무선 애드-혹 망을 위한 위치기반 라우팅에서의 맹목적 우회경로 결정문제

(Blind Detouring Problem in Geographic Routing for Wireless Ad-hoc Networks)

나종근†   김종권††
(Jongkeun Na)   (Chongkwon Kim)

요 약 최근 몇년간 무선 애드-혹 라우팅을 위한 많은 스킴들이 광범위하게 연구되었다. 위치기반 라우팅은 애드-혹 라우팅의 한 부류로써, 각 중계노드는 자신 및 목적지, 그리고 이웃노드들의 위치정보에 기반하여 독립적으로 다음-홉(next-hop) 노드를 선택한다. 이러한 위치기반 라우팅은 라우팅 상태 정보를 유지해야 하는 부담이 없기 때문에 대규모 애드-혹 망에 적합하다. 그러나, 위치기반 라우팅은 발신지와 목적지 사이에 빈공간(void)이 존재할 경우 매우 긴 우회 경로를 선택하는 문제를 안고 있다. 이 문제를 해결하기 위해, 본 논문에서는 *랜드마크를 이용하는 위치기반 라우팅 방안(GLR)*을 제안한다. GLR은 랜드마크(landmark)라고 불리우는 중계노드를 재귀적으로 발견하고 랜드마크 사이를 연결하는 서브패스(sub-path)를 구성한다. 다양한 망 위상에서 행해진 시뮬레이션 결과를 통해서, 제안된 방안(GLR)이 위치기반 라우팅의 성능을 크게 향상시킬 수 있음을 보인다.

키워드 : 위치기반 라우팅, 랜드마크를 이용한 위치기반 라우팅, 페이스 라우팅

Abstract Wireless ad-hoc routing has been extensively studied and many clever schemes have been proposed over the last several years. One class of ad-hoc routing is geographic routing where each intermediate node independently selects the next hop using the given location information of destination. Geographic routing, which eliminates the overhead of route request packet flooding, is scalable and suitable for large scale ad hoc networks. However, geographic routing may select the long detour paths when there are voids between a source and a destination. In this paper, we propose a novel geographic routing approach called *Geographic Landmark Routing (GLR)*. GLR recursively discovers the intermediate nodes called landmarks and constructs sub-paths that connect the subsequent landmarks. Simulation results on various network topologies show that GLR significantly improves the performance of geographic routing.

Key words : Geographic Routing, Geographic Landmark Routing (GLR), Face Routing

1. Introduction

Recently geographic routing has been proposed to solve the scalability problem of wireless ad hoc routing schemes. In geographic routing, it is assumed that each node knows its own location [9,10] and the location of destination nodes. Using this location information, the source and each intermediate node select the next hop closest to the destination. This basic routing technique is called greedy forwarding. Geographic routing also employs fallback (or recovery) routing in addition to greedy forwarding to provide an adequate packet delivery ratio [1]. Nodes are not always homogenously populated in ad-hoc networks, and networks may contain areas without any nodes; these areas are called voids or *communication holes*. When geographic routing encounters a void, it cannot apply greedy forwarding because there are no neighbor nodes that are closer to the destination than the

---

· 본 연구는 2005년도 두뇌한국21의 연구비 지원으로 수행하였습니다.

† 학생회원 : 서울대학교 컴퓨터공학과
jkna@popeye.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학과 교수
ckim@snu.ac.kr

논문접수 : 2006년 2월 6일
심사완료 : 2006년 8월 17일

current node. To solve this problem known as *local minimum problem*, fallback routing mechanism is used in geographic routing.

Bose et al. [2] proposed basic face routing algorithms (FACE-1 and FACE-2) that can be employed in fallback routing. On a *planar graph*[1], FACE-1 explores the faces that have edges intersecting an imaginary line $\overline{st}$ connecting source s to destination t. At each face, FACE-1 visits all the nodes of the face to find the intersecting edge closest to t and then restarts the next exploration at the edge. This procedure repeats until the destination is reached. FACE-2 eliminates the overhead of exhaustive face exploration. Instead of exploring all face boundary nodes, FACE-2 stops exploring when it finds the first intersecting edge that is closer to t than the starting point. FACE-2 restarts the face exploration from that first intersecting edge. FACE-2 generally performs better than FACE-1 but it has some pathological cases [2] i.e. selecting very long detour paths, due to its partial face exploration. Because of full face exploration overhead, slightly modified FACE-2 algorithms have been used in several geographic routing schemes such as GPSR.

GPSR [3] is a well-known representative geographic routing scheme that combines greedy forwarding and perimeter routing which is a fallback routing mechanism based on FACE-2. GPSR uses greedy forwarding as long as possible. When a packet arrives at a dead-end, GPSR applies perimeter routing to search for the first node that is closer to the destination than the dead-end. At that node, GPSR resumes greedy forwarding. When a packet bound for node t enters the perimeter routing mode at a dead-end x, GPSR forwards it along the face that intersects with the line $\overline{xt}$. GPSR basically explore the face according to *right-hand rule*, i.e. along with counter-clockwise direction about x via line $\overline{xt}$. Because GPSR also does not use full face exploration, it is not free from selecting long detour paths as in FACE-2. The same thing also appears in other geographic routing schemes such as GOAFRFC [5] since they basically share the fallback routing concept of GPSR.

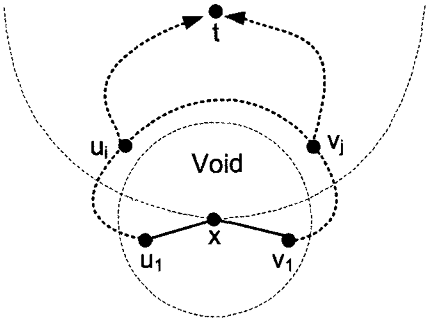


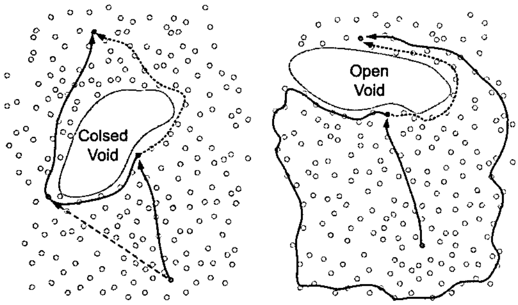
Fig. 1 Void bypassing alternatives at a dead-end x

Let us give the general example of detouring void when we combine greedy routing and fallback routing. Fig. 1 shows two alternatives to bypass a void at dead-end x. The shaded area represents the void of node x with respect to destination t. The void is contained in an interior face or exterior face when nodes and their edges are represented as network graph. If u_1-u_i, u_i-v_j and v_j-v_1 sub-paths can be connected over multi-hops, the void is located in the interior face i.e. $x-u_1-u_i-v_j-v_1-x$, otherwise it is in an exterior face. If we apply the right-hand rule in face exploration, the path, i.e. $x-u_1-u_i-t$ is selected. Otherwise, $x-v_1-v_j-t$ is by the corresponding left-hand rule. In here, u_i and v_j are nodes that stop fallback routing and re-start greedy routing toward t. Depending on the location of u_i and v_j, one path is shorter than the other. We note that it is dangerous to blindly apply right-hand rule with no full face exploration in fallback routing mode. Of course, node x can select u_1 or v_1 if the perimeter information of void is available. That information can be maintained by other means such as BoundHole [8]. Once a hole (aka void) is detected by BoundHole algorithm, the boundary information of the hole is cached in perimeter nodes around the

---

[1] The Relative Neighborhood Graph (RNG) and Gabriel Graph (GG) are well-known planar graphs. [2] and [3] provide the description of on-line algorithms that transform a non-planar graph into RNG or GG

hole. When a packet reaches a local minimum at any perimeter node, the locally stored boundary information helps the packet bypass the hole toward destination. However, proactively updating and maintaining such perimeter state information is undesirable with dynamic topology changes. In this paper, we propose a new geographic routing approach reactively bypassing voids in the end-to-end point of view, unlike proactively maintaining the local perimeter information of voids.



(a) (b)

Fig. 2 The blind detouring and triangular routing problems; (a) closed void (b) open void

Exploring face in one direction blindly, we may select long detour paths even though the paths in the other direction are much shorter. Actually, the probability of selecting longer paths is 50%. We call the problem of choosing paths blindly the *blind detouring problem*. Fig. 2(a) shows an example of the blind detouring problem. The solid line is the path from s to t selected by GPSR that applies the right-hand rule. The other detour path marked with a dotted line is shorter than the path selected by GPSR. If we use the left-hand rule, the shorter dotted line would have been selected. Geographic routing may select particularly bad paths when networks have open voids. Voids are classified into open and closed voids. A closed void is surrounded by connected nodes while an open void is not. Fig. 2(a) and Fig. 2(b) illustrate closed and open voids, respectively. Blind application of the right-hand rule (or left-hand rule) to escape open voids may result in very long detours that circle around the network boundary. In addition to the blind detouring problem, geographic routing has another problem called *triangular routing*. Fig. 2(a) illustrates the triangular routing problem. As explained before, geographic routing searches detour paths from a dead-end x to find the first node that is closer to destination t than x is. Let the first node be l in Fig. 2(a). Geographic routing forwards packets from x to l via a dead-end x and forms a triangular path. A better route would forward packets directly to l rather than take a triangular route through x.

We propose a novel geographic routing scheme called *Geographic Landmark Routing (GLR)* that solves the blind detouring problem and the triangular routing problem. To the best of our knowledge, these problems have not been fully explored in the existing literature. GLR discovers two paths, one from the source to destination direction and another in the reverse direction bypassing a void area. GLR solves the blind detouring problem by comparing the two paths and selecting the shorter path. GLR also solves the triangular routing problem. While exploring paths, GLR identifies nodes called landmarks. A *landmark node* is a node that reclaims greedy forwarding after escaping from a dead-end. Using landmarks as intermediate targets of loose source routing, GLR is able to find straight sub-paths between landmarks. We also present a preliminary simulation result. It shows that GLR always performs better than GPSR and, furthermore, closely follows the shortest paths on various network topologies containing irregular voids.

The rest of the paper is structured as follows. Section 2 contains a high-level overview of a novel routing approach to solve the blind detouring problem and triangular routing problem, also provides a detailed description of the protocol including landmark discovery and landmark routing. In Section 3, we evaluate the performance of GLR. Concluding remarks are given in Section 4.

2. Geographic Landmark Routing

Two main components of GLR are landmark discovery and landmark routing. GLR discovers landmark nodes between the source and destination using landmark discovery. It selects a path from source x to destination t, and registers landmark nodes on the selected path in a table called LC (Landmark Cache) at the source node. The entries in LC are indexed by destination nodes. GLR performs landmark routing, i.e. *loose source routing* using landmark nodes as intermediate nodes, to transmit packets through landmark nodes.

2.1 Landmark Discovery

Landmark discovery finds both forward and backward paths between the source and destination. When a new packet is generated, the source looks up the LC for landmark information to the destination. If the LC contains landmark information, then packets are forwarded to the destination via landmark routing. The landmark discovery phase is performed only if there are no entries for the destination in the LC. The source node forwards a control packet called FLD (Forward Landmark Discovery) toward the destination[2]. While being forwarded to the destination, FLD records the hop count and the landmark nodes on a path P_{st} from source s to destination t. Note that a landmark node is a node which terminates fallback routing and restarts greedy forwarding. FLD records one landmark node for each void that it encounters. Eventually, the FLD packet will arrive at the destination. The destination node, upon receiving the FLD packet, constructs a control packet called BLD (Backward Landmark Discovery) and copies the forward path information (hop count and landmark nodes) onto the BLD packet. Then the destination transmits the BLD packet to the source. Like the FLD control packet, the hop count and landmark nodes of the backward path $\widetilde{P}_{st}$ are recorded in the BLD packet.

```
L1: if( $fhop \le bhop$ )
L2:     if( $\{l_{st}\} = \phi$ ) LC_entry(t)=null;
L3:     else LC_entry(t) = $\{l_{st}\}$;
L4: else // $fhop > bhop$
L5:     if( $\{\widetilde{l}_{st}\} = \phi$ ) LC_entry(t)= $\{v\}$;
L6:     else LC_entry(t) = $\{\widetilde{l}_{st}\}$;
```

Fig. 3 GLR Path Selection Algorithm

The source will receive the BLD packet that contains information on the forward and backward paths. Based on the path information, GLR selects one of two paths for later packet transmission. Let *fhop* and *bhop* be the hop counts of forward and backward paths, respectively, and let $\{l_{st}\}$ and $\{\widetilde{l}_{st}\}$[3] be the sets of landmark nodes on the forward and backward paths, respectively. The LC entry for destination t, *LC_entry(t)* which is referenced at later packet forwarding, at source s is labeled with appropriate landmark information by the simple path selection algorithm listed in Fig. 3. In L2, LC_entry(t) is marked with null because of no need to perform landmark routing. L3 and L6 specify the selection of paths of shorter lengths if both forward and backward paths contain landmark nodes. When the backward path ($\widetilde{P}_{st}$) is shorter but $\{\widetilde{l}_{st}\} = \phi$, we need to locate one node on $\widetilde{P}_{st}$ as a virtual landmark (v) noted in L5. The reason is because the source cannot retrace the backward path using greedy forwarding based on only destination location. Without this consideration, it would make an undesirable result that follows the forward path P_{st} longer than $\widetilde{P}_{st}$. In GLR, a virtual landmark node v on $\widetilde{P}_{st}$ is acquired as a result of processing BLD packet such that the location of v is perpendicularly farthest away from line $\overline{st}$.

2.2 Landmark Routing

Landmark routing additionally solves the triangular routing problem by applying loose source routing with the landmark nodes detected during the landmark discovery phase. Whenever a source node has a packet to send, it looks up the LC_entry(t) that corresponds to destination t. There are three cases (C1-C3).

C1) LC_entry(t) does not exist ⇒Perform landmark discovery with the destination.

C2) LC_entry(t) exists but is null ⇒Forward via greedy forwarding only since no void between source and destination.

---

[2] We may piggyback a data packet in a control packet for quick data deliveries.

[3] The landmark nodes in $\{\widetilde{l}_{st}\}$ are rearranged such that the first landmark nodes from the source s appears first.

C3) LC_entry(t) specifies landmark nodes ⇒Perform loose source routing using the landmark nodes as intermediate targets.

Fig. 4 illustrates an example of GLR. As shown in Fig. 4(a), source *s* discovers forward path P_{st} and backward path $\widetilde{P}_{st}$ to and from the destination *t* during the landmark discovery phase. The forward path P_{st} has one dead-end *x* and one landmark node $\{l_{st}^{1}\}$. The backward path $\widetilde{P}_{st}$ also has one dead-end *y* and one landmark node $\{\widetilde{l}_{st}^{1}\}$. The source node selects the backward path by the path selection algorithm and store $\{l_{st}^{1}\}$ in the LC. Once the landmark node is registered in the LC, all packets bound for destination *t* are routed via loose source routing. The backward path $\widetilde{P}_{st}$ contains a triangular route between *t* and $\{\widetilde{l}_{st}^{1}\}$, but landmark routing selects a straight route as in Fig. 4(b).

2.3 Recursive GLR

We have to apply GLR recursively because there may be voids between two consecutive intermediate targets (i.e. landmark nodes) of source routing. Let us assume that there are *k* landmark nodes, $l_{st}^{1},\dots,l_{st}^{k}$ on a selected path from *s* to *t*. Given a packet destined to *t*, the source first sends the packet to l_{st}^{1}. The source applies GLR to find an efficient path from *s* to l_{st}^{1}. The recursive application of GLR should be performed at each intermediate landmark node.

Fig. 5 shows an example of the recursive application of GLR. Source *s* first discovers the landmarks on two directional paths, P_{st} and $\widetilde{P}_{st}$, using the landmark discovery protocol, as shown in Fig. 5(a). Let us assume that *s* decides to use the backward path $\widetilde{P}_{st}$. According to this decision, landmark nodes l_1 and l_2 on path $\widetilde{P}_{st}$ are recorded in the LC entry. Once the landmarks are registered, data packets are forwarded by loose source routing via $\{l_1, l_2\}$ as shown in Fig. 5(b). At node *s*, source routing to the next landmark node, l_1, may trigger the second level landmark discovery; if there is no LC entry for l_1, then node *s* creates an FLD packet destined to l_1. The second level landmark discovery is also performed at l_1. Fig. 5(c) shows the result of the second level landmark discovery. New landmark l' is found in the sub-path between l_1 and l_2. Note that we may transfer data packets while performing the second level landmark discovery. The second level source routing path is formed like Fig. 5(d). Next the third level landmark discovery starts from newly formed sub-paths, i.e. from landmark nodes l_1 and l', as shown in Fig. 5(e). The second level landmark forwarding path

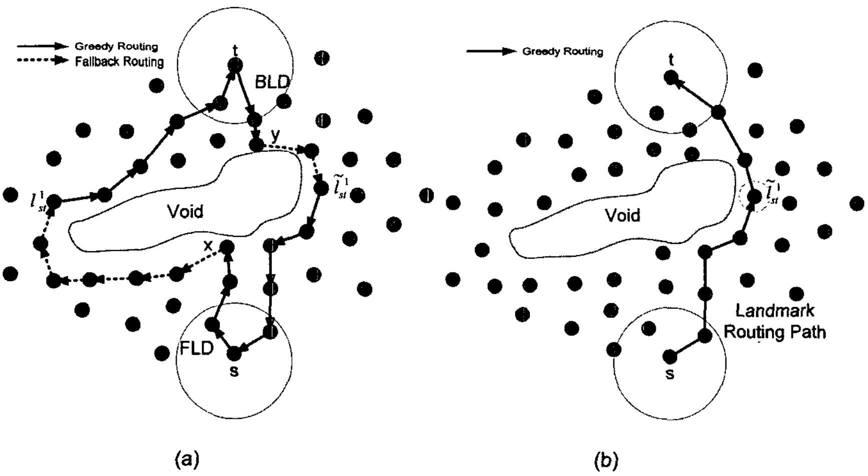


Fig. 4 An example of GLR; (a) landmark discovery phase (b) landmark routing phase

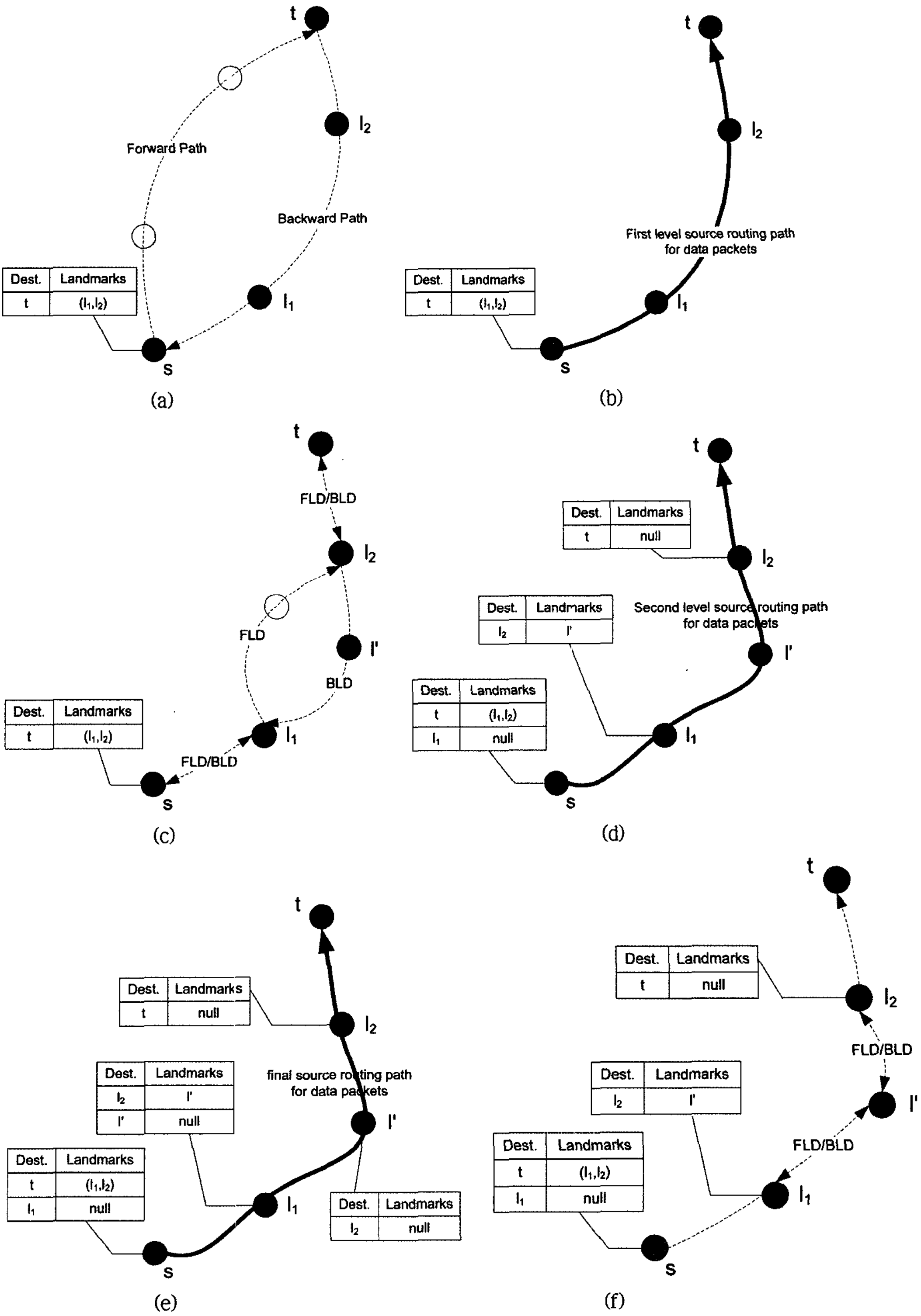


Fig. 5 An example of landmark discovery and routing

becomes the final landmark routing path between s and t because the third level landmark discovery does not find any new landmarks. Fig. 5(f) illustrates the contents of LC at all involved landmark nodes.

2.4 Adapting Topology Changes

The topology of ad hoc networks can dynamically change, thus ad hoc routing must be robust against frequent topology changes. In an ad-hoc wireless environment, topology change is caused by two major factors, node down (or sleep) and node mobility. In GLR, when a landmark node is unavailable, the previous hop node skips the landmark node and forwards packets to the next intermediate target through dynamic self-determination. A node can judge the unavailability of a landmark node when the landmark node repeatedly fails to respond to mellow packets. In that case, GFF (Greedy and Fallback Function) passes an unreachable packet to LR (Landmark Routing). LR changes *Target Addr* and *Target Location* with the next target node of loose source routing. Then, it passes the packet down to GFF so that the packet is forwarded toward the next target. At the next landmark discovery time, the stale landmark information is refreshed. In GLR protocol, landmark discovery is periodically repeated at the landmark refresh interval (*LRI*) given as a protocol parameter.

2.5 Protocol Overhead

In GLR protocol, we need to consider some additional control overhead such as FLD/BLD control packets for landmark discovery. The overhead in a node is bounded by the number of different destinations that can be represented as follows: $m_i = o_i + r_i$ for node i, where m_i means the total number of destinations, o_i the number of different destinations in packets that the applications of node i originate, r_i the number of different destinations to which the packets are relayed by landmark routing while node i is serving as a landmark. Therefore, network-wide additional control packet overhead can be expressed by $m = \sum m_i$ for $i = 1,...n$ where n is the number of nodes. For a destination that is not yet explored, a node will initiate landmark discovery to the destination in order to check if it has a blind detouring problem.

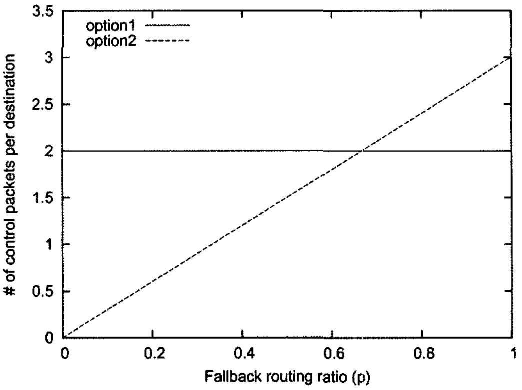


Fig. 6 The control overhead of GLR according to the trigger option of landmark discovery; c_1=2 for FLD/BLD control packets, c_2=1 for the feedback.

We may have two options in triggering landmark discovery: 1) when a packet whose destination is not yet discovered is forwarded; 2) when a source node receives any feedback signaling the occurrence of fallback routing on route for the packet it sent. In the second option, as feedback information, a stuck node which starts fallback forwarding for the packet should inform the source node of the need of landmark discovery. The cost function of each option can be represented as follows:

$$C_{option1} = mc_1$$

$$C_{option2} = pm(c_1 + c_2)\ where\ c_1 > c_2$$

In the cost functions, p denote the likelihood of experiencing fallback routing on the path to any destination, i.e. fallback routing ratio, c_1 the averaged cost for FLD/BLD message exchange, c_2 the average cost for feedback. Let us simplify these costs as the number of required control packets. As shown in Fig. 6, the option2 would be preferred if $p \le \frac{c_1}{c_1 + c_2} = 0.66$ since the cost can be greatly reduced at the expense of minimal feedback. On the other hand, if $p > 0.66$, it would be better to choose the option1 that does not require any additional feedback information. Note that we use the option1 for simplicity in the simulation of Section 3, and we leave optimizing GLR protocol such as the option2 to the future work.

3. Performance Evaluation

3.1 Simulation Model

We conducted extensive simulations using ns-2 (v2.26) network simulator. We modified the GPSR code which is previously implemented by B. Karp [3]. In our simulation, we used the standard values of the WLAN physical layer and IEEE 802.11 MAC layer parameters. In order to represent the variety of node deployment in wireless networks, we first defined simulation parameters as shown in Table 1. We introduce two hole models, the Centered Random Hole Model (CRHM) and the Free Random Hole Model (FRHM), to emulate randomized communication holes in networks where the nodes are distributed in areas of irregular shapes [6] or non-convex geometric structures [7]. The CRHM is suitable to represent one large void area surrounded by nodes, while the FRHM can model randomly dispersed communication holes.

We simulated four scenarios (I/II/III/IV) listed in Table 2. To acquire the sample paths that give the statistical significance, we created several random networks with the given simulation parameters for each simulation scenario. In each random network, n paths are sampled where n is the number of nodes in the network. The void areas are determined by the given hole model (hm) and hole factor (h). In Table II, d and h are randomly selected from the uniform distribution $U(l,u)$. Note that the hole factor h is limited to $U(0,10)$ in the FRHM hole model to ensure the unpartitioned networks. For each scenario, we collected the sample paths as follows: 988 for I, 1003 for II, 1496 for III, 1737 for IV.

We compare the performance of GPSR and GLR. In addition, we compare with an optimal routing algorithm called SHORTEST to illustrate the gaps between optimal performance and obtained results. We use an index, *fallback routing ratio (frr)*, that denotes the fraction of paths which use fallback forwarding. We use two metrics to evaluate the path optimality of routing algorithms: *path length* and *path stretch*. The path stretch is the ratio of path length to the shortest path length.

3.2 Results

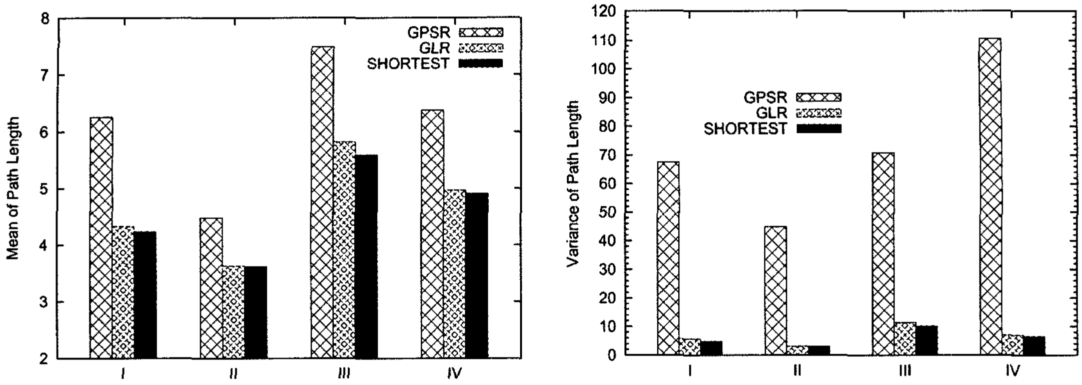
Fig. 7 shows the mean and variance of path length. In each simulation scenario, the fallback routing ratio (frr) is 0.14 for I, 0.07 for II, 0.20 for

Table 1 Simulation Parameters

Parameter	Description
Network Size (*A*)	$1500 \times 1000\text{m}^2$ and $2500 \times 1000\text{m}^2$
Transmission Range (*r*)	250m (WLAN)
Node Density (*d*)	The average number of neighbors per node. Nodes of $(d \times A)/(\pi \times r^2)$ are uniformly distributed over given topology.
Hole factor (*h*)	The number of holes randomly positioned. Each hole is represented as a circle with radius $r/2$. Nodes cannot be located within holes. Selecting random positions of the hole depends on *hole model*.
Hole Model (*hm*)	CRHM or FRHM; Centered Random Hole Model (CRHM) restricts the position of center of the hole into a rectangle area with (left=0.4X, bottom=0.4Y) and (right=0.6X, top=0.6Y). Free Random Hole Model (FRHM) does not have such a restriction.
Traffic Source	CBR source with data rate 2K bytes/s
Landmark Refresh Interval (*LRI*)	100sec; FLD message is sent periodically for adapting topology changes

Table 2 The simulation parameter values in each simulation scenario (I/II/III/IV)

Scenario	I	II	III	IV
Topology	1500×1000	1500×1000	2500×1000	2500×1000
Node Density (*d*)	U(7,20)			
Hole factor (*h*)	U(0,40)	U(0,10)	U(0,40)	U(0,10)
Hole Model (*hm*)	CRHM	FRHM	CRHM	FRHM

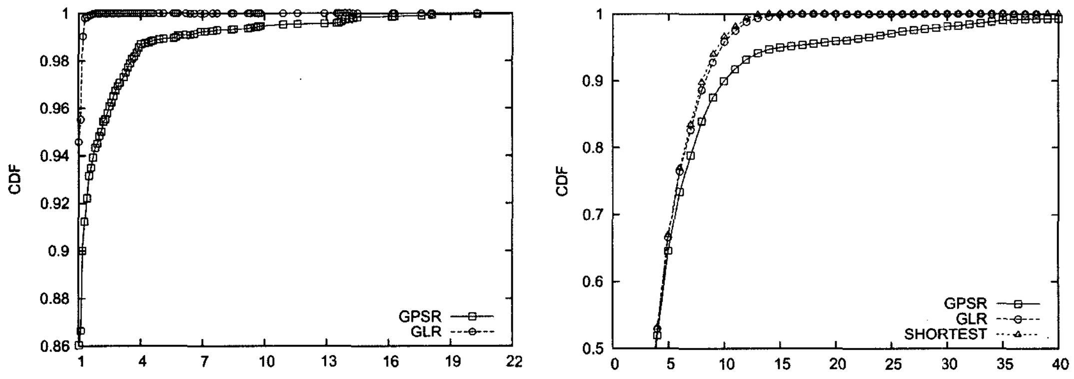


(a) (b)

Fig. 7 Mean and variance of path length in each simulation scenario (I/II/III/IV)

III, and 0.13 for IV, respectively. GPSR averagely takes more hops over 1 or 2 hops than GLR, but has very high variance. When there are voids, GPSR blindly applies counter-clockwise face exploration that may select very long path and traverses many nodes connected on planar graph. Such a long perimeter forwarding generates extremely long paths even over 100 hops in the case of open voids. For an extremely long path of GPSR, GLR reduces it to a reasonable path length by fixing the blind detouring and triangular routing problem. As a result of that, GLR shows the routing performance that is comparable to SHORTEST. Additionally, the difference of average path length between GPSR and GLR will increase with proportional to *frr* since relatively long paths are more made by GPSR when *frr* is high.

In comparison with GPSR, GLR provides much better performance in terms of path length. GPSR basically uses hybrid (i.e. greedy and fallback) forwarding. In GLR, such hybrid forwarding is converted to pure greedy forwarding. Intuitively, pure greedy forwarding makes a path comparable to the shortest. Fig. 8 shows total aggregated distribution for path length and path stretch on data collected from all simulation scenarios. As expected, GLR closely follows the distribution line of SHORTEST in path length, and almost converges to 1 in path stretch. And also, we confirm that maximum path stretch of GLR is bounded to less than 2 (1.55 for I, 1.71 for II, 1.66 for III, 1.50 for IV) but GPSR shows very large path stretch, for instance, 40 in III. Fig. 8(a) indicates that GLR provides the path quality comparable with SHORTEST by fixing the anomaly of fallback routing (to the amount of 13.5% in total, *frr*=0.135)



(a) (b)

Fig. 8 Aggregated cumulative distributions for all simulation scenarios induced by GPSR.

4. Conclusion

In this paper, we propose a new geographic routing protocol called GLR (Geographic Landmark Routing) in order to remove the side effects, i.e. blind detouring and triangular routing problem, of heuristic-based fallback mechanism in geographic routing. GLR detects void areas through simple discovery and efficiently gets around them using landmark-based loose source routing. GLR forces pure greedy forwarding to be only used in traversing landmark nodes. That makes GLR outperform GPSR as noticed in our simulation results. Our next step will provide concrete architecture, protocol and extensive performance evaluation in practical aspects.

References

[1] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," IEEE Network Magazine, 15(6):30--39, November 2001.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," In Proc. Of 3rd ACM Intl. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99, pages 48-55, 1999.

[3] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000), 2000.

[4] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically Optimal Geometric Mobile Ad-Hoc Routing," In Proc. 6th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial-M), pages 24-33. ACM Press, 2002.

[5] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Mobile Ad-Hoc Routing," In Proc. 4th Int. Symposium on Mobile Ad-Hoc Networking and Computing (Mobihoc), 2003.

[6] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker and Ion Stoica, "Geographic Routing without Location Information," MOBICOM 2003.

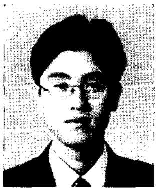
[7] Ljubica Blazevic, Jean-Yves Le Boudec, and Silvia Giordano, "A Location-Based Routing Method for Mobile Ad Hoc networks," IEEE Transtions on Mobile Computing, Vol. 4, No. 2, March/April 2005.

[8] Qing Fang, Jie Gao, Leonidas J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks", INFOCOM 2004.

[9] J. Li, J. Jannotti, D. DeCouto, D. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in Proc. 6th Annu. ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000), 2000.

[10] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage in sensornets," in 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), 2002, pages 78-87.

[11] ns Notes and Documentation. http://www.isi.edu/vint/nsnam/, 2000.



나 종 근

1992년 아주대학교 전자계산학과 학사
1995년 아주대학교 컴퓨터공학과 석사
2003년~현재서울대학교 전기.컴퓨터공학부 박사과정. 관심분야는 무선망, 라우팅 알고리즘 및 프로토콜

김 종 권

정보과학회논문지 : 정보통신 제 33 권 제 2 호 참조