Randomized AB-Face-AB Routing Algorithms in Mobile Ad Hoc Networks

Thomas Fevens, Alaa Eddien Abdallah, and Badr Naciri Bennani

Department of Computer Science and Software Engineering,
Concordia University, Montréal, Québec, Canada, H3G 1M8
{fevens,ae_abdal,b_naciri}@cse.concordia.ca

Abstract. One common design for routing protocols in mobile ad hoc networks is to use positioning information. We combine the class of randomized position-based routing strategies called AB (Above-Below) algorithms with face routing to form AB:FACE2:AB routing algorithms, a new class of hybrid routing algorithms in mobile ad hoc networks. Our experiments on unit disk graphs, and their associated Yao sub-graphs and Gabriel sub-graphs, show that the delivery rates of the AB:FACE2:AB algorithms are significantly better than either class of routing algorithms alone when routing is subject to a threshold count beyond which the packet is dropped. The best delivery rates were obtained on the Yao sub-graph. With the appropriate choice of threshold, on non-planar graphs, the delivery rates are equivalent to those of face routing (with no threshold) while, on average, discovering paths to their destinations that are several times shorter.

Keywords : mobile adhoc networks, randomized routing, position-based routing

1 Introduction

A mobile ad hoc network (MANET) is a system of wireless autonomous hosts that can communicate with each other without having any fixed infrastructure. Each node in the network can communicate with all other nodes within its transmission range [1, 2], which we will assume to be a fixed range r for all nodes. If two nodes are not able to communicate directly then a multi-hop routing protocol is needed for the nodes to send packets to each other. The absence of infrastructure in MANETs, together with possible dynamic topology changes and resource constraints, makes routing in these networks a challenging problem.

We are specifically interested in *position-based* routing (also known as online routing [3, 4] or geographic routing [5]). In position-based routing protocols, a node forwards packets based on the location (coordinates in the plane) of itself, its neighbors, and the destination [6]. The position of the nodes can be obtained using GPS, for example. There are numerous ways to use position information in making routing decisions. In one class of algorithms, *progress-based* algorithms as categorized in [7], the algorithm forwards the packet in every step to exactly one of its neighbors, which is chosen according to a specified heuristic, such as minimizing distance to the destination, or moving in a direction nearest to that to the destination.

In this paper we represent a MANET by a unit disk graph, where two nodes are connected if and only if their Euclidean distance is at most r. In GREEDY routing [8, 9], a node forwards the packet to its neighbor which is closest to the destination. COMPASS or *directional* routing [10] moves the packet to a neighboring node such that the angle formed between the current node, next node, and destination is minimized. Both of these progress-based algorithms are known to fail to deliver the packet in certain situations.

To overcome the inability of progress-based routing to always deliver their packets, position information can be used to extract a *planar* sub-graph such that routing can be performed on the faces of this sub-graph, known as *face* routing or perimeter routing [10, 3]. The advantage of this approach is that delivery of packets can always be guaranteed. The original face routing algorithm was called *Compass Routing II* in [10]. An optimization of this algorithm was given by [3] and called *Face-2*. These algorithms will be discussed further in §3. Other improvements to face routing have also been proposed. In [11], *AFR* (Adaptive Face Routing) is proposed where face routing is executed strictly within an area bounded by an adaptively sized ellipse, leading to an algorithm that is asymptotically optimal in that the cost of the algorithm steps are proportional to the square of the cost of the square of the optimal path. In [12], face routing is adapted to guarantee delivery on restricted classes of non-planar graphs.

One drawback of face routing is that, although delivery is guaranteed on a planar graph, the route discovered may be many times longer than the shortest path in the graph. To try to reduce the path length, [13] (also see [14]) and [15] use a hybrid routing algorithm that combines greedy routing with face routing. The idea is to use greedy routing until a local minimum (where all the neighbors of the current node c are further way from the destination node than c) is reached whereupon face routing is used. When the local minimum is bypassed, the algorithm switches back to greedy routing, and so on. This algorithm is termed *GFG* (Greedy-Face-Greedy) by [13] and *GPSR* (Greedy Perimeter Stateless Routing) by [15]. Other variations of this hybrid approach of combining progress based routing with face routing have appeared. *GOAFR* [16] combines greedy routing with a variation of *AFR* called *OAFR*. They show that their algorithm is worst case optimal and average case efficient.

In [17], the authors proposed a class of randomized algorithms called AB algorithms (Above-Below). Essentially, to decide on the next node to which the packet should be forwarded, the AB algorithms pick one neighbor of the current node *above* the line passing through the current node and the destination, and another neighbor *below* this line. The next node is chosen randomly from these two neighbors according to some probability distribution. The exact choice of the neighbors and the probability distribution determine the specific algorithm.

In this paper, we consider hybrid routing algorithms, called AB:FACE2:AB algorithms, that combine AB algorithms with face routing. The goal is to improve the delivery rates of randomized routing algorithms on non-planar graphs, such as the unit disk graph, while still discovering short paths to the destination. As will be shown, the performance of AB:Face2:AB algorithms is on par with that of Face2 routing on the unit disk graph but while choosing shorter paths.

Many routing strategies use a spanning sub-graph of the unit disk graph such that only the edges in the sub-graph are used for routing. These sub-graphs have fewer edges to maintain, and may provide useful properties such as planarity. We studied the performance of all the algorithms on the original unit disk graphs as well as Gabriel graphs [18] and Yao graphs [19] derived from the unit disk graphs. The rational for considering these sub-graphs is discussed at the end of §2. Our results show that the performance of nearly all algorithms in terms of the delivery rate and the length of the path discovered is best on the Yao sub-graphs.

The rest of the paper is organized as follows. The next section gives relevant definitions. Our routing strategies are discussed in §3. Then, §4 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 in §5.

2 Preliminaries

We assume that the set of n wireless nodes is represented as a point set S in the two-dimensional plane. Two nodes are connected by an edge if the Euclidean distance between them is at most r, 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 discover a path in $UDG(S)$ from s to d. At each point of the path, the decision of which node to go to next is based on the local position information of the current node c, $N(c)$, and d. Here, s is termed the source and d the destination. Frequently, we will also refer to the $\overline{cd}$ line passing through c and d. An algorithm is *deterministic* if, when at c, the next node is chosen deterministically from $N(c)$, and is *randomized* if any next step taken by a packet is chosen randomly from $N(c)$ (this description includes hybrid deterministic and randomized algorithms).

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*[1], 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 available to any routing algorithm and *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 hops in the path.

[1] This term was suggested to us by Jaroslav Opatrny to differentiate it from *stretch factor* commonly associated with t-spanner definitions

In this paper, we also consider the behavior of the routing algorithms on several sub-graphs of the unit disk graph. Define $\mathrm{P}(G)$ as a t-spanner of G if the length of the shortest path between any two nodes in $\mathrm{P}(G)$ is not more than t times longer than the shortest path connecting them in G, where t is the stretch factor. Let $G = UDG(S)$ be a unit disk graph. Define $d(x, y)$ to be the Euclidean distance between x and y. Denote the disk centered at the midpoint between the points u and v and with radius $d(u, v)/2$ by $disk(u, v)$. Then the *Gabriel Graph* of G, denoted $GG(G)$, is defined as follows [18]. 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 $disk(u, v)$. It is known that $GG(G)$ is planar and connected if the underlying graph G is a connected unit disk graph [3], Also, $GG(G)$ is a $(4\pi\sqrt{2n-4})/3$-spanner of G [20].

For a geometric graph G, a *Yao Graph* (also called a *Theta Graph* [19]) $YG_k(G)$ with an integer parameter $k \geq 6$ is defined as follows [21]. First, we will define a directed Yao graph, $DYG_k(G)$, for G. At each node u in G, k equally-separated rays originating at u define k cones. In each cone, only the directed edge (u, v) to the nearest neighbor v, if any, is part of $DYG_k(G)$. Ties are broken arbitrarily. For $G = UDG(S)$, the result is a sub-graph that may still have edges that cross. Let $YG_k(G)$ be the undirected graph obtained if the direction of each edge in $DYG_k(G)$ is ignored. The graph $YG_k(G)$ is a $1/(1 - 2sin(\pi/k))$-spanner of G [22].

Our rational for considering Gabriel sub-graphs is that they are planar graphs but their stretch factor is not bounded. Conversely, Yao graphs are not planar (although our experiments show that frequently they have very few crossing edges) and have bounded stretch factors.

3 Routing 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 consider in this paper differ in how x is chosen from among the set $N(c)$. The greedy and compass strategies have already been described in Section 1. In addition, we will use a deterministic progress-based routing algorithm, Ellipsoid-based routing [23], a variation of the Greedy routing, where the current node c forwards the packet to its neighbor n which minimizes $d(c, n) + d(n, d)$.

Next, we describe the class of algorithms called AB algorithms [17]. Each algorithm has two attributes, which is reflected in our naming convention: **AB(R,S)** where **R** is one of **C** (as in Compass), **G** (Greedy), or **E** (Ellipsoid-based), and **S** is one of **U**, **A**, or **D**. Each routing algorithm is based on initially determining two candidate neighbors, one neighbor of c from above the $\overline{cd}$ line, n_1, and, similarly, one neighbor of c below the $\overline{cd}$ line, n_2. Out of all the possible neighbors from above (below) the $\overline{cd}$ line, n_1 (n_2) is the one that would be chosen by the **R** protocol. Which of these two 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_1 and n_2. If the symbol is **A**, then 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, where $\theta_1 = \angle\, dcn_1$ and $\theta_2 = \angle\, n_2cd$. Finally, if the symbol is **D**, then the next node x is chosen from n_1 and n_2 with probability $dis_2/(dis_1+dis_2)$ and $dis_1/(dis_1+dis_2)$, respectively, where $dis_1 = d(n_1, d)$ and $dis_2 = d(n_2, d)$. If either of n_1 or n_2 is not defined, then the other neighbour is chosen by default.

As mentioned in the Introduction, message delivery on a planar graph, if the source and destination are connected, was first guaranteed with face routing [10, 3]. The central idea of face routing is that of the exploration of the interior boundary of a face using the right hand rule analogous to exploring a maze by keeping one's right hand on a wall. Face routing walks around the perimeters of the faces of a planar graph, keeping track of the points where the boundary of a face intersects the $\overline{cd}$ line connecting the source and destination nodes. The primary difference between the original face routing algorithm [10] and *Face-2* [3] is as follows. In original face routing algorithm, the entire perimeter of a face is traversed and at the intersection point p of the $\overline{cd}$ line with the perimeter that is closest to the destination, routing switches to exploring the face sharing p. Thus the face routing progresses toward the destination along the $\overline{cd}$ line. In *Face-2*, this switch occurs at the first intersection point discovered that is closer than any previously discovered intersection points to the destination.

Here, we define our primary routing algorithms which combine the randomized AB algorithms with *Face-2* [3] (our implementation will be called FACE2). Since we have nine distinct AB algorithms, we have correspondingly nine hybrid algorithms which we will term as AB(R,S):FACE2:AB(R,S) algorithms where R is one of C, G, or E, and S is one of U, A, or D as in the description of the AB algorithms previously. Define a *progress halfplane* to be the half plane whose boundary passes through c, contains d, and direction of the normal on the boundary is in the same direction as the vector from c to d. Each algorithm starts as the particular AB(R,S) routing algorithm until the algorithm reaches a node within the graph, such that there are no neighbors in (or on the boundary of) the progress halfplane. At this node, the routing algorithm switches to the FACE2 routing algorithm. Then when FACE2 reaches a vertex where the protocol would begin exploring a different face, the routing algorithm switches back to the AB(R,S). Note that our characterization of the node at which the algorithm switches to FACE2 routing is slightly different than that of a local minimum — this difference is greater the closer a packet gets to the destination whereupon FACE2 routing is resorted to less often than would be the case with a local minimum. As shown in [17], using a uniform probability distribution tended to be out-performed by using a distribution based on angle or distance, so we will only consider AB algorithms using the latter two distributions.

Table 1 shows the classification of the subset of the AB(R,S):FACE2:AB(R,S) algorithms demonstrated in this paper, based on how the AB(R,S) components of each algorithm makes their initial choice of neighbors and the probability distribution used to choose the final neighbor.

Table 1. Classification of AB:FACE2:AB algorithms.

Name of Algorithm	Selection of Neighbors	Probability Distribution
AB(C,A):FACE2:AB(C,A)	Compass	Angle
AB(C,D):FACE2:AB(C,D)	Compass	Distance
AB(G,A):FACE2:AB(G,A)	Greedy	Angle
AB(E,A):FACE2:AB(E,A)	Ellipsoid-based	Angle

4 Empirical results

In the following, GREEDY, COMPASS, ELLIPSOID-BASED, GREEDY:FACE2:GREEDY (our implementation of GFG [13]) and FACE2 are deterministic routing algorithms, and all of the other algorithms are randomized. To evaluate the relative performance of these algorithms we will consider their packet delivery rates and path dilations. We first describe our simulation environment, and then describe and interpret our results, comparing our algorithms with previous work.

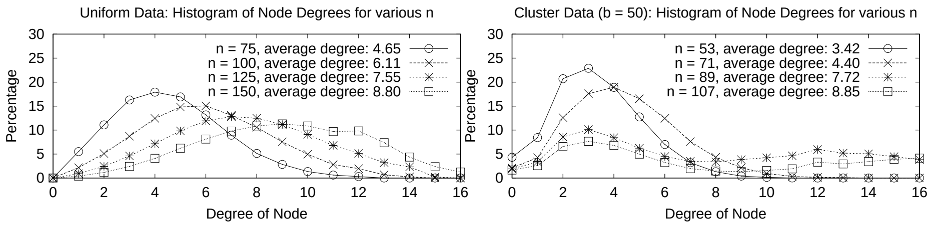
4.1 Simulation Environment

In the simulation experiments, we consider point sets that are uniformly distributed in the plane, as well as point sets that simulate clustering. The parameters like node density will determined experimentally as discussed below.

In the uniform distribution simulation experiments, a set S of n points (where $n \in \{75, 100, 125, 150\}$) is uniformly randomly generated on a square of 100 units by 100 units. For the transmission range of nodes, we use $r = 15$ units. After generating $G = UDG(S)$, we randomly choose a source node s and a destination node d from S. We determine if there exists any path from s to d in $UDG(S)$. If not, the graph is discarded; otherwise, all routing algorithms are applied in turn on G, $GG(G)$, and $YG_8(G)$.

In the clustered simulation experiments, $b = 50$ background points are uniformly randomly generated on a square of 100 units by 100 units. For the transmission range of nodes, again, we use $r = 15$ units. In addition to the fixed number of background points, three clusters, A, B, C, each with j cluster points where $j = 1 + 6 * i$, $i = 0, 1, 2, 3$, are added to the background points to give a set S of $n = b + 3j$ points. The clusters are uniformly randomly generated on a square of 20 units by 20 units and have their centers fixed such that the regions of two clusters A and B overlap (centered at $(11, 25)$ and $(25, 11)$) and the region of the third cluster C (centered at $(89, 89)$) is disjoint from the first two. Note: to facilitate comparison with the uniform distribution simulations results, within the region of a cluster, for the purposes of choosing the source and destination nodes, the additional cluster nodes are considered distinct from the background points. After generating $G = UDG(S)$, a source node s is randomly chosen from the cluster nodes of A and destination node d is randomly chosen from the cluster nodes of C. As with the uniform distribution simulations, we determine if there exists any path from s to d in $UDG(S)$. If not, the graph is discarded; otherwise, all routing algorithms are applied in turn on G, $GG(G)$, and $YG_8(G)$.

Fig. 1. The histograms of average node degrees in 10,000 generated UDGs for the indicated values of n.



In Fig. 1 are shown histograms of the node degrees for the UDGs for the chosen simulation values of n. It is suggested in [24] to consider simulations with node density per unit disk of around 5, which would correspond to graphs with average node degrees of around 4. For uniform data, graphs with $n = 75$ would most closely match this node density of interest, and for clustered data, graphs with $n = 71$ would be of particular interest for comparison. Nonetheless, we shall give experimental results for a variety of values of n to demonstrate relative behaviours with respect to n, but it may be noted in Fig. 1 that when n is larger than 100, that a substantial percentage of nodes have degrees larger than 10 which indicates highly connected graphs.

Clearly, an algorithm succeeds if a path to the destination is found. The progress-based deterministic algorithms and Face2 (without threshold) are deemed to fail if they enter a loop, while Greedy:Face2:Greedy, Face2 (with threshold), and the randomized algorithms are considered to fail when the number of hops in the path computed so far exceeds a threshold. We will use the number of nodes, n, as this threshold (in the next section, we will explore the effect of varying the threshold value). 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, out of the 100×100 runs used to compute the average packet delivery rate, the average path (hop count) dilation of successful paths is computed.

4.2 Discussion of Results

Detailed simulation results for all the routing algorithms, along with the associated standard deviations, are given in Tables 2 to 7 for unit disk graphs and their associated Gabriel graphs, and Yao graphs (with parameter 8). In particular, we are interested in the relative performance of the progress-based deterministic algorithms (Greedy, Compass, and Ellipsoid-based), the deterministic Face2-based algorithms (Greedy:Face2:Greedy, Face2 with and without threshold), the randomized AB algorithms, and the new hybrid AB:Face2:AB algorithms based on the same AB algorithms.

Table 2. Uniform data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in UDG.

	$n = 75$				$n = 100$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):FACE2:AB(C,A)	98.33	1.19	2.44	3.09	96.48	2.04	2.46	3.20
AB(C,D):FACE2:AB(C,D)	97.34	1.58	2.87	3.56	94.67	2.27	3.07	4.21
AB(G,A):FACE2:AB(G,A)	98.00	1.43	2.87	3.26	95.38	2.23	2.80	3.23
AB(E,A):FACE2:AB(E,A)	97.65	1.57	2.96	3.74	95.08	2.22	2.99	3.50
GREEDY:FACE2:GREEDY	93.25	2.12	2.26	3.28	91.28	2.04	2.35	3.18
FACE2 (with threshold)	96.79	1.65	5.16	6.82	97.13	1.40	5.84	5.21
COMPASS	72.31	5.02	1.06	0.12	77.68	4.09	1.08	0.13
GREEDY	70.73	5.16	1.01	0.05	76.43	4.04	1.02	0.07
ELLIPSOID-BASED	56.09	5.21	1.08	0.14	59.16	5.41	1.11	0.16
AB(C,A)	87.74	3.66	2.22	2.99	90.04	2.54	1.91	2.09
AB(C,D)	92.16	2.68	2.53	2.74	93.99	2.52	2.20	1.96
AB(G,A)	86.17	3.80	2.17	2.95	88.54	2.77	1.83	2.04
AB(E,A)	82.78	4.44	2.50	3.70	84.75	3.55	2.13	2.48
FACE2 (without threshold)	98.60	1.01	5.64	7.78	99.23	0.57	6.99	5.93

Performance of Algorithms in Unit Disk Graphs We will now present the comparison between different groups of algorithms in terms of packet delivery rate and path dilation. As discussed earlier, for comparison purposes, we will focus on uniform distributions with $n = 75$ and clustered distributions with $n = 71$.

In UDGs, as presented in Tables 2 and 3, for both uniform and clustered distributions, the FACE2 (without threshold) and, closely in second place, the AB:FACE2:AB algorithms (with a threshold of n) outperform all the other randomized and deterministic algorithms in terms of the packet delivery rate. As we will see later in this section, by increasing the threshold, the delivery rate for the AB:FACE2:AB algorithms can be improved to match that of the FACE2 (without threshold) algorithm. Out of the AB:FACE2:AB algorithms, on the UDG, the algorithms based on the AB(C,A) and AB(G,A) have the best performance in terms of delivery rate. All algorithms tended to perform worse on clustered distributions than uniform distributions. Next in terms of delivery rate, and roughly equivalent in performance, are the FACE2 (with threshold) and GREEDY:FACE2:GREEDY algorithms, following by the randomized AB algorithms. Finally, all the progress-based deterministic algorithms have the worst delivery rates but the best path dilations. Although the best delivery rate is for FACE2 (without threshold), this algorithm has by far the worst path dilations. The results for GREEDY:FACE2:GREEDY (without threshold) is not included in the tables but our simulations show that the path dilations are about the same as those for AB:FACE2:AB algorithms given in Tables 2 and 3, but with lower delivery rates (as we will see in Fig. 2 when we consider the effect of threshold).

The main weakness in the AB algorithms is when the current node is on the edge of a face and there may be no, or a restricted choice of, nodes on

Table 3. Clustered data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in UDG, for $b = 50$.

	$n = 71$				$n = 89$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):Face2:AB(C,A)	96.99	1.94	2.36	2.03	94.14	2.57	2.50	2.78
AB(C,D):Face2:AB(C,D)	94.87	2.54	3.10	2.30	90.06	3.05	2.81	3.29
AB(G,A):Face2:AB(G,A)	96.16	2.21	2.84	2.28	91.27	3.28	3.30	2.95
AB(E,A):Face2:AB(E,A)	95.47	2.16	2.94	2.28	91.12	3.41	3.29	3.14
Greedy:Face2:Greedy	93.23	2.46	2.17	1.66	90.50	3.65	2.44	2.54
Face2 (with threshold)	91.58	2.64	4.01	2.84	85.10	3.80	4.45	3.59
Compass	46.66	5.21	1.09	0.09	46.85	4.86	1.10	0.09
Greedy	44.06	5.29	1.03	0.05	45.02	4.73	1.03	0.05
Ellipsoid-based	23.35	3.88	1.12	0.11	23.91	4.53	1.13	0.11
AB(C,A)	82.90	3.63	2.28	1.81	84.20	3.74	2.42	2.15
AB(C,D)	85.86	3.52	3.39	2.12	87.61	3.54	3.47	2.36
AB(G,A)	80.20	3.87	2.27	1.86	81.13	4.10	2.49	2.40
AB(E,A)	74.20	3.86	2.69	2.20	74.73	4.23	2.96	2.77
Face2 (without threshold)	97.23	0.87	5.14	7.89	96.32	0.54	5.64	5.86

one side of the $\overline{cd}$ line. At this point, the algorithm is essentially deterministic and inherits a greater possibility of looping while trying to discover a path. It is at these critical points within the graph that we find nodes that have no neighbors in the progress halfplane and our AB:Face2:AB routing protocols switch to Face2 routing to bypass these critical points and greatly reduce the possibility of looping. For uniformly distributed UDGs, this in turn leads to a greater percentage of path completions within the preset threshold for number of hops, with path dilations about 150% longer than optimal. Similar behavior is observed for clustered distributions. The only routing algorithm with a larger delivery rate is Face2 (without threshold), but with path dilations that are 415–460% longer than optimal on average.

Effect of Type of Sub-graph The trend of the results in Tables 4 to 7 is similar within each type of graph studied. First, compare the performance of the algorithms between the different types of graphs in terms of delivery rates. In general, most algorithms have the best delivery rates in the Yao sub-graphs for both uniform and clustered distributions. The progress-based and AB algorithms, for uniform and clustered distributions, have their best performance on UDG graphs, their next best on Yao sub-graphs and the worst performance on Gabriel sub-graphs (occasionally much worst such as for progress-based algorithms on clustered distributions in Table 7). For the Greedy:Face2:Greedy and Face2 algorithms using a threshold, the trends are less pronounced. For uniform distributions, these algorithms have roughly the same best performance on all three graphs. For clustered distributions, the Greedy:Face2:Greedy and Face2 (with threshold) algorithms perform slightly better on Yao sub-graphs, followed by Gabriel sub-graphs and worst on the UDG. When no threshold is used for Face2, as expected, the delivery rate is 100% for Gabriel graphs and

Table 4. Uniform data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in $YG_8(UDG)$.

	$n = 75$				$n = 100$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):FACE2:AB(C,A)	99.28	0.80	2.11	2.12	98.88	0.89	2.19	3.12
AB(C,D):FACE2:AB(C,D)	98.54	1.03	2.74	2.36	98.34	1.12	2.77	3.26
AB(G,A):FACE2:AB(G,A)	99.08	0.83	2.53	3.21	98.59	0.97	2.70	3.12
AB(E,A):FACE2:AB(E,A)	99.21	0.77	2.68	3.04	98.53	1.14	2.74	3.50
GREEDY:FACE2:GREEDY	92.19	1.82	2.39	2.57	95.80	1.88	2.12	2.47
FACE2 (with threshold)	98.44	1.24	4.51	5.55	99.61	0.72	4.40	5.96
COMPASS	70.57	5.07	1.04	0.09	76.24	3.79	1.06	0.11
GREEDY	69.94	5.12	1.02	0.06	75.28	4.13	1.03	0.06
ELLIPSOID-BASED	55.75	5.50	1.06	0.12	57.54	4.81	1.08	0.13
AB(C,A)	86.37	3.70	2.22	2.80	88.62	2.77	1.89	1.83
AB(C,D)	91.11	3.03	2.61	2.62	92.81	2.42	2.27	1.88
AB(G,A)	85.16	3.97	2.22	2.85	87.61	2.82	1.90	1.95
AB(E,A)	82.28	3.82	2.45	3.25	83.60	3.27	2.01	2.09
FACE2 (without threshold)	99.30	0.88	5.67	5.90	99.25	0.64	5.75	5.19

more than 98% for all graphs and distributions except for the case of clustered distributions on UDG graphs where the delivery rate is slightly less at around 97%. Finally, for the AB:FACE2:AB algorithms, performance was best on Yao sub-graphs followed by Gabriel sub-graphs and then the UDG graphs. We understand the better performance on the Yao sub-graph to be due to its constant stretch factor [22, 25] as well as near planarity combined with preservation of the directionality of the neighbors of nodes. The constant stretch factor maintains the availability of short paths to the destination while the decreased number of edges in the subgraph reduce the number of possibly long detours discovered by randomized routing. Locally, being nearly planar allows FACE2 to work well in by-passing critical nodes while maintaining neighbors in the eight directions of each node of the Yao sub-graph allows the AB algorithms to choose pairs of neighbors that lead to the progression of packets.

In terms of path dilation, the most prevalent trend is that the best path dilations for uniform distributions occur in the Yao sub-graph and UDG, and the worst in the Gabriel sub-graph. For clustered distributions, the Yao and Gabriel sub-graphs perform equivalently well, with the UDG being worst. Typically within a class of algorithms, the difference in path dilations between types of graphs is slight. The only algorithm which did not follow this trend is FACE2 (without threshold) where the path dilation is significantly smaller on the UDG for clustered distributions and on Gabriel sub-graphs for uniform distributions.

Effect of Threshold Fig. 2 shows the effect of varying the threshold value (which was set to $2n$ in the above simulations) on the average delivery rate and average path dilation of the randomized AB:FACE2:AB and the deterministic FACE2 and GREEDY:FACE2:GREEDY (both with threshold) algorithms. The reason for using FACE2 for comparison is that it has the highest delivery rates,

Table 5. Uniform data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in $GG(UDG)$.

	$n = 75$				$n = 100$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):FACE2:AB(C,A)	98.47	1.28	2.75	2.91	97.71	1.48	2.46	2.54
AB(C,D):FACE2:AB(C,D)	97.49	1.61	3.26	3.21	96.38	1.84	3.11	3.17
AB(G,A):FACE2:AB(G,A)	97.93	1.46	3.13	3.24	97.06	1.62	2.85	3.04
AB(E,A):FACE2:AB(E,A)	98.13	1.26	3.14	3.20	96.80	1.57	2.96	3.13
GREEDY:FACE2:GREEDY	94.50	2.01	2.60	2.52	94.80	1.95	2.38	2.36
FACE2 (with threshold)	97.65	1.49	4.52	5.58	99.33	0.76	4.48	6.00
COMPASS	66.59	5.31	1.02	0.06	70.70	4.85	1.04	0.07
GREEDY	66.78	5.05	1.02	0.05	71.05	4.43	1.03	0.07
ELLIPSOID-BASED	54.00	5.27	1.03	0.07	55.71	5.23	1.04	0.08
AB(C,A)	83.42	4.01	2.37	2.59	83.83	3.48	2.03	1.69
AB(C,D)	88.86	3.09	2.83	2.51	88.84	2.92	2.54	1.87
AB(G,A)	82.85	4.40	2.36	2.63	83.90	3.75	2.06	1.78
AB(E,A)	80.70	4.13	2.44	2.85	81.58	3.77	2.08	1.86
FACE2 (without threshold)	100.00	0.00	4.76	5.37	100.00	0.00	4.86	4.44

especially if using the Gabriel sub-graph where we can achieve 100% delivery but with path dilations greater than 4.7.

First, we note that by increasing the threshold to $2n$, the relative behaviour of the algorithms is established and the differences between algorithms are clear. Therefore, for the simulations with all the algorithms, we use a threshold of $2n$. Also, it may be noted that at this threshold, the average delivery rates of the AB:FACE2:AB algorithms increase to nearly 100% with an average stretch factor in the range of 2.1–2.5, comparable (and nearly matched by the AB(C,A):FACE2:AB(C,A) algorithm) to that of GREEDY:FACE2:GREEDY but with a significantly higher delivery rate. Meanwhile, as the threshold increases, the average delivery rate of FACE2 only slowly approaches that of the AB:FACE2:AB algorithms while suffering from an average path dilation greater than 4. Note that this behaviour occurs equivalently on both distributions.

5 Conclusions

In this paper, we proposed a new class of randomized position-based algorithms for routing in mobile ad hoc networks, called the AB:FACE2:AB algorithms, based on combining AB algorithms with the FACE2 algorithm. Our simulation results demonstrate that these hybrid algorithms, on non-planar graphs like the UDG, yield a definite improvement over all the other algorithms studied, when considered both in terms of the delivery rate and path dilation. When we consider the choice of $2n$ for a threshold for the AB:FACE2:AB algorithms, the delivery rate is equivalent to the performance of FACE2 (without threshold) algorithm but with average path dilations that are several times smaller. The best average path dilations are achieved by the progress-based deterministic algorithms, followed

Table 6. Clustered data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in $YG_8(UDG)$, for $b = 50$.

	$n = 71$				$n = 89$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):FACE2:AB(C,A)	98.90	1.05	2.08	1.49	99.30	0.87	2.27	1.70
AB(C,D):FACE2:AB(C,D)	98.58	1.29	2.74	1.90	99.42	0.87	2.85	2.19
AB(G,A):FACE2:AB(G,A)	99.13	0.93	2.51	2.02	99.60	0.68	2.68	2.12
AB(E,A):FACE2:AB(E,A)	99.09	0.90	2.60	2.10	99.49	0.69	2.75	2.25
GREEDY:FACE2:GREEDY	93.85	2.49	1.92	1.04	93.64	2.54	2.23	1.08
FACE2 (with threshold)	97.08	1.78	3.52	2.34	97.83	1.35	3.78	2.55
COMPASS	45.44	5.69	1.07	0.08	45.78	4.56	1.07	0.08
GREEDY	43.81	5.45	1.03	0.05	44.18	4.17	1.03	0.05
ELLIPSOID-BASED	24.44	4.54	1.08	0.09	23.10	4.29	1.08	0.09
AB(C,A)	82.31	3.49	2.24	1.72	83.84	3.71	2.34	1.98
AB(C,D)	85.38	3.59	3.34	2.02	87.33	2.97	3.50	2.33
AB(G,A)	80.07	4.09	2.24	1.73	81.16	3.93	2.37	2.11
AB(E,A)	74.31	3.94	2.49	1.99	75.34	4.54	2.70	2.37
FACE2 (without threshold)	99.65	0.42	6.32	4.87	99.88	0.17	6.21	7.22

by GREEDY:FACE2:GREEDY, with the AB:FACE2:AB algorithms performing a close third with the latter still maintaining very high delivery rates. The best results were in terms of delivery rate and path dilations were seen when routing on the Yao sub-graph,

During our simulations, we assumed that the UDG was static. If the nodes were permitted to become mobile, we expect that, through the use of randomization, the proposed AB:FACE2:AB algorithms will continue to perform well on dynamic unit disk graphs. In essence, a dynamic graph is a version of a randomized graph and to be able to adapt to changes in topology a routing algorithm with a randomized component should fair better than a deterministic algorithm.

Acknowledgements This work was partly funded by NSERC. B.N. Bennani was supported by an NSERC Undergraduate Student Research Award.

References

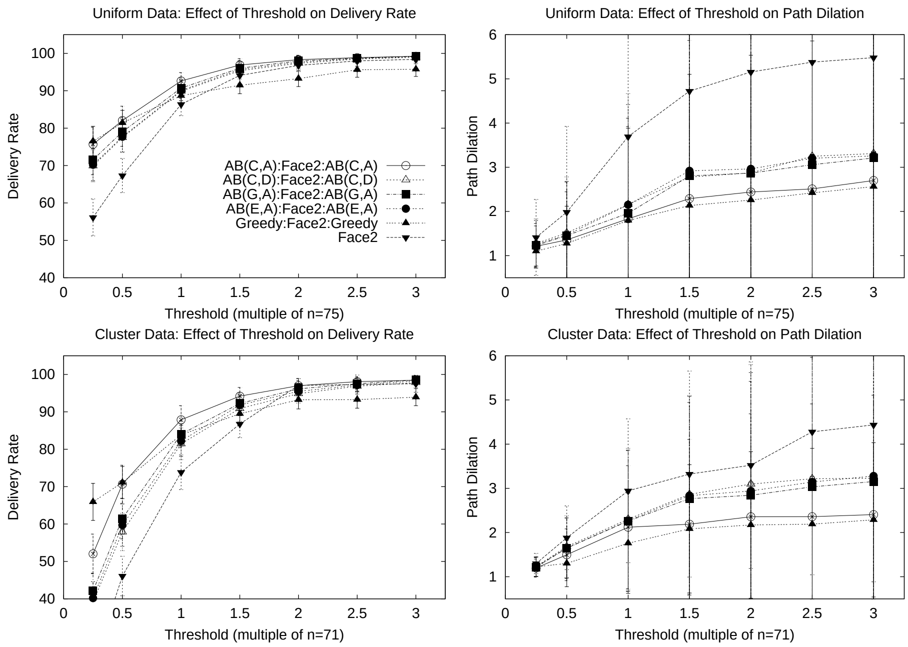
1. Basagni, S., Chlamtac, I., Syrotiuk, V., Woodward, B.: A distance routing effect algorithm for mobility (DREAM). In: Proc. of 4th ACM/IEEE Conference on Mobile Computing and Networking (Mobicom '98). (1998) 76–84
2. Barrière, L., Fraigniaud, P., Narayanan, L., Opatrny, J.: Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. Wireless Communications and Mobile Computing Journal **3** (2003) 141–153
3. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks (2001) 609–616
4. Bose, P., Morin, P.: Online routing in triangulations. In: Proc. of 10th Annual Inter. Symposium on Algorithms and Computation (ISAAC '99). (1999) 113–122

Table 7. Clustered data: Average packet delivery rate, D, and average path dilation, P, and associated standard deviations, σ, in $GG(UDG)$, for $b = 50$.

	$n = 71$				$n = 89$			
Algorithms	D	σ	P	σ	D	σ	P	σ
AB(C,A):FACE2:AB(C,A)	97.83	1.58	2.29	1.51	98.33	1.12	2.42	1.60
AB(C,D):FACE2:AB(C,D)	95.92	1.85	2.77	2.04	98.00	1.48	3.07	2.01
AB(G,A):FACE2:AB(G,A)	97.53	1.53	2.69	1.74	98.80	1.04	2.82	1.81
AB(E,A):FACE2:AB(E,A)	97.33	1.60	2.81	1.75	98.79	1.13	2.84	1.99
GREEDY:FACE2:GREEDY	92.41	2.51	2.15	1.02	93.90	2.39	2.35	1.03
FACE2 (with threshold)	95.57	1.96	3.34	2.09	97.60	1.67	3.58	2.26
COMPASS	37.27	5.27	1.03	0.04	36.79	4.71	1.04	0.04
GREEDY	36.90	5.27	1.03	0.04	36.87	3.98	1.03	0.03
ELLIPSOID-BASED	21.07	4.37	1.04	0.04	20.59	3.80	1.04	0.04
AB(C,A)	76.32	4.61	2.16	1.45	76.01	4.41	2.24	1.66
AB(C,D)	78.70	4.41	3.28	1.74	80.13	4.42	3.42	1.97
AB(G,A)	75.71	4.80	2.14	1.42	76.37	4.20	2.26	1.64
AB(E,A)	70.49	4.91	2.22	1.53	70.93	3.97	2.34	1.75
FACE2 (without threshold)	100.00	0.00	6.53	6.51	100.00	0.00	6.47	5.73

5. Jain, R., Puri, A., Sengupta, R.: Geographical routing using partial information for wireless ad hoc networks. IEEE Personal Comm. Magazine **8** (2001) 48–57
6. Giordano, S., Stojmenovic, I., Blazevic, L.: Position based routing algorithms for ad hoc networks: A taxonomy. In Cheng, X., Huang, X., Du, D., eds.: Ad Hoc Wireless Networking. Kluwer (2003)
7. Mauve, M., Widmer, J., Hartenstein, H.: A survey of position-based routing in mobile ad-hoc networks. IEEE Network Magazine **15** (2001) 30–39
8. Finn, G.: Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISU/RR-87-180, USC ISI, Marina del Ray, CA (1987)
9. Stojmenovic, I., Lin, X.: Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Transactions on Parallel and Distributed Systems **12** (2001) 1023–1032
10. Kranakis, E., Singh, H., Urrutia, J.: Compass routing on geometric networks. In: Proc. of Canadian Conf. on Computational Geometry (CCCG '99). (1999) 51–54
11. Kuhn, F., Wattenhofer, R., Zollinger, A.: Asymptotically optimal geometric mobile ad-hoc routing. In: Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DAILM '02). (2002)
12. Ansari, S., Narayanan, L., Opatrny, J.: A generalization of face routing to some non-planar networks. In: Proc. of Mobiquitous. (2005)
13. Datta, S., Stojmenovic, I., Wu, J.: Internal node and shortcut based routing with guaranteed delivery in wireless networks. Cluster Computing **5** (2002) 169–178
14. Stojmenovic, I., Datta, S.: Power and cost aware localized routing with guaranteed delivery in wireless networks. In: Proc. Seventh IEEE Symposium on Computers and Communications ISCC, Taormina, Sicily, Italia (2002) 31–36
15. Karp, B., Kung, H.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proc. of 6th ACM Conference on Mobile Computing and Networking (Mobicom '00). (2000)
16. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: Of theory and practice. In: Proc. of Principles of Distributed Comp. 2003. (2003)

Fig. 2. The effect of changing the threshold used for uniform (upper row) and clustered data (lower row) on average delivery rate (left column) and average path dilation (right column). The error bars shown indicate standard deviation. The same legend applies to all four figures.



17. Fevens, T., Haque, I., Narayanan, L.: A class of randomized routing algorithms in mobile ad hoc networks. In: Proc. of 1st Algorithms for Wireless and Ad-hoc Networks (A-SWAN). (2004)
18. Gabriel, K., Sokal, R.: A new statistical approach to geographic variation analysis. Systematic Zoology **18** (1969) 259–278
19. Bose, P., Gudmundsson, J., Morin, P.: Ordered theta graphs. Computational Geometry: Theory and Applications **28** (2004) 11–18
20. Bose, P., Devroye, L., Evans, W., Kirkpatrick, D.: On the spanning ratio of gabriel graphs and beta-skeletons. In: Proceedings of the Latin American Theoretical Infocomatics (LATIN). (2002)
21. Yao, A.C.: On constructing minimum spanning trees in k-dimensional spaces and related problems. SIAM J. Computing **11** (1982) 721–736
22. Li, X.Y., Wan, P.J., Wang, Y.: Power efficient and sparse spanner for wireless ad hoc networks. In: Proc. of IEEE Int. Conf. on Computer Communications and Networks (ICCCN01). (2002) 564–567
23. Yamazaki, K., Sezaki, K.: The proposal of geographical routing protocols for location-aware services. Electronics and Communications in Japan **87** (2004)
24. Kuhn, F., Wattenhofer, R., Zollinger, A.: Ad-hoc networks beyond unit disk graphs. In: Proc. of the 2003 joint workshop on the found. of mobile comput. (2003) 69–78
25. Bose, P., Maheshwari, A., Narasimhan, G., Smid, M., Zeh, N.: Approximating geometric bottleneck shortest paths. Comp. Geom. Theory Appl. **29** (2004) 233–249