Randomized $3D$ Position-based Routing Algorithms for Ad-hoc Networks

A.E. Abdallah, T. Fevens and J. Opatrny
Department of Computer Science and Software Engineering
Concordia University
Montréal, QC, Canada, H3G 1M8
Email: {ae_abdal,fevens,opatrny}@cse.concordia.ca

Abstract

In position-based routing algorithms for ad-hoc networks, the nodes use the geographical information to make the routing decisions. Recent research in this field primarily addresses such routing algorithms in two dimensional space $(2D)$. However, in real applications, nodes may be distributed in $3D$ space. In this paper we extend previous randomized routing algorithms from $2D$ space to $3D$ space, and we propose two new position-based routing algorithms that combine randomized AB3D routing algorithms with a deterministic CFace (coordinate face) algorithm. The first algorithm AB3D-CFace(1)-AB3D starts with AB3D routing algorithm until a local minimum is reached. The algorithm then switches to CFace routing using one projected coordinate. If CFace(1) enters a loop, the algorithm switches back to AB3D. The second algorithm AB3D-CFace(3) starts with AB3D, until a local minimum is reached. The algorithm then permanently switches to CFace routing using three projected coordinates, in order. We evaluate our mechanisms and compare them with the current routing algorithms. The simulation results show the significant improvement in delivery rate over pure AB3D randomized routing (97% compared to 70%) and reduction in path dilation (up to 50%) over pure CFace algorithm.

1. Introduction

Mobile ad-hoc networks (MANETs) consist of a collection of wireless mobile hosts that can communicate with each other without a fixed infrastructure. A node in the network can communicate directly only with its neighbors (the nodes within its transmission range). To communicate with nodes outside its transmission range, multihop routing is used utilizing intermediate communicating nodes. Since mobile ad-hoc networks may change their topology frequently and because of the resource constraints, routing in such networks is difficult. In the past decade, several adaptive routing protocols for ad-hoc networks have been proposed to address the multihop routing problem in ad-hoc networks. Each is based on different assumptions and concepts. In general, these protocols can be classified in two basic types: *topology based routing* and *position-based routing*.

Topology based routing protocols define an explicit route among nodes using the information about the links that exist in the network.

Position-based routing [1, 2, 3, 4, 5, 6, 10] or online routing [9, 16] algorithms limit the huge bandwidth required by topology based routing. The host forwards the message based on its position, the position of the destination, and the position of the hosts to which it can communicate directly. In one class of position-based routing, progress-based algorithms, the current node forwards the packet in every step to exactly one of its neighbors, which is chosen according to some heuristic such as Greedy [5] or Compass [4]. However, progress-based routing methods suffer from the so-called local minimum phenomenon, in which a packet may get stuck at a node that does not have a neighbor that makes a progress to the destination, even though the source and destination are connected in the network. Many algorithms attempt to deal with this problem. Bose *et al.* [9] described a routing algorithm which guarantees the delivery of the message in a MANET under a geometric planar graph. This algorithm, called *Face routing*, uses the right hand rule to propagate the packet along the interior of the faces of the planar graph which are intersected by the line segment connecting the source to the destination. A combination between Greedy routing and Face routing has been proposed in [9], GFG (Greedy-Face-Greedy), and in [11], GPSR (Greedy Perimeter Stateless Routing). Initially, these algorithms make greedy forwarding decisions. If the packet reaches a region where progress to the destination by greedy forwarding is impossible, the algorithm enter into recovery mode by switching to face routing. Once the packet reaches a node closer to the destination than that node where greedy forwarding previously failed for that packet, the algorithm switches back to greedy forwarding again. Face routing algorithms require a planar subgraph to guarantee the deliv-

1-4244-0499-1/06/$20.00 ©2006 IEEE

ery of the packet. In $3D$ geometric graphs there may not even exist faces, thus face routing algorithm can not easily extended to $3D$ space. Most of the previous position-based routing algorithms do not take into account the realistic $3D$ network model. Kao *et al.* [8] propose a heuristic for face routing in $3D$ space called projective face routing algorithm. Recently in [14], the authors provide a combination between partial flooding and position-based routing algorithms for $3D$ model.

In this paper, firstly, we extend the randomized localized routing algorithm AB (Above/Below) algorithm [6] from $2D$ space to $3D$ space. Secondly, we improve a new version of projective-face routing algorithms (CFace(i)) for realistic $3D$ model. Finally, we propose two new 3D-Position-based routing algorithms which combine randomized AB3D routing algorithms with CFace(i) routing.

The rest of the paper is organized as follows. In the next section we define the network model and survey previous work. In Section 3 we give a detailed description of the new routing algorithms. We present some experimental results to demonstrate the much improved performance of the proposed methods in comparison with existing techniques in Section 4. Section 5 summarizes our results.

2 MODEL AND PRELIMINARIES

2.1 Network Model

We assume that the set of n wireless hosts is represented as a point set S in the $3D$ space. All network hosts have the same communication range of r, which represented as a sphere volume of radius r. We define the $d(u, v)$ as the Euclidean distance between the points u and v, $d(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_y - v_y)^2}$. Two nodes are connected by an edge if the Euclidean distance between them is at most r. The resulting graph is called a unit disk graph (UDG). Position based routing protocols assumes that the node knows: (1) The coordinates (x, y, z) of its position, which can be easily satisfied using GPS; (2) the location of its neighbors using a periodical exchange of messages; and (3) the location of the destination, e.g., by using a location service [1].

The position-based routing task is to find a path from the source node s to the destination node d. It uses the local information at each node to determine how to route the packet. The algorithm is called randomized position-based, if any next step taken by a packet is chosen randomly.

We are interested in the following performance measures for routing algorithms: the *delivery rate*, which is the percentage of times that the algorithm succeeds in delivering its packet, and the *path dilation rate*, the average ratio of the length of the path returned by the algorithm to the length of the shortest path in the UDG. The path dilation is defined with respect to the shortest path sp in the UDG since, even when routing on a sub-graph of the UDG, sp is equal in length to, or shorter than, any shortest path that may be discovered in a sub-graph. Here the length of the path is taken to mean the number of edges in the path.

2.2 Geometric Sub-graphs

In addition to the UDG, in this paper we consider the behavior of the routing algorithms over two sub-graphs of the UDG. Firstly, the Gabriel Graph (GG) [12] of a graph G, denoted GG(G), is defined as follows. Given any two adjacent nodes u and v in G, the edge (u, v) belongs to GG(G) if, and only if, no other node $w \in G$ is located in the sphere of diameter $d(u, v)$ containing (u, v). Secondly, the Relative Neighborhood Graph (RNG) [13] of a graph G, denoted RNG(G), is defined as follows: an edge (u, v) exists in RNG(G) between the points u and v in G if no other point w in G is inside the lens formed by the intersection of the two spheres centred on u and v with radii equal to the transmission range of each node.

2.3 Related Routing Algorithms

Here we describe the related deterministic routing algorithms:

***Greedy Routing*[5]:** the current node forwards the packet to the neighbor that minimizes the remaining distance to the destination. The same procedure is repeated until the destination node is reached.

***Compass Routing*[4]:** the current node forwards the packet to the neighbor node that minimizes the angle between the current node, next node and the destination node. Fig. 1 gives examples of the neighbors chosen by these progress-based algorithms.

CFace (face coordinate) Routing: based on $2D$ face routing [9], which starts by extracting the GG from the UDG. Then the packets are routed over the faces of GG which are intersected by the line between the source and the destination, sd, using the right hand rule. That is, the boundary of f is traversed in the counterclockwise direction, unless the current edge crosses sd at an intersection point closer to the destination than any previously discovered intersection point. In this case, the algorithm switches to the next face sharing the edge and continues with the right hand rule. This algorithm is repeated until the node arrives to the destination. Face routing algorithm guarantees the delivery only over $2D$ planar geometric graph.

Since in $3D$ faces may not lie in a plane, let alone a plane containing sd, or may not even exist, this algorithm is not directly applicable for $3D$ graphs. The authors in [8] propose a heuristic using the projective approach to adapt face routing to $3D$ graphs. Their algorithm may be summarized as follows: the points are first projected onto a plane containing the line sd. The face routing is performed on this projected graph. If the routing fails, the points are then projected onto the second plane, which is orthogonal to the first

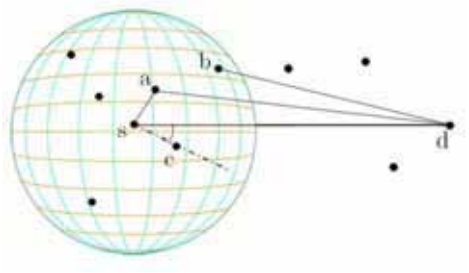

Figure 1. To route from s to d. Greedy chooses b and Compass chooses c as the next node.

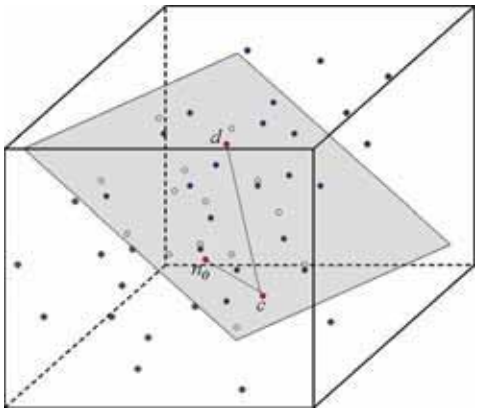

Figure 2. The plane that defined at each step in AB3D random algorithm.

plane and also contains the line sd. The face routing is again performed. Here, we define an $3D$ extension of face routing, called *CFace(3)*; Algorithm 1 gives a description of the *CFace(3)* routing algorithm.

Algorithm 1 CFace(3) Algorithm

Input *a graph $G(V, E)$, where V is the set of nodes, and E is the set of Edges, source node s, the destination node d.*

Output*: return false if the packet fails to reach the destination.*

1- Project all nodes in xy plane // node z=0
2- Call Face routing on the projected graph
3- If the packet does not arrive // loop occurs in 2
4- Project the original coordinate of all nodes in xz plane
5- Call Face Routing starting from the original source
6- If the packet does not arrive// loop occurs in 5
7- Project the original coordinate of all nodes in yz plane
8- Call Face Routing starting from the original source
9- If the message does not arrive
10- Return fail
11- Else return success

Randomized AB3D(R,S) algorithms: here we describe our extension of the *AB* algorithms presented in [6]. We assume that the current node is c, d is the destination node, n_0 is the closest point to d from $N(c)$ (n_0 could be chosen according to a compass-based measure, but in our simulations this do not lead to better results) and P is the plane that passes through c, d and n_0. See Fig. 2. Each algorithm has two attributes, which is reflected in our naming convention: *AB3D(R,S)* where R is one of C (as in Compass) or G (Greedy), and S is one of U, A, or D. Each routing algorithm is based on initially determining two candidate neighbors in addition to n_0, one neighbor (n_1) of c from above the plane P, and, similarly, one neighbor (n_2) of c below the plane P. Out of all the possible neighbors from above (below) the plane P, n_1 (n_2) is the one that would be chosen by the R protocol. Which of these three candidate neighbors is actually chosen depends on the symbol for S. If the symbol is U, then the next node x is chosen uniformly at random from n_0, n_1, and n_2. If the symbol is A or D, then the next node x is chosen from n_0, n_1 and n_2 with probability $1 - p_i/(p_0 + p_1 + p_2)$, where if the symbol is A then $p_i = \theta_i = \angle n_i c d$, or if the symbol is D, then $p_i = dis_i = d(n_i, d)$. If one of the nodes is not defined, then the algorithm uses the rest.

3 Hybrid New Algorithms

Projection face routing [8] and our version of coordinate face routing, *CFace(3)*, give a very good delivery rates which reach 95% but with a very high path dilation (around 8). See Tables 1 to 3. In the following, we define our two primary routing algorithms that combine the randomized *AB3D* algorithms with *CFace* algorithm.

3.1 AB3D-CFace(1)-AB3D

Our first hybrid algorithm starts with any one of the six distinct *AB3D(R,S)* algorithms. Once a local threshold is passed in terms of the number of hops and we arrive at a local minimum, the algorithm switches to *CFace(1)*. *CFace(1)* traverses one projective plane, which is randomly one of the xy, yz, or xz planes starting from the local minimum c as the new source node. If the destination is not reached during *CFace(1)* and looping occurs, the algorithm goes back to *AB3D(R,S)* and the count for the local threshold restarts at 0. In this algorithm, the reason for using a local threshold is based on the algorithm in [2], where although the packet reaches a local minimum it can still be forwarded to the node with the least backward (negative)

progress. In addition to the local threshold, the algorithm uses a global threshold to drop the packet if the total number of hops exceeds the global threshold. *AB3D-CFace(1)-AB3D* is given in Algorithm 2.

Algorithm 2 AB3D(R,S)-CFace(1)-AB3D(R,S)

Input *a graph $G(V, E)$, where V is the set of nodes, and E is the set of Edges, source node s, the destination node d.*

Output*: return true if the destination is reached or false other wise.*

repeat
  Call AB3D routing
  if the packet arrive return success;
  if (local min) and (path length>Local_threshold) **then**

    randomly choose one of the planes xy, yz or xz
    Project all nodes on the selected plane
    Do Face routing starting from current node c
    if the packet reach the destination return success;
    else
      Continue // back to AB3D routing
  end if
until Global_threshold is reached

3.2 AB3D-CFace(3)

The main difference between this algorithm and *AB3D-CFace(1)-AB3D* is that instead of going back to *AB3D* if the first projective plane fails, it tries other projective planes. The algorithms also start with *AB3D(R,S)*. If a threshold is reached together with a local minimum, the algorithm switches to *CFace* using the xy plane. Again if a loop happens the algorithm goes to yz plane. Finally, if yz plane fails, the algorithm switches to xz plane. *AB3D-CFace(3)* is summarized in Algorithm 3.

The key advantage of these hybrid algorithms is the improvement in performance over randomized *AB3D* algorithms and *CFace* algorithm, with a decrease of the large path dilation caused by *CFace* routing algorithm. In the next section, we show the simulation results which illustrate the advantages of our algorithms.

4 Simulation Results

In this section we describe our simulation environment, and then we show and interpret our results, comparing our algorithms with previous published deterministic routing algorithms Greedy, Compass and *CFace(3)*.

4.1 Simulation Environment

In the simulation experiments, a set S of n points (where $n \in \{65, 75, 95, 105\}$) is randomly generated in a cube of side length 100. The maximum transmission radius of each

Algorithm 3 AB3D(R,S)-CFace(3)

Input *a graph $G(V, E)$, where V is the set of nodes, and E is the set of edges, source node s, the destination node d.*

Output*: return true if the destination is reached or false other wise.*

1- Call AB3D routing
2-**if** the packet reach the destination return success
3-**if** (local min) and (path length $>$ Local_threshold)
4-Call CFace(3) algorithm

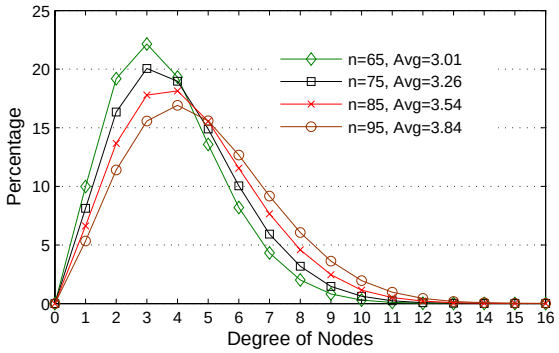

Figure 3. The histogram of average node degrees of the LCC in $10,000$ generated UDG.

host is set to 25. We set the global_threshold to $2n$ and the local_threshold to n. We first calculate all connected components in the graph. Then select the largest connected component (LCC) among all the connected components to perform the routing algorithms. The source and destination nodes are then randomly picked from LCC. It is suggested in [7] to consider simulations with node density per unit disk of around 5 in $2D$ environment, which would correspond to the graph with average node degrees of around 4. Fig. 3 illustrates a histogram of the node degrees for the graphs with the chosen simulation values n. Graphs with $n > 75$ are closest to the node density of interest. An algorithm succeeds if a path to the destination is found. To compute the packet delivery rate, this process is repeated with 100 random graphs and the percentage of successful delivers determined. To compute the average packet delivery rate, the packet delivery rate is determined 100 times and an average taken. Additionally, out of the $10,000$ runs used to compute the average packet delivery rate, the path dilation is computed. Since we have more than fifteen different combinations of the algorithms, it's difficult to show all of these combinations, thus we show the some algorithms which gave the most interesting results. We provide three separate analyzes. In each of these analyzes we study the delivery rate and path dilation versus the node density, the

Table 1. Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in UDG.

	$n = 75$				$n = 95$			
Algorithms	D	σ	P	σ	D	σ	P	σ
COMPASS	63.60	*5.06*	1.03	*0.10*	65.74	*4.47*	1.05	*0.11*
GREEDY	62.56	*5.25*	1.02	*0.07*	64.22	*4.36*	1.03	*0.09*
CFACE(3)	94.53	*2.61*	9.87	*13.00*	95.56	*1.95*	14.59	*19.79*
AB3D(C,D)	80.37	*4.34*	3.02	*2.64*	85.68	*3.65*	3.22	*3.12*
AB3D(C,A)	76.91	*4.06*	3.41	*3.14*	84.25	*3.32*	3.70	*3.59*
AB3D(G,D)	79.90	*3.89*	2.99	*2.70*	84.94	*3.53*	3.20	*3.21*
AB3D(G,A)	76.91	*3.83*	3.35	*3.06*	83.68	*3.65*	3.55	*3.51*
AB3D(C,D):CFACE(3)	97.76	*1.63*	6.25	*9.49*	98.04	*1.21*	6.91	*12.46*
AB3D(C,A):CFACE(3)	97.49	*1.65*	7.33	*11.12*	97.85	*1.40*	7.58	*12.74*
AB3D(G,D):CFACE(3)	97.54	*1.42*	6.37	*9.82*	97.92	*1.30*	6.98	*12.82*
AB3D(G,A):CFACE(3)	97.28	*1.61*	7.27	*10.43*	98.04	*1.42*	7.82	*13.68*
AB3D(C,D):CFACE(1):AB3D(C,D)	92.90	*2.77*	4.97	*6.84*	94.53	*2.09*	5.35	*8.30*
AB3D(C,A):CFACE(1):AB3D(C,A)	92.61	*2.82*	5.84	*7.44*	94.59	*2.11*	6.01	*8.75*
AB3D(G,D):CFACE(1):AB3D(G,D)	92.71	*2.74*	5.04	*6.76*	94.15	*2.31*	5.46	*8.64*
AB3D(G,A):CFACE(1):AB3D(G,A)	92.13	*2.96*	5.89	*7.77*	94.03	*2.63*	6.09	*9.01*

threshold and the subgraph type. In all graphs we just show the best algorithm in each proposed class.

4.2 Observed Result

We present the comparison between different groups of algorithms in terms of packet delivery rate and path dilation in Table 1, 2 and 3. For comparison purposes, we will focus on $n = 75$, and $n = 95$. It is immediately evident form the result given in Table 1 that deterministic progress-based algorithms (Greedy and Compass) have the lowest delivery rate (less than 65%) which yields the low path dilation because the packets that fail to arrive to the destination is not counted in the path dilation. The randomized algorithm comes after that with a delivery rate over 75% and path dilation around 3. The delivery rate of *CFace(3)* jumps to 94%, but this algorithm has by far the worst path dilation (around 9 for $n = 75$), our new algorithm *AB3D-CFace-AB3D* almost reaches the delivery rate of *CFace(3)*, but it decreases the path dilation by 50%. The best delivery rate with over 97% is found in *AB3D-CFace(3)* and also has a lower path dilation than *CFace(3)* algorithm. We find, from Tables 1 to 3, that the algorithms based on *AB3D(C,D)* and *AB3D(G,D)* have the best performance in terms of deliver rate and path dilation.

Effect of using a sub-graph of UDG for routing In Fig. 4 we can see the influence of the sub graph over the delivery rate. In Fig. 5 we show the influence of the sub-graph over the path dilation. First, in terms of delivery rate, as expected, the deterministic and randomized *AB3D* algorithms delivery rate decreased over both GG and RNG graphs, due to potentially fewer neighbours available in the progress direction. Our new hybrid algorithms have roughly the same best performance on all three graphs. *CFace(3)* has the best delivery rate on Gabriel sub-graphs, followed by RNG and then on UDG. This can be explained also by considering the number of edges; fewer edges implies fewer crossing edges in the projected face which means less chance for the packet to enter a loop. In terms of the path dilation, the algorithms depend on AB3D have the best path dilation over UDG graph, but in both Gabriel sub-graph and relative neighborhood graph the path dilation is increased for the same reason mentioned above. On these subgraphs, *CFace(3)* algorithm the path dilation has been decreased obviously because there is less chance for the packet to enter a loop, which means no new projective planes.

Effect of the network density In Fig. 6 we illustrate the effect of the number of nodes (network density) on the performance of the algorithms. In all the algorithms, as the number of nodes increased the delivery rate also increased. This is because most of the proposed algorithms depend on the randomized AB3D which means there is a better chance for a good route to the destination. When n equal to 65 the delivery rate does not follow the above trend, because LCC is very small, which implies that the path between any pair of nodes is relatively short. Fig. 7 shows how the path dilation is effected by the network density. Because increasing the number of nodes implies increasing the possibility for long detours being discovered during randomized routing, the path dilation is increased. For *CFace(3)* there is an increase in the number of crossing edges, which means greater chance for entering into loops, therefore increasing the probablity of having to project to the second and third plane.

Table 2. Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in GG.

	$n = 75$				$n = 95$			
Algorithms	D	σ	P	σ	D	σ	P	σ
COMPASS	60.95	*4.81*	1.02	*0.07*	62.53	*4.67*	1.03	*0.08*
GREEDY	60.43	*5.18*	1.02	*0.06*	62.47	*4.47*	1.03	*0.08*
CFACE(3)	95.03	*2.30*	8.73	*10.55*	94.83	*2.24*	11.58	*14.33*
AB3D(C,D)	79.96	*3.78*	3.60	*3.17*	84.27	*3.34*	3.86	*3.36*
AB3D(C,A)	75.39	*4.57*	4.20	*3.99*	81.93	*4.19*	4.56	*4.07*
AB3D(G,D)	79.60	*4.26*	3.59	*3.19*	84.38	*3.86*	3.86	*3.41*
AB3D(G,A)	75.44	*4.76*	4.18	*3.92*	81.10	*4.15*	4.48	*4.04*
AB3D(C,D):CFACE(3)	97.80	*1.59*	6.51	*8.14*	98.05	*1.22*	6.83	*9.49*
AB3D(C,A):CFACE(3)	97.47	*1.59*	7.93	*9.48*	97.64	*1.62*	8.11	*10.96*
AB3D(G,D):CFACE(3)	97.67	*1.38*	6.51	*8.24*	97.90	*1.45*	6.80	*9.34*
AB3D(G,A):CFACE(3)	97.46	*1.59*	7.84	*9.52*	97.69	*1.65*	8.27	*10.84*
AB3D(C,D):CFACE(1):AB3D(C,D)	93.70	*2.56*	5.56	*6.30*	94.55	*2.20*	5.74	*7.08*
AB3D(C,A):CFACE(1):AB3D(C,A)	92.77	*2.76*	6.90	*8.08*	94.52	*2.43*	6.99	*8.55*
AB3D(G,D):CFACE(1):AB3D(G,D)	93.12	*2.46*	5.53	*6.41*	94.52	*2.22*	5.93	*7.45*
AB3D(G,A):CFACE(1):AB3D(G,A)	92.39	*2.82*	6.85	*7.64*	94.16	*2.43*	6.90	*8.40*

Effect of the threshold Figs. 8 and Fig. 9 show the effect of varying the threshold value on the average delivery rate and average path dilation of all studied algorithms. We find that when the threshold is set to n the relative behavior of the algorithms is established and the difference between algorithms is clear. The delivery rate of all algorithms can been increased by increasing the threshold, and this is very clear with the *AB3D* algorithm. Since the increasing of the delivery rate means more success delivered packets added to the average path dilation, then the average path dilation is expected to increase. The simulation results also confirm this expectation, with an increase in average path dilation corresponding to an increase in threshold.

5 Conclusion

In this paper, we study ad-hoc routing in $3D$ space which is more realistic than routing in $2D$ space.

We summarize the contribution of this paper:

- We extend *AB* randomized algorithm from $2D$ to $3D$ environment.
- Improve a new version of project-face routing algorithms (*CFace(i)*) which decreases the path dilation to almost 50%.
- We propose two new hybrid routing algorithms, *AB3D-CFace(1)-AB3D* and *AB3D-CFace(3)*, which combine the efficiency of progress-based algorithms with high delivery rate of face routing. our experiments show that these hybrid algorithms ended increase the delivery rate to over 97% while keeping the average dilation if the route much smaller than face routing.
- Present simulation results comparing performance of deterministic, randomized and the new hybrid position-based routing algorithms on UDG and its associated spanning sub graphs in $3D$ space.

In all our simulations, we assumed that the UDG was static. However we expect that the new proposed algorithms, *AB3D:CFace(1):AB3D* and *AB3D-CFace(3)*, would also perform well on dynamic $3D$ unit disk graphs since the randomization component of these algorithms would adjust to reasonably large changes in node positions. This could be part of future work.

References

[1] M. Mauve, J. Widmer and H. Hartenstein,"A survey of position-based routing in mobile ad-hoc networks," IEEE Network Magazine, Vol. 15, No. 6, pp. 30-39, November 2001.

[2] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," IEEE Trans. on Communications, Vol. 32, No. 3, pp. 246-257, March 1984.

[3] L. Barriere, P. Fraigniaud, L. Narayanan and J. Opatrny, "Robust position-based routing in wireless ad hoc networks with irregular transmission ranges," Wireless Communications and Mobile Computing Journal, Vol. 3, No. 2, pp. 141-153, March 2003.

Table 3. Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in RNG.

	$n = 75$				$n = 95$			
Algorithms	D	σ	P	σ	D	σ	P	σ
COMPASS	52.58	*4.44*	1.02	*0.05*	51.76	*4.88*	1.02	*0.06*
GREEDY	52.81	*4.53*	1.01	*0.05*	51.94	*4.79*	1.02	*0.06*
CFACE(3)	94.57	*2.06*	8.30	*9.82*	95.40	*2.03*	10.85	*12.81*
AB3D(C,D)	71.48	*4.15*	4.14	*3.38*	75.52	*4.14*	4.87	*3.96*
AB3D(C,A)	61.12	*4.38*	4.93	*4.48*	65.07	*4.56*	5.67	*4.86*
AB3D(G,D)	71.68	*4.37*	4.13	*3.35*	75.25	*3.68*	4.84	*3.90*
AB3D(G,A)	61.75	*5.01*	4.96	*4.39*	65.82	*5.23*	5.75	*4.86*
AB3D(C,D):CFACE(3)	97.41	*1.54*	7.96	*8.70*	98.02	*1.24*	9.07	*10.27*
AB3D(C,A):CFACE(3)	97.21	*1.53*	10.26	*10.35*	97.32	*1.43*	12.01	*13.03*
AB3D(G,D):CFACE(3)	97.71	*1.59*	7.89	*8.41*	97.71	*1.61*	8.88	*10.12*
AB3D(G,A):CFACE(3)	97.18	*1.66*	10.29	*10.40*	97.47	*1.69*	11.82	*12.32*
AB3D(C,D):CFACE(1):AB3D(C,D)	91.71	*2.74*	6.85	*6.95*	93.24	*2.35*	7.70	*8.09*
AB3D(C,A):CFACE(1):AB3D(C,A)	90.40	*3.10*	9.01	*8.84*	92.24	*2.82*	10.30	*10.05*
AB3D(G,D):CFACE(1):AB3D(G,D)	91.94	*2.82*	6.89	*6.89*	93.67	*2.43*	7.74	*8.06*
AB3D(G,A):CFACE(1):AB3D(G,A)	90.43	*3.48*	8.90	*8.46*	91.86	*2.81*	10.48	*10.54*

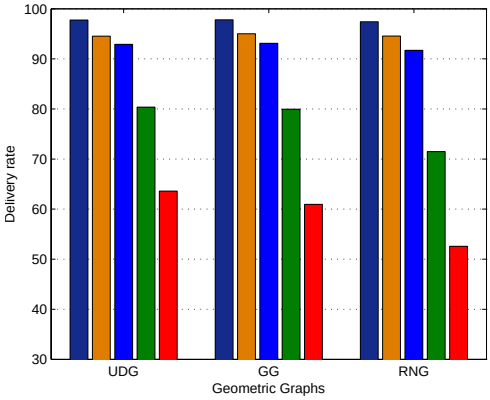

Figure 4. The packet delivery rate at different geometric graphs; see Fig. 5 for legend.

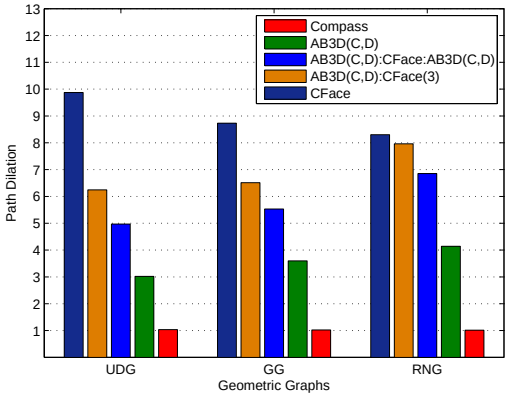

Figure 5. The average path dilation at different geometric graphs.

[4] E. Kranakis, H. Singh and J. Urrutia, "Compass routing on geometric networks," in the Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG '99) , pp. 51-54, Vancouver, August 1999.

[5] G. Fin, "Routing and addressing problems in large metropolitan-scale internetworks," Technical Report ISU/RR-87-180, USC ISI, Marina del Ray, CA 1987.

[6] T. Fevens, I. Haque and L. Narayanan, "A class of randomized routing algorithms in mobile ad hoc networks," in the proceeding of the 1st Algorithms for Wireless and Ad-hoc Networks (A-SWAN), Boston, August 2004.

[7] F. Kuhn, R. Wattenhofer and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," In the proceeding of the 2003 joint workshop on the foundation of mobile computing (DIALM-POMC), pp. 69-78, San Diego, 2003.

[8] G. Kao, T. Fevens and J. Opatrny, "Position-Based Routing on 3D Geometric Graphs in Mobile Ad Hoc Networks", in the proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05), pp.88-91, Windsor, August 2005.

[9] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless net-

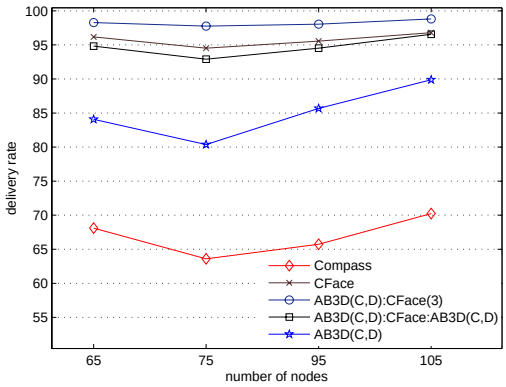

Figure 6. The packet delivery rate at different node densities.

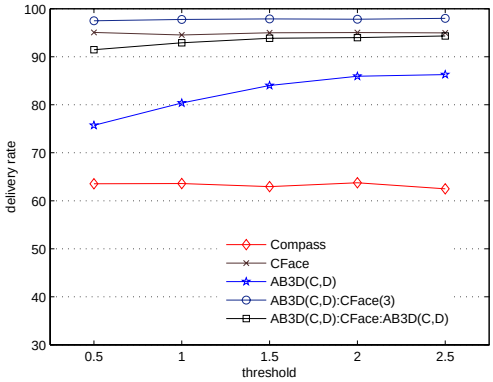

Figure 8. the packet delivery rate at different threshold.

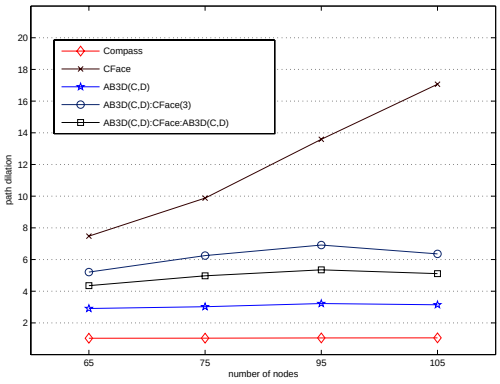

Figure 7. The average path dilation at different node densities.

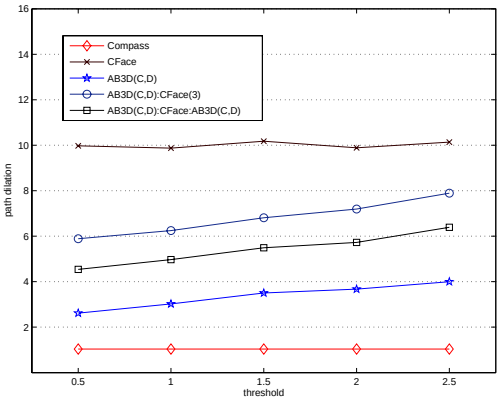

Figure 9. The average path dilation at different threshold.

works", Wireless Networks journal, Vol. 7, No. 6, PP. 609-616, November 2001.

[10] S. Giordano, I. Stojmenovic, and L. Blazevic, "Postion based routing algorithms for ad hoc networks for ad hoc networks: A taxonomy", Ad hoc wireless Networking (ed. X. Cheng, X. Huang, and D.Z. Du), Kluwer, December 2003.

[11] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless net-works," in the Proceeding of 6th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom 2000), pp. 243-254, August 2000.

[12] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," Systematic Zoology, Vol. 18, No. 3, pp. 259-278, September 1969.

[13] J. Jaromczyk and G. Toussaint. "Relative neighborhood graphs and their relatives," in the proceedings of the IEEE, vol. 80, No. 9, pp.1502-1517, September 1992.

[14] A. Abdallah, T. Fevens, and J. Opatrny: "Hybrid Position-Based 3D Routing Algorithms with Partial Flooding", in the proceeding of IEEE Canadian Conference on Electrical and Computer Engineering Networks(CCECE), Ottawa, May 2006.

[15] T. Fevens, A. Abdallah, and B. Bennani, "Randomized AB-Face-AB Routing Algorithms in Mobile Ad Hoc Network", in the Proceedings of the 4th International Conference on AD-HOC Networks and Wireless, pp. 43-56, Mexico, October 2005.

[16] P. Bose and P. Morin, "Online routing in triangulations," in the proceeding of the 10th Annual International Symposium on Algorithms and Computation (ISAAC '99), pp. 113-122, 1999.