J. Parallel Distrib. Comput. 71 (2011) 1085–1097

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Capability information: A cost-effective information model for multi-hop routing of wireless ad hoc networks in the real environment

Zhen Jiang[a,*], Zhigang Li[b], Jie Wu[c], Nong Xiao[b]

[a] Department of Computer Science, West Chester University, West Chester, PA 19383, USA

[b] Computer School, National University of Defense Technology, Changsha, China

[c] Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122, USA

ARTICLE INFO

Article history:
Received 23 July 2010
Received in revised form
10 February 2011
Accepted 8 April 2011
Available online 22 April 2011

Keywords:
Distributed algorithms
Information model
Routing
Wireless ad hoc networks (WANs)

ABSTRACT

In greedy routing, each relay node forwards the message to a neighbor (also called the successor) that is closer to the destination. However, the successor candidate set (SCS) is different every time the relative location of the relay node to the destination changes. The configuration in the entire network, when all succeeding paths from a relay node are blocked by local minima, is irregular and its concern region cannot be determined unless the routing actually initiates. In the real deployment environment of the wireless ad hoc networks, the link quality also changes dynamically. This brings a challenge for the local decision of the greedy advance to precisely adjust its SCS for the flip-flop of link quality that blocks the non-detour path ahead. This paper introduces a new information model to a non-detour routing, also called progressive routing, under the impact of dynamic blocks. As a result, each 1-hop advance, by sacrificing little routing flexibility, can avoid those unsafe situations and remains on a non-detour path. In our model, each node prepares the information in a proactive mode, but can use it for all different paths passing through, saving the cost and delay in the reactive mode. We focus on an "everyone" model, in which each node will apply the same generic process in a fully distributed manner, in order to achieve a scalable and reliable solution. In detail, we discuss how in a sample realistic environment the pattern of SCS can be interpreted in a single safety descriptor $\in [0, 1]$ at each node. It indicates the maximum probability of a successful non-detour path from this node to the edge of networks. The larger value the more likely the non-detour routing will be successful and the more stable the path will be. We illustrate the effectiveness of this indirect reference information in the corresponding routing in terms of the success of non-detour path constitution and the ability of self-adjustment for dynamics in the networks, while the cost of information construction and update propagation is minimized. The results are compared with the best results known to date.

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, medicine, and 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 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 a *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 greedily advance [16] 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 the real environment, the occurrence of detours can be caused not only by "deployment holes" such as sparse deployment and physical obstacles, but also by many dynamic factors including node failures, signal fading, communication jams, power exhaustion, interference, and node mobility [1,21,26]. In order to achieve reliability and scalability

[*] Corresponding author.
E-mail address: zjiang@wcupa.edu (Z. Jiang).

0743-7315/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2011.04.002

Acronyms

CIM	Capability information model
CLF	Capability-information-based LF routing
CR	CLF routing extended by a perimeter routing phase with the capability information
DCR	CR routing using the most stable links
GF	Geographic greedy forwarding
GMS	Reactive information model in [13] for the local greedy advance with a consideration of interference, etc.
GMSI	GMS model that collects information from the entire network
GMSM	GMS model that collects neighborhood information within a distance of 4-hops only
LAR	Location aided routing [16]
LF	Limited greedy forwarding, a specific GF that is limited with the request zone in LAR scheme 1
MAC	Media-access-control
UGD	Unit-disk-graphs communication model
WAN	Wireless ad hoc network

Notation

u, v	Nodes u and v
s/d	Source/destination
x_u/y_u	Coordinate of node u along X/Y dimension
$L(u)$	Location of node u, i.e., (x_u, y_u) in the 2D plane
$D(u, v)$	Distance between nodes u and v, i.e., $\lvert L(u) - L(v) \rvert$
$N(u)$	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 ($1 \leq i \leq 8$)
$Z_i(u, d)$	Type-i request zone ($1 \leq i \leq 8$) with respect to d
$S_i(u)$	Status for $Q_i(u)$ ($1 \leq i \leq 8$)
$S(u)$	Information tuple of node u ($S_i(u) : 1 \leq i \leq 8$)
Γ/Γ_i	Stuck nodes set/set of type-i stuck nodes
$\aleph$	An unsafe area
H	Maximum length of the boundary circling an $\aleph$
$\lambda_{u \rightarrow v}$	Reachability of a directed link $u \rightarrow v$
$\lambda_{\{u,v\}}$	Reachability of a bi-directional link $\{u, v\}$
$\eta_i(u)$	The other end node of the key link of u for $S_i(u)$, where $S_i(u) = \lambda_{\{u,\eta_i(u)\}} \times S_i(\eta_i(u))$

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 relies on accurate information for an early decision to predict all the candidates in the succeeding paths and then to know whether all of them are available. Such *capability information* can guarantee each hop to advance along a progressive routing path.

Our work provides each node with this required 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 mode (e.g., [13]). However, the neighborhood connections in the wireless communications are of irregular structure [30] at each node. A relay node will have different successor candidates as well as their availability under the impact of local minima every time its relative location to the destination changes. Consider the availability of node u_3 in Fig. 1(a) under the impact of a hole area of local minimum. The transmission from u_4 to the destination d is blocked by the mountain territory, and u_2 does not have any neighbor closer to d in the deployment. u_2 and u_4 are stuck nodes. u_1 and u_3 must be excluded from the access of the routing because their succeeding paths of progressive routing will all be blocked by stuck nodes. However, when the routing is initiated at d instead of s (see Fig. 1(b)), u_1, u_2, u_3 cannot be reached due to the repulsive force along the boundary of that hole. By the exact same local minimum, the statuses of u_1, u_2, and u_3 will not affect the routing this time. Indeed, when another routing u_4–u_1 is initiated instead of s–d (see Fig. 1(c)), the access of u_3 must be allowed for the available path u_4–u_3–u_1. Those existing methods (e.g., [13,23]) in the reactive mode require the collection of 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. Existing proactive models (e.g., boundary model [8] and convex area model [2,5,6]) are not precise enough to catch such a change of node capability status (i.e., disabled or enabled) in each routing case. Even though many nodes become capable of successfully forwarding the packet in progressive routing, they will still be disabled from the consideration of routing decisions as well as their communication ability.

While 1-hop geographic greedy forwarding [16] (GF) in the progressive routing of the entire path has been studied extensively, the variations of link availability in a real deployment environment bring new insights to local minimum and the corresponding capability of progressive 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(d), s_2 wishes to send a report to d_2 while the s_1–d_1 transmission is in progress. Any routing encountering the transmission of s_1–d_1 in the gray zone will cause signal collisions and cannot successfully forward its data packet. u_4 is a stuck node and routing is forced to take a path 6 hops long. When a "lossy link" [4] u_4–d_2 happens to be available, the routing s_2–u_2–u_4–d_2 is progressive by enabling u_4 (Fig. 1(e)). On the other hand, when the transmission s_1–d_1 ends and its channels are released from occupation, the progressive routing will have another option s_2–u_1–s_1–u_8–d_2 by enabling u_1, s_1, and u_8 (Fig. 1(e)). However, the quality of a lossy link may not be stable, causing the failure of data transmission that uses such a path. Under the competition of concurrent GFs, a node can switch its availability too often and too quickly to interpret in any capability information that tries to predict a block in advance. This uncertainty of link quality is even ignored in existing routings (e.g., [3,9,26]) that try to guarantee the delivery. It incurs a great amount of overhead in those existing information models and makes them not applicable to the deployment in the real environment.

We face three new challenges of unstable link quality to achieve a practical information model for the progressive routing.

- How does each node attain information about its capability to reach 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.
- How can the granularity of such a region be determined? As indicated in [12], 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 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.

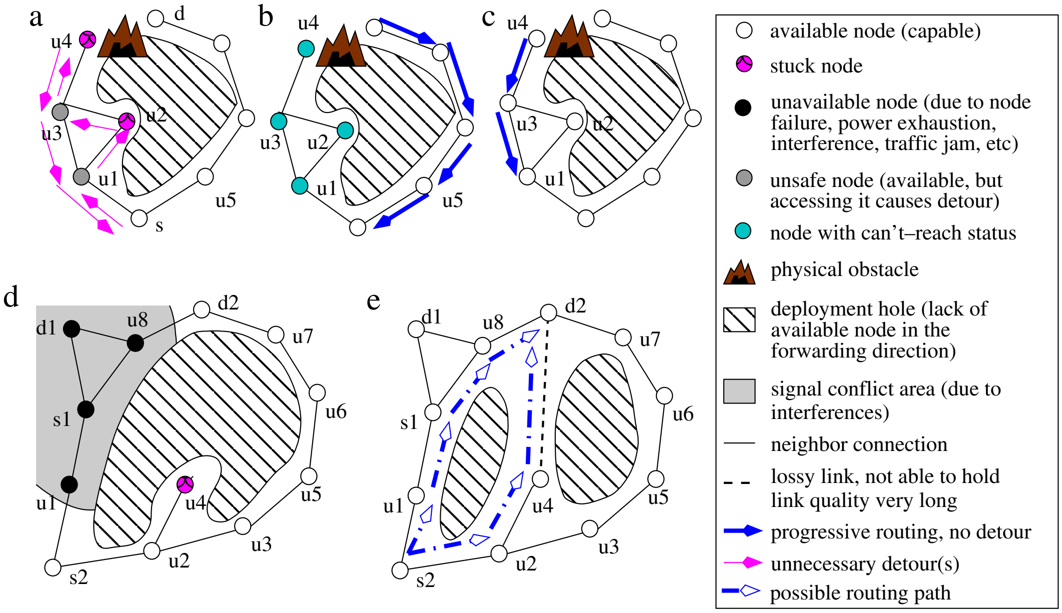

a
u4
d
u3
u2
u1
u5
s
b
u4
u3
u2
u1
u5
c
u4
u3
u2
u1
d
d1
u8
d2
u7
s1
u6
u5
u1
u4
u3
s2
u2
e
d1
u8
d2
u7
s1
u6
u1
u4
u5
u3
s2
u2
○ available node (capable)
stuck node
● unavailable node (due to node failure, power exhaustion, interference, traffic jam, etc)
unsafe node (available, but accessing it causes detour)
node with can't–reach status
physical obstacle
▨ deployment hole (lack of available node in the forwarding direction)
signal conflict area (due to interferences)
neighbor connection
lossy link, not able to hold link quality very long
progressive routing, no detour
unnecessary detour(s)
possible routing path

Fig. 1. Multiple-hop unicasting with local minima.

- How precisely does the designated information reflect the quality of a progressive routing? We need to study the effectiveness of a stable capability descriptor in the proactive mode in helping to achieve a progressive routing via dynamic links, while a complete detour solution does not exist currently in either a proactive or reactive manner. The proposed information-based routing must still be applicable even when many nodes have not updated their capability description.

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., [29]) 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 tradeoff between the precision of the 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 an edge node of the networks. As usual, edge nodes are always available to constantly provide a complete network coverage. The larger value the successor has, the more likely the progressive routing will be successful and the more stable the path will be. Third, this 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 more 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 [28], we illustrate the cost-effectiveness of our new information model in the real environment with both analysis and simulation.

Our contributions are threefold:

1. The proposed routing capability relies on the maximum of its neighbors, not on any single connection. It is relatively stable and its update can be minimized. This information is irrelevant to the positions of the source and destination in routing and can be constituted in the proactive mode, saving the cost and delay of reconstituting the information probing in the reactive mode.
2. Our capability information is a reference, indirectly inferring the local minima in a global view. It indicates a relatively better option for the local 1-hop decision to remain on a non-detour path. It will be effective in guiding the progressive routings to their destinations, even when the information is not up-to-date. The serious inefficiency that is incurred by the delay and cost in the ignored reconstruction in the reactive models can be avoided. This is the first detour solution under the proactive mode that is applicable to the dynamic networks. It is based on our comprehensive study of the impact of the local minima and the efficient routing information.
3. We achieve the balance point of the tradeoff between the simplicity of structure regularity and the precision of capability description. The proposed descriptor for the dynamic, unstructured networks can be normalized in a value $\in [0, 1]$, which can be carried in a beacon message in the MAC layer to 1-hop neighbors. In this way, the information exchanges and updates in our information model do not incur any extra message process and are also not affected by any traffic jamming or other delay factors.

The remainder of the paper is organized as follows: Section 2 discusses the existing issue in related work. Section 3 introduces some necessary notations and preliminaries. We provide details of the network model, the realistic communication in the sample environment, and the progressive routing that limits the forwarding within the request zone but achieves certain structural regularity of the successor candidate set. Section 4 presents our capacity description for the progressive routing. Its construction process is implemented in a distributed manner in the self-configuration of each node. The update of this information in a node can indicate its evolute from capable to incapable of being used in a progressive routing, and a versa process from incapable to capable. We prove it cost-effective in the same section. Section 5 provides our capacity-information-based routings, and the analysis of their properties. In Section 6, the simulation results are illustrated to prove the great reduction in construction cost and the performance improvement in our routing compared with the best results known to date. Section 7 concludes this paper and provides ideas for future research.

2. Related work

As indicated in [12], the node availability in progressive routing is relative when the source and destination change 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., [2,5,6,8]) 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 option for a progressive path still exists in other directions, forcing the routing to take unnecessary detours. The effectiveness of information and the delay of reconstruction make existing methods less applicable, in both proactive and reactive modes.

By adopting the GF that is limited within the request zone in LAR scheme 1 in [16] (also called LF routing), a proactive model presented in [12] 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 the deployment hole is considered under the well-known unit-disk-graphs (UDG) communication model only. The flip-flop of a link status in any realistic network model will affect the calculation of such statuses and make them unstable. The use of lossy links [4] 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. [11] used the most reliable path of the request zone as a reference to prevent the routing from entering the unsafe area. Its information construction does not rely on any single neighbor connection and can remain relatively stable. However, this proactive process cannot update node information for the links that are recovered from incapable statuses. A broadcast will be required to reset the initial status for each node. In the real environment, many links can be recovered (or reconnected) at the same time as others fail. The information update must have a complete lifecycle of status evolution and needs to consider both kinds of dynamic changes in the entire network.

GMS [13] 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, which can be easily achieved by multiple localized phases [20].

3. Realistic communication model

3.1. 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 indicates 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* in [22]. 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 [28] to determine each $\lambda_{u\rightarrow v}$ as follows, with respect to the distance of link (i.e., $D(u, v)$).

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

Such a link model can easily be extended to other realistic models (e.g., [17,23]) by using a different calculation of $\lambda_{u\rightarrow v}$.

3.2. Collection of 1-hop neighborhood information with the MAC reservation

The reservation MAC protocol (e.g., [29]) confirms the stable neighbor connections from node failure, signal fading, power exhaustion, and node mislocation. Each node u maintains its 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 currently connected by bi-directional links, denoted by $n(u)$, can be verified. Each node u will exchange information with its $n(u)$ neighbors and determine its own status with a normalized value $\in [0, 1]$ for all paths passing through (i.e., the proactive mode). According to the value, a node can tell whether it is disabled (a stuck node $\in \Gamma$), safe (>0), or unsafe. The information is simple enough to fit in a small beacon message while remaining efficient for the global optimization of a routing path. The information exchange can reuse the existing beacon process, without incurring any extra cost because each node needs to constantly apply the beaconing scheme to maintain the connection to its neighbors in the dynamic networks. On one hand, an urgent, short transmission [19] is usually adopted in the beaconing process. The beacon message as well as our neighborhood information is sent in a more timing-critical scheme, without any unnecessary delay. On the other hand, our information process adopts the optimistic mode. Only the neighborhood information that was received successfully will be used. In this way, the impact of failure or incompleteness can be mitigated to the minimum. Considering the interference [18] caused by any existing transmission from a node u, the reception node v will gain the knowledge of such a channel assignment with the MAC protocol. 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 links $u \rightarrow v$ and $w \rightarrow v$ have any conflict. Both end nodes of the assigned channel can use their local time and do not need any new synchronization or change of existing assignment. 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

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

Then in the routing phase, the current node u will select one of the safe $n(u)$ neighbors to make a 1-hop progressive advance. The selected successor node will take the place in the next round to continue the routing process, 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 selection. It is not necessary to

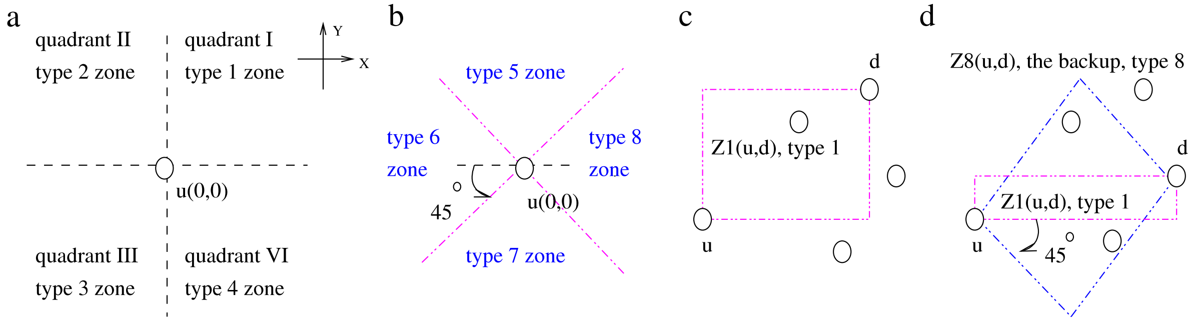


Fig. 2. (a) $Q_1(u)$, $Q_2(u)$, $Q_3(u)$, and $Q_4(u)$. (b) $Q_5(u)$, $Q_6(u)$, $Q_7(u)$, and $Q_8(u)$. (c) Request zone. (d) Backup zone.

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, as indicated in [24], 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 retransmissions. 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.

Assume that nodes are deployed on a 2D 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 easily be extended 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 1-hop per round.

3.3. Progressive routing under the realistic communication model

In [12], the selection of a forwarding successor is limited within the request zone, which has a simple regularity structure of the successor candidate set. The request zone is a rectangle in the corresponding quadrant (see Fig. 2(a)) with u and d at opposing corners (see Fig. 2(c)), as described in LAR scheme 1 in [16]. 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 $Z_i(u, d)$ $(1 \leq i \leq 4)$. 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.

As shown in Fig. 2(d), the above routing will have difficulty to select the successor when the rectangular request zone at the source has extreme disparity between the width and the length (e.g., $|x_u - x_d| \gg |y_u - y_d| \to 0$). In this paper, the forwarding is extended to increase its adaptivity with a backup request zone, simply called the *backup*. Denoted by $Z_i(u, d)$ $(5 \leq i \leq 8)$, each backup (see Fig. 2(d)) 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(d) shows a sample of node selection in $Z_8(u, d)$ after the routing fails to find any available neighbor in $Z_1(u, d)$.

The discussion in [12] 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. 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 $n(u)$ [12].

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

4. Capability information model (CIM)

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 are a sequence of neighboring nodes that connecting them will form a convex hull [7] to contain all the deployed nodes. As shown in Fig. 3, the larger the value is (of node u), the more likely the progressive routing will be successful and the more stable the path will be for communication. Such a value also implies a higher success rate of valid progressive routing to any closer destination (for instance, d in Fig. 3). 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 malfunction occurs in the networks, and the last one is applied when such a malfunction 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 by reusing the beaconing process of 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.

4.1. 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 in [7]. We assume the network is fully connected or connected at least once during the hull construction so that the

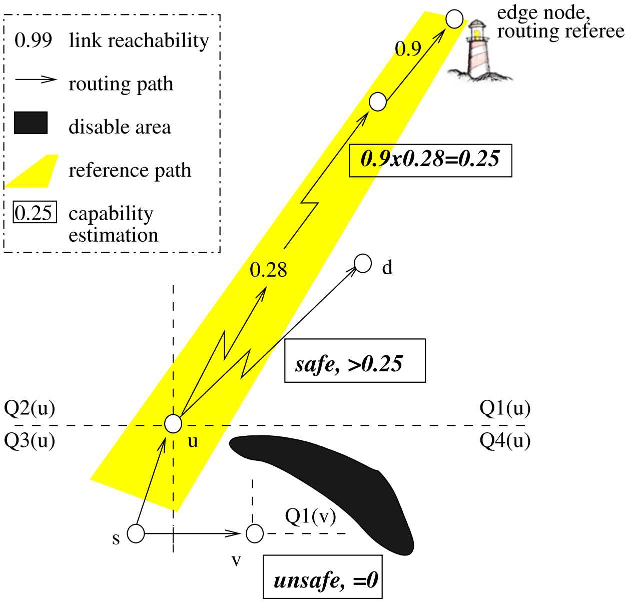


Fig. 3. Illustration of the use of the relative reference in 1-hop advance.

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 of the routing.

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

$$S_i(u) = \max\{\lambda_{\{u,v\}} \times S_i(v)\}, \quad v \in n(u) \cap Q_i(u) \wedge 1 \le i \le 8. \tag{3}$$

The selected node v is denoted by $\eta_i(u)$, and the corresponding link $\{u, v\}$ (i.e., $\{u, \eta_i(u)\}$) 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 $v \in n(u) \cap Q_i(u)$ 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 Fig. 4(a) and (b). At first, in Fig. 4(a), $n(u) = \{v_2, v_3\}$. Link $\{u, v_1\}$ is disconnected, although it has the highest probability of connection. In such a situation, $v_3 = \eta_1(u)$ and 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)), $v_1 = \eta_1(u)$ and the link $\{u, v_1\}$ becomes 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 $N(u) = \{v_1, v_2, v_3\}$.

4.2. 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, $S_i(u) = 0$. 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 [27] 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 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 in the channel assignments in the MAC layer can help to reduce the signal collision and make 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 recalculate $S_i(u)$ by using Eq. (3). 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 with Eq. (3) 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 containing 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 passing through this newly emerged 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 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 the interest area, respectively. "0" symbolizes an unsafe status; otherwise, it is safe. An unsafe node u is a node where $\exists 1 \le i \le 8, S_i(u) = 0$. 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 $(0, \ldots)$. 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 $\eta_1(u) = v_1$ and $S_1(v_1)$ does not change.

For the sample networks in Fig. 1(a), (b), and (c), $S_1(u_2)$ and $S_1(u_4)$ will be set at 0 for the type-1 forwarding from s to d. Their values will trigger the updates $S_1(u_1) = S_1(u_3) = 0$. The value $S_1(u_1) = 0$ indicates that any type-1 forwarding from u_1 is unsafe and may be blocked at a stuck node. This will help routing at s to select the right successor u_5 in the progressive routing. The routing u_4–s_1 in Fig. 1(c) is a type-4 forwarding. If s is outside of any type-4 unsafe area, $S_4(s)$ will stabilize at a positive value $\in (0, 1)$. The further updates of u_1 and u_3 will converge at $1 > S_4(s) > S_4(u_1) > S_4(u_3) > 0$, which guarantees a possible progressive routing via u_3 at u_4.

In the sample network in Fig. 1(d) and (e), during the data transmission of s_1–d_1, $S_i(s_1)$ and $S_i(d_1)$ ($1 \le i \le 8$) will change to 0 because they do not have the spectrum window to accept the signal from any other resource. $S_1(u_1)$ will change to 0 because it does not allow any additional forwarding to interfere the existing communication in its Q_1. By the same reason, $S_3(u_8) = 0$ and

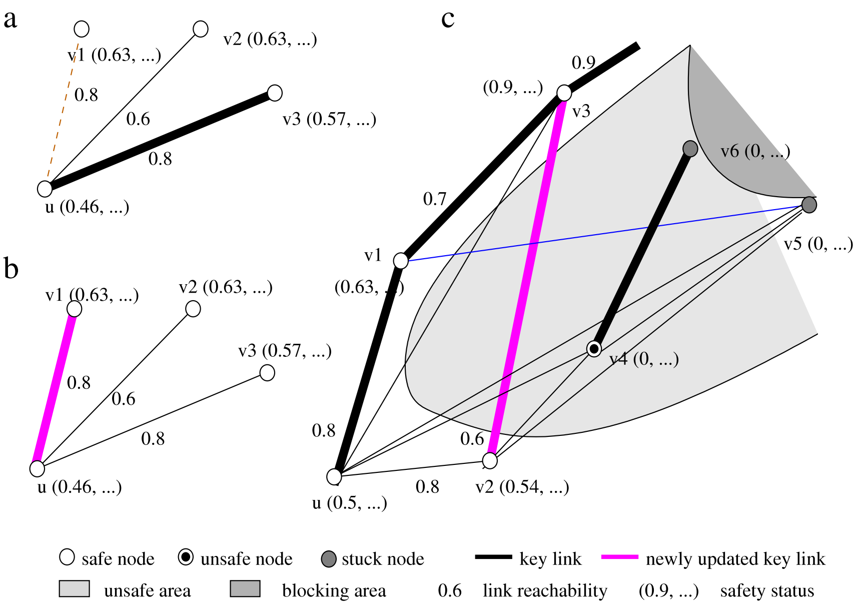


Fig. 4. Information construction of $S_1(u)$. (a) $n(u) = \{v_2, v_3\}$, (b) $n(u) = \{v_1, v_2, v_3\}$, and (c) a complicate case for $n(u) = \{v_1, v_2, v_3, v_4, v_5\}$.

this blocks any type-3 forwarding from entering u_8. According to $\lambda_{\{u_4,d_2\}}$ of a lossy link, $S_1(u_4)$ is a very small value, but it is safe enough to take the progressive advance to d_2 when $d_2 \in n(u_4)$. Note that before the transmission s_2–d_2 occupies the channels of d_2, $S_1(u_8) > 0$ because its update relies on the value of $S_1(d_2)$, not $S_1(s_1)$. The ability of communication of u_8–d_2 still remains at that time. Our information is accurate without unnecessarily sacrificing any concurrent communication. The following analysis proves 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.*

Proof. It is easy to prove that the status update that occurs by using Eq. (4) will converge when all of its $N(u)$ neighbors in the corresponding forwarding zone have been stabilized. 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 path $\notin \beta$ to a stuck node that consists of only unsafe nodes, 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 update relies on those stable, unsafe statuses of its 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 boundary 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 stable path from u to the edge nodes must use the segment that cannot be used in a progressive routing 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 can change the status value in the labeling process, meaning the 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 $\notin \aleph$).*

Proof. For any unsafe node u in $\aleph$, each of its successors in $Q_i(u)$ is in $\aleph \cup \Gamma$. For a progressive routing that reaches 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 be blocked 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. □

4.3. Self-healing phase

When a new 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 the 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 $(1, 1, \ldots, 1)$ and each node v inside the area sets $S(u)$ to a changeable $(0, 0, \ldots, 0)$. Then each node will have a 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 (i.e., $S_i(\eta_i(u))$), 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, scenario by scenario, we analyze the effectiveness of the information in helping to achieve the progressive routing. We also provide some unique properties of the capability-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, 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 (CLF—Capability-information-based LF routing): 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. When s has a safe successor to initiate the CLF 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 Fig. 5(a) and (b).

Property 1 (*Capability of Safe Forwarding*)**.** *A progressive routing can be derived by a CLF 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 $Z_i(s, d)$.*

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

For any unsafe area not blocking the forwarding at s, the routing can select the safe successor to avoid entering this dangerous region that contains local minima. Assume that a type-i forwarding is blocked from s to d. From s, the routing can always find a path of type-i forwarding to reach a type-i safe node v along the edge of network.

Before reaching v, the routing can always find a neighbor candidate v' that $1 \geq S_j(v') = S_{j+2Mod4}(v') > 0$. This is because of the existence of v and its connected edge neighbors, which keep all eight statuses safe.

Then, safe forwarding is conducted to approach d until it is blocked by the last local minimum around d due to the safety definition. The continuous selection of the safe successor may change the safety type and force the routing to route around such a local minimum. After trying all types of forwarding, the routing will meet the type-k or -$(k+2)$ safe path to d. Therefore, the statement is proven. □

Scenario of intelligent routing. Many existing routings [9,15,25] will start a perimeter routing phase when the forwarding is blocked. The perimeter routing sends 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 whose neighbors have all been tried.

Whenever a node has the status $(0, 0, \ldots, 0)$, all its progressive routings to the edge nodes are blocked. This means, the network is disconnected. When $S(s) = (0, 0, \ldots, 0)$, 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 CLF 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 $D(s, d) + H$.*

Proof. CLF routing will select a 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 CLF forwarding path in one certain type approximates to $D(s, d)$. If the routing toward d is blocked by unsafe nodes, the continuous selection of the 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 be interrupted if that safe path to d cannot be found. Assume that is the biggest block area and the length of its perimeter (boundary) is H. Routing around $\frac{3H}{4}$-distance far along the boundary will experience all four types of zones or its backups. Therefore, before it is interrupted, the length of the experienced path approximates to $D(s, d) + H$. □

Algorithm 4 (CR—extension of CLF 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 CLF routing in 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 $Z_{\hat{k}'}(v, d)$.
```

Scenario of scalable routing. For a node u contained in the unsafe area, if we find $1 \leq i \leq 8$ 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. For routing cases other than the above two scenarios (i.e., $S(u) \neq (0, \ldots) \wedge \exists S_i(u) = 0$), the CLF 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 CR can be seen in Algorithm 4.

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* $D(s, d) + \frac{H}{2}$.

Proof. Based on the proof of Property 2, the CLF forwarding will experience approximately $D(s, d)$-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 backups. Routing around $\frac{H}{2}$-distance far along the boundary will experience at least three types of zones or backups. Therefore, the length of a success path to d approximates to $D(s, d) + \frac{H}{2}$. □

Scenario of reliable routing. Note that at each intermediate node, CLF 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 stable successor candidate $v \in n(u)$ with respect of $\lambda_{\{u,v\}} \times S(v)$. This routing concerns not only the existing configuration, but also 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.

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 mode for searching the path. Its success is obvious as the above three properties for CLF and CR have been proved. The following property proves that CLF, CR, and DCR are progressing and can avoid any "livelock". The routing decisions in different relay nodes do not have the problem of "disagreement", "bad gadget", etc. [10,14]. Our routing processes are always progressing and use the channel in an efficient way. This property distinguishes our approach from those models using a Bellman–Ford-algorithm-like information collection.

Property 4 (*Efficient Use of Channel Resource*)**.** *CLF, its extension with perimeter routing CR, and the selective case DCR are livelock-free.*

Proof. In CLF, CR, and DCR, the safe forwarding phase will be conducted when a safe node is selected in the relay. After that, the routing will approach the destination greedily. For any unsafe relay node u, the routings will avoid accessing the corresponding request zone $Z(u, d)$. Such a node u will no longer appear in any successor candidate set. Due to the support of the reservation MAC protocol, any node that is concurrently used in other communication paths and has a channel conflict with the current forwarding cannot appear in its successor candidate set. In this way, those routing paths will not form a loop of channel requests in local advances. They are livelock-free. □

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.
```

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 the 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 what happens 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 5 (*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.*

Proof. The routing will be affected only when it enters an unsafe area where nodes have not been updated to unsafe. Note that the routing will advance each hop per round. For each safe node that becomes unsafe after the selection, it always has at least one 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 CR or DCR protocols. Such a process will continue until a stable safe node is selected. After that, the routing can use consistent information and then finds one of the possible paths that will reach d. Note that once any inconsistent information is used, the routing may change 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. □

Scenario of routing with information self-configuration. In the sample routing s_2–d_2 in Fig. 1(e) after the communication s_1–d_1 ends and the channel releases, s_1 and d_1 will become type-1 safe nodes. Then u_1 and u_8 will be type-1 safe as well, making the path s_2–u_1–u_8–d_2 available. Note that the update will not affect the path s_2–u_2–u_3–u_5–u_6–u_7–d_2 (or s_2–u_2–u_4–d_2) if it has been adopted.

The following property states that our information model has the ability of self-healing. By integrating the healing phase and the identification phase, we complete a lifecycle of information updates at nodes in dynamic networks. This makes our information sufficient and necessary to indicate the capability of progressive routing. We will prove as follows that such a phase will not affect any existing capability-information-based routing (i.e., sufficiency of the healing process for routing). Indeed, it heals more safe nodes and offers more options for routing (i.e., necessity of the healing process for routing).

Property 6 (*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.*

Proof. It is obvious that this self-healing phase is an opposite procedure of the 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 a synchronous or asynchronous round-based system. Since the routing selects safe nodes, the recovery from unsafe to safe status will not affect the existing routing path. □

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

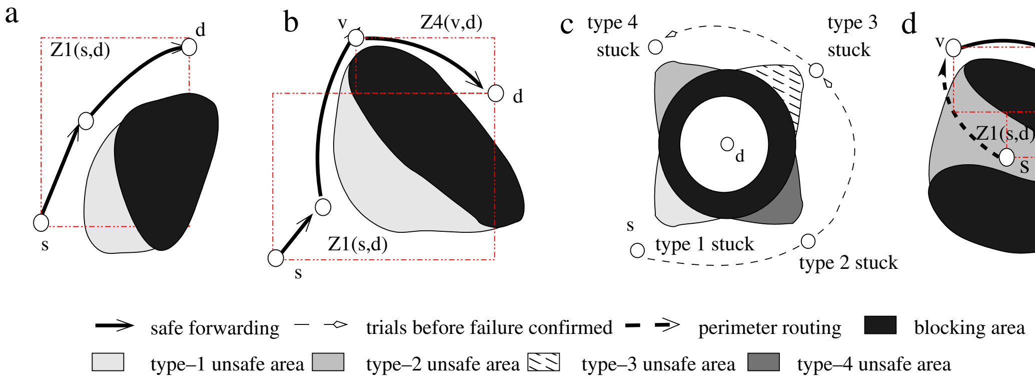


Fig. 5. (a), (b), and (c) Samples of CLF. (d) Sample of CR (CLF + perimeter routing).

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 mode. Note that there is no existing proactive solution applicable to the realistic communication model because the flip-flop of link statuses will incur the oscillation in information collection and force the routing to trust 1-hop neighbors only. As a result, they (e.g., [31]) are not better than the GMS model that collects 2-hop neighborhood information. By the results of GMS, we indirectly show that our capability information model is more effective than any existing solution in the proactive mode.

6.1. *Simulation environment*

In the simulations, 2000 nodes are deployed uniformly to cover an interest area of 200 m × 200 m in the center. The link quality model of Eq. (1) is adopted. Each node uses 4–5 synchronized channels. The deployment holes are created randomly and 5% of the nodes are selected to move and change their neighboring links. This simulates

- not only the cases in which the intermediate nodes fail or are affected by traffic,
- but also those in which succeeding nodes/paths newly emerge from incapable statuses.

In a round-based system, we simulate the node action under both the CIM and GMS models. Each node applies the Poisson distribution to determine whether it must report to a nearby sink/destination. 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 the interest area so that each initiated communication has a receiver available, but not necessarily reachable due to the dynamic blocks.

Each routing is conducted under different information models. In our CIM routing, the required information for each 1-hop advance has been prepared in a proactive way by constantly receiving beacon message from connected neighbors. Under the reactive GMS model, the routing path can be constituted at the source by a probing process to collect the global information. Due to the dynamic change of link statuses, such a probing is still needed at each intermediate node along the path. We adopt two different information collection modes. 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 mode because it achieves more accurate information and is more effective than any of them applied in our dynamic networks. Secondly, each node collects the information from all other nodes in the networks. Denoted by GMSI, this is an ideal mode for retrieving global information.

After that, our information-based routings CR and DCR, as well as forwarding under the GMSM and GMSI models will be applied. We tested the performance of these routings in terms of the success of non-detour path constitution and the corresponding overhead costs in terms of the scalability and the speed of information construction. Two different cases are considered. First, we tested

- the overhead cost required for a given routing when it is the only task in the entire network.

Second, we tested the concurrent (multiple routing) tasks in the networks and their mutual impact on the progressive routing decision, which includes:

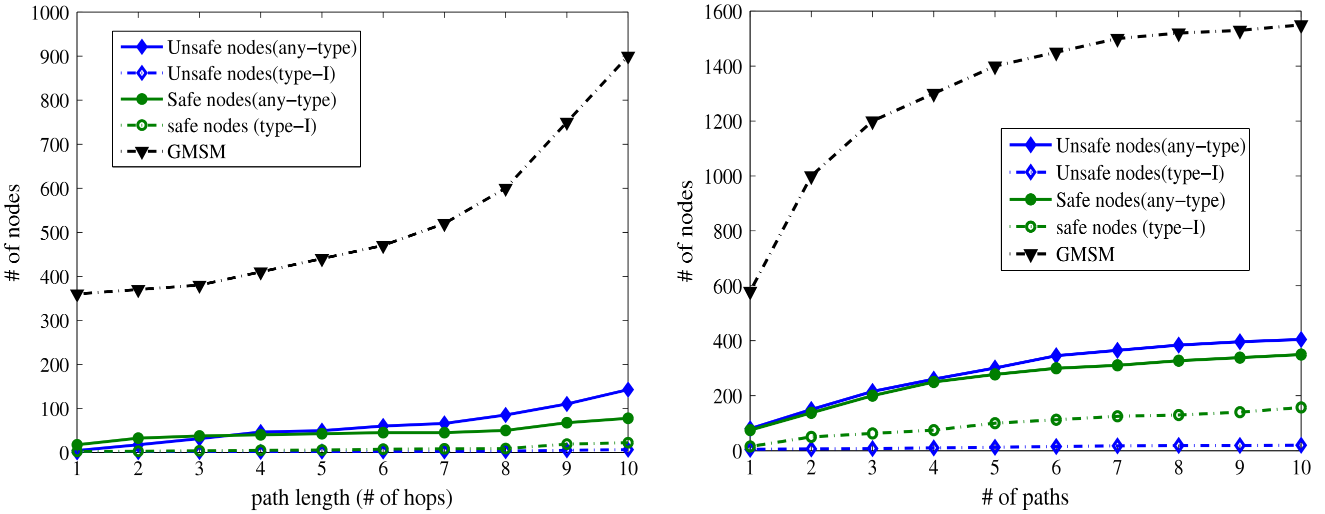
- the channel allocation and occupation,
- the interference,
- the communication accomplishment and its channel release,
- the multi-role of a single descriptor in different tasks.

This information will be collected directly by nodes in both the GMSM and GMSI models while it is incurring the label process and the self-healing process in our CIM.

When the path is longer than 10 hops, due to the use of lossy links, GMS needs information from the entire network. To compare CIM and GMS fairly, we only record the results when each path is no longer than 10 hops. We do not show the performance of DCR routing with others because it is a selective case in CR. For each case, 100 samples are tested.

6.2. *Scalability and speed of information construction*

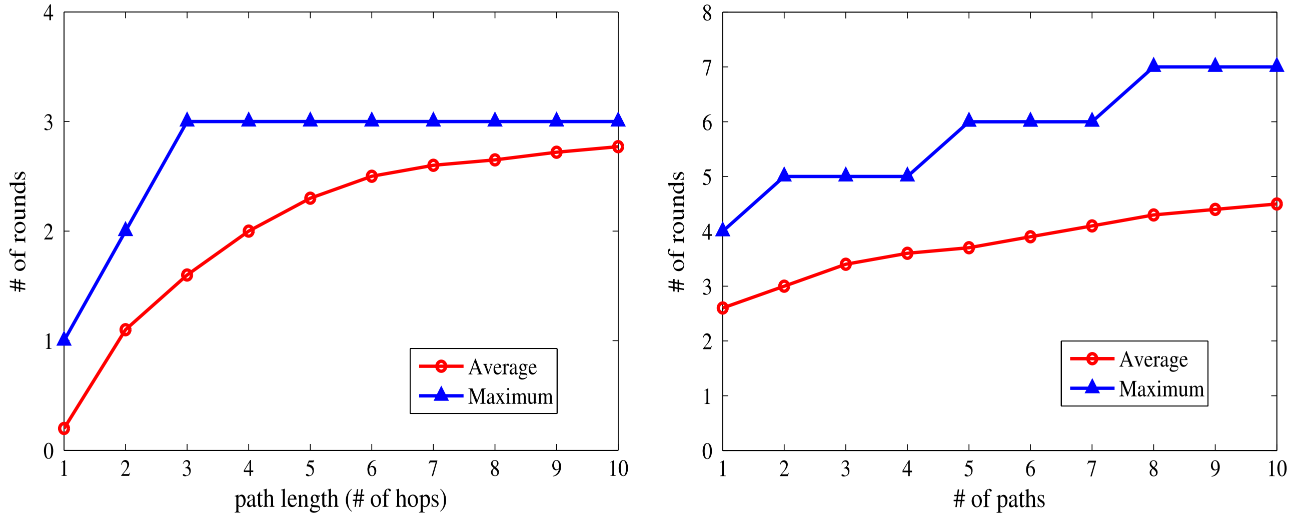
Fig. 6 shows the average number of nodes involved in the information update under both the capability information model and the GMSM model. Note that each type of status has similar results. A node having any of its eight statuses labeled as unsafe is called an "any-type" unsafe node. We show the results of both type-1 and any-type statuses. Due to the use of the lossy link connection, the node connections are relatively dense, thereby offering a greater chance to share the most reliable path segment among different routings. Therefore, few safe nodes need to update their statuses. Fig. 6(a) and (b) show the cost incurred by a single path and concurrent paths, respectively. We only compare the



(a) Single path (maximum cost per task).

(b) Concurrent paths (affected by interference, channel occupation, and channel release).

Fig. 6. Cost comparison of CIM with GMSM.



(a) Single path.

(b) Concurrent paths (affected by interference, channel occupation, and channel release).

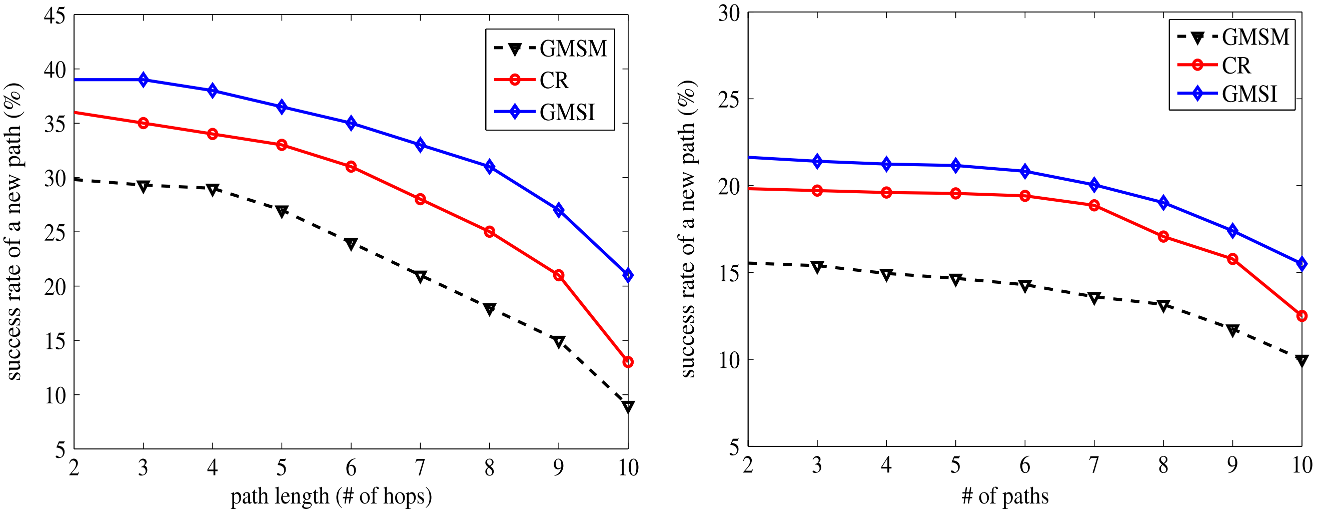
Fig. 7. Convergence of construction in CIM.

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 approach provides accurate information on the mutual impact of local minima while the GMSM model cannot.

Fig. 7 shows the average number of rounds of convergence in our information model. Note that GMSM requires fixed 4 rounds and GMSI requires the information to be collected along the network diameter. CIM utilizes fewer nodes and the number of rounds is reasonably low, compared to those under the GMSI and GMSM models. For a single path that is constituted without interference, CIM is the fastest. 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 of CIM is slower, as shown in Fig. 7(b). But the result is very close to GMSM and still in an acceptable range. As we observed in our results, most unsafe nodes can determine their statuses within 4 rounds. The CR routing can be applied immediately as the GMSM forwarding is initiated, although some inconsistent information may be used, causing a longer routing path. This addition can be ignored as we next proved by the performance improvement of CIM in the cases with concurrent paths.

6.3. *Routing performance*

Fig. 8 shows the percentage of each routing under the CIM, 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, many GMSI advances will have a progressive routing. Among these successful cases of GMSI, the GMSM forwarding will



(a) Single path (affected by holes and mobility).

(b) Concurrent paths (affected by interference, channel occupation, and channel release).

Fig. 8. Success rate of CR routing compared with GMS forwarding.

also sometimes 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 implies that our approach is more effective than any existing information model in the proactive mode.

7. Conclusion

A localized information model CIM 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. This approach is effective even when the information collection is asynchronous or is deferred due to the distance or any incorrect detection of the neighbor availability. 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 even shorter paths can be achieved.

Acknowledgments

We would like to thank Ms. Devan Parker for her assistance in the paperwork preparation. This work was supported in part by NSF grants CNS 0626240, CCF 0840891, and CCF 0936942.

References

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

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

[3] P. Bose, P. Morin, I. Stojmenovic, Routing with guaranteed delivery in ad hoc wireless networks, in: Proc. of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, pp. 48–55.

[4] A. Cerpa, J. Wong, M. Potkonjak, D. Estrin, Temporal properties of low power wireless links: modeling and implications on multi-hop routing, in: Proc. of the 6th ACM MobiHoc, 2005, pp. 414–425.

[5] C. Chang, K. Shih, S. Lee, S. Chang, RGP: active routing guiding protocol for wireless sensor networks with obstacles, in: Proc. of the 3rd IEEE MASS, 2006, pp. 367–376.

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

[7] Definition, Convex hull. http://en.wikipedia.org/wiki/Convex_hull.

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

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

[10] T. Griffin, F. Shepherd, G. Wilfong, The stable paths problem and interdomain routing, IEEE/ACM Transactions on Networking 10 (2) (2002) 232–243.

[11] Z. Jiang, Z. Li, N. Xiao, J. Wu, CR: Capability information for routing of wireless ad hoc networks in the real environment, in: Proc. of the IEEE International Workshop on Networking, Architecture and Storages, NAS'10, 2010, pp. 155–164.

[12] Z. Jiang, J. Ma, W. Lou, J. Wu, An information model for geographic greedy forwarding in wireless ad-hoc sensor networks, in: Proc. of the 27th IEEE INFOCOM, 2008, pp. 825–833.

[13] C. Joo, X. Lin, N. Shroff, Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks, in: Proc. of the 27th IEEE INFOCOM, 2008, pp. 1103–1111.

[14] I. Kalaydjieva, The border gateway protocol and its convergence properties, Technique Report, 2003. Document also available at: http://www8.in.tum.de/teaching/SS03/routing/final/4.pdf.

[15] B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless sensor networks, in: Proc. of the 6th Annual International Conference on Mobile Computing and Networking, ACM/IEEE MobiCOM'00, 2000, pp. 243–254.

[16] Y. Ko, N. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: Proc. of the 4th ACM/IEEE MOBICOM, 1998, pp. 66–75.

[17] J. Kuruvila, A. Nayak, I. Stojmenovic, Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical layer, IEEE Journal on Selected Areas in Communications 23 (6) (2005) 1267–1275.

[18] H. Lee, A. Cerpa, P. Levis, Improving wireless simulation through noise modeling, in: Proc. of the 6th International Conference on Information Processing in Sensor Networks, ACM IPSN'07, 2007, pp. 21–30.

[19] C. Liang, A. Terzis, Rethinking multi-channel protocols in wireless sensor networks, in: Proc. of HotEmNets, 2010. Document also available at: http://hinrg.cs.jhu.edu/joomla/uploads/Main/mc.pdf.

[20] C. Lin, B. Liu, H. Yang, C. Kao, M. Tsai, Virtual-coordinate-based delivery-guaranteed routing protocol in wireless sensor networks with unidirectional links, in: Proc. of IEEE INFOCOM, 2009, pp. 351–355.

[21] K. Liu, N. Abu-Ghazaleh, K. Kang, Location verification and trust management for resilient geographic routing, Journal of Parallel and Distributed Computing 62 (2) (2007) 215–228.

[22] Y. Liu, Q. Zhang, L. Ni, Opportunity-based topology control in wireless sensor networks, in: Proc. of ICDCS, CD-ROM, 2008.
[23] M. Lukic, B. Pavkovic, N. Mitton, I. Stojmenovic, Greedy geographic routing algorithms in a real environment, in: Proc. of the Fifth International Conference on Mobile Ad-Hoc and Sensor Networks, 2009.
[24] M. Marina, S. Das, Routing performance in the presence of unidirectional links in multihop wireless networks, in: Proc. of the 3rd ACM MobiHoc, 2002, pp. 12–23.
[25] M. Nesterenko, A. Vora, Void traversal for guaranteed delivery in geometric routing, in: Proc. of the 2nd IEEE MASS, 2005.
[26] S. Olariu, I. Stojmenovic, Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting, in: Proc. of the 25th IEEE INFOCOM, 2006.
[27] K. Tang, M. Correa, M. Gerla, Isolation of wireless ad hoc medium access mechanisms under tcp, in: Proc. of the ICCCN, 1999, pp. 77–82.
[28] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proc. of ACM SenSys, 2003, pp. 14–27.
[29] S. Yessad, F. Nait-Abdesselam, T. Taleb, B. Bensaou, R-MAC: reservation medium access control protocol for wireless sensor networks, in: Proc. of IEEE LCN, 2007, pp. 719–724.
[30] G. Zhou, T. He, S. Krishnamurthy, J. Stankovic, Impact of radio irregularity on wireless sensor networks, in: Proc. of ACM MobiSys, 2004, pp. 125–138.
[31] M. Zorzi, R. Rao, Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance, IEEE Transactions on Mobile Computing 2 (4) (2003) 337–348.

Zhen Jiang received B.S. degree from Shanghai Jiaotong University, China, in 1992, Master degree from Nanjing University, China, in 1998, and Ph.D. degree from Florida Atlantic University, USA, in 2002. Currently, he is the associate professor of Computer Science Department at West Chester University of Pennsylvania, USA. He is also a faculty member of Information Assurance Center at West Chester University. His research interests are in the areas of information system development, routing protocols, and wireless networks. He is a member of the IEEE.

Zhigang Li received his M.S. degree in College of Computer from National University of Defense Technology (NUDT), China, in 2005. He is currently working toward the Ph.D. Degree in Computer School at NUDT. His research interests include pervasive computing, wireless sensor networks and distributed computing.

Jie Wu is a professor and the chairman of the Department of Computer and Information Science, Temple University. His research interests include the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 450 papers in various journals and conference proceedings. He is on the editorial board of the IEEE Transactions on Mobile Computing. He was on the editorial board of the IEEE Transactions on Parallel and Distributed Systems and was a co-guest editor of the IEEE Computer and Journal of Parallel and Distributed Computing. He is the author of the textbook "Distributed System Design" published by the CRC Press. He has served as an IEEE Computer Society distinguished visitor and is the chairman of the IEEE Technical Committee on Distributed Processing (TCDP). He is a fellow of the IEEE.

Nong Xiao received his B.S. degree in 1990 and Ph.D. in 1996 from the College of Computer at the National University of Defense Technology, China. He is a professor in Computer School at the National University of Defense Technology. His research interests include Grid computing, Storage and Architecture.