SR: A Cross-Layer Routing in Wireless Ad Hoc Sensor Networks

Zhen Jiang
Dept. of Computer Sci.
West Chester University
West Chester, PA 19383

Zhigang Li, Nong Xiao
Dept. of Computer Sci.
Natl. Univ. of Defense Tech.
Changsha, P. R. China

Jie Wu
Dept. of Computer Sci. & Eng.
Florida Atlantic University
Boca Raton, FL 33431

***Abstract*—In this paper, we extend our early work on safety information in wireless ad hoc sensor networks to the realistic communication model in which each node has the opportunity to receive the signal from any node in the entire network. The safety information is a value $\in [0, 1]$ at each node which is calculated in a fully distributed manner based on 1-hop neighborhood information. It indicates the maximum probability of a successful straightforward path from this node to a destination along the edge of the networks. Such information can be used to mitigate the impact of local minima and achieve more straightforward paths in the geographic greedy forwarding. The whole construction process can converge quickly and be controlled in a limited area. By integrating with the MAC protocols, its cost can be minimized. Both analytical and experimental results illustrate that our new approach is cost-effective, compared with the best results known to date.**

***Index Terms*—distributed algorithms, greedy advance, information model, routing, wireless ad-hoc sensor networks.**

I. Introduction

Geographic greedy forwarding (GF) [4], [7], as a simple, efficient and scalable strategy, is the most promising routing scheme in wireless ad-hoc sensor networks (WASNs). In such a multi-hop unicasting, the path from the source to the destination is determined at each intermediate node in a fully-distributed manner by selecting its successor in the forwarding. The packet advances greedily along a *straightforward* path hop by hop. Not only can it avoid wasting energy in detours, but also less interference occurs in other transmissions when fewer nodes are involved in the transmission. This plays an important role in WASN applications that require a reliable data steaming service.

An important challenge often faced in GF routing is the "local minimum phenomenon" [1] where the forwarding process is blocked at a node called *stuck node*. The occurrence of block can be caused by not only the "deployment hole" such as sparse deployment and physical obstacles, but also many dynamic factors, including node failures, signal fading, communication jamming, power exhaustion, interference, and node mobility [1], [8], [12].

In our early work [5], the safety information model is provided for the forwarding (also called LF routing) that is limited within the request zone in LAR scheme 1 in [7]. Not only stuck nodes but also those nodes that their succeeding forwarding are all blocked by stuck nodes are identified as unsafe nodes. A straightforward path can be achieved if and only if safe nodes are used. However, the calculation of such a boolean value of status relies on a fixed, ideal network topology. In this paper, we extend such a safety information model and the corresponding routing to a realistic communication model [14], in which each node has the opportunity to receive the signal from any node in the entire network.

Our contributions is twofold: **(a)** We study the effect of local minima on LF routing in dynamic networks and its structural regularity in local description at each node that can precisely infer the global topology evolution for constituting a straightforward path in the multi-hop systems, for any possible pair of the source and destination. **(b)** Considering the variation of link status, we propose a scalable, distributed method to collect the information for the construction of such a local descriptor at each node.

The implementation of our new information model is integrated with the reservation MAC protocol (e.g., [15]). While a stable topology link is confirmed via the beaconing scheme, its two end nodes also exchange the information of their safety status by the beacon messages. According to such 1-hop neighborhood information that is newly collected, each node updates its own status. The safety status under the realistic communication model (SR) is a value $\in [0, 1]$, which indicates the maximum probability of a successful straightforward path from this node to a destination along the edge of the networks, via bi-directional links. The larger the value, the more likely forwarding will be successful and the more reliable the path will be for communication. "0" stands for the *unsafe* status of this node in a greedy advance. Otherwise, the node is called *safe*. Such a safe value also implies a higher success rate of valid forwarding to any closer destination. By extending the LF routing under SR model, more straightforward paths can be achieved to avoid unnecessary detour so that the performance and quality of data communication can be improved.

Note that the routing only needs the information for one successful path. It is not necessary to collect the information of all unstable links. The bi-directional link is used in our approach: the outgoing link is for packet forwarding and the

This work was supported in part by NSF grants CNS 0422762, CNS 0434533, CNS 0531410, CNS 0626240, and CCF 0840891. Contact E-mail: zjiang@wcupa.edu.

incoming link is for guaranty information collection. There may be cases when differences in transmission power give rise to unidirectional links. However, as indicated in [10], the main difficulty of using unidirectional links comes from the asymmetric knowledge about message reception at its end nodes, which requires a three-party agreement. This usually causes unexpected delay or unnecessary re-transmissions in real applications. On the other hand, with our SR information, the routing can take advantage of any alternative path and avoid being stuck with any unidirectional link. Because each node needs to apply beaconing scheme constantly to maintain the connection to its neighbors, the construction cost of our SR information can be ignored. However, the information, which is a local representative of neighboring nodes, needs to be simple enough to fit in a small size message while staying efficient for the global optimization of the entire path.

II. Related Work

As indicated in [5], the node availability in forwarding is relative when the source and destination changes the relative locations. Considering the signal that can be received via lossy links [2], the realistic communication model [14] increases the complexity of the forwarding at each node and makes it even more difficult to precisely catch the diverse availability of a node in the description of topological evolution.

By adopting LF routing, the safety level model presented in [5] achieves a balance point of tradeoff between the structure regularity of node status and the adaptive flexibility. For a given network configuration, the safe/unsafe status of each node is deterministic and unique. A straightforward path can be achieved if and only if safe nodes are used. However, the flip-flop of a link status in the reality of WASN application that is caused by any dynamic factor will affect the calculation of such statuses and make them unstable. A more accurate safety description of dynamic variation in WASNs that does not rely on any single connection link in neighborhood is required.

GMS [6] provides a greedy advance solution by looking ahead for the node statuses within a k-hops distance. It adopts a reactive model which requires a probing process. The GMS cannot achieve global optimization until k is set as the diameter of the networks. Moreover, when the realistic communication model [14] is adopted to utilize the lossy link connection, each node will have too many neighbors due to its possible connection to all of the nodes in the entire network. Thus, a more scalable model in which the information construction can be controlled in a limited area, while inferring the global configuration of each straightforward path, is required for a practical routing solution.

III. Realistic Communication Model

Network model. We model a WASN as a directed graph $G = (V, E)$, where V is a set of vertices including all of the nodes and E is a set of directed links, each of which indicate the link between two nodes and the direction of the data flow on this link. Each node u has the location (x_u, y_u), simply denoted by $L(u)$. In a communication, assume node s is the source node, u is the current node, d is the destination node. For each link $u \to v \in E$, $\lambda_{u\to v} \in [0, 1]$ indicates the probability that the signal from node u can be successfully received at node v, called the *link reachability* in [9]. Its value is affected by node failure, energy depletion, signal fading, or node mobility. We adopt the quality model observed from the Berkeley Mica mote platform [14] to determine each $\lambda_{u\to v}$ as follows, with respect to the distance of link (i.e., $D(u, v) = \mid L(u) - L(v) \mid$).

$$\lambda_{u\to v} \begin{cases} \in & (0.9, 1], \quad D(u,v) \le 10\ feet \\ \simeq & 0, \qquad\quad D(u,v) > 40\ feet \\ \in & (0, 1), \qquad otherwise \end{cases} \tag{1}$$

The reservation MAC protocol (e.g., [15]) confirms the reliable connections to neighbors. Each node u maintains its reliable incoming links $\in E$ and the corresponding channel assignment. $N(u)$ denotes the corresponding 1-hop neighbor at the other end of these links. Among $N(u)$, neighbors that are connected by bi-directional links, denoted by $n(u)$, can be verified. Each node u will exchange the information with its $n(u)$ neighbors and update its own safety status. According to the value, it determines whether it is disabled (a stuck node $\in \Gamma$), safe (> 0), or unsafe. Considering the interference caused by any existing data transmission from a node u, the reception node v will gain the knowledge of such a channel assignment with the MAC protocol and exclude itself from the n set of its neighbors, say $n(w)$ set at any node w, when the quantum windows of both links $u \to v$ and $w \to v$ have conflict. Both end nodes of assigned channel can use their local time and does not need any new synchronization or change of assignment. Then, in the routing phase, u will select one of safe $n(u)$ neighbors for the forwarding to make a one-hop advance. The selected successor node will take the place of the preceding node in the next round. This occurs continuously until the packet is delivered to d.

Figure 1 shows a sample of action of one node under such a network model. Figure 1 (a) shows the sequence of node actions in each slot in the local time. Figure 1 (b) shows the action of node u by using the synchronous beaconing slots. During the interval of each round, say round-j, node

s / d	source / destination
u	the current node of the routing from s to d
$L(u)$	location of node u, i.e., (x_u, y_u) in the 2-D plane
$N(u)$	neighbor set of u connected through directed links
$n(u)$	current successor set of u ($\subset N(u)$)
$Q_i(u)$	type-i forwarding zone ($1 \le i \le 8$)
$Z_i(u, d)$	type-i request zone ($1 \le i \le 8$) with respect to d
$S_i(u)$	safety status for $Q_i(u)$ ($1 \le i \le 8$)
$S(u)$	safety tuple of node u ($S_i(u) : 1 \le i \le 8$)
Γ / Γ_i	stuck nodes set / set of type-i stuck nodes
$\aleph$	an unsafe area
$\wp$	maximum length of the boundary circling an $\aleph$
λ_e	reachability of a directed/undirected link e

TABLE I
List of notions used.

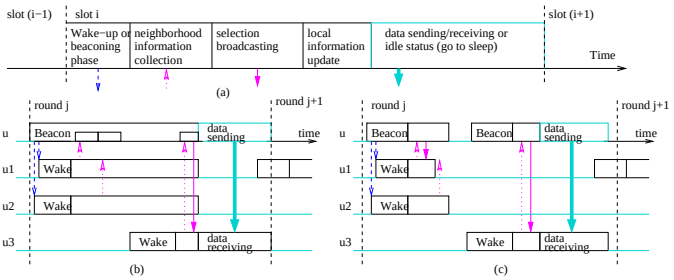

Fig. 1. (a) Action in active mode in each MAC slot. (b) Illustration of node action (of u), from neighborhood information construction to packet forwarding, with synchronous beaconing scheme. (c) Illustration of node action (of u) to synchronize its neighbors and to select the forwarding successor with asynchronous beaconing scheme.

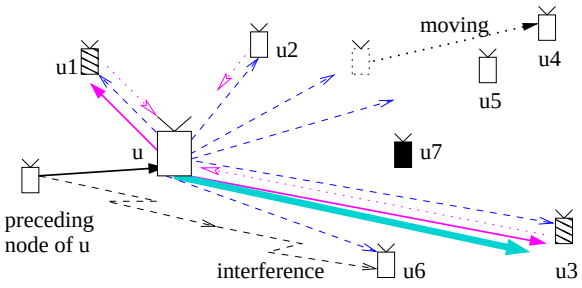

Fig. 2. Construction of $n(u)$ among all possible $N(u)$ neighbors.

u will keep sending beacon message to connect all its $N(u)$ neighbors. When a node u_1 (u_2, or u_3) wakes up, it will try to listen to its neighbors' beacon messages first and to respond to such a neighbor. The response message via broadcasting will be initiated at the reserved period so that the hand-shaking between u and this node can be accomplished individually without conflict. After node u selects the forwarding successor, it will send out its selection while all its neighbors are waiting. Due to miss-schedule or unsuccessful response message receiving (see node u_2 in Figure 1 (b)), node u may not be able to synchronize all its $N(u)$ neighbors in its $n(u)$ set in one round. Note that the successor of u is selected among $n(u)$. Any node can be idle and go to sleep when it does not receive the confirmation from u that notifies the selection of itself. Figure 1 (c) shows the action of node u by asynchronous beaconing slots. Whenever a node wakes up and receives the beacon message of node u, it will immediately respond to u. More likely, such a response will be affected by dynamic factors and cannot get through to u. When u receives such a response after its relatively shorter beaconing phase, u will selected the successor immediately (node u_1 with a successful response) or apply a certain time for a better neighbor candidate in its prediction (node u_2 in Figure 1 (c)). After that, the notification will be sent out to synchronize the select node for data transmission.

Note that when any node v fails to connect with u, u will not have up-to-date information of v. This will reduce the flexibility of the routing process in regards to selecting

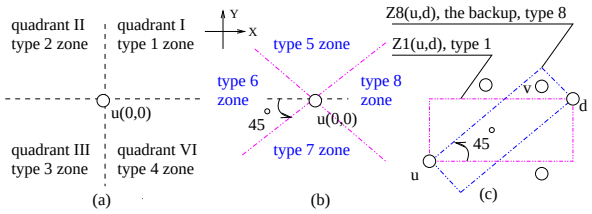

Fig. 3. (a) Forwarding zone $Q_1(u)$, $Q_2(u)$, $Q_3(u)$, and $Q_4(u)$. (b) Forwarding zone $Q_5(u)$, $Q_6(u)$, $Q_7(u)$, and $Q_8(u)$. (c) Definition of the request zone and its backup.

successors at u, but will not affect the correctness of the selection. Figure 2 shows a sample of selection of u among all its $N(u)$ neighbors ($\{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$). Due to the unstable nature of wireless signal, u_2's response is not received by u. Node u_4 moves out of the range of u from its original position so that it is also excluded from $n(u)$ set. Temporarily, the beacon message of u cannot be received by node u_5 due to the signal fading and by node u_7 due to the node failure. u_6 has already received signal from other nodes and does not need to respond to u to avoid signal conflict (i.e., interference). Eventually, node u will receive the acknowledgement from nodes u_1 and u_3 successfully (i.e., $n(u) = \{u_1, u_3\} \subset N(u)$). The forwarding successor will be selected between these two nodes.

Note that $n(u)$ is changeable. The ratio of the times that a node v appears in $n(u)$ to the total number of elapsed rounds can be measured by the Monte Carlo Method and determines the reachability

$$\lambda_{\{v,u\}} \approx \lambda_{u \to v} \times \lambda_{v \to u}, \quad \forall v \in n(u). \tag{2}$$

Table I summarizes all of the notions used in this paper. Assume that nodes are deployed on a 2-D plane. All the schemes are described in a round-based system, either synchronous or asynchronous. They can be extended easily to a more general system. However, to make our schemes clear, we do not pursue the relaxation. Every node can keep its status stable during each interval. Each packet is transmitted via a single channel and advances at a rate of one hop per round.

Forwarding under the realistic communication model. In [5], the selection of forwarding successor is limited within the request zone which has a simple regularity structure. The request zone is a rectangle in the corresponding quadrant (see Fig. 3 (a)) with both u and d at the opposite corners (see Fig. 3 (c)), as described in LAR scheme 1 in [7]. Such a scheme is also called limited forwarding routing, or simply LF routing. The request zones with respect to d in quadrants I, II, III, and IV are of types 1, 2, 3, and 4, denoted by $Z_i(u, d)$ ($1 \leq i \leq 4$). Respectively, each corresponding quadrant is called a type-i *forwarding zone*, denoted by $Q_i(u)$. A greedy advance with $Z_i(u, d)$ is called the type-i forwarding.

The above routing will have difficulty selecting the successor when the rectangular request zone at the source has extreme disparity between the width and the length. In this

paper, the forwarding is extended to increase its adaptivity with a backup request zone, simply called the *backup*. Denoted by $Z_i(u,d)$ $(5 \le i \le 8)$, each backup (see Fig. 3 (c)) is a rectangle where two opposing corners are u and d after self-rotating $Z_{i-4}(u,d)$ $45°$ in the counter-clockwise direction. Respectively, the corresponding forwarding zone is denoted by $Q_i(u)$ (see Fig. 3 (b)). The routing will be given a second chance to continue the forwarding (types 5-8) in the backups. Fig. 3 (c) shows a sample of node selection in $Z_8(u,d)$.

The discussion in [5] focuses on the networks where the sensing/communication range is a disk of uniform radius, simply called the uniform disk model. It is not suitable for the lossy link connection. Algo. 1 shows the details of zone-based routing under the realistic model of Eqs. (1) and (2). Each round, a successor is selected within the request zone or its backup, by the rectangle area with two opposing corners being the current and destination nodes. Note that a single forwarding may experience different types when the relative position of d to u changes and d is located in different types of request zones. The discussion in this paper focuses on type-1 forwarding and the corresponding information collection. The rest of the results can easily be derived by rotating the plane.

Algorithm 1 (LF routing): Determine the successor of node u (including node s) with respect to $n(u)$ [5].

1) If $d \in n(u)$, $v = d$.
2) Determine the request zone $Z_k(u,d)$ $(1 \le k \le 4)$ and its backup $Z_{k'}(u,d)$ $(5 \le k' \le 8)$, according to $L(u)$ and $L(d)$.
3) Select $v \in n(u) \cap Z_k(u,d)$; otherwise, $v \in n(u) \cap Z_{k'}(u,d)$.

IV. SR Model

Inspired by an earlier work on *safety levels* [5], in this paper, we describe the maximum probability that a packet can be successfully forwarded from a node u to the edge nodes of the networks with a type-i forwarding in the safety status $S_i(u)$ $\in [0:1]$ $(1 \le i \le 8)$. As shown in Fig. 4, the larger the value, the more likely the forwarding will be successful and the more reliable the path will be for communication. Such a value also implies a higher success rate of valid forwarding to any closer destination. In the following discussion, we will show the details of the labeling process by which each node u determines its statuses. The labeling process has three phases: one is applied during the network initialization of deployment, one is applied when any dysfunction of node and/or link occurs in networks, and the last one is applied when such a dysfunction is recovered (e.g., an occupied channel is released when its communication task is accomplished).

Initialization phase. We assume that all communication actions occur inside the *interest area*. The interest area is an inner part of the deployment area encircled by its edge which can easily be constructed by the hull algorithm. We assume the network is connected or connected at least once during the hull construction so that the interest area and those edge nodes can be determined. Any edge node has a fixed status and does not affect the labeling. In this phase, each node determines the initial value only, regardless of the unsafe/safe status.

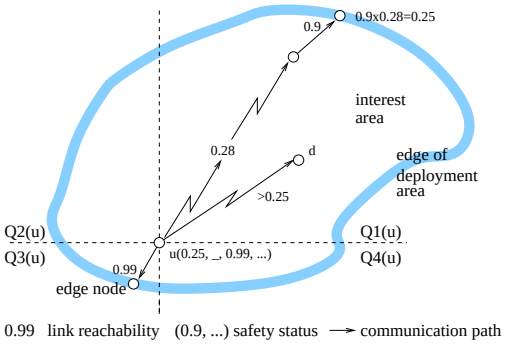

Fig. 4. Illustration of the definition of $S(u)$.

Each edge node outside the interest area sets its fixed safety status to $(1, 1, \cdots, 1)$. Each node u inside the interest area sets a changeable $(0, 0, \cdots, 0)$. After this, u will update $S_i(u)$ once with:

$$S_i(u) = \max\{\lambda_{\{u,v\}} \times S_i(v)\}, \quad 1 \le i \le 8 \tag{3}$$

where $v \in n(u) \cap Q_i(u)$, and the selected link $\{u,v\}$ is called the *key link* of u for $S_i(u)$. Then, $S_i(u)$ will stabilize by repeating:

$$S_i(u) = \max\{S'_i(u),\ \lambda_{\{u,v\}} \times S_i(v)\}, \quad 1 \le i \le 8 \tag{4}$$

where $S'_i(u)$ is the original value before the update of $S_i(u)$, and $v \in n(u) \cap Q_i(u)$. Note that $n(u)$ is changeable. Eq. (3) initiates the update. Eq. (4) will catch the maximum overall value. Starting from the edge nodes of the networks with a fixed status, the whole initialization phase converges.

A sample of the update of $S_1(u)$ is shown in Figs. 5 (a) and (b). At first, $n(u) = \{v_2, v_3\}$, and link $\{u, v_1\}$ is disconnected, although it has the highest probability of connection. In such a situation, link $\{u, v_3\}$ is selected as the key link (which is highlighted). Assume $S'_1(u) = 0$. We have $S_1(u) = S_1(v_3) * \lambda_{\{u,v_3\}} \simeq 0.46$ by using Eq. (3). When node v_1 appears in $n(u)$ (see Fig. 5 (b)), the link $\{u, v_1\}$ is selected as the key link. $S_1(u) = S_1(v_1) \times \lambda_{\{u,v_1\}} \simeq 0.5$ by using Eq. (4) and it is the final stable value with $N(u) = \{v_1, v_2, v_3\}$.

Identification phase. First, the stuck nodes where the local minimum can occur in the LF routing are identified as unsafe nodes. Specifically, a node u will be set as a type-i stuck node $(\in \Gamma_i)$ when there is no successor available in its type-i request zone $(n(u) \cap Q_i(u) = \phi,\ 1 \le i \le 8)$. Obviously, $S_i(u) = 0$. Under the realistic communication model, the packet is forwarded at a specifical time that is reserved with the MAC protocol. Due to the broadcasting nature of wireless communication, a node u can receive the signal from v and will cause the signal conflict when it is used as a successor of w at the same time. To avoid any hidden terminal or exposed terminal [13], in the update of $S_i(w)$, node u will be excluded from the $n(w)$ set when the quantum window of link $w \to u$ has conflict with that of link $v \to u$, which has been occupied by any exiting forwarding. This reservation can be easily implemented by the beacon messages that carry the

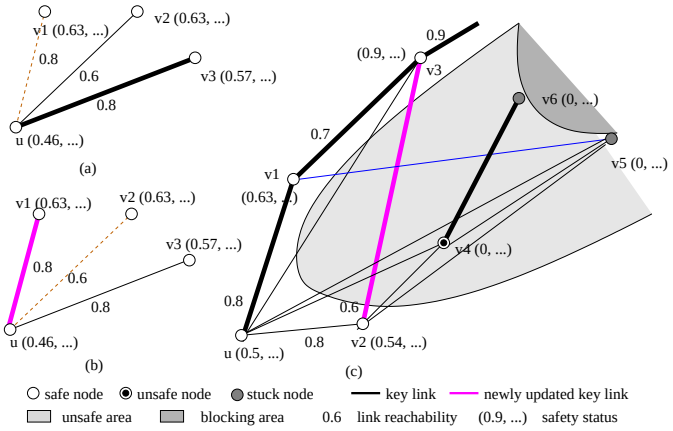

Fig. 5. Sample of safety information construction. (a) Calculation of $S_1(u)$ with Eq. (3). (b) Update of $S_1(u)$ with Eq. (4) due to a newly connected neighbor with higher quality. (c) A complex case of type-1 safety status.

information of the occupied quantum window. Note that our goal is to make a smart decision to avoid interference with redundant deployed resources, not to conduct a conflict-free channel assignment by the MAC protocol.

Second, we identify many nodes near these stuck nodes that should also be avoided in LF routing because their successors all are stuck nodes. A node u neighboring stuck nodes in its $Q_i(u)$ will re-calculate $S_i(u)$ by using Eq. (3). If u cannot find a $n(u)$ neighbor v such that $v \in Q_i(u)$ and $S_i(v) > 0$, we have $S_i(u) = 0$. u is identified as type-i *unsafe node*. The update of $S_i(u)$ will force a re-calculation of its $n(u)$ neighbors via their key links to u and contribute further changes in the next round. After all the unsafe nodes are identified, the rest of the nodes will have $S_i > 0$ and are identified as type-i *safe nodes*. The corresponding area that contains unsafe nodes is called an *unsafe area* (see Fig. 5 (c)). The above process will also initiate the updates in safe nodes because their most reliable forwarding path via the newly emerging area (with the highest probability described in the original status value) is blocked. If a safe node u has a new safety status $S_i(u) > 0$, it keeps the safe status but requires to obtain a stable value with Eq. (4). The above recalculation initiated by the status change of neighbor will continue until there is no node that needs status change in Eq. (3). Note that a type-i unsafe node could still be safe in other types. The setting of a unsafe node depends on whether a safe neighbor is always found among snapshots of dynamic connections of such a node, not on the existence of any single safe neighbor.

Definition 1: *Any node u is called a type-i stuck node ($\in \Gamma_i$) and set $S_i(u) = 0$ iff $n(u) \cap Q_i(u) = \phi$. $S_i(u)$ is the maximum probability of type-i forwarding from u to the nodes along the edge of interest area, respectively. "0" symbolizes an unsafe status; otherwise, it is safe. An unsafe node u is a node where $\exists 1 \leq i \leq 8$, $S_i(u) = 0$. Specifically, it is called type-i unsafe. Any node u is called a (type-i) safe node when $S_i(u) > 0$.*

In the example shown in Fig. 5 (c), when v_5 and v_6 are identified as stuck nodes, $S(v_5)$ and $S(v_6)$ are set to $(0, \cdots)$. When node v_4 receives the changes of $S_1(v_5)$ and $S_1(v_6)$, it will update $S_1(v_4)$ to 0 by using Eq. (3) and reach a stable (unsafe) status. Because of the update at v_4, v_2 will continue this process and update $S_1(v_2)$. Note that v_2 is still safe because $S_1(v_2) > 0$. Such an updating propagation for type-1 safety statuses will stop at node u because the other end of its key link $\{u, v_1\}$ does not have any change. The following analysis shows that our SR information is cost-effective to provide a straightforward path.

Theorem 1 (Convergence of the identification phase): *For a fixed configuration, the identification phase of the labeling process converges.*

Theorem 2 (Use of safety status): *A local minimum will occur if and only if any type-i unsafe node ($\in$ an unsafe area $\aleph$) is used in the type-i forwarding ($d \in Q_i(s)$ but $\notin \aleph$).*

Self-healing phase. When a new neighbor link occurs or the occupied channel of an existing link is released, the corresponding stuck node may change its status. In our approach, a stuck node will initiate the self-healing phase of the labeling process when it detects such a link change. The process applies Eq. (4) directly to reset the safety status of stuck nodes and relevant unsafe nodes. It is a reverse-process of the identification phase. Thus, its properties will still hold as the ones we proved in Theorems 1 and 2.

The details of all three phases are shown in Algo. 2.

Algorithm 2 (Labeling process).

1) **Initialization phase**. Each node u outside the interest area sets $S(u)$ to a fixed (1, 1, $\cdots$, 1) and each node v inside the area sets $S(u)$ to a changeable (0, 0, $\cdots$, 0). Then, each node will have stable status by applying Eqs. (3) and (4).
2) **Identification phase**. Any node u is called a type-i stuck node ($\in \Gamma_i$) and set $S_i(u) = 0$ iff $n(u) \cap Q_i(u) = \phi$. Upon detecting a change of the other end of the key link, a node u with $S_i(u) > 0$ recalculates its type-i status by using Eq. (3) and inform all of its neighbors in the next round. When the new value $S_i(u) = 0$, u is called a type-i unsafe node and no longer changes its status. Otherwise, u is still a type-i safe node and $S_k(u)$ will eventually stabilize by using Eq. (4).
3) **Self-healing phase**. Any node u (stuck, unsafe, or safe nodes) will recalculate $S_i(u)$ by using Eq. (4), until the value becomes stable.

V. SR FORWARDING (SF)

In this section, we first extend the LF routing under the SR model. Then, we analyze the effectiveness of SR information in helping achieve straightforward paths in routing scenarios.

Safety information based routing. In Theorem 2, we proved that using any unsafe node will cause the block of local minimum in LF routing. By selecting a safe successor only, such a safety-information-based routing can guarantee a successful straightforward path. Basically, for each current node u, a neighbor within its request zone $Z_k(u, d)$ that is safe to the destination (i.e., $S_{\mathbb{k}}(v) > 0$) is always preferred. Otherwise, the second chance will be given to seek v in the backup

$Z_{k'}(u,d)$ such that $S_{\mathbb{k}'}(v) > 0$. $\mathbb{k}$ and $\mathbb{k}'$ denotes the types of the request zone and the backup at that selected forwarding successor. Note that k and $\mathbb{k}$, and k' and $\mathbb{k}'$ are not necessarily the same. The details are shown in Algo. 3.

Algorithm 3 (SF routing): Determine the successor of node u (including node s) with respect to $n(u)$.

1) Apply steps 1) and 2) of Algo. 1.
2) Select $v \in n(u) \cap Z_k(u,d)$ (otherwise $n(u) \cap Z_{k'}(u,d)$), where the forwarding from v to d is safe with respect to request zone $Z_{\mathbb{k}}(v,d)$ and its backup $Z_{\mathbb{k}'}(v,d)$.

Scenario 1 (Safe forwarding). Regardless of the safety status of the source s, when a source that has a safe successor to initiate the SF routing, that safety status guarantees all the greedy advances along the routing path. When the destination d is not in any unsafe area, the forwarding will reach a node currently connecting with d and then deliver the packet to d in the same round. Thus, a straightforward path is achieved. Samples of this safe forwarding from s to d can be seen in Fig. 6 (a) and (b). We summarize the capability of the SF routing in finding a straightforward path in the following property. Note that all the properties can be derived from the above theorems. Detailed proofs are omitted due to the space limitation.

Property 1 (Straightforward path): *A straightforward path can be derived by SF routing from a safe node when the destination d can be in one type of safe area. Such a forwarding, say type-i, can be initiated at a source that has a safe successor, i.e., a type-i safe $n(u)$ neighbor in $Z_i(s,d)$.*

Scenario 2 (Intelligent routing decision). Many existing routings [3], [11] will start a perimeter routing phase when the forwarding is blocked. The perimeter routing routes the packet counter-clockwise along a face of the planar graph that represents the same connectivity as the original network by the "right-hand" rule, until it reaches a node that is closer to the destination than that stuck node. Due to the mutual impact of concurrent local minima, s and d can be disconnected. In such a case, the perimeter routing may experience too many unnecessary nodes before ending at a node that all its neighbors have been tried.

Whenever a node has the status $(0,0,\cdots,0)$, all its forwarding routings to the edge nodes are blocked. That is, the network is disconnected. When $S(s) = (0,0,\cdots,0)$, our routing will stop immediately. In a cautious way, we avoid any unnecessary trial of perimeter routing and wait for a more suitable configuration for data transmission. When the destination is in an unsafe area and disconnected with the source, the above safe forwarding will experience all four types of request zones or backups (see Fig. 6 (c)) and then stop. Among all $O(n)$ nodes in the neighborhood that may be tried by the perimeter routing, our routing only uses $O(\sqrt{n})$ perimeter nodes around that unsafe area. Due to the limited size of each unsafe area, our approach reduces the number of unnecessary trials before the routing fails. Overall, with the safety information, our routing can predict the failure ahead and avoid wasting time and channel resources.

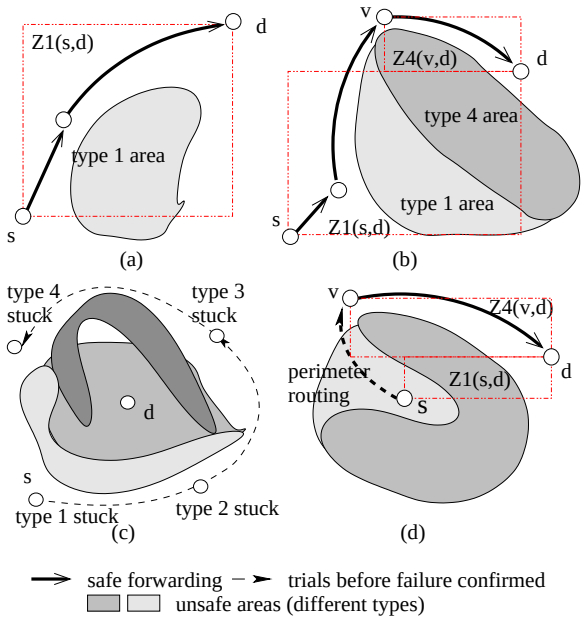

Fig. 6. Samples of SF routing.

Algorithm 4 (ESF, extension of SF with perimeter routing phase): Determine the successor of node u (including node s) with respect to $n(u)$.

1) Apply steps 1) and 2) of Algo. 3.
2) Select $v \in n(u)$ such that $\exists S_i(v) > 0$, until the forwarding from v to d is safe with respect to request zone $Z_{\mathbb{k}}(v,d)$ and its backup $Z_{\mathbb{k}'}(v,d)$.

Property 2 (Forwarding path validation): *The initiated SF routing may interrupt when the destination is in an unsafe area and disconnected with the source. Before the retransmission starts, the length of the path approximates to $D(s,d) + \wp$.*

Scenario 3 (Sub-optimal forwarding). For a node u that is contained in the unsafe area, if we have $S_i(u) > 0$, the routing from u can use the type-i forwarding to approach to the edge of this unsafe area and then leave away from such an area. For the routing cases other than the above two types (i.e., $S(u) \neq (0,\cdots) \wedge \exists S_i(u) = 0$), the SF routing is extended with a guided perimeter routing phase to reach an intermediate node so that the safe forwarding can continue (see Fig. 6 (d)). Due to the limited size of each unsafe area, the number of detours can be controlled as well as the length of the entire path (see the following property). The details of the extension can be seen in ESF routing in Algo. 4.

Property 3 (Scalable routing): *When s is inside an unsafe area, a successful routing will achieve a path shorter than $D(s,d) + \frac{\wp}{2}$.*

Scenario 4 (Reliable forwarding). Note that at each intermediate node, SF and ESF routings may have several options

Algorithm 5 (DSF routing): Determine the successor of node u (including node s) with respect to $n(u)$.

1) Same as step 1) of Algo. 3.
2) select $v \in Z_k(u,d) \cup Z_{k'}(u,d)$ where v has the highest forwarding probability to d indicated by $S(v) \times \lambda_{\{u,v\}}$.
3) Same as step 2) in Algo. 4, but always prefer to the use of key link(s).

that satisfy the safety requirement. This flexibility allows any existing routing scheme to apply in selecting the successor. To build a more reliable forwarding path, we modify ESF routing to select the most reliable link based on the information propagated along the key links. Such a routing concerns not only the existing configuration but also the history of a successful path for the forwarding. Therefore, the whole path can still be reliable even when many dynamic changes occur during the data communication. For each hop along the forwarding path, the selection is deterministic, so the routing is called "deterministic SR forwarding" (DSF). The details are shown in Algo. 5.

Note that DSF routing is just one selective case in Algo. 4 that is along a special path. Due to the directional construction of safety statuses, the value at an intermediate node will increase while the routing approaches d. The routing is under an optimistic model in searching the path for transmission. The success of DSF routing is obvious as Properties 1, 2, and 3 have been proved for ESF routing.

The above results rely on the stable safety statuses. When concurrent routings advance head-to-head, some safe nodes selected in routing may not satisfy the safety condition in Definition 1 after they become stable. That is, the information used in that routing selection is *inconsistent*. This is also the situation when our approach is applied to an asynchronous round-based system.

Definition 2: *Any node selected in the LF forwarding may not satisfy the safety condition in Definition 1 after it becomes stable. This outdated information used by the routing is called inconsistent information.*

Property 4 (Forwarding with inconsistent information): *If our forwarding advances can reach the destination d with consistent safety information, a path can also be constructed with inconsistent information.*

Property 4 ensures the success of our routing and its extensions in an asynchronous round-based system, as well as in the synchronous system. The following statement ensures that the self-healing phase will not affect any existing safety-information-based routing. Indeed, such a process heals more safe nodes and offers more options for routing.

Property 5: *The self-healing phase converges in a limited number of rounds and will not affect any existing safety-information-based routing.*

VI. Experimental Results

In this section, we study the performance of the SR model and the routing algorithms, using a custom simulator built in C#. The metrics used are the convergence rounds and the nodes involved in the information update (i.e., scalability of the information model), and the success rate of straightforward path construction (i.e., performance of the routing). The results are compared with those of GMS − the best localized solution for the local minimum problem known to date.

Simulation environment. In the simulations, 2,000 nodes are deployed uniformly to cover an interest area of 200m $\times$ 200m in the center. The link quality model of Eq. (1) is adopted. Each node uses 4-5 channels with $Maxslot = 10$ on each channel. $Maxslot$ is the number of slots in the MAC scheduling. Each round, we simulate the actions of each node u under both the SR model and the GMS model. The deployment holes are created randomly and 5% of the nodes are selected to move inside the interest area and change their neighboring links. This also simulates the cases when the nodes fail or are affected by the traffic. The labeling process of the SR model and the information collection process of the GMS model are applied. Note that we only collect 1-hop neighborhood information under the SR model in each round. For GMS greedy advance, the information is collected under different models. First, a node collects the information within 4-hops distance which is the minimum distance to prevent two head-to-head routings from accessing a pair of neighboring successors simultaneously, causing interference. Denoted by GMSM, this information model requires the least construction cost. Secondly, a node collects the information from all of the nodes in the interest area. Denoted by GMSI, this is an ideal model to retrieve global information.

Each node applies the Poisson distribution to determine whether it must initiate a communication to another node. We assume each communication has the same amount of data to send and elapses a long, fixed period in the networks. Thus, not only the number of communications created per round, but also the number of existing communication paths (i.e., service and waiting time in average) can be controlled. Then, our information-based routings ESF and DSF, and greedy forwarding under the GMSM and GMSI models will be applied. When a communication is accomplished, the occupied channels are released. Such information will be collected directly by nodes in both the GMSM and GMSI models while it incurring the self-healing process in our SR Model.

For each communication launched, the destination will be selected randomly. However, when the path is longer than 12 hops, due to the use of lossy link connections, GMS needs the information from the entire network in many cases. Therefore, to compare SR and GMS fairly, we only record the experimental results when each path is no longer than 12 hops long. We do not compare the DSF routing with others because it is a selective case in ESF only. For each case, 100 samples are tested.

Scalability of information construction. Fig. 7 shows the

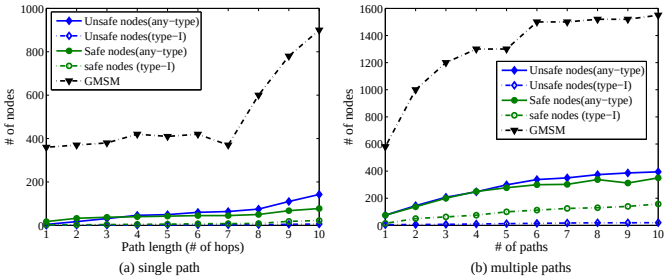

Fig. 7. Cost comparison of SR with GMSM: (a) single path and (b) concurrent paths.

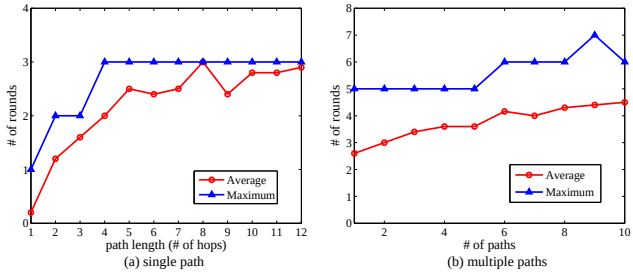

Fig. 8. Convergence speed of SR: (a) single path and (b) concurrent paths.

average number of nodes involved in the information update under both the SR model and the GMSM model. Note that each type of safety status has similar results. A node having any of its eight safety statuses labeled as unsafe is called an "any-type" unsafe node. We show the results of both type-1 and any-type statuses. Due to the use of the lossy link connection, the density of nodes is relatively high, thereby offering a greater chance of sharing the most reliable path on different forwarding paths. Therefore, few safe nodes need to update their statuses. Figs. 7 (a) and (b) show the cost incurred by a single path and concurrent paths, respectively. We only compare the result of our SR model with that of the GMSM model, which has a minimum cost required for a successful greedy advance, being aware of each intermediate node in the use. The results show that for a single path, the total cost of safe and unsafe nodes under the SR model is less than that of GMSM, in which the update has been controlled ideally to a minimum. For concurrent paths, the cost of the SR model is less than two times that of GMSM. Note that the SR information provides the accurate information on the mutual impact of local minima while the GMSM model cannot.

Fig. 8 shows the average number of rounds in convergence in the SR model. Although both the GMSM and GMSI models require fixed rounds, the SR model involves fewer total nodes. Fig. 8 (a) shows that the number of rounds in the SR model is reasonably low, compared with that under the GMSI model. When concurrent paths occur in the networks , the mutual impact of disabled nodes will incur unsafe areas to merge and create a bigger unsafe area. The converging speed is decreased, as shown in Fig. 8 (b). However, as we observed in the

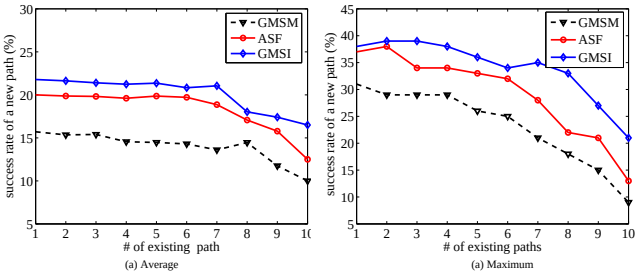

Fig. 9. Success rate of ESF routing, compared with GMS greedy forwarding: (a) average and (b) maximum.

experimental results, most unsafe nodes can determine their statuses within 4 rounds. The ESF routing can be applied immediately, although the inconsistent information may be used, causing a longer routing path.

Routing Performance. Fig. 9 shows the percentage of each routing under the SR, GMSI, or GMSM models in successfully achieving a straightforward path with other paths existing in the networks. Note that the local minima may disconnect the networks. With global information, 22% GMSI greedy advances will have a straightforward path. Among these successful cases of GMSI, the GMSM forwarding will fail when it happens to enter a large unsafe area where all the dead ends are 4-hops away from the entry point. The more concurrent paths there are, the more local minima and the more forwarding failures. In most of the cases where GMSI forwarding succeeds, a straightforward path can still be found under the SR model. Compared with GMS methods, our new approach is more cost-effective and practical.

VII. Conclusion

Our work provides a practical routing to improve the quality and performance of communication in WASN applications. A localized model, SR, is provided to describe the impact of interferences as other factors that incur local minima blocking the transmission in dynamic networks. The SR information provides a certainty of neighborhood topology under the opportunistic communication model, and can be used to achieve more straightforward paths in routings. In our future work, we will study the performance of our approach. The maximum throughput achieved in concurrent communications will be the focus. We will also conduct further study on more accurate information for unsafe areas so that shorter paths can be achieved.

References

[1] N. Ahmed, S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: A survey. *ACM Sigmobile Mobile Computing and Communication Review*, 9(2):4–18, 2005.

[2] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wirelss links: Modeling and implications on multi-hop routing. *Proc. of the 6th ACM MobiHoc*, pages 414–425, 2005.

[3] H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. *Proc. of the 12th ACM/IEEE MOBICOM*, pages 390–401, 2006.

[4] S. Gasagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. A distance routing effect algorithm for mobility (DREAM). *Proc. of the 4th ACM/IEEE MOBICOM*, pages 76–84, 1998.
[5] Z. Jiang, J. Ma, W. Lou, and J. Wu. An information model for geographic greedy forwarding in wireless ad-hoc sensor networks. *Proc. of the 27th IEEE INFOCOM*, pages 825–833, 2008.
[6] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. *Proc. of the 27th IEEE INFOCOM*, pages 1103–1111, 2008.
[7] Y. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Proc. of the 4th ACM/IEEE MOBICOM*, pages 66–75, 1998.
[8] K. Liu, N. Abu-Ghazaleh, and K. Kang. Location verification and trust management for resilient geographic routing. *Journal of Parallel and Distributed Computing*, 62(2):215–228, 2007.
[9] Y. Liu, Q. Zhang, and L. Ni. Opportunity-based topology control in wireless sensor networks. *Proc. of ICDCS*, 2008. CD-ROM.
[10] M. Marina and S. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. *Proc. of the 3rd ACM MobiHoc*, pages 12–23, 2002.
[11] M. Nesterenko and A. Vora. Void traversal for guaranteed delivery in geometric routing. *Proc. of the 2nd IEEE MASS*, 2005.
[12] S. Olariu and I. Stojmenovic. Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting. *Proc. of the 25th IEEE INFOCOM*, 2006.
[13] K. Tang, M. Correa, and M. Gerla. Isolation of wireless ad hoc medium access mechanisms under tcp. *Proc. of the ICCCN*, pages 77–82, 1999.
[14] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliabile multihop routing in snesor networks. *Proc. of ACM SenSys*, pages 14–27, 2003.
[15] S. Yessad, F. Nait-Abdesselam, T. Taleb, and B. Bensaou. R-MAC: Reservation medium access control protocol for wireless sensor networks. *Proc. of IEEE LCN*, pages 719–724, 2007.

Appendix

Proof of Theorem 1:

It is easy to prove that the status update by using Eq. (4) will converge when all of its $N(u)$ neighbors in the corresponding forwarding zone have been stable. Note that the process labeling each type of unsafe node is independent and will not have any sort of cross-impact on other nodes.

We can find a rectangle β with four corners (x_1, y_1), (x_2, y_1), (x_2, y_2), and (x_2, y_1) to exactly cover each unsafe area $\aleph$. Otherwise, for any unsafe node $\notin \beta$, we can always find a connected path $\notin \beta$ that consists of only unsafe nodes to reach a stuck node, due to the use of a rectangular forwarding zone. That is, a larger rectangle $\beta' > \beta$ is needed to cover ξ. Thus, the unsafe areas are limited as well as the number of unsafe nodes. When any node changes to unsafe, its status update ends and the need for such an final update relies on those stable unsafe statuses of neighbors. Therefore, the process will converge in a limited number of rounds inside unsafe areas.

Then, we prove that the status updates among safe nodes are limited. Assume that a is the average length of the edge of rectangle β. Assume a safe node u which needs to update $S(u)$ is a γ-distance away from $\aleph \subset \beta$. The most reliable path from u to edge nodes must use the segment that cannot be used in a forwarding from u through $\aleph$. Therefore, the probability of such a replacement relies on the ratio $(\frac{a}{\gamma})^2$; that is, $a \sim \gamma$. Therefore, γ is limited as well as a and only a limited number of nodes change the status value in the labeling process. That is, the process converges. ■

Proof of Theorem 2:

For any unsafe node u in $\aleph$, each of its successors in $Q_i(u)$ is in $\aleph \cup \Gamma$. For a forwarding path reaching d from accessing u, there must be a node v along this path whose successor is outside of $\aleph$. According to the labeling process for unsafe nodes, the nodes from v to u along the path will all be safe. This conflicts with the fact that u is unsafe. Therefore, the forwarding will have a signal collision at a node $\in \Gamma$.

Now we prove that using a type-i safe node u indicates the availability of at least one type-i interference-free forwarding from u. If any type-i forwarding is blocked at a dead end, say v, v will be type-i unsafe in the first round. In the labeling process, node u must also be labeled type-i unsafe. Therefore, the statement is proven. ■

Proof of Property 1:

Assume the routing starts from s because $s' \in n(u)$ and $S_i(s') > 0$. We will prove that when $S_i(d), S_{(i+2Mod4)}(d) > 0$, the routing path can be found in SF routing and no detour is needed. Assume $1 \leq i \leq 4$. The proof for the rest cases $4 \leq i \leq 8$ can be derived after self-rotating $45°$.

For any unsafe area not blocking the forwarding, the routing can select the safe successor to avoid entering its region. Assume that a type-i forwarding is blocked from s to d. From d, we can always find a path using type-i forwarding to reach a safe node v along the edge of network, where $S_i(v) = S_{(i+2Mod4)}(v) = 1 > 0$. In opposite direction from v, a type-$(i + 2Mod4)$ forwarding can be find to reach the type-$(i + 2Mod4)$ safe node d; that is, any node along the latter path will have $1 > S_{(j+2Mod4)} > 0$ and $1 > S_j > 0$, based on the Definition 1. It is obvious that a SF routing can be found for any pair of source and destination along the edge of networks. Thus, a SF routing path from s to d exists. If the SF routing towards to d is blocked by unsafe nodes, the continuous selection of safe successor may change the safety type and force the routing to route around. After trying all types, the routing will meet that type-$(i + 2)$ path to d, at a closer node to d than v along such a path. Therefore, the statement is proven. ■

Proof of Property 2:

SF routing will select safe successor only while approaching the destination. Each hop is a progressive, greedy advance, unless the distance in one dimension has been exhausted. The length of a SF forwarding path in one certain type approximates to $D(s, d)$. If the SF routing towards to d is blocked by unsafe nodes, the continuous selection of safe successor may change the safety type and force the routing to route around. After trying all four types of request zones or all four types of backup zones, the routing may interrupt if that safe path to d cannot be found. Assume that is the biggest block area and the length of its perimeter (boundary) is $\wp$. Routing around $\frac{3\wp}{4}$-distance far along the boundary will experience all four types of zones or its backups. Therefore, before it is interrupted, the length of experienced path approximates to $D(s, d) + \wp$. ■

Proof of Property 3:

Based on the proof of Property 2, the SF forwarding will experience approximate $D(s,d)$-distance far before its path to d is blocked. However, the safe forwarding successor can still be selected in other types. By routing around the unsafe area, if the routing can find that safe path to d, it experiences at most three of four types of request zones or the backups. Routing around $\frac{\wp}{2}$-distance far along the boundary will experience at least three types of zones or its backups. Therefore, the length of a success path to d approximates to $D(s,d) + \frac{\wp}{2}$. ■

Proof of Property 4:

The routing will be affected only when it enters an unsafe area whose nodes have not been updated to unsafe. Note that the routing will advance each hop per round. For each safe node that turns to unsafe later after the selection, it always have at least a safe neighbor, its preceding node, to retreat from the expanding unsafe area. Each backtracking is selected according to the current neighborhood information at that time, still following ESF or DSF protocols. Such a process will continue until a stable safe node is selected. After that, the routing can use consistent information and will find one of the possible paths to reach d. Note that once any inconsistent information is used, the routing may change the the routine and access different nodes. However, each of its segments built with consistent information is always one of the possible options in the routing after all information is up-to-date. ■

Proof of Property 5:

It is obvious that this self-healing phase is an opposite procedure of unsafe labeling process. Proven in Theorem 1, the convergence area of that labeling process is limited. Therefore, the region of status recovery is limited and the corresponding safety status adjustment with Eq. (4) can also be controlled within a limited area and in a limited number of rounds, no matter whether we use synchronous or asynchronous round-based system. Since the routing selects safe nodes, the recovery from unsafe to safe status will not affect existing routing path. ■