only in recovery mode operation since a packet in greedy mode will always take positive advancement and progressively converge to D if greedy mode keeps working. This requires that a packet in recovery mode carry a x_{HC} field.

Specifically, the hop counter is initialized to zero, $x_{HC} \leftarrow 0$, when routing operation for a packet is changed from greedy to recovery. It increases by one, $x_{HC} \leftarrow x_{HC} + 1$, when a node receives the packet in recovery mode and the packet is immediately dropped if x_{HC} is equal to some given maximum hop count constant M_{HC}.

It is clear that M_{HC} should be set to a large number in order to support a high probability of success in delivering a packet to destination. However, this will incur a significant cost of hop counts before dropping a packet if there is no route to D in a current network graph or loop routing has occurred. We thus develop a more efficient routing stop rule next.

2) 2-d stop rule: To save energy in futile routing, we consider a finite two-dimensional region for recovery mode operation, defined by total angle movement y_{TA} and hop count x_{HC}. The total angle y_{TA} is the sum of each angle from the direction of edge $v_{i-1}v_i$ to that of edge v_iv_{i+1}, $i \geq 0$, where v_{i-1}, v_i, and v_{i+1} are the previous, current, and next relay nodes, respectively, v_0 is the stuck node, and v_{-1} is the previous relay node before the packet first reaches v_0.

Specifically, let $\Delta\theta_i$, $i \geq 0$, be the angle from $\overrightarrow{v_{i-1}v_i}$ to $\overrightarrow{v_iv_{i+1}}$ and $\Delta\theta_0 = 0$ if the local minimum event occurs at the source node S where v_{-1} does not exist. The total angle movement is first initialized at node v_0 to be $y_{TA} \leftarrow 0$ and then updated at node v_i by

$$y_{TA} \leftarrow y_{TA} + \Delta\theta_i \tag{4}$$

For given positive constant angle factor C_A and maximum hop counts M_{HC}, the packet in recovery mode operation is dropped if (x_{HC}, y_{TA}) is not in the convex region defined by

$$x_{HC} < M_{HC} \tag{5}$$

$$|y_{TA}| \leq C_A \log_2(1 + x_{HC}). \tag{6}$$

V. Simulations and results

A. Simulation models

We consider two types of network topology of the same size 40×40 in the first quadrant of a two-dimensional Cartesian system. One without artificial voids is distributed uniformly with 4800, 4200, or 3600 nodes for three different node density levels. The other with two artificial voids shown in Fig. 4 (Refer to [12], [15] for the geometry of the two holes) is distributed

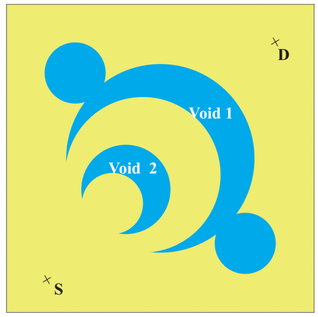


Fig. 4. A network area of 40×40 in the first quadrant has two artificial voids with geometry defined in [12], [15].

uniformly with 4320, 3840, or 3360 nodes over areas other than the two voids. In each network, the node closest to position $(5, 5)$ or $(35, 35)$ is designated as the source S and destination D of a packet, respectively.

In a network topology, any two nodes, say v_i and v_j, have a bidirectional link between them if and only if node v_i receives from v_j signals with power level larger than some given threshold p_t and vice versa; Otherwise, they have no direct communication link.

We consider Rayleigh fading, fix the mean received signal power to be 0 dB at distance 1, the reference distance [18], and use two path loss exponents 3 and 4 separately to create networks of different connection regularity [15]. Furthermore, we set $p_t = \ln(10)/2$ to be the power level threshold for success of receiving signals. Thus, for a signal transmitted by node v_i and path loss exponent 4, node v_j will receive mean signal power $1/|v_i v_j|^4$. The probability of the received signal power larger than p_t is then $e^{-p_t |v_i v_j|^4}$ [19], [20]. By our assumption of links with bidirectional communication, nodes v_i and v_j with $|v_i v_j| = 1$ have a link between them with probability $e^{-\ln(10)/2} \times e^{-\ln(10)/2} = \frac{1}{10}$. Note that under the constraint of the same received signal power level at reference distance, a higher transmit power is required for path loss exponent 4 than that for path exponent 3. Therefore, a network created based on path loss exponent 4 will support a higher degree of connection regularity than that based path loss exponent 3.

Besides, we consider a quasi-static network, in which a network topology and connection graph is created once and remains the same in a full session of routing a packet from S to D. For sending another packet, another random network topology and connection graph is created. Totally, there are 4×10^4 network graphs created or packets sent, which represents the size of a simulation sample for a given network setting in node density and path loss exponent. In each sample, the number of network graphs in which a path exists between S and D is given in columns 2 and 4 of Table II for path loss exponents 4 and 3, respectively. Those numbers are to be used in computing routing success probabilities (RT SP). They are obtained by performing the Dijkstra's shortest path algorithm [21] using global state information of each network graph. Their mean *shortest* path hop counts are then listed in columns 3 and 5 of the table accordingly. We see from the table that for a given number of nodes deployed, each sample of random networks created with path loss exponent 4 provides stronger connectivity between S and D (or shorter mean path hop counts) than that with path loss exponent 3, due to the setting of distance-path loss model mentioned previously.

B. Performance with α-CO_mTT

1) The effect of memory size m: In simulation, we set $m = 9600$ as a hypothetically unlimited memory size for a history list of visited nodes and the maximum hop counts $M_{HC} = 9600$ whenever recovery scheme α-CO_∞TT is invoked. Results on the proportion of packets successfully delivered from S to D are listed in columns 3 and 5 of Table III, under the term RT SP, for path loss exponents 4 and 3, respectively. They are the maximum achievable performance of RT SP for each greedy scheme working with α-CO_∞TT of no memory constraint nominally. Corresponding to each entry in the same row of the table, the greedy mode scheme employed is listed in the second column under xGF and optimal minimum SC size η_L, distance threshold d_t and/or enhance factor ω are listed in columns 4 and 6.

The impact of finite list sizes $m = 2^1, 2^2, \cdots, 2^8$ in recovery scheme α-CO_mTT on the performance of routing success probabilities is shown in Figs. 5(a)-(b) and 6(a)-(b) for networks without and with two artificial voids, respectively. Those in figures (a) and (b) are separately plotted for loss exponents 4 and 3. In each figure, the case for unlimited memory size $m = \infty$ is plotted on the rightmost, over x-coordinate 1024. All are optimal performance for each given network sample, total number of nodes, and m. We can see from all cases that routing success probabilities increase with memory length m and this trend is leveling off at about $m = 32$. These reveal a worthy tradeoff of extended packet headers for routing success rate.

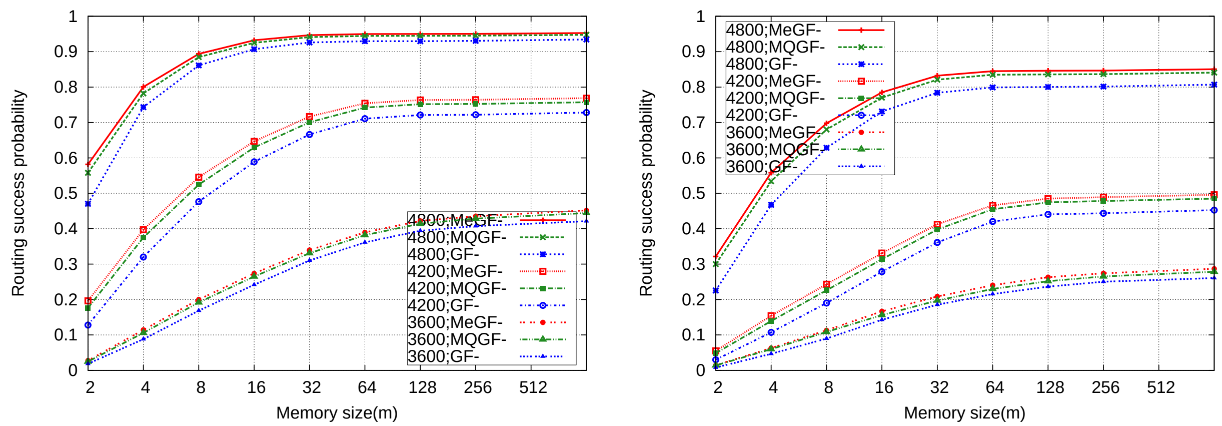
Comparing the performance of a chosen greedy scheme in Fig. 5(a) (resp. 6(a)) for path loss exponent 4 with that in Fig. 5(b) (resp. 6(b)) for path loss exponent 3, we see that the routing success probability is highly correlated with the regularity of connection between nodes in a network besides node densities. It should be noted that under UDA [11], [12] which can be viewed as the extreme case of connection regularity, all of those schemes support packet delivery guarantee without the need of memory overhead m. We also see from Figs. 6(a) and (b) that with recovery method α-CO$_m$TT, each routing scheme is highly capable of bypassing such large and peculiar voids as long as memory size m and node densities are not too low.

Exploiting RAI for more successful forwarding advancement, MeGF- or MQGF-αCO$_m$TT outperforms GF-αCO$_m$TT in routing success probabilities for all cases of node densities and list size m, as can be seen from Figs. 5 and 6. This advantage is less significant when network voids are obviously located between S to D, as shown in Figs. 6 (a) and (b).

Figs. 7 (a) and (b) show the mean hop counts of delivering a packet from S to D versus list size m for networks without and with two artificial voids, respectively. They are all for path loss exponent 4. Because of similarity, we omit the plots for path loss exponent 3. We can easily see that the cost of larger m for higher routing success performance is path hop counts for resolving local minimum issues. The cost of using those localized schemes is significant, as compared with the mean hop counts of shortest path route from S to D listed in column 3 of Table II. In particular, the mean hop count is extremely high for networks with low total numbers of nodes, 3600 or 3360. We also see that MeGF- or MQGF-αCO$_m$TT is able to significantly cut down the cost of mean hop counts for delivering a packet, agreeing with the design objective in [12] under UDA. With more fine-tuned use of RAI, MeGF-αCO$_m$TT performs the best in routing success rate and mean hop counts.

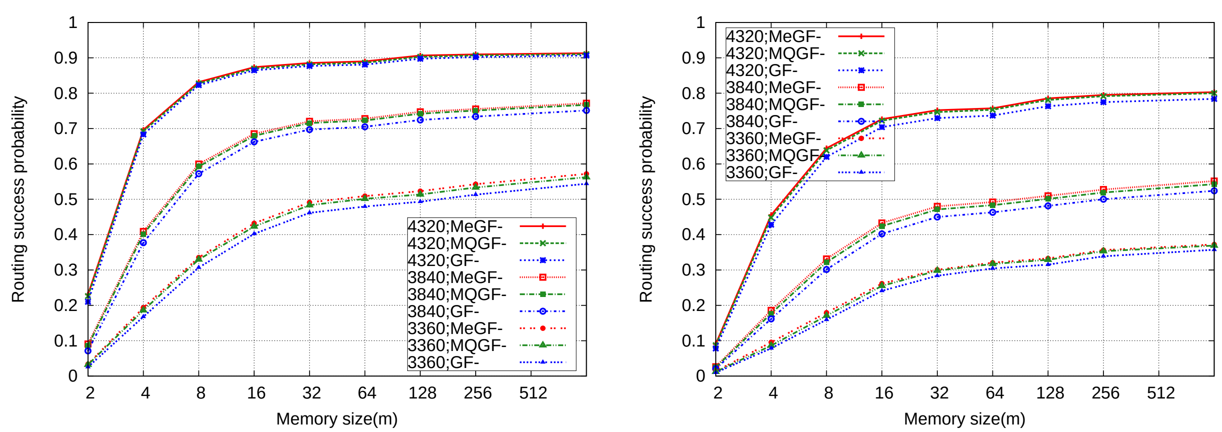
Finally it should be noted that those routing success probabilities result from the setting of a large distance between S and D or large artificial voids between them. They should be higher in general cases. For example, if S and D are the nodes closest to $(12.5, 12.5)$ and to $(27.5, 27.5)$, respectively, in the sample of no artificial void networks with 4200 nodes and path loss exponent 4, then optimal routing success probability by GF-αCO$_{16}$TT is 82.6% (not shown here) as compared to 58.9% (see Fig. 5(a)) for the previous setting of S and D. On the other hand, if S and D are the nodes closest to $(15, 15)$ and to $(30, 30)$, respectively, in the sample of two artificial void networks with 3840 nodes and path loss exponent 4, then optimal routing success probability by GF-αCO$_{16}$TT is 69.6% (not shown here) as compared to 66.3% (see Fig. 6(a)). The latter is because the mean hop count of shortest SD routing path is still 104.4 hops as compared to the previous 111.4 hops, although the direct distance $|SD|$ is half the previous one.

Because of complexity increased with involving one more state variable RAI, we only present results for routing scheme GF-αCO$_m$TT in the following. To reduce space, we further limit presentation for path loss exponent 4 and for 4200 and 3840 nodes in networks without and with artificial voids, respectively.



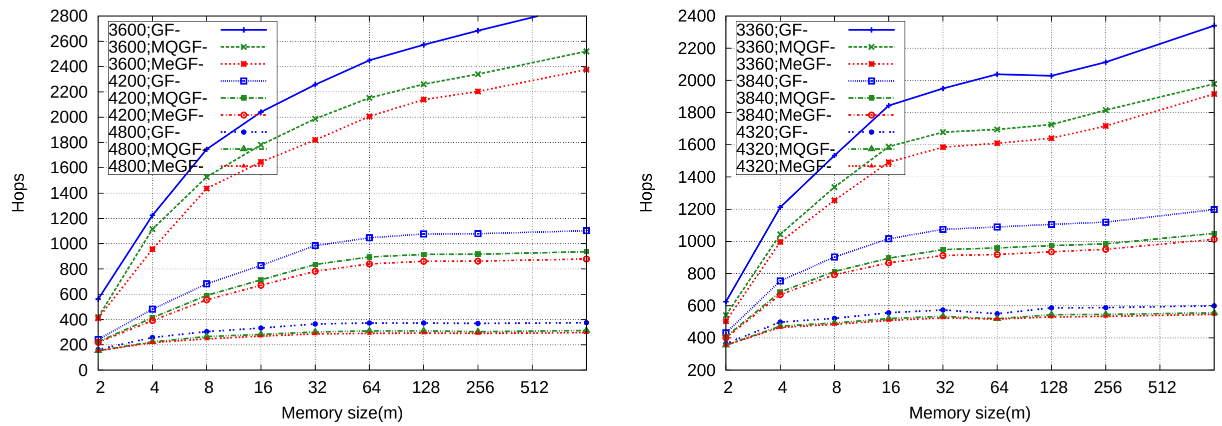
(a) (b)

Fig. 5. The probability of successful routing from S to D versus memory length (m) for path loss exponent (a) 4 and (b) 3, in networks without artificial voids.



(a) (b)

Fig. 6. The probability of successful routing from S to D versus memory length (m) for path loss exponent (a) 4 and (b) 3, in networks with two artificial voids.

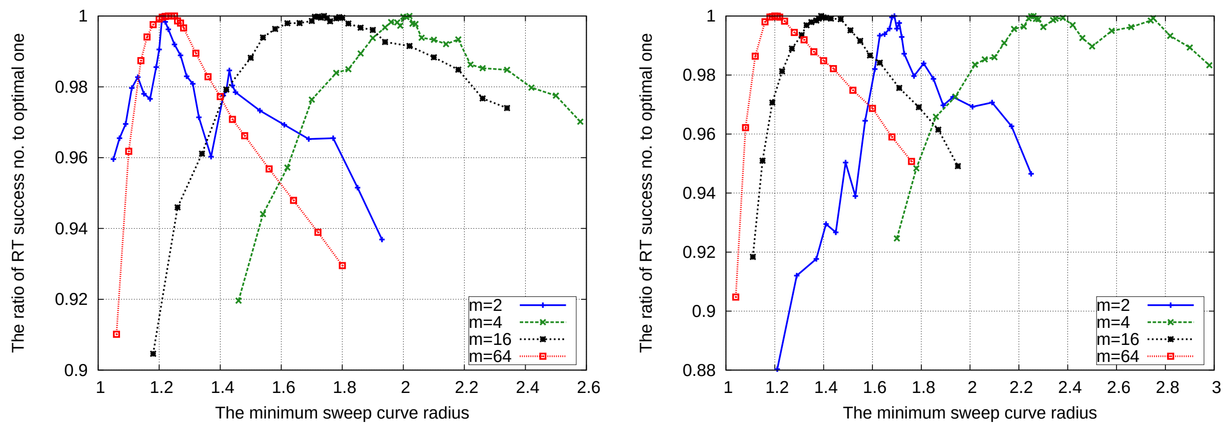


(a) (b)

Fig. 7. Average SD-path hop counts versus memory length (m) for path loss exponent 4 and networks (a) without artificial voids and (b) with two artificial voids.

2) The effect of minimum sweep curve size, η_L: Fig. 8(a) and (b) show the ratio of optimal routing success probability for a given minimum SC size η_L to the global optimal result versus η_L, for networks with no artificial voids and 4200 nodes and with two artificial voids and 3840 node, respectively. For simplicity, only results for list size $m = 2, 4, 16, 64$ and path loss exponent 4 are plotted. We see that for each given list size m, there exists a global optimal SC size, denoted as $\eta_L^*(m)$, for maximum routing success performance and that except $m = 2$, $\eta_L^*(m)$ generally decreases with $m \geq 4$ and approaches to the corresponding entry listed in column 4 of Table III for unlimited m. We also see that the loss of routing success performance is more sensitive for assigned minimum SC size $\eta_L(m) < \eta_L^*(m)$. The reason is mainly that if $\eta_L(m)$ is assigned too small, relay nodes selected by such an SC tend to lie further out of a network void boundary, which violates the criteria for performing rotational sweeps to successfully bypass the void. On the other hand, if $\eta_L(m)$ is too large, the SC may miss hitting some potential edge-intersection node [11] that may have an edge leading to D, causing performance loss.

3) The effect of distance threshold, d_t: For each $m = 2, 4, 16, 64$ and optimal minimum SC size $\eta_L^*(m)$ assigned, the ratio of routing success numbers to optimal ones versus d_t is shown in Figs. 9 (a) and (b) for the previous samples of networks. Here the setting $d_t = 64$ on the x-coordinate implies that the routing scheme always seeks to operate in greedy mode since $64 > |SD| = 30\sqrt{2}$. We see that for each m there exists a range of optimal d_t^* for maximum



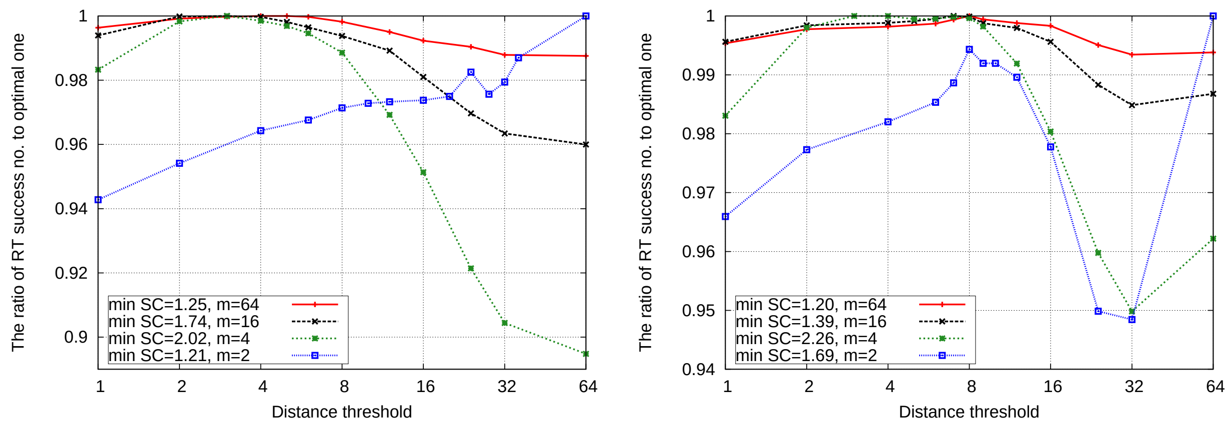
(a) (b)

Fig. 8. Routing success performance relative to optimal achievable one versus minimum sweep curve size (η_L) for path loss exponent 4 and networks (a) without artificial voids and with 4200 nodes deployed and (b) with two artificial voids and with 3840 nodes deployed. Results are optimized with regard to d_t, for routing scheme GF-αCO$_m$TT.

routing success performance. Except $m = 2$, d_t^* is at about 4 for the sample of networks without artificial voids and about 8 for that with two artificial voids. This agrees with the design objective of using d_t described in Sec II-B4. Recovery scheme α-CO$_m$TT with insufficient memory size $m = 2$ is unable to resolve most local minimum issues through taking more detour hops, which may then lead to facing further routing problems. This thus supports the strategy of switching to greedy mode as early as possible, by setting $d_t^* = 64$.

C. *The efficiency of routing stop rule*

Through simulations of GF-αCO$_m$TT in a given sample of networks with optimal setting of η_L and d_t, we search the minimum number of hundreds, $M_{HC}^* = \min_{i \in Z}(i \times 100)$, for maximum hop counts allowed such that the number of routing success is still equal to that achieved for any maximum hop count $M_{HC} > M_{HC}^*$ set in recovery mode operation. We can roughly view M_{HC}^* as the best number of M_{HC} for *1-d* routing stop decision under a given network and *S-D* setting. With such an M_{HC}^*, shown in the legends of Fig. 10, the mean number of hops per failure of packet delivery, that may due to no *S-D* path or a routing loop, is plotted over the rightmost x-coordinate, 9, in the figures. This mean hop number is regarded as the minimum mean wastage per futile packet delivery, based on *1-d* routing stop rule. For example, it is 2051.2 hops for list size $m = 4$ and *1-d* routing stop limit $M_{HC}^* = 1900$ in Fig. 10 (a) for the sample of networks



(a) (b)

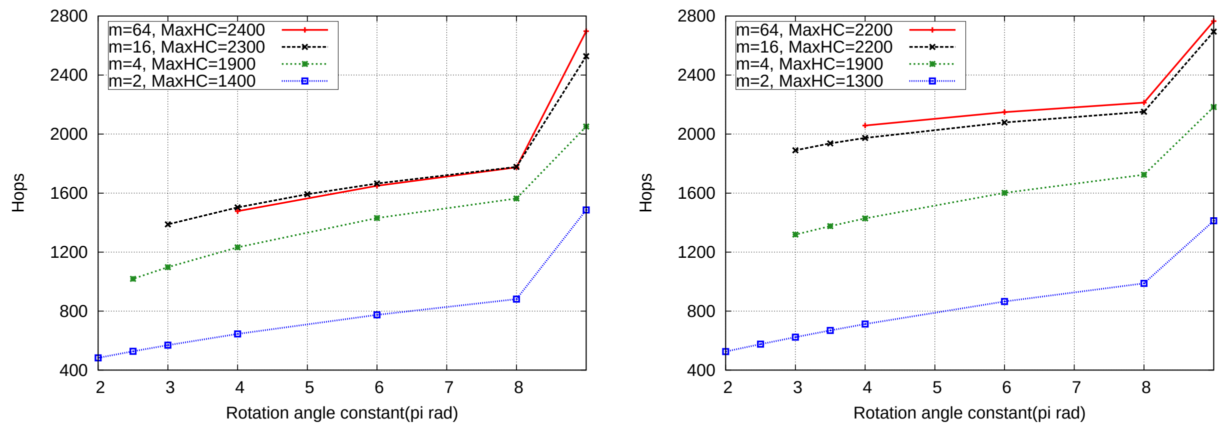
Fig. 9. Routing success performance relative to optimal achievable one versus distance threshold (d_t) for path loss exponent 4 and networks (a) without artificial voids and with 4200 nodes deployed and (b) with two artificial voids and with 3840 nodes deployed. Optimal minimum SC size η_L for each m is employed. The routing scheme is GF-αCO$_m$TT.

with 4200 nodes, no artificial voids and loss exponent 4.

We then set angle factor $C_A = x\pi$ for each given $x = \frac{i}{2}, i \in Z$, and $M_{HC} = M^*_{HC}$ found previously for *2-d* routing stop rule defined by (5) and (6) in simulations. Figs. 10 (a) and (b) show the mean number of hops per failure of packet delivery versus C_A for the sample of networks without and with two artificial voids, respectively. In these two figures, we only plot results for C_A giving rise to no loss of routing success numbers. It is obvious that routing with the proposed *2-d* stop rule imposed on operating α-CO$_m$TT will save significant energy resource for futile transmissions. The saving relative to that by *1-d* stop rule with best setting is more significant for lower memory list size m. If the previous example for $m = 4$ is changed to use *2-d* stop rule based on such constants $C_A = \frac{5}{2}\pi$ and $M_{HC} = 1900$, the mean number of hop counts per futile packet delivery is 1018.8, the y-coordinate over x-coordinate $\frac{5}{2}\pi$ in Fig. 10 (a). This number is a little less than half the previous 2051.2 with *1-d* stop rule. However, this advantage is at the cost of extended packet header to carry one more variable, y_{TA} defined by (4).

VI. Conclusion

We have presented an adaptive and cooperative rotational sweep algorithm based on circular arcs of varying sizes (α-CO$_m$TT) for bypassing routing holes and recovering from greedy forwarding failure in geographic routing under realistic scenarios of wireless coverage. It retains



(a) (b)

Fig. 10. Average hop counts upon terminating S-to-D routing (per routing failure) versus rotation angle constant C_A (π radian) used to delimit a 2-d operation regions for path loss exponent 4 and networks (a) without artificial voids and with 4200 nodes deployed and (b) with two artificial voids and with 3840 nodes deployed. Optimal η_L and d_t for each m is employed, as shown in Fig. 9. The routing scheme is GF-αCO$_m$TT.

the feature of localized routing and thus scalability. Results have shown that a reasonable routing success probability can be achieved at the cost of packet overheads carrying a few visited node positions in recent history of recovery mode operation. To reduce the wastage of energy for futile transmissions, a two dimensional routing stop rule has been proposed and imposed in recovery mode operation, successfully achieving an early cutoff of a potential loop routing. Future research challenges include an analytic way to resolve the minimum size of circular arc employed in α-CO$_m$TT for optimal performance and also a practical way to determine the constants with two-dimensional stop rule.

References

[1] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Tran. Commun.*, vol. 32, pp. 246–257, Mar. 1984.

[2] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conf. Computational Geometry,* Vancouver, BC, Canada, Aug. 1999, pp. 51-54.

[3] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. IEEE/ACM Mobicom*, Boston, MA, Aug. 2000, pp. 243–254.

[4] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Univ. Southern California, Tech. Rep. ISI/RR-87-180, Mar. 1987.

[5] D. Chen and P.K. Varshney,“A survey of void handling techniques for geographic routing in wireless networks,” *IEEE Comm. Surveys and Tutorials,* vol. 9, no. 1, pp. 50-67, 2007.

[6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia,“Routing with guaranteed delivery in ad hoc wireless networks,” in *Proc. 3rd Int. Workshop Discrete Algorithms and Methods for Mobile Computing and Commun.*, 1999, pp. 48–55.

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

[8] B. Karp, “Geographic routing for wireless networks,” Ph.D. thesis, Harvard Univ., Cambridge, MA, 2000.

[9] Wen-Jiunn Liu, Kai-Ten Feng,“Greedy routing with anti-void traversal for wireless sensor networks”, *IEEE Trans. Mobile Computing,* vol. 8, no. 7, pp. 910-922, 2009.

[10] S. Rüehrup and I. Stojmenović, “Optimizing communication overhead while reducing path length in beaconless georouting with guaranteed delivery for wireless sensor networks,” *IEEE Tran. Comput.,* vol. 62, no. 12, pp. 2440-2453, Dec. 2013.

[11] A. Mostefaoui, M. Melkemi, and A. Boukerche, “Localized routing approach to bypass holes in wireless sensor networks,” *IEEE Tran. Comput.,* vol. 63, no. 12, pp. 3053-3065, Dec. 2014.

[12] J.-T. Tsai and Y.-H. Han “Geographic routing with enhanced local information for wireless networks”, *Journal of Information Science and Engineering*, vol. 32, no. 5, pp. 1261-1288, Sept. 2016.

[13] Q. Fang, J. Gao, and L. Guibas,“Locating and bypassing routing holes in sensor networks,” in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp. 2458-2468.

[14] S. Rührup, H. Kalosha, A. Nayak, and I. Stojmenović, “Message-efficient beaconless georouting with guaranteed delivery in wireless sensor, ad hoc, and actuator networks,” *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 95-108, Feb. 2010.

[15] J.-T. Tsai and Y.-H. Han “Cooperative rotational sweep schemes for geographic routing”, in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, Kuala Lumpur, Malaysia, May 2016.

[16] M. Zorzi and R. R. Rao, “Geographic random forwarding (GeRaF) for Ad Hoc and sensor networks: multihop performance,” *IEEE Trans. Mobile Computing,* vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.

[17] J. A. Sanchez, P. M. Ruiz, and R. Marin-Perez,“Beacon-less geographic routing made practical: challenges, design guidelines, and protocols,” *IEEE Commun. Mag.,* vol. 47, no. 8, pp. 85-91, Aug. 2009.

[18] T. S. Rappaport, *Wireless Communications: Principles and Practice.* NJ:Prentice- Hall, 1996.

[19] T. Aguilar, S.-J. Syue, V. Gauthier, H. Afifi, and C.-L. Wang, “CoopGeo: A beaconless geographic cross-layer protocol for cooperative wireless ad hoc networks,” *IEEE Trans. Wireless Commun.,* vol. 10, no. 8, pp. 2554–2565, Aug. 2011.

[20] J.-T. Tsai, ”Transmission rate scheduling and stopping time for time-sensitive multicast stream traffic in cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 1754–1765, April 2014.

[21] D. Bertsekas and R. Gallager, *Data Networks*, 2^{nd} Ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[22] Y.B. Ko and N.H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks,” in *Proc. ACM/IEEE MobiCom,* 1998, pp. 66-75.

[23] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, “Geometric routing: Of theory and practice,” in *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC),* 2003.

[24] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang, “GLIDER: Gradient landmark-based distributed routing for sensor networks,” in *Proc. IEEE INFOCOM*, 2005, vol. 1, pp. 339-350.

[25] Y-J Kim, R. Govindan, B. Karp, and S. Shenker, “On the pitfalls of geographic face routing,” in *Proc. DIALM-POMC’05*, Sept. 2005, pp 34-43.

[26] S. S. Lam and C. Qian, “Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 663–677, April 2013.

Algorithm 2: Adaptive Rotational Sweep Scheme, α-TT

Require: Node v_i has a packet with state information $\{D, \text{mode=Recovery}, v_A, T_r, B_I\}$. The minimum SC size is η_L.

Ensure: Select a node $v_{i+1} \in \mathcal{N}_i$. $\{$Suppose $|\mathcal{N}_i| \geq 1\}$

Find h_i on the start-sweep line $v_i h_i$ by Algorithm 1, where $h_i = h(D, v_i)$ if $T_r = \emptyset$ and $h_i = h(T_r[|T_r| - 1], v_i)$ otherwise.
$\eta \leftarrow \max_{v_\ell \in \mathcal{N}_i}\{|v_\ell v_i|\}$
if $\eta \leq \eta_L$ **then**
    $n_a \leftarrow$ the node in $\mathcal{N}_i$ first hit by SC(η_L) swept from line $v_i h_i$.
else
    $n_a \leftarrow$ the node in $\mathcal{N}_i$ first hit by SC(η) swept from line $v_i h_i$
    repeat
        $\eta \leftarrow |v_i n_a|$
        $n_b \leftarrow n_a$
        $n_a \leftarrow$ the node in $\mathcal{N}_i$ first hit by SC($\max\{\eta, \eta_L\}$) swept from line $v_i h_i$
    until $n_b == n_a$ or $\eta \leq \eta_L$
end if
$v_{i+1} \leftarrow n_a$ $\{$Except updating T_r next, α-TT ends.$\}$
if $|v_{i+1}D| < |v_A D|$ **then**
    mode$\leftarrow$ GREEDY $\{$Perform xGF at next node.$\}$
else
    if $|T_r| < m$ **then**
        Push v_i from the tail into list T_r.
    else
        Pop out and drop the header entry $T_r[0]$ while push v_i from the tail into list T_r.
        $B_I \leftarrow$ TRUE if B_I is FALSE. $\{$TRUE: Buffer T_r ever overflows.$\}$
    end if
end if
return v_{i+1}

Algorithm 3: Adaptive & Cooperative Rotational Sweep

Require: Node v_i has a packet with state information $\{D, \text{mode=Recovery}, v_A, T_r(0, i-1), B_I\}$. The minimum SC size is η_L.

Ensure: Select next relay $v_{i+1} \neq v_i$ not in history order $T_r \cup \{v_i\}$ by performing α-CO$_m$ or from N_i by calling α-TT.

$k \leftarrow -1$
$j \leftarrow 0$ if B_I ==FALSE, otherwise $j \leftarrow 1$.
while $k < 0$ and $j < |T_r|$ **do**
    Find h_j on start-sweep line $T_r[j]h_j$ by Algorithm 1, where $h_j = h(D, T_r[0])$ if $j == 0$ and B_I ==FALSE and $h_j = h(T_r[j-1], T_r[j])$ otherwise.
    $\eta \leftarrow \max\{|T_r[j]T_r[j+1]|, \eta_L\}$ if $j < |T_r| - 1$; otherwise $\eta \leftarrow \max\{|T_r[j]v_i|, \eta_L\}$
    if $\eta \leq \eta_L$ **then**
        $n_a \leftarrow$ the node in $T_r \cup \mathcal{N}_i \cup \{v_i\}$ first hit by SC(η_L) swept from line $T_r[j]h_j$.
    else
        $n_a \leftarrow$ the node in $T_r \cup \mathcal{N}_i \cup \{v_i\}$ first hit by SC(η) swept from line $T_r[j]h_j$
        repeat
            $\eta \leftarrow |T_r[j]n_a|$
            $n_b \leftarrow n_a$
            $n_a \leftarrow$ the node in $T_r \cup \mathcal{N}_i \cup \{v_i\}$ first hit by SC($\max\{\eta, \eta_L\}$) swept from line $T_r[j]h_j$
        until $n_b == n_a$ or $\eta \leq \eta_L$
    end if
    if n_a is $T_r[j+1]$ for $j+1 < |T_r|$ or v_i for $j+1 == |T_r|$ **then**
        $j \leftarrow j+1$
    else
        $k \leftarrow j$
    end if
end while
if $k \geq 0$ **then**
    Flush $\{T_r[\ell]; \ell = k+1, k+2, \cdots, |T_r|-1\}$ out of T_r to obtain the update $T_r[\ell] \neq \emptyset, \ell = 0, 1, \cdots, k$.
end if
if $k \geq 0$ and $n_a \neq v_i$ **then**
    Select node $v_{i+1} \leftarrow n_a$\{a hidden node discovered where α-CO$_m$ is to be performed.\}
else
    Call procedure α-TT (Algorithm 2) to obtain v_{i+1}.
end if
return v_{i+1}

TABLE I

ROUTING OPERATION CHANGES TO RECOVERY MODE α-CO_2TT AT NODE v_0 IN THE NETWORK OF FIG.2 AND CONTINUES IN THIS MODE FROM NODE(ROW) v_0 TO v_6. AT NODE(ROW) v_7, IT CHANGES TO GF MODE.

Pkt@	Method	SC size	hit	T_r, B_I
v_0	α-TT	$\eta_0 = \|v_0 v_{15}\| > \eta_L$	v_1	
		$\eta_1 = \|v_0 v_1\| > \eta_L$	v_1	$\{v_0\}, F$
v_1	α-CO_2	$\eta_1 = \|v_0 v_1\| > \eta_L$ @$T_r[0]$	v_1	
	α-TT	$\eta_3 = \|v_1 v_{16}\| > \eta_L$	v_8	
		$\eta_4 = \|v_1 v_8\| > \eta_L$	v_2	
		$\eta_L > \|v_1 v_2\|$	v_2	$\{v_0, v_1\}, F$
v_2	α-CO_2	$\eta_1 = \|v_0 v_1\| > \eta_L$ @$T_r[0]$	v_1	
		$\eta_L > \|v_1 v_2\|$ @$T_r[1]$	v_3	
v_3	α-CO_2	$\eta_1 = \|v_0 v_1\| > \eta_L$ @$T_r[0]$	v_1	
		$\eta_L > \|v_1 v_3\|$ @$T_r[1]$	v_3	
	α-TT	$\eta_L > \|v_3 v_4\|$	v_4	$\{v_1, v_3\}, T$
v_4	α-CO_2	$\eta_L > \|v_3 v_4\|$ @$T_r[1]$	v_5	
v_5	α-CO_2	$\eta_L > \|v_3 v_5\|$ @$T_r[1]$	v_6	
v_6	α-CO_2	$\eta_L > \|v_3 v_6\|$ @$T_r[1]$	v_6	
	α-TT	$\eta_L > \|v_6 v_7\|$	v_7	mode=GF
v_7	GF	NA	D	NA

TABLE II

TOTALLY 4×10^4 RANDOM NETWORK GRAPHS ARE CREATED FOR EACH SAMPLE OF NETWORK SETTING IN NODE DENSITY AND WIRELESS PATH LOSS EXPONENT. EACH ENTRY IN COLUMN 2 OR 4 SHOWS THE NUMBER OF THEM HAVING A PATH BETWEEN THE NODE CLOSEST TO $(5,5)$ AND THAT CLOSEST TO $(35,35)$ FOR A GIVEN NUMBER OF NODES, LISTED IN COLUMN 1, UNIFORMLY DEPLOYED. CORRESPONDINGLY THE MEAN SHORTEST PATH HOP COUNT IS LISTED IN COLUMN 3 OR 5.

Path Loss E.	4		3	
Nodes	Connected	Avg. hops	Connected	Avg. hops
No artificial void				
4800	38155	76.25	36670	79.57
4200	33072	87.11	27455	94.18
3600	9095	108.00	2141	112.55
Two artificial voids				
4320	38105	101.84	36080	105.14
3840	29588	111.44	21405	116.94
3360	7374	124.95	2214	130.37

TABLE III

MAXIMUM ROUTING SUCCESS PROBABILITY (RT SP) UNDER UNLIMITED HISTORY LIST SIZE m AND OPTIMAL η_L, d_t AND/OR ω FOR EACH GREEDY FORWARDING METHOD (xGF) COMBINED WITH α-CO$_\infty$TT .

Networks without artificial void					
Path Loss E.		4		3	
Nodes	xGF	RT SP	$\eta_L; d_t; \omega$	RT SP	$\eta_L; d_t; \omega$
4800	GF	.9344	1.26; 2	.8068	1.28; 4
	MQGF	.9476	1.21; 3(5)	.8407	1.29; 2
	MeGF	.9526	1.18; 2; 0.8	.8505	1.26; 6; 1.1
4200	GF	.7281	1.18; 4(5)	.4526	1.26; 5
	MQGF	.7572	1.18; 4	.4851	1.26; 5
	MeGF	.7687	1.19; 4; 0.9	.4958	1.26; 6; 1.1
3600	GF	.423	1.14; 8	.261	1.21; 19(17)
	MQGF	.444	1.16; 9	.278	1.21; 17
	MeGF	.452	1.16; 8; 1.2	.287	1.21; 17; 0.6
Networks with two artificial voids					
4320	GF	.9066	1.20; 3	.7838	1.28; 9
	MQGF	.9104	1.19; 4(8)	.7999	1.28; 7
	MeGF	.9131	1.17; 8; 0.6	.8027	1.25; 7(9); 0.9
3840	GF	.7513	1.18; 8	.5239	1.23; 10
	MQGF	.7670	1.18; 6	.5427	1.23; 10
	MeGF	.7717	1.18; 8; 0.7	.5516	1.23; 8; 0.8
3360	GF	.544	1.19; 9	.357	1.25; 64
	MQGF	.562	1.20; 3(4)	.369	1.26; 64
	MeGF	.571	1.19; 10; 1.3	.372	1.25; 64; 1.3

106年度專題研究計畫成果彙整表

計畫主持人：蔡榮宗					**計畫編號：**106-2221-E-003-005-		
計畫名稱：應用於實際無線網路幾何繞徑的調變式合作循環橫掃方法							
成果項目					量化	單位	質化（說明：各成果項目請附佐證資料或細項說明，如期刊名稱、年份、卷期、起訖頁數、證號...等）
國內	學術性論文	期刊論文			0	篇	
		研討會論文			0		
		專書			0	本	
		專書論文			0	章	
		技術報告			1	篇	MOST 2018 project report has been in a state of manuscript suitable for submission to some top Journal.
		其他			0	篇	
	智慧財產權及成果	專利權	發明專利	申請中	0	件	
				已獲得	0		
			新型/設計專利		0		
		商標權			0		
		營業秘密			0		
		積體電路電路布局權			0		
		著作權			0		
		品種權			0		
		其他			0		
	技術移轉	件數			0	件	
		收入			0	千元	
國外	學術性論文	期刊論文			1	篇	Alternative forwarding strategies for geographic routing in wireless networks, International Journal of Ad Hoc and Ubiquitous Computing, Vol 27, No. 4, 2018, pp295~307
		研討會論文			0		
		專書			0	本	
		專書論文			0	章	
		技術報告			0	篇	
		其他			0	篇	
	智慧財產權及成果	專利權	發明專利	申請中	0	件	
				已獲得	0		
			新型/設計專利		0		
		商標權			0		

		營業秘密	0		
		積體電路電路布局權	0		
		著作權	0		
		品種權	0		
		其他	0		
	技術移轉	件數	0	件	
		收入	0	千元	
參與計畫人力	本國籍	大專生	3	人次	開發視覺化繞徑方法電腦軟體程式
		碩士生	1		將所提出的無線繞徑方法應用於 Ad Hoc Network Beaconless Routing
		博士生	0		
		博士後研究員	0		
		專任助理	0		
	非本國籍	大專生	0		
		碩士生	1		降低計算複雜度可行性
		博士生	0		
		博士後研究員	0		
		專任助理	0		
其他成果 （無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。）					

科技部補助專題研究計畫成果自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現（簡要敘述成果是否具有政策應用參考價值及具影響公共利益之重大發現）或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標
□未達成目標（請說明，以100字為限）
　　□實驗失敗
　　□因故實驗中斷
　　□其他原因
說明：

2. 研究成果在學術期刊發表或申請專利等情形（請於其他欄註明專利及技轉之證號、合約、申請及洽談等詳細資訊）

論文：□已發表　□未發表之文稿　■撰寫中　□無
專利：□已獲得　□申請中　■無
技轉：□已技轉　□洽談中　■無
其他：（以200字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性，以500字為限）

過去相關研究及學術界重要文章發表的幾何路由繞徑方法, 幾乎完全建立在理想的Unit Disk Graph 的無線網路連結假設之下, 結果這些方法在實際的無線連結環境
中並不可行。我們直接以實際非常不完美的無線連結環境為基礎,所開發提出的繞境方法,透過耗時的電腦模擬, 數據成果顯示我們新創的調變暨合作螺旋橫掃繞徑決策模式有相當高的可行性,可進一步發展在beaconless routing。

4. 主要發現

本研究具有政策應用參考價值：■否　□是，建議提供機關
（勾選「是」者，請列舉建議可提供施政參考之業務主管機關）
本研究具影響公共利益之重大發現：□否　□是
說明：（以150字為限）
1. 調變暨合作螺旋橫掃繞徑決策模式適用於實際無線連結環境中作為幾何繞徑貪婪傳遞模式失敗的補救措施
2. 解決螺旋橫掃繞徑補救措施時無窮迴圈中斷的有效方式
3. 找出並確定繞徑效能影響因素:封包表頭長度，橫掃圓曲線大小，貪婪切換運作模式的終點距離設定