CR: Capability Information for Routing of Wireless Ad Hoc Networks in Real Environment

Zhen Jiang
Dept. of Comp. Science
West Chester University
West Chester, PA 19383

Zhigang Li, Nong Xiao
Dept. of Comp. Science
Natl. Univ. of Defense Tech.
Changsha, China

Jie Wu
Dept. of Comp. & Info. Sciences
Temple University
Philadelphia, PA 19122

1 Introduction

Wireless ad-hoc networks (WANs) have great long-term economic potential and the ability to transform our lives. Consider the WAN application of emergent disaster recovery. Before delivering food, water, and medicine, as well as doctors to the survivors, we need to know where and how many of these things are needed. The most efficient way is to send rescue teams carrying portable equipment, to search for the victims and survivors. The environment information will be collected through the wireless communication in order to estimate the amount of need at the base. In many cases, the surveillance reports cannot be sent directly to the base/sink and they require a multi-hop relay path. It is life-critical to send surveillance data without delay. The key issue is to avoid accessing a node called *stuck node* of the "local minimum phenomenon" [1] which causes detours and wastes time.

A detour-free multi-hop routing, which is also called *progressive routing*, requires each hop to advance to a closer successor to the destination. The progress routing not only avoids any unnecessary detour delay, but also allows more concurrent reporting processes in the networks when fewer nodes are involved in the transmission. Note that a progressive routing does not necessarily have the shortest path due to the redundant neighbors available in node selection. In real environment, the occurrence of detour can be caused not only by "deployment holes" such as sparse deployment and physical obstacles, but also by many dynamic factors in real environment, including node failures, signal fading, communication jams, power exhaustion, interference, and node mobility. In order to achieve reliability and scalability in dynamics, the path in progressive routing is built by the independent decision of each intermediate node that selects the successor from its 1-hop neighbors. This selection relies on accurate information to predict all the candidates in its succeeding paths and then to know whether all them are available. Such *capability information* can guarantee each hop to advance along a progressive routing path.

Our work provides each node this required capability information in a proactive manner with a structural regularity for all different paths passing through, saving the cost and delay of reconstituting the probing process in the reactive model (e.g., [9]). However, the neighborhood connections at each node are of irregular structure. A relay node will have different successor candidates, as well as their availability under the impact of local minima, every time when its relative location to the destination changes. Consider the availability status of node u_3 in Fig. 1. In the routing from s to d, using node u_1 will cause the routing to be blocked at node u_4, which is called a stuck node. In this case, not only the stuck node u_4, but also its nearby nodes u_1, u_3, and u_4 must be excluded from the access of the routing because their succeeding paths are blocked. However, when the routing from u_7 to u_4 is initiated, the access of u_3 must be allowed to keep the routing progressive. Those existing methods (e.g., [9, 11]) in the reactive model require to collect the information from the entire network in an on-demand manner to ensure the node capability. They face the problem of delay and cost in reconstituting the information for each newly initiated routing. Exiting proactive models (e.g., boundary model [6] and convex area model [2, 4, 5]) are not precise enough to catch such a dual role of node in each case. Even though many nodes become capable to successfully forward the packet in progressive routing, they will still be disabled from the consideration of routing decision as well as their communication ability.

The variations of link availability in real deployed environment bring new insights to local minimum and the corresponding capability information for routing. In such an environment each node has the opportunity to receive the signal directly from any node in the entire network, while each link can change its status by those dynamic factors, making the capability uncertain. In Fig. 1, when a "lossy link" [3] $u_4 - d$ happens to be available, the routing $s - u_1 - u_3 - u_4 - d$ is progressive by enabling u_1, u_3, and u_4. However, such a path may not be stable, causing the failure of data transmission. Even in existing routings that

978-0-7695-4134-1/10 $26.00

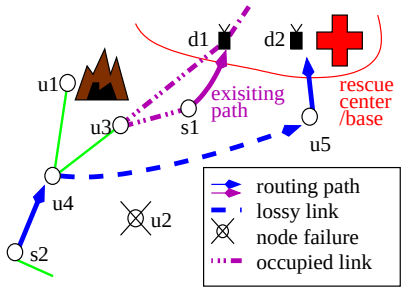

Figure 1. Multiple-hop unicasting with local minima caused by dynamics.

can guarantee the delivery the link quality in long term is ignored and a stable path for practical use cannot be achieved.

Our information model faces three new challenges of unstable link quality. First, how does each node attain information about its capability to a destination and then control the cost of its collection process? Due to the unstable link status, the capability information may not propagate to those affected nodes in time. The information will be collected by exchanging information among neighbors only, without using any global control. In order to complete the collection quickly, we need to control the scalability of information collection (i.e., within a limited area) even when many links are unstable. Second, how can the granularity of such a region be determined? As indicated in [8], the node availability in progressive routing is relative and will change as well as the relative locations of the source and destination. After introducing the use of lossy links, the neighborhood of a node can expand to be as large as the entire network, but each time it is unable to hold for very long. The above limited area must be relatively stable in calculation to avoid changing the node status too often and too quickly. Third, how does the designated information reflect the quality of a progressive routing? We need to study the effectiveness of a stable capability descriptor in helping to achieve a progressive routing via dynamic links. The proposed information-based routing must still be applicable even when many nodes have not updated their information.

We focus on an "everyone" model, in which each node will apply the same genetic process in a fully distributed manner. We first adopt the reservation MAC protocol (e.g., [13]) to confirm the available 1-hop neighbors. Second, for each neighbor candidate, we provide a simple safe-or-not answer to the existence of a progressive path in a region. The region size is a trade-off between precision of capability description and cost of information construction. We use $\lambda \in [0, 1]$ (or a product of λ for links along the path) to accommodate the quality of the link (or the path). It indicates the maximum probability of a successful non-detour path from this node to a stable node nearby, such as the sink or edge node of the networks. The larger value the successor has, the more likely the progressive routing will be successful and the more reliable the path will be. Like a lighthouse guiding boats to the harbor at night but unnecessarily illuminating everywhere. Third, such information guides our routing to approach its destination greedily in the predefined region with a higher success rate, in order to advance in a relatively reliable direction while staying along the path of a progressive routing. When dynamics occur after the network initialization phase, the updates of such information in the networks will converge quickly in a limited area. Our routing can make an alternative selection to avoid those newly emerged blocks. When some channels are recovered or released from their occupation, our information model will heal more safe nodes and offer more options for progressive routing. Strictly speaking, we provide a segmented progressive routing in dynamic situations that is guided by indirect referees. By applying this approach in a sample realistic communication model [12], we illustrate the effectiveness of our new information model in real deployed environment with both analysis and simulation.

Our contributions are threefold: First, the routing capability relies on the maximum of its neighbors, not on any single connection. It is relatively stable and its update process can be minimized. This is the first detour solution in the dynamic networks under the proactive model. It is based on our comprehensive study of local minimum impact and efficient routing information. Second, its construction is irrelevant to the positions of the source and destination in routing. It is implemented with the beaconing scheme in the MAC layer between 1-hop neighbors, which does not incur any extra message process and is not affected by any traffic jamming or other delay factors. Third, our capability information infers the local minima in a global view. It will be effective to guide the progressive routings to their destinations, even when the information is not up-to-date. The ignored reconstruction in reactive models, which has seriously inefficient problem, is considered here. Due to the space limitation, the detailed proofs of theorems and properties can be seen in [7] and are omitted.

2 Related Work

As indicated in [8], the node availability in progressive routing is relative when the source and destination changes their relative locations. Existing methods ignore such a fact and require the information to be reconstituted for each source and destination pair. Many of them (e.g., [2, 4, 5, 6]) lack the accuracy to describe those nodes whose succeeding progressive routings are all blocked by stuck nodes. They allow the routing to enter such an unsafe area even when the progressive path still exists, forcing the routing to take

unnecessary detours. The effectiveness of information and the delay of re-construction will make existing methods less applicable, in both proactive and reactive models.

By adopting the geographic greedy forwarding (GF) that is limited within the request zone in LAR scheme 1 in [10] (also called LF routing), a proactive model presented in [8] achieves a balanced point of tradeoff between the structure regularity of the capability of progressive routing and the routing flexibility. A boolean value stored at each node indicates whether such a node can safely be used in progressive LF routings. However, the calculation relies on a stable, ideal network topology where the link never changes its available status and only the deployment hole is considered under the well-known unit-disc-graphs (UDG) communication model. The flip-flop of a link status in any realistic model will affect the calculation of such statuses and make them unstable. The use of lossy links [3] increases the complexity of the forwarding at each node and makes those existing methods more difficult to precisely catch the diverse capability of a node in the description of topological evolution. A more accurate description of dynamic variation is required so that its construction can remain relatively stable and does not rely on any single neighbor connection.

GMS [9] provides a reactive solution by looking ahead for the node statuses within a distance of k-hops. It requires a probing process. GMS cannot achieve global optimization until k is set as the diameter of the networks. Under the realistic communication model, each node will have too many neighbors due to its possible connection to all the nodes in the entire network. Therefore, a more scalable, effective model in which the information construction can be controlled in a limited area, is required for a practical routing solution. Note that our goal is to achieve global optimization of the entire path, not just the reachability.

3 Realistic Communication Model

Communication model. We model a WAN as a directed graph $G = (V, E)$, where V is a set of vertices including all 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 $L(u)$. For a communication, assume node s is the source node, u is the current node, and d is the destination node. For each link $u \rightarrow v \in E$, $\lambda_{u\rightarrow v} \in [0, 1]$ indicates the probability that the signal from node u can be successfully received at node v, called *link reachability*. 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 [12] to determine each $\lambda_{u\rightarrow v}$ as follows, with respect to the distance of link (i.e., $D(u, v)$).

u, v	Nodes u and v
s, d	source and destination
x_u, y_u	coordinate of node u along X and Y dimension
$L(u)$	location of node u, i.e., (x_u, y_u) in a 2-D plane
$D(u, v)$	distance between u and v, i.e., $\mid L(u) - L(v) \mid$
$N(u)$	neighbor set of u connected by directed links
$n(u)$	current successor set of u ($\subset N(u)$)
$Q_i(u)$	type-i forwarding zone ($1 \leq i \leq 8$)
$Z_i(u, d)$	type-i request zone in $Q_i(u)$ with respect to d
$S_i(u)$	status for $Q_i(u)$
$S(u)$	info. tuple of node u ($S_i(u) : 1 \leq i \leq 8$)
Γ	stuck nodes set
Γ_i	set of type-i stuck nodes
$\aleph$	an unsafe area
H	maximum length of the boundary circling an $\aleph$
λ_e	reachability of a directed/undirected link e

Table 1. List of notions used.

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

Such a link model can easily be extended to other realistic models by using different calculation of $\lambda_{u\rightarrow v}$ in [11].

Collection of 1-hop neighborhood information with the reservation MAC. The reservation MAC protocol (e.g., [13]) confirm the reliable neighboring connections and can avoid the effect of node failure, signal fading, power exhaustion, and node mislocation. 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 information with its $n(u)$ neighbors and update its own 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. In such a case, node v will be excluded from the n set of its neighbors, say $n(w)$ set at any node w, when the quantum windows of both links $u \rightarrow v$ and $w \rightarrow v$ have conflict. Note that both end nodes of the assigned channel can use their local time and do not need any new synchronization or change of assignment. Then in the routing phase, u will select one of the safe $n(u)$ neighbors to make a one-hop progressive 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.

Note that when any node v fails to connect with u, u will not have up-to-date information for v. This will reduce the flexibility of the routing process in regards to selecting successors at u, but will not affect the correctness of the selection. 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 incoming link is for collecting guaranteed information. There may be cases when differences in transmission power give rise to unidirectional links. However, 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 delays or unnecessary re-transmissions. On the other hand, with our capability information, as we will show later, the routing can take advantage of any alternative path and avoid being stuck with unidirectional links. 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 a highly trusted reachability for coming data transmission

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

Because each node constantly applies a beaconing scheme to maintain the connection to its neighbors, the construction cost of our capability 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 beacon message while remaining efficient for the global optimization of the entire path.

Table 1 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. In a synchronous system, each round is the period a node needs to synchronize all its neighbors at least once. In an asynchronous system, each round is the sleep-wake cycle of a node. These schemes can be extended easily to a more general system. However, to make our schemes clear, we do not pursue 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.

Progressive routing under the realistic communication model. In [8], the selection of a 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. 2 (a)) with both u and d at opposing corners (see Fig. 2 (c)), as described in LAR scheme 1 in [10]. 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 $Z_i(u,d)$ $(1 \le i \le 4)$. Each corresponding quadrant is called a type-i *forwarding zone*, denoted by $Q_i(u)$. An advance within $Z_i(u,d)$ is called type-i forwarding.

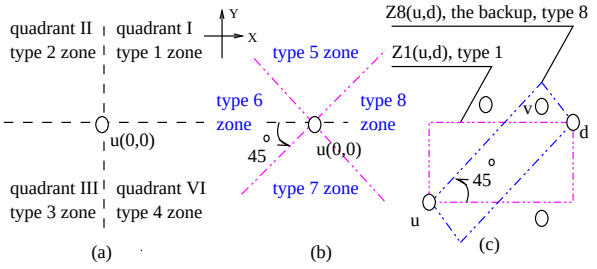

Figure 2. (a) $Q_1(u)$, $Q_2(u)$, $Q_3(u)$, and $Q_4(u)$. (b) $Q_5(u)$, $Q_6(u)$, $Q_7(u)$, and $Q_8(u)$. (c) Request zone and backup.

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 (e.g., $\mid x_u - x_d \mid >> \mid y_u - y_d \mid$). 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. 2 (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. The corresponding forwarding zone is denoted by $Q_i(u)$ (see Fig. 2 (b)). The routing will be given a second chance to continue the progressive forwarding (types 5-8) in the backups. Fig. 2 (c) shows a sample of node selection in $Z_8(u,d)$.

The discussion in [8] 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. Algorithm 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 routing may experience different types of forwarding 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 be derived easily by rotating the plane.

Algorithm 1 (LF routing, extended with backup zone and realistic communication model): Determine the successor of node u (including node s) with respect to $n(u)$ [8].

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, $v \in n(u) \cap Z_{k'}(u,d)$.

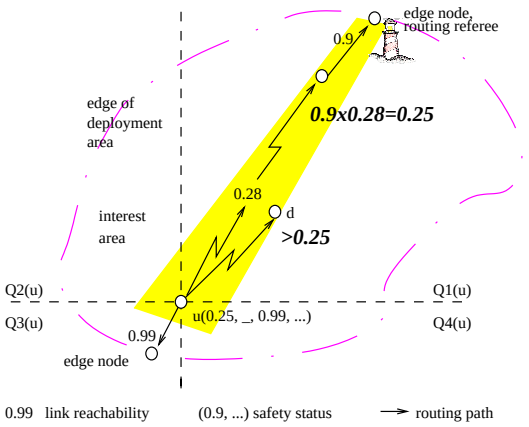

Figure 3. Illustration of the definition of $S(u)$.

4 Capability Information Model

Our capability information describes the maximum probability of a type-i progressive routing from a node u to the edge nodes of the networks in the status $S_i(u) \in [0:1]$ ($1 \leq i \leq 8$). The edge nodes can be determined easily with the hull algorithm. As shown in Fig. 3, the larger the value is, the more likely the progressive routing will be successful and the more reliable the path will be for communication. Such a value also implies a higher success rate of valid progressive routing 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 node and/or link dysfunction occurs in the 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). All three phases are implemented with the 1-hop information exchanges in the MAC layer and does not require any extra construction cost. These information processes supersede any transmission for data packets and will not be affected by problems such as traffic jamming. The details are shown later in Algorithm 2.

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 be constructed easily by the hull algorithm. We assume the network is fully 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 capability information.

Each edge node outside the interest area sets its fixed 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:

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

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:

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

where $v \in n(u) \cap Q_i(u)$ and $S'_i(u)$ is the original value before the update of $S_i(u)$. Note that $n(u)$ is changeable. Eq. (3) initiates the update. Eq. (4) will determine 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. 4 (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. 4 (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 ($n(u) \cap Q_i(u) = \phi$, $1 \leq i \leq 8$). Obviously, $S_i(u) = 0$. Due to the broadcasting nature of wireless communication, a node u can receive the signal from v and will cause a signal conflict when it is used as a successor of w at the same time. To avoid any hidden or exposed terminal in the update of $S_i(w)$, node u will be excluded from the $n(w)$ set when the quantum window of link $w \rightarrow u$ has conflict with that of link $v \rightarrow u$, which has been occupied by any existing routing. This reservation can be easily implemented by the beacon messages that carry the information of the occupied quantum window. Note that our goal is to make a smart decision to avoid interference and communication jamming with redundant deployed resources, not to conduct a conflict-free channel assignment in the MAC protocol. The latter one is difficult to achieve in dynamic networks. However, any improvement of channel assignment in MAC synchronization can help to reduce the signal collision and leave more neighbors available for the routing selection.

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).

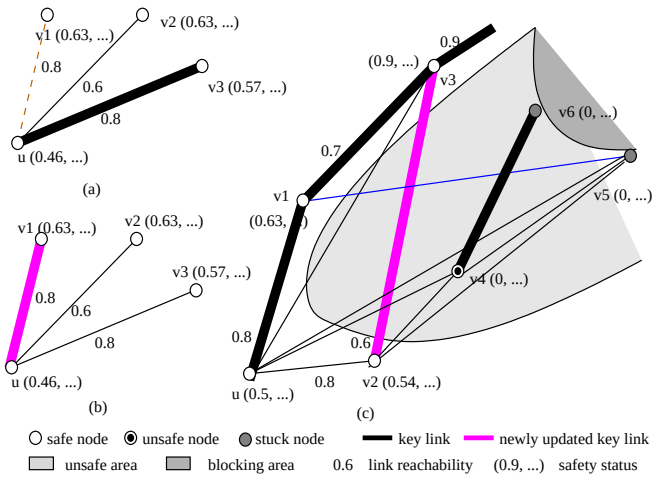

Figure 4. Information construction of $S_1(u)$. (a) $N(u) = \{v_2, v_3\}$. (b) $N(u) = \{v_1, v_2, v_3\}$. (c) A case with local minimum.

If u cannot find an $n(u)$ 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 recalculation 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. 4 (c)). The above process will also initiate the updates in safe nodes because their most reliable progressive routing 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 status $S_i(u) > 0$, it maintains its safe status, but needs to obtain a stable value with Eq. (4). The above recalculation initiated by the neighbors' status change will continue until there is no node that needs a status change in Eq. (3). Note that a type-i unsafe node could still be safe in other types. The setting of an 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 a type-i progressive routing 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. 4 (c), where 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 statuses will stop at node u because the other end of its key link $\{u, v_1\}$ does not change. For the sample network in Fig. 1, $S_1(u_1)$, $S_1(s_1)$, $S_1(d_1)$, and $S_1(u_2)$ will stabilize at 0. According to $\lambda_{\{u_3,u_4\}}$ of a lossy link, $S_1(u_3)$ is a very small value, but it is safe enough to take the progressive advance to u_4 when $u_4 \in n(u_3)$. The following analysis shows that our information is cost-effective.

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

Theorem 2 (Effectiveness of safety statuses): *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 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.

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 informs 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.

---

5 Capability Information based Routing

In this section, we first extend the LF routing under the capability information model. Then we, scenario by scenario, analyze the effectiveness of the information in helping to achieve the progressive 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, the routing can guarantee a successful progressive routing. Basically, for each current node u, a neighbor within its request zone $Z_k(u,d)$ that is safe with respect to the destination (i.e., $S_{\hat{k}}(v) > 0$) is always preferred. Otherwise, the progressive routing will still be available from a node v in the backup $Z_{k'}(u,d)$ so that $S_{\hat{k}'}(v) > 0$. $\hat{k}$ and $\hat{k}'$ denote the types of request zone and the backup at that selected successor, respectively. Note that k and $\hat{k}$, and k' and $\hat{k}'$ are not necessarily the same. These details are shown in Algorithm 3.

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

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

Scenario of safe forwarding. Regardless of the status of the source s, when s has a safe successor to initiate the CR routing, that status guarantees a progressive routing. 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 progressive routing is achieved. Samples of this safe forwarding from s to d can be seen in Figs. 5 (a) and (b). We summarize this capability of the CR routing in the following property.

Property 1: Capability of safe forwarding. *A progressive routing can be derived by a CR 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 of intelligent routing. Many existing routings will start a perimeter routing phase when the forwarding is blocked. The perimeter routing routes the packet counterclockwise 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 whose neighbors have all been tried.

Whenever a node has the status $(0,0,\cdots,0)$, all its progressive routings to the edge nodes are blocked. This means, the network is disconnected. When $S(s) = (0,0,\cdots,0)$, our routing will stop immediately. To be more intelligent, 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 becomes disconnected from the source, the above safe forwarding will experience all four types of request zones or backups (see Fig. 5 (c)) and then stop. We prove in the following property that 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. With the information collected, our routing can predict the failure ahead and avoid wasting time and channel resources.

Property 2: Ability to avoid unnecessary detours. *The initiated CR routing may interrupt when the destination is in an unsafe area and disconnected from the source. Before the retransmission starts, the length of the path approximates to $D(s,d) + H$.*

Scenario of scalable routing. For a node u contained in the unsafe area, if we find $1 \le i \le 8$ such that $S_i(u) > 0$, the routing from u can use the type-i forwarding to approach the boundary of this unsafe area and then leave away. For routing cases other than the above two scenarios (i.e., $S(u) \neq (0,\cdots) \wedge \exists S_i(u) = 0$), the CR routing is extended with a guided perimeter routing phase to reach an intermediate node so that safe forwarding can continue (see Fig. 5 (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 Algorithm 4.

Algorithm 4 (CR$^+$, extension of CR 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 Algorithm 3.
2. Select $v \in n(u)$ such that $\exists S_i(v) > 0$, until the progressive routing from v to d is safe with respect to request zone $Z_{\hat{k}}(v,d)$ and its backup $Z_{\hat{k}'}(v,d)$.

Property 3: Converging of guided perimeter routing, i.e., routing scalability. *When s is inside an unsafe area, a successful routing will achieve a path shorter than $D(s,d) + \frac{H}{2}$.*

Scenario of reliable routing. Note that at each intermediate node, CR and CR$^+$ routings may have several options to satisfy the necessity for safety. This flexibility allows any existing routing scheme to be able to select the successor. To build a more reliable progressive routing we modify the CR$^+$ routing to select the most reliable link based on the information propagated along the key links. This routing concerns not only the existing configuration, but also

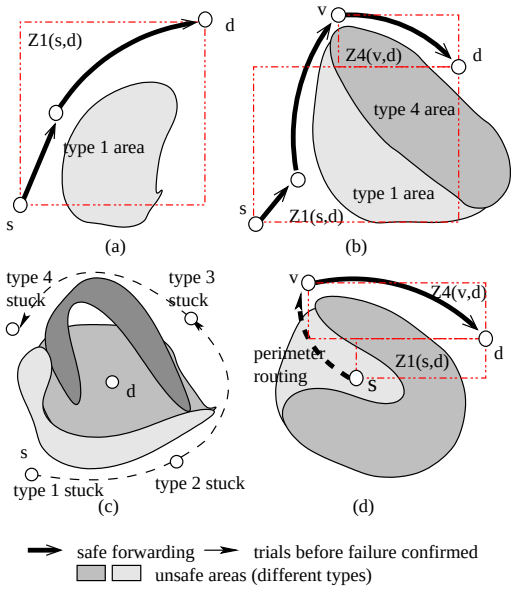

Figure 5. (a), (b), and (c) Samples of CR. (d) Sample of CR$^+$.

the history of a successful progressive routing. Therefore, the whole path can still be reliable even when many dynamic changes occur during the data communication. For each hop along the path, the selection is deterministic, so the routing is called "deterministic CR forwarding" (DCR). The details are shown in Algorithm 5.

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

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

Note that DCR routing is just one selective case along a special path in Algorithm 4. Due to the directional construction of statuses, the value at each node will increase as the routing approaches d. The routing is under an optimistic model for searching the path. Its success is obvious as the above three properties for CR and CR$^+$ have been proved.

Scenario of forwarding with inconsistent information. The above results rely on stable statuses. When concurrent routings advance head-to-head, some safe nodes selected in routing may not satisfy the safe 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, in which a certain fraction of information can be lost due to message delay.

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

In the following property , we prove the success of our routing when the information collection is deferred by distance, failure of neighbor status detection, or other factors. It also guarantees the success of such a routing when it is extended in an asynchronous round-based system.

Property 4: Robustness and effectiveness in dynamic networks. *If our progressive advances can reach the destination d with consistent information, a path can also be constructed with inconsistent information.*

Scenario of routing with information self-configuration. In the sample routing $s_2 - d_2$ in Fig. 1 after the communication $s_1 - d_1$ ends and the channel releases, d_1 will become a type-1 safe node. Then u_2 will be type-1 safe as well, making the path $s_2 - u_2 - d1$ available. Note that this update will not affect the the path $s_1 - u_3 - u_4 - d_2$.

The following statement proves that our information model has the ability of self-healing. Such a phase will not affect any existing capability-information-based routing. Indeed, it heals more safe nodes and offers more options for routing.

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

6 Simulation Results

In this section, we study the performance of the capability information 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 progressive routing (i.e., performance of the routing). The results are compared with those of GMS – the complete solution in the reactive model. Note that there is no existing proactive solution applicable to the realistic communication model because the flip-flop of link status will incur the oscillation in information collection and force the routing to trust 1-hop neighbors only. As a result, they are not better than the GMS model that collects 2-hop neighborhood information. By the results of GMS, we indirectly show that our safety model is more effective than any existing solution in the proactive model.

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 synchronized channels. Each round, we simulate the node action under both the CR and GMS models. The deployment holes are created randomly and 5% of the nodes are selected to move and change their neighboring links. This also simulates the cases in which nodes fail or are affected by traffic. In the labeling process of the information model, we only collect information from 1-hop neighbors at each round. For GMS advance, different information collection models are used. First, each node collects the information within a distance of 4-hops, which is the minimum distance to be able to prevent two head-to-head routings from accessing a pair of neighbors simultaneously, causing interference. Denoted by GMSM, this information model requires the lowest construction cost in the reactive manner. It is also a performance reference of existing information models in the proactive model because it achieves more accurate information and is more effective than any of them applied in such dynamic networks. Secondly, each node collects the information from all the other nodes in the networks. Denoted by GMSI, this is an ideal model to retrieve global information.

Each node applies the Poisson distribution to determine whether it must report to a nearby sink. We assume each communication has the same amount of data to send. They elapse a long, fixed period. Thus, not only the number of communications created per round, but also the number of existing paths (i.e., service and waiting time in average) can be controlled. Then we deploy enough sinks in the center of interest area so that each initiated communication has an available receiver. After that, our information-based routings CR$^+$ and DCR, as well as forwarding under the GMSM and GMSI models will be applied. When any communication is accomplished, the occupied channels are released. This information will be collected directly by nodes in both the GMSM and GMSI models while it is incurring the self-healing process in our CR Model.

When the path is longer than 12 hops, due to the use of lossy links, GMS needs information from the entire network. To compare CR and GMS fairly, we only record the results when each path is no longer than 12 hops. We do not compare the DCR routing with others because it is a selective case in CR$^+$. For each case, 100 samples are tested.

Scalability of information construction. Fig. 6 shows the average number of nodes involved in the information update under both the capability information model and the GMSM model. Note that each type of status has similar results. A node having any of its eight 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 node density is relatively high, thereby offering a greater chance of sharing reliable path(s) among different routings. Therefore, few safe nodes need to update their statuses. Figs. 6 (a) and (b) show the cost incurred by a single path and concurrent paths, respectively. We only compare the results of our information model with those of the GMSM model, which ideally knows all intermediate nodes and requires the minimum cost of information collection. The results show that for a single path, the total cost of the capability information model is less than that of GMSM, in which the update has been controlled ideally to a minimum. For concurrent paths, the cost of our new model is less than two times that of GMSM. Note that our information provides the accurate information on the mutual impact of local minima while the GMSM model cannot.

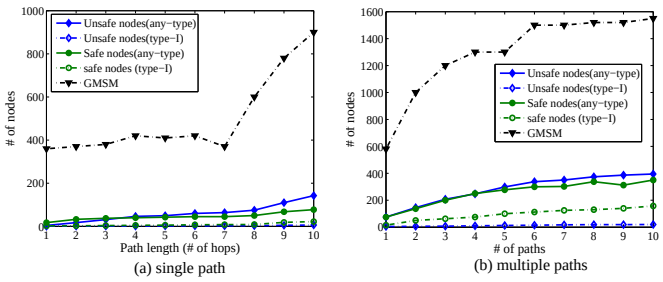

Figure 6. Cost comparison of CR with GMSM: (a) single path and (b) concurrent paths.

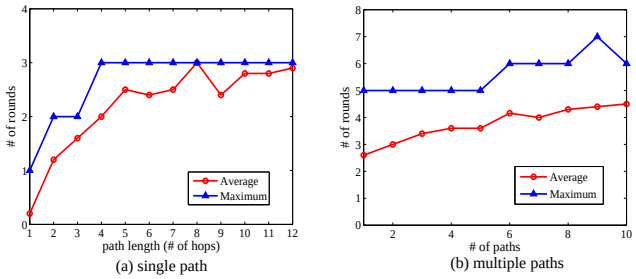

Figure 7. Convergence of CR construction: (a) single path and (b) concurrent paths.

Fig. 7 shows the average number of rounds of convergence in our information model. Although both the GMSM and GMSI models require fixed rounds, our information model involves fewer total nodes. Fig. 7 (a) shows that the number of rounds in our model is reasonably low, compared with those 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

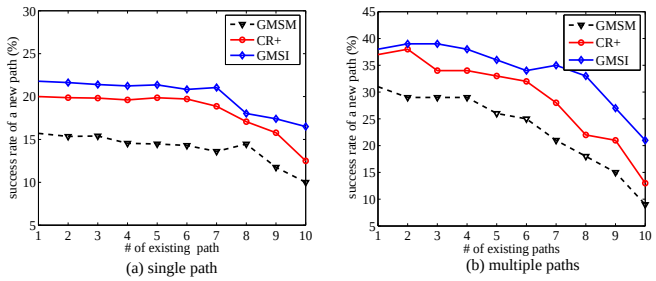

Figure 8. Success rate of CR$^+$ routing, compared with GMS forwarding: (a) average and (b) maximum.

unsafe area. The converging speed is decreased, as shown in Fig. 7 (b). As we observed in the results, most unsafe nodes can determine their statuses within 4 rounds. The CR$^+$ routing can be applied immediately, although the inconsistent information may be used, causing a longer routing path.

Routing Performance. Fig. 8 shows the percentage of each routing under the CR, GMSI, or GMSM models in successfully achieving a progressive routing with other paths existing in the networks. Note that the local minima may disconnect the networks. With global information, 22% GMSI advances will have a progressive routing. 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 forwarding failures are present. In most of the cases where GMSI forwarding succeeds, a progressive routing can still be found in CR$^+$. Compared with GMS methods, our new approach is more cost-effective and practical than the reactive information model. The comparison with GMSM also indicates that our approach is more effective than any existing information model in proactive model.

7 Conclusion

A localized capability information model is provided to describe the impact of local minima in dynamic networks. The information provides a certainty of neighborhood topology under the opportunistic communication model, while its construction cost is reduced to the minimum by the support of MAC protocols. Such information can be used to achieve more progressive routings. It is effective even when its collection process is deferred due to the distance or any incorrect neighbor status detection. In our future work, we will study the performance of our approach in traffic workload and provide more comprehensive results. The throughput achieved in concurrent communications will be the focus. We will also conduct further studies on more accurate information for unsafe areas so that shorter paths can be achieved.

Acknowledgment

This work was supported in part by NSF grants CNS 0626240, CCF 0840891, and CCF 0936942. Contact E-mail: zjiang@wcupa.edu.

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] N. Arad and Y. Shavitt. Minimizing recovery state in geographic ad-hoc routing. *Proc. of the 7th ACM MobiHoc*, 2006.

[3] 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.

[4] C. Chang, K. Shih, S. Lee, and S. Chang. RGP: Active routing guiding protocol for wireless sensor networks with obstacles. *Proc. of the 3rd IEEE MASS*, pages 367–376, 2006.

[5] S. Chen, G. Fan, and J. Cui. Avoid "void" in geographic routing for data aggregation in sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 1(4):168–178, 2006.

[6] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. *Mobile Networks and Applications IEEE INFOCOM*, 11(2):187–200, 2006.

[7] Z. Jiang, Z. Li, N. Xiao, and J. Wu. SR: A cross-layer routing in wireless ad hoc sensor networks. *Technique Report*, 2008. Document also available at http://www.cs.wcupa.edu/∼zjiang/safety2.pdf.

[8] 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.

[9] 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.

[10] 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.

[11] M. Lukic, B. Pavkovic, N. Mitton, and I. Stojmenovic. Greedy geographic routing algorithms in a real environment. *Proc. of the Fifth International Conference on Mobile Ad-Hoc and Sensor Networks*, 2009.

[12] 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.

[13] 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.