International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064

Randomized Geographic Routing with Guaranteed Delivery and Low Stretch

A. Felicia[1], **K. Brindha**[2]

[1]Research Scholar, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-5, India

[2]Research Supervisor, Department of Computer Science and Information Technology, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts and Science, Coimbatore-5, India

Abstract: ***A wireless network is critical for evaluating the performance of network protocols and improving their designs. Many protocols for wireless networks routing, topology control, information storage/retrieval and numerous other applications have been based on the idealized unit-disk graph (UDG) network model. The significant deviation of the UDG model from many real wireless networks is substantially limiting the applicability of such protocols. A more general network model, the quasi unit disk graph (quasi-UDG) model, captures much better the characteristics of wireless networks. However, the understanding of the properties of general quasi-UDGs has been very limited, which is impeding the designs of key network protocols and algorithms. In this paper, we present results on two important properties of quasi-UDGs: separability and the existence of power efficient spanners. Network separability is a fundamental property leading to efficient network algorithms and fast parallel computation. We prove that every quasi-UDG has a corresponding grid graph with small balanced separators that captures its connectivity properties. We also study the problem of constructing an energy efficient backbone for a quasi-UDG. We present a distributed localized algorithm that, given a quasi-UDG, constructs a nearly planar backbone with a constant stretch factor and a bounded degree. We demonstrate the excellent performance of these auxiliary graphs through simulations and show their applications in efficient routing.***

Keyword: quasi unit disk graph, GPSR, MANET, 4D network

1. Introduction

We define a geographic routing algorithm to base its decision solely on the position of the current node, the neighbours, and the destination, and we require the network nodes to be memory less, i.e. not to store any state for messages they see. This not only binds the routing state uniquely to the messages, but also removes an additional storage overhead from the nodes, which could limit the number of messages forwarded by a node if its memory is too small. The problem of storing message state is that this data arrives dynamically, and it is hard to predict how much of this data needs to be stored at any given time. Dynamic memory allocation would partially solve the problem, but introduces a computational overhead that many devices cannot afford. Consequently, the numbers of messages for which a node may store the state needs to be determined at compile time, geographic routing success if more messages than anticipated need to be handled.

Another important property of geographic routing algorithms is that their decisions are only based on local information, which can easily be refreshed upon changes in the network. This stands in sharp contrast to routing algorithms that relay in some way on a global view of the network. Whereas these global routing schemes provide excellent routing paths, the construction of their routing information is rather expensive, and any change of the network may require a complete, network wide reconfiguration of the routing information. As a result, these routing algorithms are an excellent choice for static networks, but not for (wireless) ad hoc networks, where a continuous change of the network topology is unavoidable.

2. Problem Statement

The problem of storing message state is that this data arrives dynamically, and it is hard to predict how much of this data needs to be stored at any given time. Dynamic memory allocation would partially solve the problem, but introduces a computational overhead that many devices cannot afford. Consequently, the numbers of messages for which a node may store the state needs to be determined at compile time, geographic routing success if more messages than anticipated need to be handled.

3. Related Works

A routing algorithm that does not label the nodes and desires a routing stretch below 3 needs routing tables of (n) bits per node [1].the routing algorithms for ad hoc networks can be roughly classified as proactive or reactive. Reactive routing schemes determine the route only on demand using flooding [2] to find a path to the destination. Whereas this approach does not generate a static overhead due to changes in the network topology, it introduces an excessive cost for route discoveries. In proactive routing schemes, on the other hand, routes are determined ahead of time and stored in routing tables on the nodes. They are efficient only if the network is stable for a long time, as topology changes may require a network-wide reconfiguration. The probably most prominent members in this class constitute the compact routing schemes, which guarantee routes of nearly optimal length with moderate sized routing tables of polylogarithmic size in the number of network nodes [2]-[3]. Compact routing schemes nearly always go along with a node labelling, i.e. each node is assigned a label. Just as with our geographic routing scheme, where the sender needs to determine the position of the destination node, compact routing requires the sender to learn the label of the target node, which is an integral part of the routing algorithm.

Whereas compact routing schemes try to minimize the size of routing tables, geographic routing does not need them at all, as messages are forwarded based only on local position information. it is possible to assign to each network node a virtual coordinate in the hyperbolic plane (the label), and perform greedy routing with respect to these virtual coordinates, not needing any routing tables at all. However, the construction of the virtual coordinates is based on a (non-local) spanning tree, introducing a worst case stretch of (n). In addition, the virtual coordinates need to be re-evaluated upon any change in the network, which makes the scheme impractical.

GPSR present the common form of greedy forwarding in ad hoc networks. Packets contain the position of the destination and nodes need only local information about their position and their immediate neighbors' positions to forward the packets. Each Wireless node forwards the packet to the neighbour closest to the destination among its neighbours (within radio range) that are closer to the destination as shown in Figure 1.

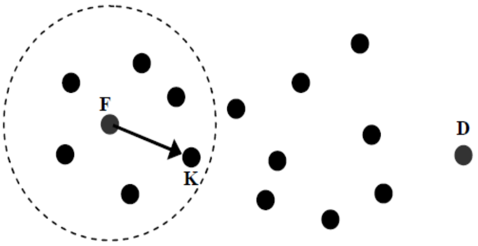

Figure 1: Greedy forwarding: Node F forwards the packet to neighbour K, which is the neighbour closest to the destination D

Greedy forwarding is very efficient in dense uniform networks, where it is possible to make progress at each step. Greedy forwarding, however, fails in the presence of voids or dead-ends, when reaching a local maximum, a node that has no neighbours closer to the destination (Figure 1). In this case, it will fail to find a path to the destination, even though paths to the destination through further nodes may exist. Previous protocols deals with dead-ends in different ways. In MFR, if no progress could be made in the forward direction, the dead-end node sends the packet to the least backward neighbour, which is the neighbour closest to the destination among its neighbours. This could cause looping and nodes need to detect when they get the same packet for a second time. Finally proposed using limited flooding for a number of hops to overcome dead-ends. When a node is reached that has no neighbours closer to the destination, it sends a search packet for n hops away. Closer nodes to the destination reply back and the closest node to the destination among those nodes is chosen to forward the packet.

The non-existence of local, memory less routing algorithms that deliver messages deterministically [4] has many direct and indirect consequences. Whereas it is possible to deterministically traverse a planar subdivision and report all nodes and faces [5] there is no corresponding algorithm in 3D. However, it has been shown in [5]-[7] that for any undirected graph, it is possible to assign each node a local ordering of its edges such that a routing algorithm can visit all nodes in O(n) time (deterministically!) by leaving a node through the edge succeeding the edge through which it entered. Unfortunately, the construction of the local edge-orderings requires a global view of the graph and has construction time cubic in the number of nodes.

4. Proposed System

To guarantee easy extensibility, offers a set of extension points, the so called models. The following list gives an overview of the available models, to each of which you may add your own extension. The mobility model describes how the nodes change their position over time. Examples are random waypoint, random walk, random direction, and many others. The connectivity model defines when two nodes are in communication range. The best known examples are the unit disk graph (UDG) and the quasi-UDG (QUDG).The distribution model is responsible to initially place the network nodes in the simulation area. E.g. place the nodes randomly, evenly distributed on a line or grid or according to a stationary regime of a mobility model.

The interference model is to define whether simultaneous message transmissions may interfere. The reliability model is a simplified form of the interference model and lets you define for every message whether it should arrive or not. E.g. drop one percent of all messages the transmission model lets you define how long a message takes until it arrives at its destination.

4.1 Network model

Unit Disk Graph (UDG): A UDG is a special instance of a graph in which each node is identified with a disk of unit radius r=1, and there is an edge between two nodes u and v if and only if the distance between u and v is at most 1. The model is depicted. Each node's transmission range is drawn as a dotted circle. The edges, which connect nodes, are drawn as straight lines. The neighbours of node u are node v, node w, node y and node z is shown in the simplified graph.

Quasi Unit Disk Graph (QUDG): In a QUDG, each node is identified with two disks, one with unit radius r=1 and other with radius q= (0, 1). It can be observed that a QUDG with q=1 is an UDG. The edges between nodes d away from each other are identified with respect to the below listed rules:

There is an edge between two nodes if d= (0, q).
There is a possible edge connecting two nodes if d= (q, 1)
There is no edge between two nodes if d= (1)

Lower bound for the performance of geographic routing algorithms in three and four dimensions. Nodes represented by solid squares lie on the surface of a sphere with mutual distance at least 2. Nodes printed as diamonds lie also on the surface and connect these points. The round (red colour) nodes lie on lines leading from the surface nodes towards the centre. A single dedicated surface-node w has an extended line leading to node t in the centre of the sphere.

4.2 Routing in 4D Networks

The greedy geographic routing we would like to apply to our 4D graphs is actually close to optimal - as long as the message does not fall into a local minimum. But because there is no deterministic local memory less routing algorithm, there is also no deterministic recovery algorithm that could lead our messages out of local minima. In this section, we take a short excursion to random walks, which we propose to use to escape from such local minima.

Whereas a message moving around randomly in our network may seem very inefficient and too simplistic, there is quite some work in this area indicating that random walks need not be as bad as it looks at a first glance. The two prominent models to capture a random walk on a graph G = (V, E) are (1) the Markov chain, and the flow in an electrical network obtained from G by replacing every edge by a resistance of 1. In the following, we use n: = |V | and m: = |E|.

4.2.1 Grid Graph of Quasi-UDG

Present a distributed algorithm for constructing a grid graph for any quasi-UDG, and prove that the grid graph is well separable. The grid graph, whose node density and edge density are both upper bounded by constants, is an abstraction of the quasi-UDG. A quasi-UDG may have highly variable node and edge densities, which prevent it from having small separators. The grid graph is a “sparsified” version of the quasi-UDG, which retains the distance information for vertices and represents well the deployment region of the quasi-UDG. As a result, the connectivity results for the grid graph can be easily mapped to results for the quasi-UDG. An example of a quasi-UDG and its corresponding grid graph is shown in parts (a) and (b) of Fig. 2, respectively.

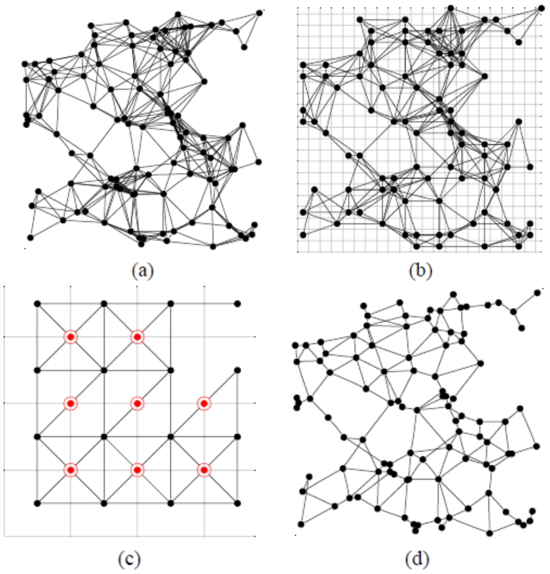

Figure 2: (a) A quasi-UDG G with 100 vertices and R=r = 0:5; **(b)** The grid graph corresponding to G; **(c)** The auxiliary graph used to find the top level separator of G; **(d)** The backbone of G.

4.2.2 Construction of the Grid Graph Algorithm

INPUT: G: a quasi-UDG with parameters R and r
OUTPUT: B: a backbone of G

1. Planarize the sub graph induced by short edges of G The sub graph B will contain the same vertex set as G. Initially, the edge set of B is set to empty. For each edge e = fu; vg in G, if there is no common neighbour of u and v in G residing in the disk whose diameter is the edge e, we add e into B. this process can be done in a distributed manner by exchanging no more than O(m) messages where m is the number of edges in G.

2. Reduce the number of short edges incident to each vertex Let G0 be the sub graph of B that includes all the vertices and short edges of B. Note that here G0 is in fact the Gabriel graph constructed from a UDG (with communication range r); so G0 is planar. We apply the algorithm described on G0. Here is a brief description of the algorithm that is performed by each vertex: Direct the edges in G0 (using the classical acyclic orientation of a planar graph) so that every vertex in G0 has at most 5 incoming edges; Perform a standard Yao step on the set of outgoing edges; Select certain edges that form large angles with consecutive edges (see [8] for details); Finally, communicate with all the neighbours of the vertex and keep edges that have been selected by least one of their ends.

When the above algorithm ends, we remove from B those edges that have been removed by the algorithm from G0. This step will reduce the number of short edges incident to every vertex to a constant k+5, where k is a parameter, and it can be done locally. Compared to the sub graph of G that contains all the short edges of G, B increases the minimum communication cost between any two vertices by a factor of at most 1 + (2 sin(¼=k)), where k is a parameter, and is the path loss exponent.

3. Reduce the number of long edges incident to each vertex Add the entire long edges of G to B. We impose a grid of cell-size r/2 X r/2 on the plane. Clearly, any long edge must be connecting vertices in two different cells. For each pair of cells, we remove from B all the long edges between them except for the shortest one.

4.3 Topology Control Models

Interference Tree: Interference is one of the major challenges in wireless networks and thereby MANETs. It alters or disrupts the message as it is transmitting along a channel between source and destination. Since the message is disrupted when the interference occurs, it has to be detected and the interfered message has to be retransmitted. In particular, in multi hop communications, the nodes dissipate energy and time due to interference. The interference in MANETs mostly occurs from concurrent message transmission. Since the nodes in MANETs generally use the Omni directional antennas, the sent messages from a node are received by all nodes which are in transmission range of sender node. When two messages are sent concurrently by two neighbouring nodes, they affect each other and interference occurs.

4.4 Mobility Models

Mobility models are designed to represent the movements of mobile users, and how their location, acceleration change over time. They are used to evaluate the performance of ad hoc network protocols. Since the performance of protocol depends on the mobility model, it is important to choose the suitable model for the evaluated protocol. Various mobility models have been proposed so far but the most common models are 1) Random Waypoint 2) Random Direction model. 3) Random Walk Model.

5. Experimental Results

In order to validate our geographic routing algorithms for 3D and 4D networks, we performed a series of simulations, a Java-based simulation framework for testing and validating network algorithms. We chose a fairly large simulation area of 20 × 20 × 10 units and deployed between 2000 and 40000 nodes to cover the range between very sparse and dense networks. In order to obtain more realistic networks, we first placed 100 randomly rotated and randomly positioned cuboids of 2 × 2 × 1 units in the simulation area. The cuboids were areas where no node could be placed, and they enforced holes in the network, such that, especially for dense networks, the messages could not be forwarded greedily without surrounding any holes. Sparse graphs tend to be heavily twisted, which challenges our GRG routing algorithms with many local minima.

To account for this fact, we performed more simulations for sparse networks, which can also be seen by the accumulation of samples for small n. For each initial deployment of n nodes, we first connected the nodes to a UBG, and kept only the giant component, the largest connected part of the network. For ease of interpretation, we plotted against the overhead of the flooding algorithm, such that the y-axis shows how much more overhead the other routing algorithms induced.

First important observation is that limiting the RW to the surface of the hole does in fact not help at all. The reason is two-fold:

First, unless the network is very dense, it tends to have a single huge face covering nearly the entire network. I.e. the holes in the network are nearly never completely closed and most of them are interconnected over the surface. As a result, the restriction to the face does not reduce the number of nodes to visit.

Secondly, we needed to drop the “power of choice” for the RW on the surface, which boosts the RW considerably, especially for sparse, tree-like networks. Thus, limiting the RW to the surface is not worth its price.

Table 1: Table showing the power stretch factor for the backbones (b= 2)

	Stretch Factor			
N	R/r=10	R/r=3	R/r=2	R/r=1.5
1000	1.048	1.141	1.190	1.184
1500	1.044	1.155	1.198	1.129
2000	1.046	1.176	1.239	1.204

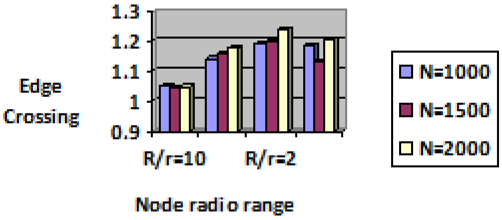

Figure 3: The Maximum degree, the average degree, and the average number of edges crossing an edge for quasi-UDGs and their backbones**.**

Table 2: Routing techniques compared

Number of deployed nodes	0	5000	10000	15000	20000	25000	30000	35000	40000
RW Graph	4	4	2.8	2.5	2.2	2.9	2.2	1.5	1
RW Dual	22	1	1	0.9	12	1.6	1.4	0.8	5
DFS Tree	45	22	2.1	1.8	15	1.56	1.45	0.6	5
RW Surface	4.1	32	3	2.5	2.1	2.4	2	1.5	1

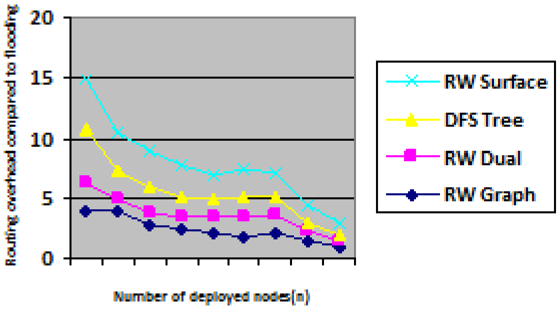

Figure 4: The overhead of our routing techniques compared against a non memories flooding algorithm

6. Conclusion

Geographic routing schemes are both memory less and which makes them highly suitable for mobile ad hoc networks and sensor networks. The geographic routing schemes for 3D networks, however, cannot be translated to 4D networks directly. For instance, limiting the recovery algorithm to visit only nodes on the surface of the network hole which caused the local minima makes little sense in 4D, as most networks tend to have a single huge surface. Whereas the analysis of the surface detection was limited to UBG, the greedy-random greedy routing scheme is applicable for real networks.

Further observe that the overhead of the RW on the dual is below the overhead of the RW on the graph, which shows that we achieved our goals of obtaining a sparse network graph via the dual graph. The astonishing good performance of the DFS on the spanning tree for the sparse networks can be traced back to the fact that the nodes of these networks have very low degree, resulting in a tree-like network.

For denser networks, however, the RW approaches perform much better. In particular, the RW on the dual and the RW on the surface perform even better than the flooding for very dense networks, as they operate on a sparser network.

References

[1] S. M. N. Alam and Z. J. Haas, “Coverage and Connectivity in Three Dimensional Networks,” In Proc. ACM MobiCom, 2006, pp. 346–357.

[2] P. Bose, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, A. Lopez- Ortiz, P.Morin, and J.Munro, Online Routing In Convex Subdivisions,” Int. J. Comput. Geom. Appl., vol. 12, no. 4, pp. 283–295, 2002.

[3] P.Bose and P. Morin, “Online Routing in Triangulations,” SIAM J. Comput., vol. 33, no. 4, pp. 937–951, 2004.

[4] P.Bose, P.Morin, I. Stojmenovic, and J.rrutia, “Routing with Guaranteed Delivery in Ad Hoc Wireless Networks,” in Proc. DIALM, 1999, pp. 48–55.

[5] M. Caesar, M. Castro, E. B. Nightingale, G. O’Shea, and A. Rowstron, “Virtual Ring Routing: Networking Routing Inspired by DHTs,” in Proc. ACM SIGCOMM, 2006, pp. 351–362.

[6] R. Draves, J. Padhye, and B. Zill, “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks,” in Proc. ACM MobiCom, 2004, pp. 114–128.

[7] Durocher, Kirkpatrick , and L. Narayanan, “On Routing With Guaranteed Delivery in Three- Dimensional Ad Hoc Wireless Networks,” in Proc. ICDCN, 2008, pp. 546–557.

[8] R. Flury and R. Wattenhofer, “Randomized 3D Geographic Routing,” in Proc. IEEE INFOCOM, 2008, pp. 834–842.

[9] S. Fortune, J. E. Goodman and J. O’Rourke, Eds., “Voronoi Diagrams and Delaunay Triangulations,” In Handbook of Discrete and Computational Geometry, 2nd ed. Boca Raton, FL: CRC Press, 2004.

Author profile

A. Felicia is currently pursuing the M.Phil in Computer science from Sri Jayendra Saraswathi Maha Vidyalaya College of Arts and Science, Coimbatore. She holds the degree of M.Sc. in Information Technology from Kongu Engineering College, Perudurai.

K. Brinda M.C.A., M. Phil., Working as an Assistant professor in department of computer Science from Sri Jayendra Saraswathi Maha Vidyalaya College of Arts and Science, Coimbatore. She has 6 years of teaching experience. Her areas of interest are networking and data mining.