

RANDOMIZED ROUTING ALGORITHMS IN MOBILE AD HOC NETWORKS

T. Fevens, I.T. Haque, and L. Narayanan
*Department of Computer Science and Software Engineering,
Concordia University,
Montreal, Quebec, Canada, H3G 1M8*
{fevens,it_haque,lata}@cs.concordia.ca

Abstract Position-based routing is a well-known paradigm for routing in mobile ad hoc networks. We give several new randomized position-based strategies for routing in mobile ad hoc networks. Our algorithms combine the greedy heuristic of minimizing the distance remaining to the destination and the directional heuristic of staying close to the direction of the destination with the use of randomization to retain some flexibility in the chosen routes. Our experiments show that randomization gives a substantial improvement in the delivery rate over the deterministic greedy and directional routing algorithms. For some of the algorithms we propose, this improvement comes at the expense of only a small deterioration in the stretch factor of the routes.

Keywords: Wireless networks, mobile ad hoc networks, routing, position based routing, stretch factor, delivery rate.

Introduction

A mobile ad hoc network (MANET) is a collection of autonomous mobile devices that can communicate with each other without having any fixed infrastructure. Each node in the network has an omni-directional antenna and can communicate using wireless broadcasts with all nodes within its transmission range. Thus a MANET can be represented by a unit disk graph, where two nodes are connected if and only if their Euclidean distance is at most the transmission range [Barriere et al., 2001]. Since nodes may not directly communicate with all other nodes because of the limited transmission range, multi-hop communication is needed in the network. The nature of MANETs include issues such as dynamic topology changes, absence of infrastructure, autonomous heterogeneous nodes, and resource constraints that contribute to making the problem of routing in these networks a tremendous challenge.

In the last few years, a plethora of routing protocols for MANETs has been proposed in the literature. As yet there is no consensus and no standards have been adopted. The proposed protocols can be divided into two main categories: *proactive* and *reactive* protocols. Proactive or table-driven protocols [Perkins and Bhagwat, 1994] are based on Internet distance-vector and link-state protocols, and maintain consistent and updated routing information about the entire network by exchanging information periodically. Randomized versions of some of these proactive protocols have been proposed, such as R-DSDV [Boukerche et al., 2001; Choi and Das, 2002; Boukerche and Das, 2003]. Reactive or on-demand routing protocols [Perkins and Royer, 1999; Johnson et al., 2002] discover routes only when data needs to be sent or the topology is changed. Reactive protocols typically use less bandwidth in terms of control packets to discover topology information, but even so, packets to discover new routes must sometimes be flooded through the network, which consumes a huge amount of bandwidth [Chlamtac et al., 2003].

One way of limiting flooding is by using information about the position of nodes in the network. In position-based routing protocols, a node forwards packets based on the location (coordinates in the plane) of itself, its neighbors, and the destination [Giordano et al., 2003]. The position of the nodes can be obtained using GPS, for example, if the nodes are outdoors. There are numerous ways of using position information in making routing decisions. For instance, in DREAM [Basagni et al., 1998] and LAR [Ko and Vaidya, 1998], information about the position of the destination is used to limit the extent of flooding. Nodes whose position makes it unlikely for them to be on a shortest path to the destination will simply not forward packets. In Terminode and Grid [Liao et al., 2001], position-based routing is used to cover long distances and non-position based algorithms are used for shorter distances. In another class of algorithms, which has been termed *progress-based* algorithms in [Giordano et al., 2003], the algorithm forwards the packet in every step to exactly one of its neighbors, which is chosen according to a specified heuristic. Finally, position information can be used to extract a planar sub-graph such that routing can be performed on the perimeter of this sub-graph as in [Bose et al., 1999] and [Karp and Kung, 2000]. The advantage of this last approach is that delivery of packets can always be guaranteed.

In this paper we shall focus on the progress-based routing algorithms. In *greedy routing* [Finn, 1987; Stojmenovic and Lin, 2001], a node forwards the packet to its neighbor which is closest to the destination. *Compass* or *directional* routing [Kranakis et al., 1999] moves the packet to a neighboring node such that the angle formed between the current node, next node, and destination is minimized. Clearly the next node selected by the two heuristics is not always the same (see Figure 1 for an example). Both of these algorithms are

known to fail to deliver the packet in certain situations. For examples of such situations, see [Bose and Morin, 1999; Karp and Kung, 2000].

In this paper, we propose several variations of the greedy and directional heuristics. To improve on the delivery rate of these algorithms, we use randomization. Our heuristics combine in various ways the two goals of covering as much distance as possible to the destination (as in greedy routing) and staying as close as possible to the direction of the destination (as in directional routing), while using randomization to allow flexibility with respect to the actual path followed. We evaluate all our heuristics in terms of delivery rate and stretch factor (ratio of the number of hops of the path given by the algorithm to the shortest path in the network). For purposes of comparison, we also study the performance of the greedy and compass routing strategies, as well as RCOMPASS. RCOMPASS chooses the next node uniformly at random from the two nodes that satisfy the directional heuristic on each side of the line from the current node to the destination (see Figure 1 for an example; a precise definition is given in Section 1.1). Our results show that randomization leads to a definite improvement in the delivery rate. Conversely, the best stretch factors is achieved by the deterministic algorithms. However, some of the randomized strategies do very well in terms of both measures of performance. In particular, one of our algorithms, WEIGHTEDRCOMPASS has the best delivery rate of all the algorithms while having one of the best stretch factors.

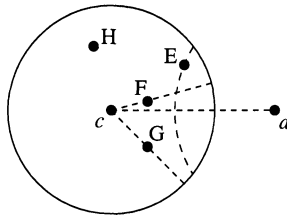


Figure 1. GREEDY chooses E , and COMPASS chooses F as next node, while RCOMPASS chooses uniformly at random from F and G to find the next node.

The rest of the paper is organized as follows. The next section gives relevant definitions including our routing strategies. Section 1.2 gives the empirical results of our simulations and provides an interpretation of the behavior of the algorithms. We conclude with a discussion of the results and future directions of this research in Section 1.3.

1.1 Definitions of Routing Algorithms

We assume that the set of wireless nodes is represented as a set S of n points in a two-dimensional plane. Two nodes are connected by a link if the Euclidean

distance between them is at most r , where r represents the transmission range of the nodes. The resulting graph $UDG(S)$ is called a unit disk graph. For node u , we denote the set of its neighbors by $N(u)$. Given a unit disk graph $UDG(S)$ corresponding to a set of points S , and a pair (s, d) where $s, d \in S$, the problem of online position-based routing is to construct a path in $UDG(S)$ from s to d , where in each step, the decision of which node to go to next is based only on the current node c , $N(c)$, and d . Here, s is termed the source and d the destination. A position-based routing algorithm is *randomized* if the next neighboring node is chosen randomly out of the neighbors of the current node [Bose and Morin, 1999]. The routing algorithm may or may not succeed in finding a path from s to d . The performance measures we are interested in are the *delivery rate*, that is, the percentage of times that the algorithm succeeds, and the *stretch factor*, the average ratio of the length of the path returned by the algorithm to the length of the shortest path in the graph. Here the length of the path is taken to mean the number of hops in the path; while other papers consider stretch factor based on Euclidean distance, we do not consider it here. Finally, nodes are assumed to be static for the duration of the packet transmission.

Given a node u , we denote the disk centered at node u with radius ℓ by $disk(u, \ell)$. Given an angle θ such that $0 \leq \theta \leq 2\pi$, we define $Sector(c, d, \theta)$ to be the sector given by angle θ in $disk(c, r)$ that is bisected by the line segment \overline{cd} . Further, given an α such that $0 \leq \alpha \leq 1$, we define *Periphery*, *Core*, and *Wing* as follows. Also, Figure 2 illustrates the given definitions.

- $Periphery(c, d, \theta, \alpha) = Sector(c, d, \theta) \cap disk(d, R)$ where R is chosen such that α is the ratio of the area of $Periphery(c, d, \theta, \alpha)$ to the area of $disk(c, r) \cap disk(d, R)$.
- $Core(c, d, \theta, \alpha) = Sector(c, d, \theta) - Periphery(c, d, \theta, \alpha)$.
- $Wing(c, d, \theta, \alpha) = (disk(c, r) \cap disk(d, R)) - Periphery(c, d, \theta)$.

1.1.1 Randomized Algorithms

In what follows, we always assume that the current node is c , the next node is x , and the destination node is d . The algorithms below differ in how to choose x from among the set $N(c)$.

FARINSECTOR: The next node x is chosen uniformly at random from the first non-empty set in the following sequence: $Periphery(c, d, \theta, \alpha)$, $Wing(c, d, \theta, \alpha)$, $Core(c, d, \theta, \alpha)$, $disk(c, r)$.

GREEDYINSECTOR: If $Sector(c, d, \theta)$ is not empty then x is chosen to be the neighbor of c with minimum distance from d , in $Sector(c, d, \theta)$. Otherwise, x is chosen uniformly at random from the set $N(c)$.

RANDOMINSECTOR: The next node x is chosen uniformly at random from

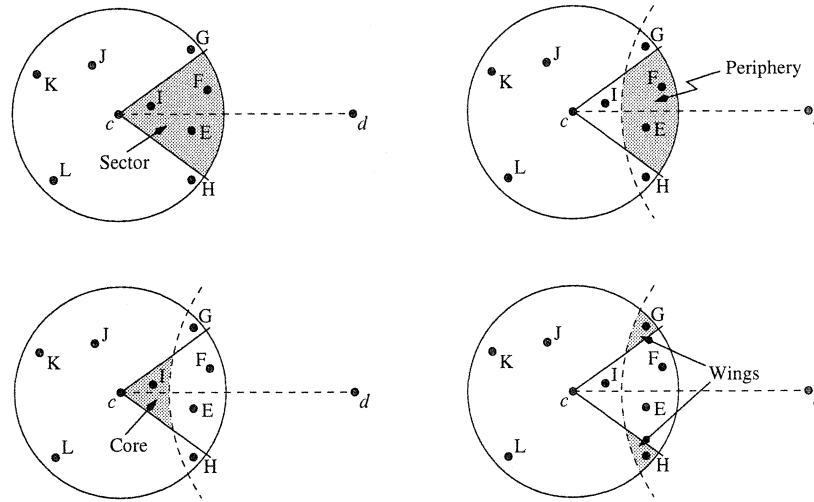


Figure 2. Illustrations of the definitions of *Sector*, *Periphery*, *Core*, and *Wing* (represented by the shaded regions). Note that $Sector(c, d, \theta) = Core(c, d, \theta, \alpha) \cup Periphery(c, d, \theta, \alpha)$. In this example, FARINSECTOR selects x uniformly at random out of nodes E and F , the only nodes in $Periphery(c, d, \theta, \alpha)$, while GREEDYINSECTOR picks $x = F$ since F is closest to d out of E, F , and I , the nodes in $Sector(c, d, \theta)$.

the first non-empty set in the following sequence: $Sector(c, d, \theta_1)$, $Sector(c, d, \theta_2)$, $disk(c, r)$, where $0 \leq \theta_1 \leq \theta_2 \leq 2\pi$.

RCOMPASS: Let $\angle xuy$ denote the angle formed by x , u , and y measured counterclockwise. Let n_1 be the neighbor of c above the line \overline{cd} such that $\angle dc n_1 = \theta_1$ is the smallest among all such neighbors. Similarly, n_2 is the neighbor of c below \overline{cd} such that $\angle n_2 cd = \theta_2$ is the smallest among all such neighbors. The next node x is chosen uniformly at random from n_1 and n_2 . This algorithm differs from the algorithm *Random Compass* proposed in [Bose and Morin, 1999] in the context of triangulations in a small way. In *Random Compass*, one neighbor chosen, $ccw(c)$, is the neighbor of u that minimizes $\angle dc\{ccw(c)\}$, and the second neighbor, $cw(c)$, is the neighbor of c that minimizes $\angle \{cw(c)\}cd$. Therefore, both neighbors could lie on the same side of the \overline{cd} line, whereas in RCOMPASS, if there are no neighbors on one side of the \overline{cd} line, only one neighbor is considered.

WEIGHTEDRCOMPASS: Let n_1 be the neighbor of c be defined as in RCOMPASS. The next node x is chosen from n_1 and n_2 with probability $\theta_2/(\theta_1 + \theta_2)$ and $\theta_1/(\theta_1 + \theta_2)$, respectively.

BEST2COMPASS: Let n_1 and n_2 be the neighbors of c such that $\angle dc n_1$ (or $\angle n_1 cd$) and $\angle n_2 cd$ (or $\angle dc n_2$) are the two smallest such angles among all

neighbors of c . Then the next node x is chosen uniformly at random out of n_1 and n_2 . This algorithm also differs from the algorithm *Random Compass* [Bose and Morin, 1999] in that the directions of the smallest angles are not considered.

BEST2GREEDY: Let n_1 and n_2 be the closest and second closest neighbors of c to the destination d . The next node is chosen uniformly at random out of these two nodes.

All the algorithms above take $O(k)$ time to find x , where d is the degree of c . The behavior of the algorithms is illustrated in Figures 2 and 3.

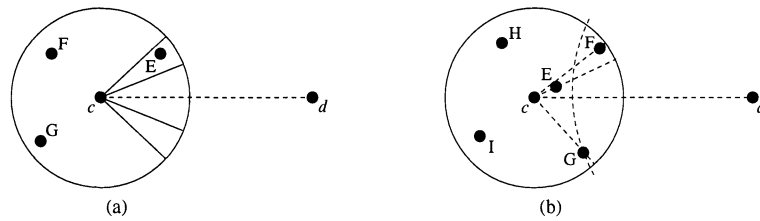


Figure 3. In (a) **RANDOMINSECTOR** selects E as the next node since $Sector(c, d, \theta_1)$ is empty and E is the only node in $Sector(c, d, \theta_2)$. In (b) **WEIGHTEDRCOMPASS** sets $n_1 = E$ and $n_2 = G$, **BEST2GREEDY** sets $n_1 = F$ and $n_2 = G$ whereas **BEST2COMPASS** sets $n_1 = E$ and $n_2 = F$.

1.2 Empirical results

We have implemented, in C++, all the algorithms discussed in Section 1.1. With the exceptions of **GREEDY** and **COMPASS**, all of the algorithms considered here are randomized. To evaluate the performance of these randomized algorithms we will consider their packet delivery rates and stretch factors. We first describe our simulation environment, including the choice of algorithm-specific parameters, and then describe and interpret our results, comparing our algorithms with previous work, as well as with each other.

1.2.1 Simulation Environment

In the simulation experiments, a set S of n points (where $n \in \{75, 100, 125, 150\}$) is randomly generated on a square of 100m by 100m. For the transmission range of nodes, we use 15m or 18m (experiments showed that with lower transmission radii, the graph was too often disconnected, and with higher transmission radii, the generated graphs were so dense that the delivery rate of all algorithms approached 100%). After generating $UDG(S)$, a source and destination node are randomly chosen. If there is no path from s to d in $UDG(S)$, the graph is discarded, otherwise, all routing algorithms are applied

in turn. Clearly, an algorithm succeeds if a path to the destination is discovered. The deterministic algorithms are deemed to fail if they enter a loop, while the randomized algorithms are considered to fail when the number of hops in the path computed so far exceeds the number of nodes in the graph. To compute the packet delivery rate, this process is repeated with 100 random graphs and the percentage of successful deliveries determined. To compute an average packet delivery rate, the packet delivery rate is determined 100 times and an average taken. Additionally, over the 100×100 runs, the average hop stretch factor is computed.

Several of the randomized routing algorithms use experimentally optimized parameters. In particular, FARINSECTOR depends on the parameters θ and α . A smaller value of θ clearly means a smaller number of eligible neighbors of c . Similarly, a smaller value of α means the area of the periphery is smaller compared to the area of the sector, which changes the number of eligible neighbors that are closer to the destination. We use $\theta = \pi/3$ and $\alpha = 0.6$ after comparing the performance of the algorithm with θ varying from $\pi/3$ to π and α from 0.1 to 0.9. For GREEDYINSECTOR, $\theta = \pi/3$ was found experimentally to give the best performance. Also, RANDOMINSECTOR depends on the size of the nested sectors, and the experimentally determined optimum values are $\pi/6$ and $\pi/3$, respectively.

1.2.2 Discussion of Results

Detailed simulation results for all the routing algorithms, along with the associated standard deviations, are given in Tables 1 and 2 for the case when the transmission radius is 15¹. In particular, we are interested in the performance of our proposed randomized routing algorithms with the previously published routing algorithms GREEDY, COMPASS and RCOMPASS.

BEST2GREEDY and BEST2COMPASS are straightforward randomizations of the greedy and compass strategies, where the next node is chosen randomly from the top two candidates according to the respective heuristics. BEST2GREEDY is the worst of the randomized strategies in terms of delivery rate, but the best in terms of the stretch factor for both values of transmission radius. BEST2COMPASS has the second-best stretch factor and the second-worst delivery rate.

The sector-based algorithms improve significantly on the delivery rate of GREEDY, COMPASS and the above two strategies. The key idea behind the sector-based algorithms is to restrict the extent to which we stray away from the direction of the destination while keeping some flexibility regarding exactly which neighbor to forward to next. The three algorithms differ in the

¹The trend of the results is the same when the transmission radius is 18 so the details are omitted here.

Table 1. Average packet delivery rate and standard deviation, σ , in terms of percentages, for transmission radius $r = 15\text{m}$.

Algorithms	$n = 75$		$n = 100$		$n = 125$		$n = 150$	
	Aver.	σ	Aver.	σ	Aver.	σ	Aver.	σ
GREEDY	70.67	4.61	75.92	3.66	85.70	3.47	92.37	2.53
COMPASS	72.10	4.42	77.14	4.07	87.83	3.16	93.59	2.37
RCOMPASS	86.93	3.07	92.61	2.66	97.97	1.37	99.51	0.64
FARINSECTOR	80.60	3.92	85.51	3.44	94.14	2.37	97.60	1.58
GREEDYINSECTOR	80.73	3.93	86.16	3.50	94.33	2.13	97.82	1.68
RANDOMINSECTOR	80.51	3.87	85.90	3.63	94.25	2.32	97.80	1.61
BEST2GREEDY	78.97	4.08	82.64	3.81	90.67	2.97	95.13	1.98
BEST2COMPASS	80.94	3.84	85.18	3.42	91.92	2.70	96.51	1.77
WEIGHTEDRCOMPASS	87.53	3.08	91.97	2.71	97.09	1.65	99.10	0.95

Table 2. Average stretch factor and standard deviation, σ , for transmission radius $r = 15\text{m}$.

Algorithms	$n = 75$		$n = 100$		$n = 125$		$n = 150$	
	Aver.	σ	Aver.	σ	Aver.	σ	Aver.	σ
GREEDY	1.01	0.05	1.02	0.06	1.02	0.07	1.02	0.07
COMPASS	1.06	0.11	1.08	0.13	1.10	0.14	1.10	0.14
RCOMPASS	2.72	2.47	2.69	2.47	2.34	2.27	1.95	1.75
FARINSECTOR	2.03	1.98	2.02	2.18	1.83	1.99	1.62	1.75
GREEDYINSECTOR	2.21	2.17	2.28	2.45	2.03	2.25	1.75	2.05
RANDOMINSECTOR	2.31	2.22	2.39	2.49	2.18	2.39	1.89	2.08
BEST2GREEDY	1.79	1.26	1.60	1.07	1.41	0.83	1.28	0.59
BEST2COMPASS	1.86	1.43	1.68	1.18	1.48	0.92	1.35	0.65
WEIGHTEDRCOMPASS	1.73	1.44	1.74	1.58	1.55	1.35	1.42	1.22

choice of the neighbor within the sector². GREEDYINSECTOR chooses the neighbor closest to the destination from within the sector, while FARINSECTOR chooses a node randomly from among the nodes closest to the destination ($Periphery(c, d, \theta, \alpha)$), and RANDOMINSECTOR chooses randomly from among nodes closest to the direction of the destination. FARINSECTOR also allows for the choice of next neighbor from $Wing(c, d, \theta, \alpha)$, if the sector is empty, on the grounds that it may be worthwhile straying outside the sector provided that we cover a lot of distance to the destination. While the delivery rates of all sector-based algorithms are almost identical for both values of transmission radius, FARINSECTOR appears to have a slight edge over the

²For simplicity, we use “the sector” to refer to $Sector(c, d, \theta)$ in this paragraph.

other two in terms of stretch factor. However FARINSECTOR also is slightly more complicated to implement than the other two.

The RCOMPASS algorithm proposed in [Bose and Morin, 1999] is the best in terms of delivery rate but is significantly worse than all other randomized algorithms in terms of the stretch factor. Recall that RCOMPASS chooses the next node with equal probability among the two nodes making the smallest angles in clockwise and counterclockwise directions from the the \overline{cd} line. This means that if there is a problem with the route on one side of the line, there is a good chance of avoiding it, which gives a good delivery rate. However, by potentially moving far away from the direction of the destination, we may end up increasing the path length. In contrast, WEIGHTEDRCOMPASS still allows for moving on both sides of the \overline{cd} line, but by weighting the probability of the choice of each neighbor x on the angle $\angle xcd$, it reduces the chances of moving too far away from the right direction and therefore taking too long a path. As a result, WEIGHTEDRCOMPASS achieves the same delivery rate as RCOMPASS but significantly improves on its stretch factor.

In summary, the experimental results show that all the randomized algorithms perform better than the deterministic algorithms in terms of average packet delivery rate. In particular, WEIGHTEDRCOMPASS and RCOMPASS outperform the other randomized algorithms. However, in terms of average hop count stretch factors, the deterministic algorithms outperform all the randomized algorithms, with GREEDY having the lowest stretch factors. The fact that the randomized algorithms are able to continue and find alternative and possibly longer routes even when encountering the same node again in a path accounts for both the higher delivery rates and the higher stretch factors as compared to the deterministic algorithms. However, it is interesting to note that our best algorithm WEIGHTEDRCOMPASS has not only the best delivery rate but also one of the best values of stretch factor. In fact, when computing the stretch factor over only those instances where GREEDY or COMPASS succeed as well, the stretch factor of WEIGHTEDCOMPASS is even better than the values shown in Table 2.

1.3 Summary

In this paper, we proposed six new randomized position-based strategies for routing in mobile ad hoc networks. We compared the performance of our algorithms with the previously proposed GREEDY, COMPASS, and RCOMPASS algorithms. Our simulation results demonstrate that randomization yields a definite improvement over the deterministic algorithms in terms of the delivery rate. Conversely, the best stretch factors are achieved by the deterministic algorithms. One of our new algorithms WEIGHTEDRCOMPASS achieves the best delivery rate and has one of the best stretch factors.

We are currently working on improving further the the performance of WEIGHTEDRCOMPASS by varying the choice of the qualifying candidates for the next node as well as the weighting function. We have recently completed a study of a class of such randomized algorithms [Fevens et al., 2004]. We are also interested in the performance of WEIGHTEDRCOMPASS on other types of graphs, such as planar graphs and triangulations. Using weighted probabilities to choose the next neighbor in a sector-based algorithm is another interesting avenue for research. Finally we are interested in characterizing the kinds of networks for which our randomized algorithms deliver the packet with probability 1.

Acknowledgments

The authors wish to thank the anonymous referees for their helpful comments. The research of T. Fevens and L. Narayanan is supported in part by NSERC.

References

- Barriere, L., Fraignaud, P., Narayanan, L., and Opatrny, J. (2001). Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. In *Proc. of 5th ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*.
- Basagni, S., Chlamtac, I., Syrotiuk, V.R., and Woodward, B.A. (1998). A distance routing effect algorithm for mobility (DREAM). In *4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98)*, pages 76–84.
- Bose, P. and Morin, P. (1999). Online routing in triangulations. In *10th Annual International Symposium on Algorithms and Computation (ISAAC '99)*, pages 113–122.
- Bose, P., Morin, P., Stojmenovic, I., and J.Urrutia (1999). Routing with guaranteed delivery in ad hoc wireless networks. In *3rd Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM)*.
- Boukerche, A. and Das, S.K. (2003). Congestion control performance of R-DSDV protocol in multihop wireless ad hoc networks. *ACM/Kluwer Journal on Wireless Networks*, 9(3):261–270.
- Boukerche, A., Das, S.K., and Fabbri, A. (2001). Analysis of randomized congestion control scheme with DSDV routing in ad-hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995.
- Chlamtac, I., Conti, M., and Liu, J. (2003). Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Network Journal*, 1(1):13–64.
- Choi, W. and Das, S.K. (2002). Performance of randomized destination-sequence distance vector (R-DSDV) protocol for congestion control in ad hoc wireless network routing. In *Proc. of Applied Telecommunications Symposium - Wireless Track (ATS)*.
- Fevens, T., Haque, I.T., and Narayanan, L. (2004). A class of randomized routing algorithms in mobile ad hoc networks. Submitted for publication.
- Finn, G.G. (1987). Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISU/RR-87-180, USC ISI, Marina del Ray, CA.

- Giordano, S., Stojmenovic, I., and Blazevic, Lj. (2003). Position based routing algorithms for ad hoc networks: A taxonomy. In Cheng, X., Huang, X., and Du, D.Z., editors, *Ad Hoc Wireless Networking*. Kluwer.
- Johnson, D., Maltz, D., Hu, Y-C., and Jetcheva, J. (2002). The dynamic source routing protocol for mobile ad hoc networks. Technical Report Internet Draft, draft-ietf-manet-dsr-07.txt (work in progress), IETF.
- Karp, B. and Kung, H. (2000). GPSR: greedy perimeter stateless routing for wireless networks. In *Proc. of 6th ACM Conference on Mobile Computing and Networking (Mobicom '00)*.
- Ko, Y.B. and Vaidya, N.H. (1998). Location-aided routing (LAR) in mobile ad hoc networks. In *4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98)*, pages 66–75.
- Kranakis, E., Singh, H., and Urrutia, J. (1999). Compass routing on geometric networks. In *Canadian Conference on Computational Geometry (CCCG '99)*, pages 51–54.
- Liao, W.H., Tseng, Y.C., and Sheu, J.P. (2001). GRID: A fully location-aware routing protocols for mobile ad hoc networks. *Telecomm. Systems*, 18(1):37–60.
- Perkins, C.E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244.
- Perkins, C.E. and Royer, E. (1999). Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA.
- Stojmenovic, I. and Lin, X. (2001). Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032.