1 Routing with guaranteed delivery in geometric and wireless networks

JORGE URRUTIA

Instituto de Matemáticas, Universidad Nacional Autónoma de México, Mexico D.F. México

ABSTRACT

In this paper we study on-line local routing algorithms for communication networks. Our algorithms take advantage of the geometric properties of planar networks. We pay special attention to *on-line local routing* algorithms which guarantee that a message reaches its destination. A message cosists of packets of data that have to be sent to a destination node, i.e. the *message itself* plus a finite amount of space used to record a constant amount of data to aid it in its traversal, e.g. the address of the starting and destination nodes, a constant number of nodes visited, etc. *Local* means that at each site we have at our disposal only *local information* regarding a node and its neighbors, i.e. no global knowledge of the network is available at any time, other that the network is planar and connected. We then develop *location aided local routing algorithms* for wireless communication networks, in particularly cellular telephone networks.

—

1.1 INTRODUCTION

The vertices of a geometric network are points on the plane, and its edges straight line segments joining them. A geometric network is called planar if it contains no two edges that intersect other than perhaps at a common end point. In the remainder of this paper we will assume that all our graphs, unless otherwise stated, are planar geometric networks.

Our main goal here is that of studying routing algorithms that take advantage of the location of the nodes of geometric networks. Early papers on routing ignored information regarding the physical location of the nodes of the networks. With the advent of new technologies such as Global Positioning Systems (GPS), the user's location is becoming common information that can be retrieved from GPS, and then used to develop better routing algorithms.

For other applications, we can use the location of a node as part of its label. This can in turn can be used to obtain efficient routing algorithms. In many applications, such as wireless cellular networks, Internet service providers, and others, many nodes have fixed locations. Networks such as cellular communication networks consist of a backbone sub-network, and a collection of mobile users that move around freely, and connect through fixed switches. In many of these networks, the use of global positioning systems allow users to obtain the physical location or *geographical information* regarding users and switches of a network [18].

Information regarding the position of the nodes of a network can, and indeed has been used to obtain new routing schemes that take advantage of this information. A number of papers proposing various types of routing algorithms using geographical data have been written [3, 5, 7, 12, 14, 15, 22, 27].

In this paper we will focus on *on-line* or *local routing* algorithms for *connected planar geometric graphs* that take advantage of the physical location of the nodes of the networks. We are mainly interested in on-line routing algorithms that use geographic information on the nodes and links of a network, and that in addition *guarantee that messages arrive at their destination*. Our approach differs from similar algorithms studied in the literature, particularly in the context of wireless networks in which numerous routing schemes have been developed and mostly tested experimentally.

Some earlier work such as [11], and [7] proposed location-based algorithms based on various notions of progress. Most of those routing protocols do not necessarily guarantee message delivery. Indeed some of the routing schemes proposed recently [2, 15] can also lead to the same problem [27]. In many cases, e.g. flooding routing algorithms [10], multiple redundant copies of the messages are sent in the hope that one of them will eventually reach its destination. Sending multiple copies of messages creates other problems such as network congestion. We believe that the usage of algorithms such as those presented here will become paramount as the number of users of communication networks increases. In [14] another method called *compass routing* is proposed that is shown to work for some specific types of networks. Briefly if a message is located at a node v, and wants to reach node t, compass routing will send it to the neighbor u of v such that the slope of the line segment joining u to v is the closest to the slope of the segment joining v to t. While occasionally compass routing may fall into infinite loops failing to reach t, it works for some important classes of networks. In particular it is shown in [14] that compass routing works correctly for Delaunay Triangulations, a result that will be useful to develop routing algorithms for wireless communication networks. We will also study variations of compass routing that will enable it to work for planar geometric networks.

In [20] similar problems are studied. Shortest-path problems are studied in which a map is not known in advance. They seek dynamic decision rules that optimize the worst-case ratio of the distance covered to the length of the shortest paths.

We will show how our results can be used to solve some routing problems in wireless communication networks which are not necessarily planar. To this end, we will develop fully distributed techniques to calculate planar sub-networks of wireless communication networks. This will be achieved by using some standard tools in Computational Geometry. The resulting algorithms are also guaranteed to deliver messages to their destination. Some future lines of research are pointed out at the end of our paper.

It has been proposed that the algorithms presented here can be consider as a safeguard method to be used when heuristic techniques such as those proposed in [13, 11, 19], and [28] fail. We argue that algorithms of the type of those presented here should become standard, as they not only guarantee that a message gets to its destination, but also tend to create little overhead, which in turn solves other problems arising from broadcasting multiple copies of data messages.

1.1.1 Local position aided routing algorithms

In this section we present some of the basic ideas used in the development of our *location aided* or *geometric* on-line routing algorithms on *planar geometric networks*. Some of these algorithms have been refined and improved, yet the basic ideas remain. By a *location aided* or *geometric* on-line routing algorithms, we understand an algorithm that works under the following restrictions:

1. A typical message contains the location of its starting point s, the location of its destination t, the contents of the message, e.g. the text of an e-mail, and perhaps a constant amount of extra storage in which a *constant* amount of information regarding some data concerning the route that a message has traveled is recorded.

2. At each node of the network, a processor has some geographical local information concerning only the location of its neighbors.

3. Based only on the local information stored at the nodes of the network, the location of s, t and the information stored in the extra memory the message itself carries, a decision is taken regarding on where to send the message next.

It is not straightforward to develop a routing algorithm that satisfies the above restriction, and yet guarantees that a message arrives at its destination. In fact in some earlier papers on the subject [5], seemed to assume that their algorithms guaranteed message delivery!

Our objective in this section is to develop such an algorithm.

1.1.2 Compass routing

Suppose that we want to travel from an initial vertex s to a destination vertex t of a planar geometric network. Assume that all the information available to us at any point in time is:

1. The coordinates of our starting and destination points.

2. Our current position.

3. The directions of the edges incident with the vertex where we are located.

With this information available, we define the following rule to route in geometric networks:

Compass Routing *Starting at s, we will in a recursive way choose and traverse the edge of the geometric graph incident to our current position and with the slope closest to that of the line segment connecting the vertex we are standing at to t. Ties are broken randomly.*

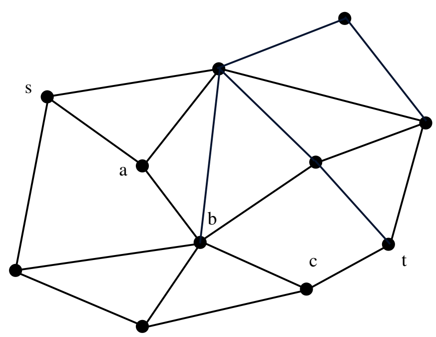

Fig. 1.1 Traveling from s to t using compass routing will follow the path s, a, b, c, t.

Unfortunately compass routing does not guarantee arrival to the destination. This is evident if we use it in geometric graphs with low connectivity, or graphs with non-convex faces. What is somewhat unexpected is that compass routing fails even in geometric graphs in which all of its faces are triangles and the external face is bounded by a convex polygon The geometric graph shown in Figure 3 has these properties, and yet when we try to use compass routing to go from $s = u_0$ to t we get stuck around the cycle with vertex set $\{v_0, w_i; i = 0, \ldots, 3\}$. The graph consists of two concentric squares, one of which is rotated slightly. The line segment $t - v_i$ is orthogonal to the edge joining v_i to w_i, and w_i lies on $t - v_i$, $i = 0, \ldots, 3$. It is now easy to see that under these conditions, if we are at point v_i (resp. w_i), compass routing will choose next the edge connecting v_i to w_i (resp. w_i to v_{i+1}, addition taken mod 4). Similar constructions exist in which instead of using a square to start the construction, we use a regular polygon with n vertices, $n \geq 4$.

At this point, we would like to mention that our initial motivation to study on-line location aided routing algorithms arose from an interesting routing scheme called *interval routing* introduced by Santoro and Khatib [23]. The goal in interval routing is that of finding, whenever possible, a labeling of the vertices of a graph with the integers $1, \ldots, n$ such that for every vertex i of the graph, we can assign to each edge e_i incident to i a disjoint interval $[a_i, b_i]$ with the property that if $j \in [a_i, b_i]$

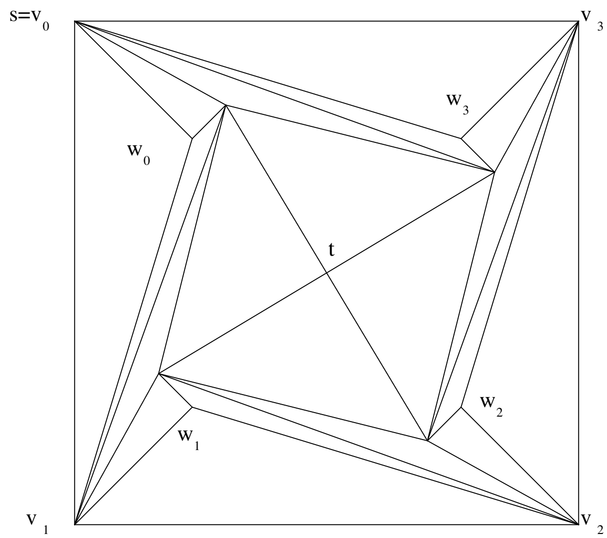

Fig. 1.2 Compass routing will not reach t from $u_i, i = 0, \ldots, 3$.

then there is a shortest path from i to j containing e_i. Each edge is assigned two intervals, one at each of its endpoints. See Figure 1.3. One of the motivations for interval routing was that of having a fast and efficient method to forward information received at a node whose final destination was not the node itself. Interval routing reduces the forwarding problem to that of performing a simple search on the set of intervals assigned to the edges incident to a vertex of a graph. Observe that compass routing also reduces the forwarding problem to a search problem. It is easy to see that as is the case with compass routing, most graphs have no labeling scheme that supports interval routing. However when interval and compass routing work, they give efficient, fast and reliable routing protocols.

We say that a geometric graph G supports compass routing if for every pair of its vertices s and t, compass routing (starting at s) produces a path from s to t.

The *Delaunay triangulation* $\mathcal{D}(P_n)$ of a set P_n of n points on the plane, is the partitioning of the convex hull of P_n into a set of triangles with disjoint interiors such that

- the vertices of these triangles are points in P_n
- for each triangle in the triangulation, the circle passing through its vertices contains no other point of P_n in its interior.

It is well known that when the elements of P_n are in general *circular position*, i.e. no four of then are co-circular, then $\mathcal{D}(P_n)$ is well defined. For the rest of this

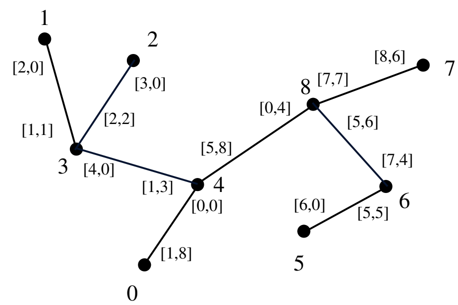

Fig. 1.3 An interval routing scheme for a tree with 9 vertices. The intervals are taken $mod\, 9$. For example interval [7,4] consists of the elements $\{$7, 8, 0, 1, 2, 3, 4$\}$

section we will assume that P_n is in general circular position. The next result was proved in [14]:

Theorem 1.1.1 *Let P_n be a set of n points on the plane; then $\mathcal{D}(P_n)$ supports compass routing.*

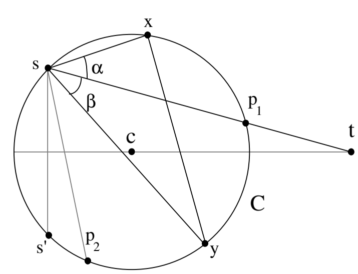

Fig. 1.4 Routing on Delaunay triangulations.

The proof relies on the fact that each time we move along an edge, the Euclidean distance to t always decreases. This can be easily seen from Figure 1.4. Indeed suppose that s and t are not adjacent, and that the line connecting s to t intersects the triangle with vertices $\{s,\ x,\ y\}$ of $\mathcal{D}(P_n)$. By definition t does not belong to the circle passing through s, x, and y, and the segment $s - t$ intersects the segment $x - y$. It is easy to see now that if compass routing chooses to move from s to say x, then the distance from x to t is strictly smaller than the distance from s to t. Experimental results by P.R. Morin [17] show that the average link and distance dilatation of compass routing on Delaunay triangulations of randomly generated point sets in the unit square with up to 500 points, are less than 1.4 and 1.1 respectively.

1.1.3 Compass routing on convex subdivisions

A geometric graph is called a *convex subdivision* if all its bounded faces are convex, and the external face is the complement of a convex polygon. By randomizing compass routing Morin [17] was able to guarantee message delivery not only in triangulations, but in convex subdivisions.

Morin's modification is indeed simple. Suppose that we want to reach vertex t, and that a message is currently located at vertex v. Let $cw(v)$ and $ccw(v)$ be the two vertices defined as follows: $cw(v)$ is the vertex adjacent to v that minimizes the clockwise angle $\angle^{cw} t, v, u$, and $ccw(v)$ the vertex adjacent to v that minimizes the counterclockwise angle $\angle^{ccw} t, v, u$, see Figure 1.5. Random Compass sends the message with equal probability to $ccw(v)$ or to $cw(v)$.

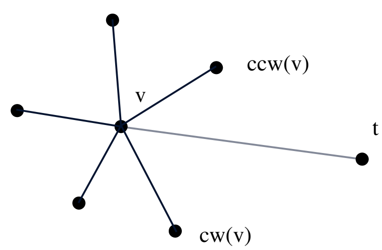

Fig. 1.5 Defining $ccw(v)$ and $cw(v)$.

Morin proved:

Theorem 1.1.2 *Random Compass guarantees message delivery in any convex subdivision.*

It should be mentioned that the previous result guarantees that using Random Compass a message will eventually reach its destination. In theory it could take an arbitrarily large amount of time before a message arrives at its destination. However experimental results also presented in [17] show that Random Compass performs well on the average. Its dilation is better than 1.7 for Delaunay triangulations with up to 500 vertices. No experimental results are reported for convex subdivisions.

Although Compass routing fails for triangulations, we now show how a slight modification of it will enable it to work in convex subdivisions.

Compass Routing on Convex Subdivisions: [14]

The following procedure stops upon reaching t.

1. Starting at s determine the face F incident to s intersected by the line segment $s - t$. Pick any of the two edges of F incident to s, and start traversing the edges of F until we find the second edge of F intersected by $s - t$.

2. Update F to be the second face of the geometric graph containing $u - v$ on its boundary.

3. Traverse the edges of our new F until we find a second edge $x - y$ intersected by $s - t$. At this point we update F again as in the previous point. We iterate our current step until we reach t

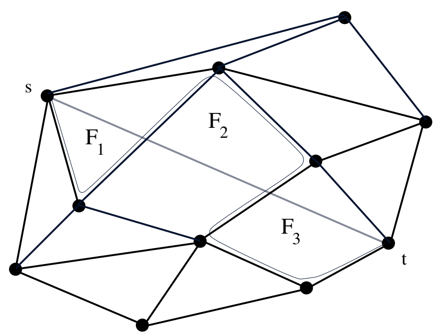

Fig. 1.6 Routing using Compass routing on convex subdivisions.

To prove that a message always gets to its destination, we proceed as follows: Let us label the faces intersected by the line segment joining s to t by $\{F_1, \dots, F_m\}$ according to the order in which they are intersected. Initially $F = F_1$. Observe that each time we update F we move from F_i to F_{i+1} for some i. Thus eventually we reach the face F_m containing t, and thus t. See Figure 1.6. Observe that our algorithm traverses each edge of our graph at most once. It is easy to see that if the faces of a geometric graph are not convex, the previous algorithm may fall into a loop. In the next section we show how to modify compass routing so that it will also work for arbitrary geometric graphs. The price we pay is that in general the paths we have to traverse might increase substantially in length. This is a consideration to have in mind when using the results in the next subsection for particular applications.

1.1.4 Compass routing on geometric graphs

Observe first that the vertices and edges of any geometric graph G induce a partitioning of the plane into a set of connected regions with disjoint interiors, not necessarily convex, called the faces of G. The boundary $\mathcal{B}_i$ of each of these faces is a closed polygonal in which we admit some edge of G to appear twice. For example in the graph shown in Figure 1.7, in the polygonal bounding the external face the edge $u-v$ appears twice.

Suppose now that we want to travel from a vertex s to a vertex t of G. As before, calculate the line segment joining s to t, and determine the face $F = F_0$ incident to s intersected by $s - t$. We now traverse the polygonal determined by F_0. Each time we intersect $s - t$ at a point p, while traversing the boundary of $F(0)$, we calculate the distance from p to s. Upon returning to s, (unless we reach t, in which case we stop) all we need to recall is the point p_0 at which the polygonal bounding F_0 intersects $s - t$, which maximizes its distance to s. We now traverse the boundary of F_0 again until we reach p_0, at which point we update F to be the second face whose boundary contains p_0. We repeat our procedure using p_0 and our new F instead of s and $F(0)$. It is straightforward to see that we eventually reach t. Notice that each edge of our graph is contained in at most two faces. Observe that if the edges of a face are traversed, they are traversed at most twice. It follows that each edge is traversed at most four times. A slight modification can be used so that each edge is traversed at most three times [3].

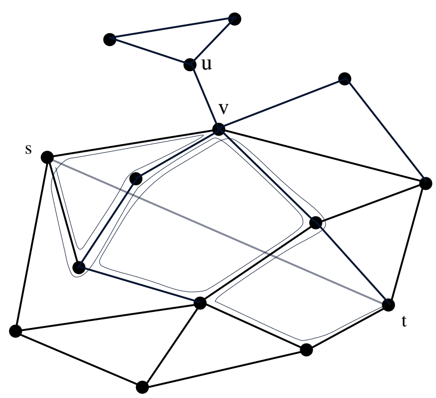

Fig. 1.7 Routing using Compass routing on non-convex subdivisions. Observe that the length of the path traversed from s to t is considerably longer than the one we obtained for convex subdivisions.

Thus we have proved:

Theorem 1.1.3 *[14] There exists a local information routing algorithm on geometric graphs which guarantees that we reach our destination. Moreover, our algorithm is such that we traverse a linear number of edges.*

It should be pointed out that the main objective of the algorithms presented in this section is that of finding on-line local routing algorithms that guarantee message delivery. This implicitly implies that the routes generated by our algorithms will be in general not the shortest paths connecting s to t. In fact it is straightforward to see that for every k we can construct examples in which the lengths of the paths found by our algorithms are k times longer than that of the shortest paths connecting s to t. This can be achieved if the length of a path is measured either in terms of the sum of the lengths of its edges, or the number of edges used in the path. In practice however this does not happen often. For details see [3, 17].

We stress this point here, as there are numerous papers in which many *ad hoc* routing techniques are proposed and tested for numerous types of communication; ad hoc, wireless etc. networks. A common parameter measure in most of these methods is the *success rate*, i.e. the percentage of messages that arrive at their destination. In addition, many of these algorithms broadcast multiple copies of a message in hope that at least one of them will reach its destination. Observe that this creates a large overload in terms of the amount of traffic generated. In time this will become an important factor to be avoided. In contrast our algorithms have a 100% success rate, while sending only one copy of each message. In the next section we will show how the results presented in this section are used to obtain routing algorithms in wireless communication networks such as cellular telephone networks. Our algorithms guarantee message delivery.

1.2 APPLICATIONS TO AD HOC WIRELESS COMMUNICATION NETWORKS

A wireless communication network can be modeled as a set of radio stations located on a set of points $P_n = \{p_1, \ldots, p_n\}$, each of which has associatedto it a real number r_i, its *transmission power*, such that two points p_i, p_j are connected if their distance is smaller than the minimum of $\{r_i, r_j\}$. We now address the problem of developing an on-line local routing algorithm for wireless cellular communication networks.

Cellular telephone communication networks consist of a set of fixed, low-powered radio stations located on $P_n = \{p_1, \ldots, p_n\}$, all with the same transmission power $r(i) = 1$, and a set of mobile users that move freely. The mobile users connect to the network through the closest fixed radio station. The set of fixed radio stations defines a *unit wireless communication network* $UW(P_n)$ on P_n in which two elements $p, q \in P_n$ are connected if their distance is at most 1.

We proceed now to develop an on-line local routing algorithm for unit wireless communication networks. Observe first that $UW(P_n)$ is not necessarily planar. For instance if P_n consists of 12 points contained within a circle of radius 1, $UW(P_n)$ is not planar.

In order to use the results presented in the previous section, we should be able to *extract* a planar subnetwork from any $UW(P_n)$. Two requirements must be satisfied by the method we use to extract the planar subgraph to fully ensure its functionality for real life applications:

- If a cellular communication network is connected, the resulting planar subgraph must be connected.

- We must have a *local protocol* so that each node of the network can decide in a consistent manner which neighbor connections to keep, and ensure that collectively, and without the need to communicate, the set of edges chosen individually by the nodes of the network form a planar graph.

The necessity for the second condition follows from our desire to have fully distributed protocols that avoid the use of any kind of centralized protocols.

The problem of extracting or even deciding if a graph contains a planar connected subgraph is a well known *NP-complete* problem [16]. Fortunately $UW(P_n)$ networks always have such a subgraph, and in fact, finding it is relatively straightforward.

The key to our result arises from the use of *Gabriel* garphs [1]. Given two points p and q on the plane, let $C(p,q)$ be the circle passing through them such that the line segment joining p to q is a diameter of $C(p,q)$. Given a set of n points $P_n = \{p_1, \ldots, p_n\}$ on the plane, the Gabriel graph of P_n is the graph whose set of vertices is P_n in which two points u and v of P_n are adjacent iff the $C(p,q)$ contains no other points of P_n. Let $G'(P_n)$ be the graph with vertex set P_n such that two vertices p and q are adjacent in $G'(P_n)$ iff $C(p,q)$ contains no other points of P_n and p and q are adjacent in $W(P_n)$, that is $G'(P_n)$ is the intersection of the Gabriel graph of P_n with $W(P_n)$. The following result was proved in [3]:

Theorem 1.2.1 *If $UW(P_n)$ is connected then $G'(P_n)$ is also connected.*

The easiest proof of this result proceeds as follows. Let p and q be such that they are adjacent in $UW(P_n)$, and there is no path connecting them in $G'(P_n)$. Suppose further that their distance is the smallest possible among all such pairs of points in P_n. Since p and q are not connected in $G'(P_n)$, $C(p,q)$ contains at least a third point $r \in P_n$. Observe that the distances from r to p and q are smaller than the distance from p to q, and thus there is a path P' in $G'(P_n)$ connecting r to p and a path P'' connecting r to q. The concatenation of these paths produces a path from p to q in $G'(P_n)$. Our result follows.

It is obvious that each node p in $UW(P_n)$ can decide locally which of its neighbors in $UW(P_n)$ should be its neighbors in $G'(P_n)$. It simply collects the locations from all its neighbors (i.e. the elements of P_n at distance at most 1 from p, and tests for each q of them if the circle $C(p,q)$ is empty. This can be done using standard algorithms in Computational Geometry in $O(k \ln k)$ where k is the number of neighbors of p in $UW(P_n)$ [21].

We now have the general tools to obtain an on-line local routing algorithm on unit wireless communication networks. First find $G'(P_n)$, and then use the routing algorithm in Theorem 1.1.3 to send messages. The calculation of $G'(P_n)$ can be done only once, or periodically in cases where node failures can happen.

Thus we have proved;

Theorem 1.2.2 *There exists an on-line routing algorithm for unit wireless communication networks that guarantees delivery. Any message takes at most a linear number of steps to reach its destination.*

Some fine-tuning of the algorithm resulting from the previous theorem was done in [3, 17]. These papers make some modifications to compass routing for arbitrary planar geometric networks that improve the worst case scenario regarding the number of edges traversed. The reader interested in the details can consult [3, 17]. In the same papers, experimental results that show that in practice our algorithms perform well are available. Details of simulations of our algorithms, and variations of them are also included in those papers.

Another routing algorithm using similar ideas to those presented before, was presented in [3]. The main idea of their algorithm is as follows. Start routing using a greedy type algorithm such as compass routing, until a problem arises, e.g. none of the possible candidates to visit next is strictly *closer* to our destination than our current position. At this point, we switch to a routing algorithm that guarantees delivery, e.g. use geometric routing on arbitrary geometric graphs, until a node strictly closer to our destination than our current position is reached. At this point we switch back to compass routing.

Another modification to our algorithms was presented in [5] in which they use some of the edges in $UW(P_n)$ that are not present in the Gabriel graph of P_n as *shortcuts*. Further they also use and refine techniques presented in [29] that make use of independent sets of vertices of graphs to obtain an algorithm that in practice performs very well.

Stojmenovic and Lin [27] also studied a hybrid single path/flooding algorithm that guarantees delivery of a message.

1.3 DELAUNAY TRIANGULATIONS

A common approach in serial network design is that of finding good architectures that guarantee good performance, e.g. hypercubes, and then building networks that satisfy those architectures. In many applications of wireless communication networks, the cost of the actual radio stations is relatively cheap. In those applications the best way to tackle routing problems is suggested by Theorem 1.1.1. If a wireless network, not necessarily a unit wireless communication network, does not contain the Delaunay triangulation as a subgraph, make it do so. This can be achieved in two different ways. In the first we can deploy extra stations until our objective is reached. The second method to achieve this would be to increase, if the conditions of our application allow us to do so, the transmission power of our stations until the Delaunay triangulation is contained in our wireless communication network. In some instances, e.g. when all nodes of a wireless communication network can communicate with each other, the Delaunay triangulation $\mathcal{D}(P_n)$ can be calculated locally [25]. This follows from that fact that once we have calculated the Voronoi diagram of P_n we also have the Delaunay triangulation [1]. Once the Delaunay trinagulation is calculated, for each vertex we can define for each element of P_n the parameter $Del(p_i)$ to be the distance from p_i to its furthest neighbor in $\mathcal{D}(P_n)$. This value can then be used to determine the minimum transmission power required by p_i so that its furthest neighbor in $\mathcal{D}(P_n)$ can be reached. This in turn will help save energy which is essential in several wireless communication networks [4, 26, 8]. In case that direct communication is not possible, it is still possible to run a distributed set-up procedure to calculate $Del(p_i)$ by forwarding the position of all the nodes of our network to each vertex. The value of $Del(p_i)$ can then be used to adjust the transmission power of p_i.

1.4 CONCLUSIONS

In this paper we reviewed *on-line routing algorithms* on geometric networks and wireless communication networks that *guarantee that a message arrives to its destination*. In practice our algorithms are also competitive, and have the advantage of sending only one copy of a message, in contrast to many of the algorithms developed to date. The algorithms presented here thus eliminate the overhead created by many existing algorithms that send multiple copies of a message, that in turn may lead to traffic problems. A more ample review of routing algorithms in ad hoc networks appears in this issue [24].

Acknowledgments

Supported by a grant from CONACyT-REDII, Universidad Nacional Autónoma de México.

References

1. F. Aurenhammer, and R. Klein, "Voronoi diagrams", in Handbook of Computational Geometry, J.R. Sack and J. Urrutia eds. Elsevier Science Publishers, 2000, pp.201-290.

2. S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)", Proc. MOBICOM, 1998, 76-84.

3. P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks", Proc. of 3rdACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99, Seattle, August 20, 1999, 48-55.

4. J.H. Chang, and L. Tassiulas,"Routing for maximum system lifetime in wireless ad hoc networks", Proc. 37^{th} Annual Alerton Conf. on Communication, Control and Computing, Monticello, IL, Sept. 1999.

5. S. Datta, I. Stojmenovic, and J. Wo, "Internal node and shortcut based routing with guaranteed delivery in wireless networks", Proc. IEEE Int. Conf. on Distributed Computing and Systems (Wireless Networks and Mobile Computing Workshop), Phoenix, AR, April 16-19, 2001, to appear in Cluster Computing.

6. Dijkstra, E.W., "A note on two problems in connection with graphs." *Numer. Math.*, 1 (1959).

7. G.G. Finn, "Routing and addressing problems in large metropolitan-scale network", ISR research report ISU/RR87-180, March 1987.

8. P. Gupta, and P.R. Khumar,"Critical power for asymtotic connectivity in wireless networks", Stochastic Analysis, Control, Optimization and Applications: A volume in honor of W.H. Fleming, W.M. McEneaney, G.Yin, and Q. Zhang (eds.) Birkhauser, Boston, 1998.

9. Hedetniemi, Sandra M.; Hedetniemi, Stephen T.; Liestman, Arthur L. , "A survey of gossiping and broadcasting in communication networks", *Networks*, **18**, 319-349, 1988.

10. C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath,“Flooding for reliable multicast in multiple-hop ad hoc networks” Proc. MOBICOM, 243-254, August 1999.

11. T.C. Hu, and V.O.K. Li, “Transmition range control in multihop packet radio networks”, IEEE Transactions on Communications, **34**,1, 1986, 38-44.

12. T. Imielinski, and J.C. Navas,“GPS-based addressing and routing”, IETF RFC 2009, Rutgers University Computer Science, November 1996.

13. Y-B. Ko, and N.H. Vaidya, “Using location information in wireless ad hoc networks”, IEEE Vehicular Technology Conference (VTC’99), May, 1999.

14. E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks”, Proc. 11^{th} Canadian Conference on Computational Geometry, pp. 51-54, Vancouver Aug. 15-18, 1999.

15. Ko, Y.B., and N.H. Vaidya, “Location-aided routing in mobile ad hoc networks”, Proc. MOBICOM, 1998, 66-75.

16. Liu, P.C. and Geldmacher, “On the deletion of non-planar edges of a graph”, SIAM J. Comp. XXXXX.

17. P.R. Morin, “On line routing in geometric graphs”, Ph.D. Thesis, School of Computer Science, Carleton University, 2000.

18. J.C. Navas, and T. Imielinski, “Geocast-Geographic addressing and routing”.

19. R. Nelson, and L. Kleinrock, “The spatial capacity of a slotted ALOHA multihop packet radio network with capture”, IEEE Transations on Communications, **32**, 6, 1984, 684-694.

20. Papadimitriou, Christos H., Yannakakis, Mihalis, “Shortest paths without a map”. Theoret. Comput. Sci. 84 (1991), no. 1, Algorithms Automat. Complexity Games, 127–150.

21. F.P. Preparata, and M.I. Shamos, “Computational Geometry, and Introduction”, Springer-Verlag 1985.

22. S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," ACM/Baltzer Mobile Networks and Applications, Vol. 1, No. 2, pp. 89-103.

23. Santoro, N., and R. Khatib, “Labeling and implicit routing in networks” *The Computer Journal*, **28**, 1 (1985), 5-8.

24. I. Stojmenovic, “Location updates for efficient routing in ad hoc networks”. In this issue.

25. I. Stojmenovic,“Voronoi diagram and convex hull based geocasting and routing in wireless networks”, Tchnical report TR-99-11, December 1999, SITE, University of Ottawa.

26. Ivan Stojmenovic and Xu Lin,“Power aware localized routing in wireless networks”, IEEE International Parallel and Distributed Processing Symposium, Cancun, Mexico, May 1-5, 2000, 371-376.

27. I. Stojmenovic, and X. Lin, “GEDIR: loop free location based routing in wireless communication network”, IASTED Int. Conf. on Parallel and Distributed Computing Systems, Nov. 3-6, 1999, Boston, MA, USA, 1025-1028.

28. H. Takagi, and L. Kleinrock, “Optimal transmission rates for randomly distributed packet radio terminals”, IEEE Transactions on Communications, **32**, 3, 1984, 246-257.

29. J. Wu, and H. Li,“On calculating connected dominating sets for efficient routing in ad hoc wireless networks”, Proc. DIALM, Seattle, Aug. 1999, 7-14.