Local Solutions for Global Problems in Wireless Networks

J. Urrutia*†

May 1, 2006

Abstract

In this paper, we review a recently developed class of algorithms that solve *global* problems in unit distance wireless networks by means of *local algorithms*. A local algorithm is one in which any node of a network only has information on nodes at distance at most k from itself, for a constant k. For example, given a unit distance wireless network $\mathcal{N}$, we want to obtain a planar subnetwork of $\mathcal{N}$ by means of an algorithm in which all nodes can communicate only with their neighbors in $\mathcal{N}$, perform some operations, and then halt. We review algorithms for obtaining planar subnetworks, approximations to minimum weight spanning trees, Delaunay triangulations, and relative neighbor graphs. Given a unit distance wireless network $\mathcal{N}$, we present new local algorithms to solve the following problems:

1. Calculate small dominating sets (not necessarily connected) of $\mathcal{N}$.
2. Extract a bounded degree planar subgraph $\mathcal{H}$ of $\mathcal{N}$ and obtain a proper edge coloring of $\mathcal{H}$ with at most 12 colors.

The second of these algorithms can be used in the channel assignment problem.

1 Introduction

Let P_n be a set of points in general position on the plane. The unit distance graph $UDG(P_n)$ associated to P_n is a graph whose vertex set consists of the elements of P_n, two of which are connected if they are at distance at most one (see Figure 1). Unit distance graphs are used to model various types of wireless networks, including cellular networks, sensor networks, ad-hoc networks, and others in which the nodes represent broadcast stations with a uniform broadcast range. We shall refer to networks that can be modeled using unit distance graphs as *unit distance wireless* networks, abbreviated as UDW networks. A central assumption in the algorithms presented here is that each element p (a processor in these wireless networks) of P_n will be assumed to have available the coordinates of the position of p. We note that this is a quite realistic assumption for many real-life networks of this type. (For example, the main computers at the Universidad Nacional Autónoma de México are located in Mexico City, not in Tokyo.) For this reason, algorithms of the type presented here are known as *position-based* algorithms. A few years ago, this supposition would

*Supported by CONACYT of Mexico, Proyecto 45876, and PAPIIT (UNAM) of Mexico, Proyecto IN110802.
†Instituto de Matemáticas, Universidad Nacional Autónoma de México, México D.F. México

have been hard to justify, but the advent of systems such as GPS make it a realistic assumption in many cases. Moreover, GPS systems are constantly improving in precision, cost, and size; i.e. the devices used to calculate the position of a processor are becoming ever smaller, cheaper to produce, and more precise.

In this paper we will review a new class of algorithms that have been developed to solve several problems in UDW networks. We consider algorithms that employ only *local computations* and the position of the vertices of the networks to solve *global problems* on UDW networks. By a *local algorithm* we mean an algorithm in which the only information possessed by the processor at each node of a network is information on its own neighbors (in general, nodes at hop distance at most k, k a constant, in most cases less than or equal to 4) and the fact that they belong to a UDW network. No further information regarding the rest of the network is available to the processor, e.g. neither the number of nodes, where they are located, the topology of the network, nor any other global information.

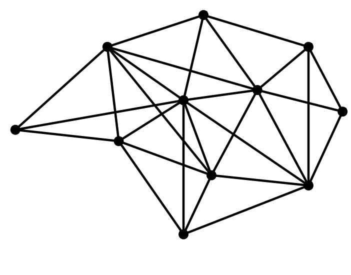

Figure 1: A unit distance graph

A key consideration that led us to impose the *local knowledge* restriction on UDW networks is the fact that modern networks are highly dynamic. The advent of networks such as the Internet has imposed new constraints that are very challenging from a theoretical as well as from an applied point of view. The traditional assumption that, up to some degree, we know the topology of the networks is now less relevant to real-life networks and thus ever harder to justify. The inherent nature of communication networks is becoming more dynamic; that is, nodes appear and disappear constantly. In networks such as cellular phone networks, many of the nodes are not static; that is, they change their position frequently or continually. As a consequence, the application of algorithms such as Dijkstra's shortest path algorithm [17] is no longer possible in many real life problems.

A natural question that comes to mind is whether under such stringent restrictions, we can solve any meaningful and useful, or at least theoretically interesting problems? Surprisingly the answer is yes. What is even more surprising is that many of the algorithms we will review here have real applications to wireless networks that can be modeled with unit distance graphs, e.g. cellular networks, sensor networks, ad-hoc networks, and others. We are confident that many of the results presented here will be generalized in the near future to geometric networks—networks whose vertices are sets of points on the plane, and whose edges are straight line segments joining pairs of adjacent vertices.

Among the problems we will review here are the following: Given a connected UDW network, can a local algorithm be designed to extract a connected planar subgraph from it? Can a connected subgraph of a UDW network be obtained such that the weight of the graph (the sum of the lengths of its edges) is close to the weight of a minimum weight spanning tree of the UDW network? Some new results such as the the calculation of small dominating sets (not necessarily connected), edge colorings, and vertex colorings with few colors (with important applications to the frequency assignment problem in wireless networks) in UDW networks will also be presented.

We begin by reviewing the basic mathematical structures that allow us to develop local algorithms in unit distance wireless networks. Many important challenges remain before the results presented here can be fully implementable in real wireless networks, such as the fact that in many cases the nodes in real networks are not as uniform as one might want, and the presence of physical obstacles (e.g. city buildings, mountains, etc.) that have an effect on the effective range of a broadcast station; see [6, 12, 34, 37, 38]. An inspection of the list of references provided in this paper gives an indication of the many issues arising from implementations of results similar to those reviewed here.

2 The first algorithms: Face Routing and planarization of wireless networks

Let us recall that a *geometric graph* G is a graph whose vertices are points on the plane and whose edges are straight line segments joining pairs of adjacent vertices in G. We say that G is planar if no two of its edges intersect except perhaps at their end points. The key factor that encouraged the development of *local algorithms* for UDW networks, and thus for many types of wireless networks, was the solution to the following problem [36].

Problem 1 *Let G be a geometric planar graph. Assume that at each vertex v of G we have available the position (coordinates) of v and the position of its neighbors in G. Is there a deterministic algorithm that will allow an agent $\mathcal{A}$ standing at a vertex u to travel to a vertex v of G under the following conditions:*

1. *$\mathcal{A}$ has a constant amount of memory; that is, at any point in time $\mathcal{A}$ knows the position of u and v, and the positions of a constant number of nodes on G.*
2. *When the agent visits a vertex w of G, it can use the list of vertices (and their positions) adjacent to w.*
3. *$\mathcal{A}$ is not allowed to leave any marks along its way.*

Let us further clarify the first restriction. Under this condition, $\mathcal{A}$ has a constant amount of memory in which it can store a constant number of positions of elements of G. After $\mathcal{A}$ has filled the memory available to it, to remember a new position it must erase or forget the position of some other node in its memory. It is clear from this restriction that the agent will never have global knowledge of G.

It might seem that some of these restrictions are unnecessarily strong, but if we consider the behavior of networks such as the Internet, then it becomes clear that these conditions are relevant. For example, if a message left a mark each time it passed through a given node in the network, all the memory available at that node would soon be used up, hence the necessity of the third condition. The restriction on learning is due to the limitation on having any kind of global knowledge about G, some of the advantages of which will be highlighted presently. The same criterion applies to the second condition, but in this case the restriction is imposed on the nodes of a network.

Some troublesome questions come to mind when trying to solve this problem, for example: How will $\mathcal{A}$ detect that it has entered a dead end, and how can it escape from the cul-de-sac? Recall that our agents cannot learn much, and if $\mathcal{A}$ returns to a previously visited node, it will likely not remember that it has already been there. Problem 1 was solved in [36], where the algorithm now known as *Face Routing* was first introduced (in that paper, the algorithm was called *Geometric Routing II*). It turns out that in traveling from u to v, $\mathcal{A}$ will need to remember, aside from the positions of u and v, the following information:

- The position of the last two vertices it visited.
- A point s, initially set to u.
- A distance d, initially set to 0.

Let ℓ be the line segment joining u and v. Observe that ℓ will always be available to $\mathcal{A}$, as $\mathcal{A}$ always remembers the coordinates of u and v. $\mathcal{A}$ will now proceed as follows:

First it will detect the face $\mathcal{F}$ incident to u that is intersected by ℓ. $\mathcal{A}$ will now traverse $\mathcal{F}$ until it returns to u or, if v belongs to $\mathcal{F}$, it arrives at v, in which case it stops. Each time an edge e of $\mathcal{F}$ is traversed, $\mathcal{A}$ proceeds as follows:

i) If e intersects ℓ, calculate the distance d' of the intersection point of ℓ and e to u. If $d' > d$ then reset d to d'.

ii) If e does not intersect ℓ, continue the traversal of $\mathcal{F}$.

If v is not a vertex of $\mathcal{F}$, the agent $\mathcal{A}$ will return to $s = u$ carrying a value d. At this point the agent will again traverse $\mathcal{F}$ until arriving at the edge e such that the intersection point p of e and ℓ is at distance d from s. Edge e is in the boundary of two faces $\mathcal{F}$ and $\mathcal{F}'$. The agent now resets $\mathcal{F}$ to be $\mathcal{F}'$, s to p, and restarts from s, i.e. it now proceeds to traverse $\mathcal{F}'$. It is straightforward to see that following this strategy, if G is connected, $\mathcal{A}$ will always get to v. Note that it is necessary to check for intersections of edges of G with ℓ, not with the line determined by u and v. Figure 2 illustrates the route that will be taken by the agent traveling from u to v.

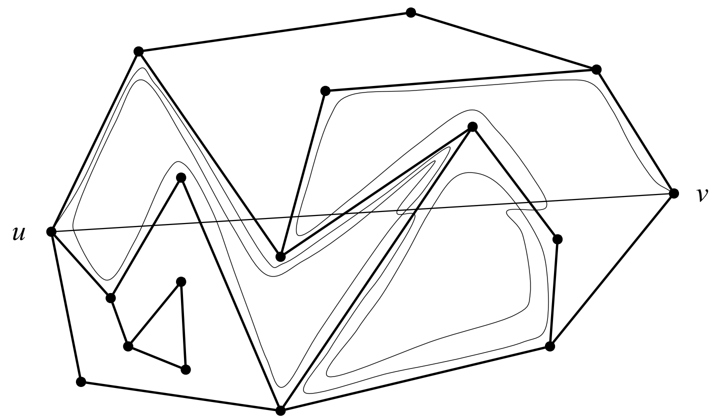

Figure 2: Traversal from u to v.

2.1 Extracting a planar subgraph from a unit distance graph

To simplify the presentation, all networks considered from this point on will be *UDW* networks. Suppose now that we wish to route in a *UDW* network $\mathcal{N}$. The existence of Face Routing makes the following problem very appealing:

Problem 2 *Given a UDW network* $\mathcal{N}$*, can a local algorithm be found to extract a planar subgraph such that if* $\mathcal{N}$ *is connected, then the subgraph is also connected?*

Solving Problem 2 and using face routing in the resulting planar subgraph would immediately give a powerful on-line local algorithm for routing in *UDW* networks [10]. Recall that in a local algorithm a processor of $\mathcal{N}$ can communicate only with its at distance at most k, and that the only information it can learn about them is their positions (not even their degree or any further information), k a constant. After learning its neighbor's positions, each processor performs some operations and then terminates. We may note that to solve Problem 2, a node of $\mathcal{N}$ only needs information about its neighbors at most one hop away; in fact it never receives or collects information about neighbors two or more hops away.

To obtain the planar subgraph of $\mathcal{N}$, we will calculate what is known as its *Gabriel subgraph* [27]. An edge joining two vertices u and v in $\mathcal{N}$ is called a *Gabriel* edge if the circle whose diameter is the line segment joining u to v contains no other vertex of $\mathcal{N}$. The Gabriel subgraph of $\mathcal{N}$ is the subgraph containing all the Gabriel edges of $\mathcal{N}$; all other edges are discarded. For example, if P_4 is the set of points shown in Figure 3(a), then the edges joining p to r and t will be kept, and the edge joining p to q will be discarded. The Gabriel subgraph of the graph shown in Figure 1 is given in Figure 3(b). It is well known that the Gabriel subgraph of any unit distance network $\mathcal{N}$ is planar, and that if $\mathcal{N}$ is connected, then its Gabriel subgraph is also connected [10]. It is clear that if a node v of a *UDW* network knows the position of all its neighbors, it can easily identify its Gabriel edges (if an edge $v-w$ is eliminated by a node r of $\mathcal{N}$, r is closer to v than to w, i.e. r is also adjacent to v). This solves Problem 2.

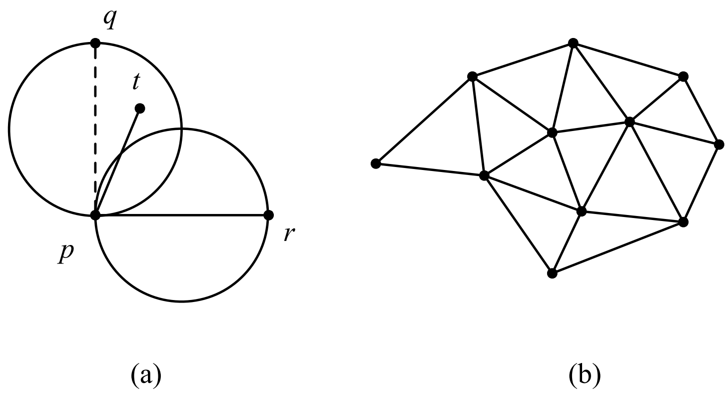

Figure 3: Edges $p-t$ and $p-r$ will be kept; edge $p-q$ will be removed.

2.2 Proximity graphs and UDW networks

Although the first way found to planarize UDW networks was by using Gabriel graphs, it is by no means the only method known. In the computational geometry literature, there are several classes of planar geometric graphs arising from what are known as *Proximity Graphs* [33, 59], most of which can be extracted from UDW networks using local algorithms. We briefly review some of them.

2.2.1 Relative neighbourhood graphs

Given two points p and q on the plane, let $\delta(p,q)$ be the distance from p to q. We define $C(p,\delta(p,q))$ as the circle with center at p and radius $\delta(p,q)$. The *lune* Λ_{pq} is now defined as $C(p,\delta(p,q)) \cap C(q,\delta(q,p))$. The relative neighbourhood graph $RNG(P_n)$ of a point set P_n is defined as follows: The vertices of $RNG(P_n)$ are the elements of P_n, two of which, say p and q, are adjacent if there are no elements of P_n in Λ_{pq}. The edges of $RNG(P_n)$ are called *rng* edges. RNGs, introduced by Toussaint [59] in 1980, can be constructed in $O(n \ln n)$ time [58].

In a similar way to that used to show that the Gabriel subgraph of a connected unit distance graph G is connected, it is easily seen that the subgraph of G containing only those edges of G that are *rng* edges is always planar, and that if G is connected, the subgraph is also connected. It is also clear that *rng* edges can be identified locally [2, 43]. Figure 4 gives the *rng* subgraph of the unit distance graph shown in Figure 1.

It is easy to see that any *rng* edge is also a Gabriel edge of G, but the converse is not true. As a consequence, for some applications such as routing, it is usually better to work with the Gabriel subgraph of G.

In a different setting, RNGs have also been analyzed in [41] in connection to minimum weight spanning trees in unit distance graphs. Li shows that in general, the weight of $RNG(P_n)$ can

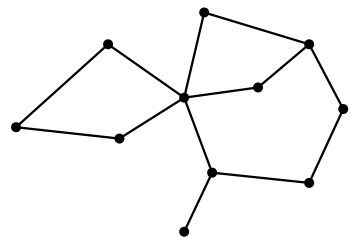

Figure 4:

be arbitrarily large compared to the weight of the minimum weight spanning tree, abbreviated $MWST$. It should be pointed out however, that in general the number of edges in the $RNG(P_n)$ is close to $(1.278 + o(1))n$ (the expected degree of almost all vertices of a RNG is 2.5575...); see [16]. Moreover, since the RNG of any point set contains a minimum weight spanning tree, the expected number of edges in a RNG that are not in a minimum weight spanning tree is approximately $.27n$, and thus in general we can expect RNG graphs to be good approximations to $MWST$.

2.2.2 Delaunay graphs

Given a set of points P_n in general position, the Delaunay triangulation $Del(P_n)$ is the graph with vertex set P_n such that two vertices p and q of $Del(P_n)$ are adjacent if there is a circle passing through p and q that does not contain any other element of P_n. It is easy to see that $Del(P_n)$ induces a triangulation of the convex hull of P_n, $Conv(P_n)$, in which for any triangle t of this triangulation, the circle passing through the vertices of t contains no other element of P_n (see Figure 5).

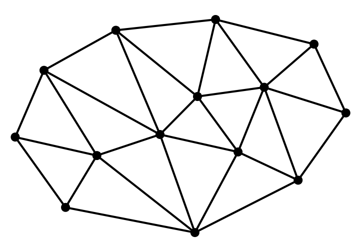

Figure 5: A Delaunay triangulation.

Although it is not possible to construct the Delaunay subgraph of a UDW network locally (the circle through three almost aligned points whose pairwise distances are less than or equal to one may be arbitrarily large), they have an important property that makes them desirable for routing; namely that Compass Routing works on them [36]. In other words, the most elementary form of greedy routing (directional, hence the name Compass Routing) works for for Delaunay subgraphs. There is, however, a closely related family of graphs, *localized Delaunay* triangulations [28, 20, 41, 44] that inherits many properties of Delaunay triangulations, and which *can* be calculated locally. Let $\mathcal{N}$ be a UDW network, and u, v, w three nodes of $\mathcal{N}$. We say that the triangle $\Delta(u, v, w)$ is a k-localized Delaunay triangle of $\mathcal{N}$ if the circle passing through u, v and w does not contain any node in $\mathcal{N}$ at distance less than or equal to k hops from u, v, or w, $k \geq 1$. The k-localized subgraph of $\mathcal{N}$

is the graph with the same set of nodes as $\mathcal{N}$, and whose edges are all those edges of $\mathcal{N}$ that are contained in a k-localized Delaunay triangle of $\mathcal{N}$.

There are several reasons why k-localized Delaunay triangulations are desirable. First of all, they are planar, and perhaps more importantly, they contain all the Gabriel edges of $\mathcal{N}$. This is desirable, as Face Routing will, in general, terminate faster in geometric graphs with many edges (in particular if many of these faces are triangles); see [44]. Moreover, as mentioned above, compass routing works very well in Delaunay triangulations [36]. This idea was used in [20] to route in k-localized Delaunay graphs.

2.2.3 Minimum weight spanning trees

It is easy to see that it is not possible to calculate the $MWST$ of a UDW network with a local algorithm. For example, if we have n points on a circle C such that the distances between consecutive nodes on C are $1 - \epsilon_i$, $i = 1, \ldots, n$, finding the $MWST$ of $\mathcal{N}$ involves identifying the smallest ϵ_i. On the other hand, it is possible to find a planar subgraph $\mathcal{H}$ of $\mathcal{N}$ such that the weight of $\mathcal{H}$ (i.e. the sum of the lengths of its edges) is within a constant factor of the weight of a $MWST$. The following algorithm was proposed by Li [43].

Given a vertex $v \in \mathcal{N}$, let $H(v, k)$ be the subgraph of $\mathcal{N}$ induced by the vertices of $\mathcal{N}$ at distance less than or equal to k from v, including v itself. Given a constant k, proceed as follows:

First, define a linear ordering on the edges of $\mathcal{N}$ as follows: an edge $u - v$ is smaller than an edge $r - s$ if the length of $u - v$ is less than the length of $r - s$. In the event that they have the same length, the edge with the rightmost topmost node is taken to be longer. If they have the same length and the same rightmost vertex, the one with the leftmost topmost vertex is considered as the longer.

Algorithm $LocalMST_k$

1. Each node v computes the minimum spanning tree $T_{k,v}$ of the subgraph of $H(v, k)$ (according to the linear ordering defined above).
2. An edge $u - v$ of $\mathcal{N}$ is kept if it appears in both $T_{k,u}$ and $T_{k,v}$.

In [43], it is proved that the weight of the resulting subgraph is within a constant factor of the weight of a minimum weight spanning tree of $\mathcal{H}$. In [12] it is proved that the weight of the solution obtained is in fact at most $\frac{k+1}{k-1}$ times the optimal solution, $k \geq 2$. The various local $MWST_k$ can be used to perform energy-efficient broadcast [43].

2.2.4 Some comments on the complexity, robustness and maintenance of Face Routing, the extraction of planar subgraphs, and future work

It is well known that the Gabriel subgraph of a geometric graph can be calculated in $O(n \ln n)$ time [33, 47]. The same is true of RNGs, Delaunay graphs, and minimum weight spanning trees of geometric graphs. However in practice, most UDW networks have the property that their vertices tend to be configured in such a way that all their nodes have few other nodes within distance one (in most cases "few" means a constant). Under these circumstances, to extract the edges incident to each vertex, it is preferably to use an algorithm that is simple, easy to implement and maintain. The obvious quadratic time algorithms are recommended. Moreover, since the graphs are all planar, the average number of edges a vertex must keep is at most six. Thus under most practical circumstances, the amount of work that a node in a UDW network has to do to extract and maintain the chosen subgraph is minimal.

On the other hand, the advantages of using Face Routing on planar subgraphs of UDW networks are well known and considerable. The first is based on the robustness of Face Routing. Consider an agent $\mathcal{A}$, running face routing on a geometric planar graph G while traversing from node u to node v. Suppose that before $\mathcal{A}$ reaches v, a failure occurs somewhere on the network, i.e. a node w goes permanently down (this is commonly the case in sensor networks). Observe that by the nature of Face Routing, unless the failure happens just where the agent is located and *kills* $\mathcal{A}$ (i.e. a failure occurs at the node or edge where an agent is located) or disconnects G, it will not prevent $\mathcal{A}$ from reaching v. The reason for this is obvious. To make the situation more interesting, assume that the node w is on what would have been the path followed by $\mathcal{A}$ in its traversal of G from u to v, and that w fails before it is reached by $\mathcal{A}$. Since until $\mathcal{A}$ reaches any node of G, it is never aware of its existence, Face Routing will simply proceed as if the initial graph was $G - w$ instead of G. Thus if $G - w$ is connected, $\mathcal{A}$ will reach v. More potentially disruptive to Face Routing are failures of nodes or edges followed by recoveries. This could make a message fall into a cycle away from ℓ.

Face routing also eliminates the need to maintain global structures such as *routing tables* which, in highly dynamic networks such as UDW networks, have to be updated on a regular basis. Finally, we observe that in ideal conditions, Face Routing sends a single copy of a message. This considerably reduces the amount of traffic generated, especially when compared to broadcast algorithms.

We believe that one of the most important problems to solve in this area of research is to overcome Face Routing's limitation of working only on planar networks. For some reason that we have not yet fully understood, it is extremely hard to deal with even a small number of edge crossings in an otherwise almost planar network. An extremely challenging area of future research is that of developing routing algorithms for UDW networks in R^3. Is there an equivalent to Face Routing in R^3? Another problem that we believe is very interesting is to develop algorithms similar to face routing for networks in which the broadcast stations have different broadcast ranges. A satisfactory solution is lacking even for networks in which any station has either one or the other of two possible broadcast ranges. The basic problem here is that this kind of network has directed edges. Face Routing looks deceptively easy, but as soon as the conditions on the planarity of a network are relaxed even slightly, we are essentially back to square one, that is, we must return to older techniques such as broadcast algorithms. The alert reader will infer from these comments that the problem of generalizing Face Routing to R^3 is, at least for now, impossible to achieve, as it requires development of a technique to deal with non planar graphs.

3 New local algorithms for edge colorings and dominating sets

We now present two new algorithms; the first colors the edges of a planar subgraph of a *UDW* network, and the second obtains small dominating sets of vertices. These algorithms are local, and use a new idea that employs tilings of the plane. The basic idea is to subdivide the plane into squares of size $k \times k$. Since each processor in a *UDW* network has its coordinates available, we can solve a particular problem, i.e. finding a dominating set, by first solving it within each square, and, in a second iteration, *merge* the solutions obtained for each square.

3.1 Edge coloring

We now proceed to solve the following problem: Given a *UDW* wireless network $\mathcal{N}$, using a local algorithm, extract a planar subgraph $\mathcal{H}$ from $\mathcal{N}$ and produce a proper edge coloring of $\mathcal{H}$. A proper edge coloring of a graph G is a coloring of its edges such that any two edges that share a common vertex receive different colors. The chromatic index of G is the smallest integer k such that there is a proper k-edge coloring of G. We proceed as follows.

Begin by obtaining the subgraph $\mathcal{H}$ of $\mathcal{N}$ generated by the $LocalMST_2$ algorithm. It is known that the maximum degree in $\mathcal{H}$ is at most 5; see [12]. A classical graph theory result known as Vizing's Theorem [8] asserts that the chromatic index of a graph is less than or equal to its maximum degree plus one. It follows that the chromatic index of $\mathcal{H}$ is at most six. To apply Vizing's Theorem however, we would need to know all of $\mathcal{H}$, and we are allowed to use only a local algorithm. We show next how to obtain an edge coloring of $\mathcal{H}$ using at most 12 colors.

Subdivide the plane into subsquares of size 2×2; for example take the set of squares $S_{i,j}$ such that the vertices of $S_{i,j}$ are the points $\{(2i, 2j), (2(i+1), 2j), (2(i+1), 2(j+1)), (2i, 2(j+1))\}$. A node v with coordinates (x, y) in $\mathcal{N}$ is assigned to $S_{i,j}$ if $2i \leq x < 2(i+1)$ and $2j \leq y < 2(j+1)$.

Let $\mathcal{H}_{i,j}$ be the subgraph of $\mathcal{H}$ induced by the vertices of $\mathcal{N}$ assigned to $S_{i,j}$. $\mathcal{H}_{i,j}$ is not necessarily connected. In fact two vertices within a given square can have an arbitrarily large hop distance in $\mathcal{N}$! Observe, however, that the hop distance between any two vertices in a component of $\mathcal{H}_{i,j}$ is at most 4 in $\mathcal{H}_{i,j}$. Then any vertex v of $\mathcal{H}$ can calculate the connected component of $\mathcal{H}_{i,j}$ to which it belongs by collecting the coordinates of all its neighbors at (hop) distance at most 4 in $\mathcal{H}$, discarding those elements that do not belong to the subraph $\mathcal{H}_{i,j}$. We can now select a vertex in each component of $\mathcal{H}_{i,j}$ (e.g. the topmost leftmost vertex) that will then apply Vizing's Theorem locally and 6-color the edges of its component using colors $\{1, \ldots, 6\}$.

The remaining problem is to color the edges that join vertices in adjacent squares. It is well known that any two edges incident to a vertex v in a minimum weight spanning tree of a geometric graph induce angles of size at least $\frac{\pi}{3}$ at v. It follows easily that $\mathcal{H}$ has the same property. The following lemma is an immediate consequence.

Lemma 1 *Let v be any vertex assigned to $\mathcal{H}$ that belongs to $S_{i,j}$. Then v is adjacent to at most* 3 *vertices of $\mathcal{H}$ above or on the horizontal line $y = 2j + 2$. See Figure 6.*

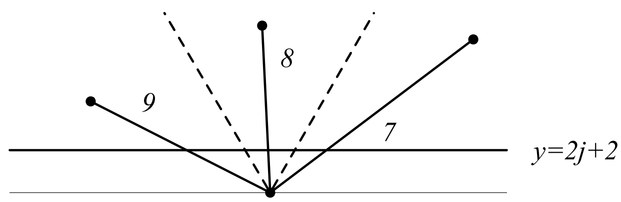

Figure 6:

Let v be as in this lemma, and v—w an edge such that w is above $y = 2j+2$. We now assign a color to v—w according to the following criterion. If the angle in the counter clockwise direction between $y = 2j+2$ and the line segment joining v to w is less than or equal to $\frac{\pi}{3}$, we color v—w with color 7; if the angle is greater than $\frac{\pi}{3}$ but smaller than or equal to $\frac{2\pi}{3}$, v—w is colored 8, otherwise it is colored 9; see Figure 6. It is now easy to see that if a vertex w above $y = 2j+2$ is adjacent to two vertices below $y = 2j+2$, the edges connecting w to these vertices receive different colors. In a similar way we can color the remaining *uncolored edges* joining vertices of H that cross vertical lines of the form $x = 2i+2$ with colors 10, 11 and 12. Thus we have proved:

Theorem 1 *Let $\mathcal{N}$ be a UDW network. Then a planar subgraph $\mathcal{H}$ of $\mathcal{N}$ with maximum degree* 5 *can be extracted using a local algorithm; indeed if $\mathcal{N}$ is connected, $\mathcal{H}$ is also connected. Furthermore, the edges of $\mathcal{H}$ can also be* 12*-colored using a local algorithm. The local algorithm is such that each vertex of $\mathcal{N}$ collects information of vertices at distance at most* 4 *from itself.*

To conclude this section, we note that this algorithm can be modified to obtain edge colorings of a *UDW* network $\mathcal{N}$ in which, for any two vertices at distance less than or equal to k, the edges incident to them receive different colors. This algorithm is useful in the channel assignment problem for wireless networks. The details will be given in a forthcoming paper. The number of colors used will obviously depend on the *density* of the vertices of $\mathcal{N}$. For example, if there are at most m nodes within any unit square and all of them broadcast with the same power, at least cm colors are required, for a constant c.

3.2 Dominating sets

Given a graph G with vertex set $V(G)$, a dominating set is a subset $S \subset V(G)$ such that every vertex of G is in S or is adjacent to a vertex in S. There has been much research on calculating *connected dominating sets* for unit distance wireless networks; see [2, 3, 63, 43, 54, 52, 61, 64]. One of the motivations for studying connected dominating sets is the construction of a backbone network in a wireless network. The algorithms used to obtain connected dominating sets are, however, not local in our terminology, since in these algorithms, there are times when a node has to wait for some information to disseminate through large portions of the network.

In this section, we develop a local algorithm to obtain a dominating set of a *UDW* network that generates a dominating set whose size is within a constant of the optimal solution. One of our motivations is the fact that these vertices can be used to store backup information on a network; e.g. see [29]. We prove the following.

Theorem 2 *Given any UDW network, a dominating set of size at most* 15 *times the size of the optimal dominating set of* $\mathcal{N}$ *can be calculated using a local algorithm.*

We remark again that the dominating sets obtained here are not necessarily connected. The algorithm used follows a similar idea to the edge coloring algorithm above; that is, the plane is partitioned into convex regions of bounded size. The size of the dominating sets obtained is bounded using an approach similar to that used in [24]. An independent set I of a graph G is a subset of vertices of G such that no two of them are adjacent. We say that I is maximal if any vertex of G not in I is adjacent to at least one element in I.

We use the following well known lemmas.

Lemma 2 *Let G be any graph, and D a minimal dominating set of G, and let I be a maximal independent set of G. Then $|D| \leq |I|$.*

The following lemma is easy to prove for unit distance graphs.

Lemma 3 *Let D be the smallest dominating set of a unit distance graph $\mathcal{N}$. Then the size of any independent set of $\mathcal{N}$ is at most* $5|D|$.

Proof: Let D be a dominating set of $\mathcal{N}$. For each vertex v of D consider the circle of unit radius centered at v. Then all vertices of $\mathcal{N}$ lie in at least one of the circles centered at an element of D. Observe now that an independent set S can have at most five elements in each of these circles. The result follows. ∎

To obtain a small independent set using a local method, we now proceed as follows.

In a procedure similar to that given in the previous section, split the plane into a set of regular hexagons with edges of unit length. Assume that the center of one of these hexagons is the origin; see Figure 7.

For each hexagon H_k of this hexagonal partition of the plane, let $\mathcal{N}_k$ be the subgraph of $\mathcal{N}$ induced by the nodes of $\mathcal{N}$ within H_k. In a similar way to that described in the previous section, the elements of $\mathcal{N}$ can decide to which cell H_k of the hexagonal partition they belong to. Proceed to calculate a minimum dominating set D_k of $\mathcal{N}_k$ in each hexagon. Let I_k be the largest independent set in each $\mathcal{N}_k$. By Lemma 2, $|D_k| \leq |I_k|$. Therefore to bound the size of the union of the dominating sets it is enough to bound the size of the union of the sets I_k.

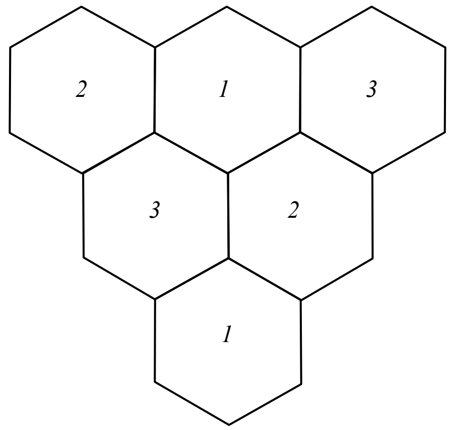

Figure 7:

We now prove that the cardinality of the union of the set of independent sets I_k over the set of hexagons of the partition is at most 15 times the size of the smallest dominating set of $\mathcal{N}$. To see this, observe that the set of hexagons of the hexagonal partition can be 3-colored in such a way that adjacent hexagons receive different colors; see Figure 7. Observe that for each color i, $1 \leq i \leq 3$, the union of the independent sets of all hexagons with the same color also forms an independent set in $\mathcal{N}$, and thus by Lemma 3 has at most five times the size of the smallest dominating set of $\mathcal{N}$, $i = 1, \ldots, 3$. Thus the union of all the dominating sets of all the hexagons has cardinality at most 15 times that of the cardinality of the smallest dominating set of $\mathcal{N}$. This proves Theorem 2. ■

Figure 8 illustrates an example in which the error is five times the optimal. To construct the example, we take a hexagon with sides of length 1, and five points within it such that the distances between them are greater than one; see Figure 8(a). We then paste multiple copies of this configuration, as shown in Figure 8(b). The local algorithm would then produce five elements per hexagon. However in the global solution, we need to take only one point per hexagon, the one close to the bottom vertex of the leftmost edge of each hexagon (except for those hexagons lying on the boundary of the union of the hexagons).

It is worth noting that that in several examples we have tried, the local algorithms for obtaining dominating sets produce sets of size at most twice the size of a minimum dominating set. An interesting problem is to improve on the 15 times factor with respect to the minimum dominating set. From an algorithmic point of view, within each hexagon H_k, the size of the largest independent set contained in it is at most five, and thus the smallest dominating set is bounded above by the same constant. This implies that we can calculate it in polynomial time.

A closing comment

We would like to mention that in a forthcoming paper [13] using similar techniques to those introduced in Section 3 of this paper, we obtain a local algorithm to extract a connected planar subgraph of a UDW network and 7-color its vertices.

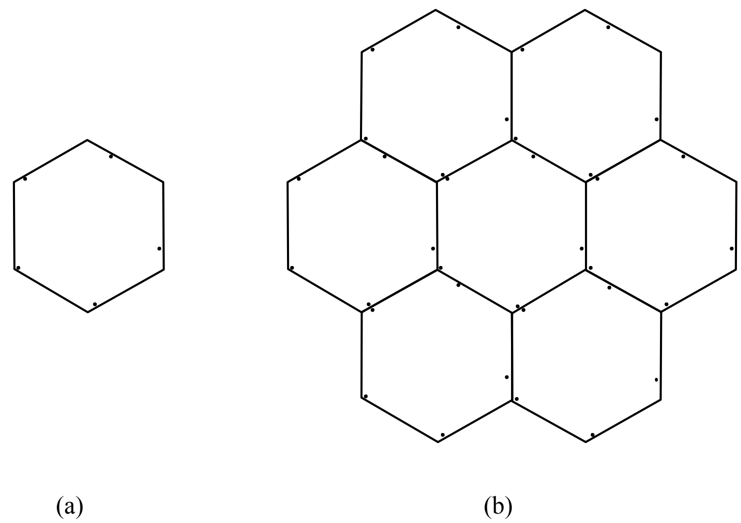

Figure 8:

3.3 Conclusions

In this paper, we have reviewed and presented some local algorithms on UDW networks. We concentrated only on the basic mathematical principles of *local solutions for global problems.* There are numerous issues involved in improving or implementing the algorithms presented here, however a complete review is beyond the scope of the present paper, as hundreds of papers in this area have been published since 2000. Many papers have studied strategies to improve on Face Routing, in some cases developing strategies that extract planar spanners of UDW networks with more edges, such as localized Delaunay graphs. In other cases, mixed strategies have been proposed. For example, one might start with a greedy strategy such as Compass Routing, and if a message gets stuck somewhere or the distance to the destination does not decrease, switch to Face Routing. Energy consumption is also an important factor in networks such as sensor networks. Research in this direction has focused on developing networks closely related to minimum weight spanning trees, e.g. connected dominating sets, Voronoi diagrams and geometric spanners with small stretch factor, among others. We provide a list of references, by no means exhaustive, that can be helpful to those interested in the areas of research covered in this paper. Finally, we would like to thank the organizers of Adhoc Now 2005, Violet R. Syrotiuk and Edgar Chavez, for inviting and encouraging us to write this paper.

References

[1] A. Agarwal and S.R. Das, Dead reckoning in mobile ad hoc networks, Wireless Communications and Networking, 2003. WCNC **Vol. 3** (2003). 2003 IEEE, March 6–20, 2003, pp. 1838–1843.

[2] K.M. Alzoubi, X.Y., Li, Y. Wang and P.J. Wan, Geometric spanners for wireless ad hoc networks, IEEE TPDS **14** (2003), pp. 408–421.

[3] K.M. Alzoubi, K.M. and P.J. Wan, O. Frieder, Maximal independent set, weakly-connected dominating set, and induced spanners in wireless ad hoc networks, *Int. J. Foundations of Computer Science* **14** (2003), pp. 287–304.

[4] S. Ansari, L. Narayanan and J. Opatrny, A generalization of the face routing algorithm to a class of non-planar networks, *Proceedings of Mobiquitous 2005* (2005) pp. 213–225.

[5] M. Barbeau, E. Kranakis, D. Krizanc and P. Morin, Improving distance-based geographic location techniques in sensor networks, Ad-Hoc, Mobile, and Wireless Networks: Third International Conference, ADHOC-NOW 2004, Vancouver, Canada, July 22–24, 2004. *Lecture Notes in Computer Science 3158*, (2004) pp. 197–210.

[6] L. Barriere, P. Fraigniaud, L. Narajanan and J. Opatrny, Robust position-based routing in wireless ad hoc networks with irregular transmission ranges, *Wireless Communications and Mobile Computing* **3**(2003), pp. 141–153.

[7] L. Blazevic, , S. Giordano and J.-Y. Le Boudec, Self-organized terminode routing, *Cluster Computing* **5(2)** (2002), pp. 205–218.

[8] A. Bondy and and U.S.R. Murty, *Graph Theory With Applications*, (Macmillan Press, 1976).

[9] P.E. Boone, E. Chavez, L. Gleitzky, E. Kranakis, J. Opatrny, G. Salazar and J. Urrutia, Morelia Test: Improving the efficiency of the Gabriel test and Face Routing in ad-hoc networks, in: *Proceedings of SIROCCO 2004*, Springer Verlag, LNCS, 3104, R. Kralovic and O. Sykora, eds. June 2004, Smolenice Castle, Slovakia.

[10] P./ Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks', in: *Proc. of 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99*, Seattle, August 20, (1999) pp. 48–55.

[11] J. Cartigny, F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic, Localized LMST and RNG based minimum energy broadcast protocols in ad hoc networks, in: *Proc. IEEE INFOCOM, San Francisco, CA, USA*, April 1-3, 2003. Ad Hoc Networks, **3(1)** (2005) pp. 1–16.

[12] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho and J. Urrutia, Local construction of planar spanners in unit disk graphs with irregular transmission ranges. LATIN 2006, 7th Latin American Symposium, Valdivia, Chile, March 2006, LNCS 3887, pp. 286–297.

[13] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho and J. Urrutia, Local coloring of the vertices of planar spanners in unit disk graphs, in preparation.

[14] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, H. Tejeda and J. Urrutia, Half-space proximal: a new local test for extracting a bounded dilation spanner of a unit disk graph. OPIDIS, 9th International Conference on Principles of Distributed Systems, Dec. 2005, Pisa, Italy.

[15] S.M. Das, H. Pucha, Y.C. Hu, Performance comparison of scalable location services for geographic ad hoc routing, in: *Proceedings of IEEE INFOCOM 2005*, Miami, FL, March 13–17, 2005.

[16] L. Devroye, On the expected size of some graphs in computational geometry, *Computers and Mathematics with Applications* **15** (1988), pp. 53–64.

[17] E.W. Dijkstra, A note on two problems in connexion with graphs. *Numerische Mathematik* **1** (1959), pp. 269–271.

[18] E. Erwan, A. Fladenmuller and P. Guy, Geographical routing for heterogeneous wireless systems, IFIP Med-Hoc-Net Sardaigna, Italy, Sept. 2002.

[19] F. Fan and E.S. Biagioni, An approach to data visualization and interpretation for sensor networks, in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences 2004*, Jan. 5–8, 2004, pp. 63–71.

[20] Q. Fang, J. Gao and L. J. Guibas, Locating and bypassing routing holes in sensor networks, in *Proceedings of IEEE INFOCOM'04, March 2004*, **23(1)** (2004), pp. 2458–2468.

[21] M. Fayed and H.T. Mouftah, Characterizing the impact of routing holes on geographic routing, ICW, 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05), (2005) pp. 401–406.

[22] T. Fevens, A. Eddien and B. Naciri Bennai, Randomized AB-Face-AB routing algorithms in mobile ad-hoc networks, 4th. Ad-hoc, mobile and wireless Networks, Cancun, Mexico, Oct. 2005. LNCS **3738** (2005), pp. 43–56.

[23] T. Fevens, G. Kao and J. Opatrny, Position-based routing on 3-d geometric graphs in mobile ad hoc networks, in *Proceedings of CCCG 2005* (2005), pp. 88–91.

[24] S. Funke, A. Kesselman, U. Meyer and M. Segal, A simple improved distributed algorithm for minimum CDS in unit disk graphs, 1st IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) 2005, Montreal.

[25] H. Fußler, M. Mauve, H. Hartenstein and M. Kasemann, A comparison of routing strategies for vehicular ad hoc networks, TR-02-003, Dept. Comp. Sci., Univ. of Mannheim, July 2002.

[26] Fußler, H., J. Widmer, M. Kasemann, M. Mauve, H. Hartenstein, Beaconless position based routing for mobile ad hoc networks, TR-03-001, Dept. Comp. Sci., Univ. of Mannheim, Feb. 2003.

[27] K.R. Gabriel and R.R. Sokal, A new statistical approach to geographic variation analysis, *Systemic Zoology* **18** (1972), pp. 259–278.

[28] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang and A. Zhu, Geometric spanner for routing in mobile networks, ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc, Long Beach, California, USA, October 4–5, 2001, pp. 45–55.

[29] C. Georgiou, E. Kranakis, R. Marcelin-Jimenez, S. Rajsbaum and J. Urrutia, Distributed dynamic storage in wireless networks, to appear in *International Journal of Distributed Sensor Networks.*

[30] S. Giordano and I. Stojmenovic, Position-based routing algorithms for ad hoc networks: A taxonomy, in: *Ad Hoc Wireless Networking* X. Cheng, X. Huang and D.Z. Du, eds., Kluwer, 2004, pp. 103–136.

[31] X. Guan, Face traversal routing on edge dynamic graphs, in: *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)* - Workshop 12 (2005) p. 244b.

[32] M. Heissenbuttel and T. Braun, Optimizing neighbor table accuracy of position-based routing algorithms, IEEE INFOCOM 2005.

[33] J.W. Jaromczyk and G.T. Toussaint, Relative neighborhood graphs and their relatives, in: *Proc. IEEE*, **80(9)** September 1992, pp. 1502–1517.

[34] Y. Kim, J.-J. Lee and A. Helmy, Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks, Special Issue on Wireless Pan and Sensor Networks, ACM SIGMOBILE MC2R, **8(1)** (2004), pp. 48–60.

[35] Y.B. Ko and N.H. Vaidya, Location-aided routing in mobile ad hoc networks, in *Proc. MOBICOM, 1998*, pp. 66–75.

[36] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, in: *Proc. 11th Canadian Conference on Computational Geometry* Vancouver, Aug. 15–18, 1999, pp. 51–54.

[37] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, Geometric ad-hoc routing: of theory and practice, in: *ProcṖODC 2003.*

[38] F. Kuhn, R. Wattenhofer and A. Zollinger, Ad-hoc networks beyond unit disk graphs, ACM DIALM-POMC, San Diego, Sept. 2003, pp. 69–78.

[39] F. Kuhn, R. Wattenhofer and A. Zollinger, Asymptotically optimal geometric mobile ad-hoc routing, in *Proc. of the 6th international workshop on discrete algorithms and methods for mobile computing and communications (Dial-M)* (2002), pp. 24–33.

[40] F. Kuhn, T. Moscibroda and R. Wattenhofer, Unit disk graph approximation. ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), Philadelphia, PA, October 2004.

[41] X.Y. Li, Applications of computational geometry in wireless ad hoc networks, In *Ad Hoc Wireless Networking*, Kluwer, XiuZhen Cheng, Xiao Huang and Ding-Zhu Du, eds., 2003.

[42] X.Y. Li, Localized Delaunay triangulation is as good as unit disk graph, Workshop on Ad Hoc Networks, IEEE Local Area Network Conference, Tampa, November 2001.

[43] X.Y. Li, Approximate MST for UDG locally, in: *Computing and Combinatorics: 9th Annual International Conference, COCOON 2003 Big Sky, MT, USA, July 25–28, 2003 Proceedings*, pp. 364–373.

[44] X.Y. Li, G. Calinescu, P.J. Wan and Y. Wang, Localized Delaunay triangulation with application in ad hoc wireless networks, in: *IEEE Transactions on Parallel and Distributed Systems*, **14** 10 Oct. 2003, pp. 1035–1047.

[45] X.Y. Li, Y. Wang and O. Frieder, Localized routing for wireless ad hoc networks, in: *Proc. IEEE ICC 2003.*

[46] X. Lin and I. Stojmenovic, Location based localized alternate, disjoint and multi-path routing algorithms for wireless networks, *Journal of Parallel and Distributed Computing* **63(1)** (2003), pp. 22–32.

[47] D.W. Matula and R.R. Sokal, Properties of Gabriel graphs relevant to geographical variation research and the clustering of points on the plane, *Geographical Analysis* **12** (1984), pp. 205–222.

[48] J. Newsome and D. Song, GEM: Graph embedding for routing and data-centric storage in sensor networks without geographic information, in: *Proc. First ACM Conf. on Embedded Networked Sensor Systems, SenSys*, Redwood, CA, Nov. 2003.

[49] R. O'Dell and R. Wattenhofer, Theoretical aspects of connectivity-based multi-hop positioning systems, *Theoretical Computer Science* **344** (2005), pp. 47–68.

[50] K. Saeda, A. Helmy and R. Govindan, On the effect of localization errors on geographic face routing in sensor networks, in: *Proc. ACM SenSys*, Los Angeles, November 2003.

[51] K. Seada and A. Helmy, Efficient geocasting with perfect delivery in Wireless Networks, in: *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2004.

[52] J. Shaikh, J. Solano, I. Stojmenovic and J. Wu, New metrics for dominating set based energy efficient activity scheduling in ad hoc networks, in: *Proc. WLN Workshop at IEEE Conf. on Local Computer Networks*, Bonn, Germany, Oct. 20–24, 2003.

[53] I. Stojmenovic, Position-based routing in ad hoc networks, *IEEE Communications Magazine* **40(7)** (2002), pp. 128–134.

[54] I. Stojmenovic, M. Seddigh and J. Zunic, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, in *IEEE Transactions on Parallel and Distributed Systems*, **13(1)** (2002), pp. 14–25.

[55] M.T. Sun, L. Huang, A. Arora and T.H. Lai, Reliable MAC layer multicast in IEEE 802.11 wireless networks, *Wirel. Commun. Mob. Comput.* **3** (2003), pp. 439–453.

[56] I. Stojmenovic, Localized network layer protocols in sensor networks based on optimizing cost over progress ratio, *IEEE Network* **20(1)** (2006), pp. 21–27.

[57] I. Stojmenovic and N. Zaguia, Bluetooth scatternet formation in ad hoc wireless networks, Ch. 9 in: *Performance Modeling and Analysis of Bluetooth Networks: Network Formation, Polling, Scheduling, and Traffic Control* (*J. Misic and V. Misic*), Auerbach Publications (Taylor and Francis Group, 2006), pp. 147–171.

[58] K.J. Supowit, The relative neighbourhood graph, with application to minimum weight spanning trees, *J. Assoc. Comput. Mach.* **30(3)** (1983), pp. 428–448.

[59] G.T. Toussaint, The relative neighbourhood graph of a finite planar set, *Pattern Recogn.* **12** (1980), pp. 261–268.

[60] Y.C. Tseng and T.Y. Hsieh, Fully power-aware and location-aware protocols for wireless multi-hop ad hoc networks, *IEEE ICCCN 2002*.

[61] Y. Wang and X.Y. Li, Localized construction of bounded degree planar spanner, *ACM DialM, 2003*.

[62] J. Widmer, M. Mauve, H. Hartenstein and H. Füßler, Position based routing in ad hoc wireless networks, in: *Handbook of Ad Hoc Wireless Networks*, M. Ilyas, ed., (CRC Press, 2003), pp. 219–232.

[63] J. Wu and H. Li, A dominating set-based routing scheme in ad hoc wireless networks, *Telecommunication Systems* **18(1/3)** (2001), pp. 13–36.

[64] J. Wu, F. Dai, M. Gao and I. Stojmenovic, On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks, *IEEE/KICS Journal of Communication Networks* **4(1)**(2002), pp. 59–70.

[65] M. Yuksel, R. Pradhan and S. Kalyanaraman, An implementation framework for trajectory-based routing in ad hoc networks, *Ad Hoc Networks*, Elsevier Science, **4(1)** (2006), pp. 125–137.