Computing 2-Hop Neighborhoods in Ad Hoc Wireless Networks

Gruia Calinescu

Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616
calinesc@iit.edu

Abstract. We present efficient distributed algorithms for computing 2-hop neighborhoods in Ad Hoc Wireless Networks. The knowledge of the 2-hop neighborhood is assumed in many protocols and algorithms for routing, clustering, and distributed channel assignment, but no efficient distributed algorithms for computing the 2-hop neighborhoods were previously published.
The problem is nontrivial, as the graphs induced by ad-hoc wireless networks can be dense. We employ the broadcast nature of the wireless networks to obtain a distributed algorithm in which every node gains knowledge of its 2-hop neighborhood using a total of $O(n)$ messages, where n is the total number of nodes in the network, and each message has $O(\log n)$ bits, which we assume is enough to encode the ID and the geographic position of a node. Our algorithm operates in an asynchronous environment, and makes use of the geographic position of the nodes.
A more complicated algorithm achieves the same communication bounds when geographical positions are not available, but nodes are capable of evaluating the distance to neighboring nodes or the angle of signal arrival. We also discuss updating the knowledge of 2-hop neighborhoods when nodes join or leave the network.

1 Introduction

Wireless ad hoc networks can be flexibly and quickly deployed for many applications such as automated battlefield, search and rescue, and disaster relief. Unlike wired networks or cellular networks, no physical backbone infrastructure is installed in wireless ad hoc networks. A communication session is achieved either through a single-hop radio transmission if the communication parties are close enough, or through relaying by intermediate nodes otherwise.

In this paper, we assume that all nodes in a wireless ad hoc network are distributed in a two-dimensional plane and have an equal maximum transmission range of one unit. The topology of such wireless ad hoc network can be modeled as a *unit-disk graph, or UDG* (see [11] for many interesting properties of unit-disk graphs), a geometric graph in which there is a link between two nodes if and only if their distance is at most one.

The 1-hop neighborhood of a node v (denoted by $N_1(v)$) is simply the set of nodes adjacent to it in the UDG. We use $N_2(v)$ to denote the set of nodes

S. Pierre, M. Barbeau, and E. Kranakis (Eds.): ADHOC-NOW 2003, LNCS 2865, pp. 175–186, 2003.

of the UDG 2-hops away from v. The *2-hop neighborhood* of v is the bipartite graph with node set $N_1(v) \cup N_2(v)$ in which all the links of the UDG with one endpoint in $N_1(v)$ and the other endpoint in $N_2(v)$ are included.

Knowledge of the 2-hop neighborhoods is assumed in many distributed algorithm and protocols such as constructing structures [24,6], improved routing [20], broadcasting [9], and channel assignment [3]. The clusters used for channel control typically have diameter at most two [19]. The knowledge of the set of 2-hop neighbors is helpful in frequency assignment to avoid secondary interference. Also distributed algorithms for $L(2,1)$-Labeling ([12,8,10]) can use the information about 2-hop neighborhoods stored by every node. Knowledge of the 2-hop neighborhood can be used for efficient computation of multipoint relays, used for example in [14].

Our distributed algorithms operate in an asynchronous environment, and we use the number of messages as the measure of the efficiency of the algorithm. In our model a message can hold the ID of a node, the geographical position of a node, and $O(\log n)$ bits, where n is the total number of nodes in the network. Concentrating on the number and the length of the messages is justified by the limited resources available to wireless nodes. We assume nodes have $O(n)$ memory available.

In this model, computing the set of 1-hop neighbors with $O(n)$ messages is trivial: every node broadcasts a message announcing its ID. One can easily compute the 2-hop neighborhood with $O(n)$ messages of size $\Delta \log n$ each, where Δ is the maximum number of 1-hop neighnors. But we insist on messages of size $O(\log n)$ each, and therefore, as UDGs can be dense, computing the 2-hop neighborhood is not trivial.

The broadcast nature of the communication in ad hoc wireless networks is however very useful when computing local information. To our knowledge no distributed algorithm for computing 2-hop neighborhoods has been previously proposed and analyzed.

First we assume that each static wireless node knows its position information, either through a low-power Global Position System (GPS) receiver or through some other ways. Then to construct the 2-hop neighborhoods it is enough to know the IDs and positions of the 1-hop and 2-hop neighbors. With these assumptions, we present a simple distributed algorithm which allows every node to compute the positions of its 2-hop neighbors. The total number of $O(\log n)$-bit messages of the algorithm is $O(n)$.

Second, we assume that position information is not available, but every two adjacent nodes are capable of estimating their pairwise distance. Probing - lowering the transmission power over an interval of time - is one way which allows the computation of pairwise distances. A detailed discussion of location systems appears in [13]. Under this assumption, we present a distributed algorithm which allows every node to compute its 2-hop neighborhood. The total number of $O(\log n)$-bit messages of the algorithm is $O(n)$. The algorithm is based on triangulation and can be immediatly updated to work when the angle-of-arrival information is available (an assumption justified in [18] or [15]) instead of pairwise distances.

Our approach is based on the specific connected dominating set introduced by Alzoubi, Wan, and Frieder [2,21]. This connected dominating set is based on a maximal independent set (MIS), whose role in algorithms for unit-disk graphs was discovered by Marathe et. al [16]. An MIS is a dominating set: every node must have a 1-hop neighbor in the maximal independent set. In our algorithm, each node uses its adjacent node(s) in the MIS to broadcast over a larger area relevant information. Listening to the information about other nodes broadcast by the MIS nodes enables a node to compute its 2-hop neighborhood. There is a direct (without using a MIS) solution when node positions are available, but it is more complicated and requires synchronization in order to achieve $O(n)$ messages each of size $O(\log n)$ bits.

The example in Figure 1 shows that $\Theta(n/\log n)$ time is sometimes necessary for computing 2-hop neighborhoods (assuming one "step" allows the transmission of $O(\log n)$ bits), as the center node has to transmit $\Theta(n)$ bits to show the existance (or non-existance) of each node on one side to the nodes on the other side. This justifies our concentration on communication complexity, and not time complexity. And while our algorithms use heavily the nodes in the connected dominating set, the same example shows that overloading certain nodes is sometimes unavoidable.

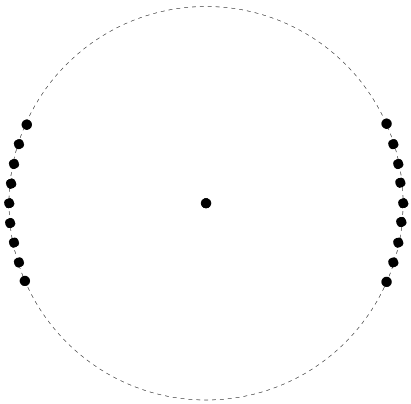

Fig. 1. The center node of this disk of radius 1 must send $\Theta(n)$ bits to allow the correct computation of the 2-hop neighborhoods

We also describe a straightforward procedure of updating the 2-hop neighborhoods when nodes join or leave the network. When leaving the network, the communication cost is $O(\log n)$ bits. When joining the network, the number of messages is bounded by a small constant times the number of nodes in the 2-hop neighborhood of the new node.

The paper is organized as follows. The next section clarifies the notation and explains the properties of the connected dominating set our algorithms use.

Section 3 describe the algorithm for the situation when geographic position is available. Section 4 describes the generalization of the algorithm to the situation when only pairwise distance in between adjacent nodes is available. Section 5 describes the recomputation of 2-hop neighborhoods due to changes in the network configuration. We conclude with Section 6.

2 Preliminaries

In this paper by broadcast we understand local broadcast - a packet send by a node, and received by every other node within the transmission range.

Recently [2,21] introduced a virtual backbone of the network, and our algorithms make heavy used of this virtual backbone. The next subsection quickly reproduces their construction, and lists the important properties of the virtual backbone.

2.1 The Virtual Backbone

The virtual backbone is a connected dominating set in the UDG. It is based on a maximal independent set (MIS), and we call the nodes in the maximal independent set *MIS nodes*. MIS nodes cannot be 1 hop away; if two MIS nodes are two or three hops away, we call them *virtually-adjacent*. One or two *connector* nodes are used to establish a path corresponding to a pair of virtually-adjacent MIS nodes. A node can participate as a connector for several pairs of virtually-adjacent MIS nodes. Only the links in between a connector node and the MIS nodes it connects, or in between two connector nodes which together establish the path corresponding to a pair of virtually-adjacent MIS nodes are added to the virtual backbone.

In [2,21] it is shown how the virtual backbone (including adding the connector nodes) can be constructed distributely with $O(n)$ messages, where the message size is $O(\log n)$ bits. They also show how to maintain the virtual backbone when the topology of the network changes.

Wan et. al. [2,21] proved that the virtual backbone is connected. Using an area argument, [2,21] proved that within three hops of an MIS node there could be at most 47 MIS nodes, and therefore the maximum degree of the virtual backbone is bounded by a constant we call Δ. Please refer to Figure 2 for intuition on the virtual backbone described above.

It was first proved in [16] that the size of any maximal independent set is at most five times the minimum dominating set in the UDG, as in fact for any node x can have at most five neighbors in an MIS. Alzoubi et al. [2,21] noticed that their virtual backbone is also within a constant the size of the minimum connected dominating set.

In addition, it is immediate that the virtual backbone of [2,21], together with links from every node to an MIS node adjacent to it, is a hop-spanner. Precisely, for every path in the UDG, there is a path on the virtual backbone with at most three times as many links from an MIS node adjacent to the origin of the path

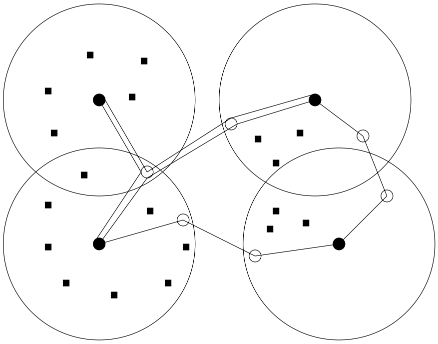

Fig. 2. An illustration of the virtual backbone of Alzoubi, Wan, and Frieder. The solid round nodes are the MIS node, which form a dominating set. Two virtually-adjacent MIS nodes are connected by paths of length at most three through connector nodes - the empty tiny circles in the figure. Nodes not in the virtual backbone are small solid squares in the figure

to an MIS node adjacent to the destination of the path. This fact was noticed by Alzoubi [1], and by Wang and Li [22], which also planarize the virtual backbone while keeping all its attractive properties.

3 Geographic Position Available

In this section we describe the distributed algorithm which allows every node to construct the list of its 2-hop neighbors, assuming every node knows its geographical position. With this information, every node can also easily compute the links between its 1-hop and 2-hop neighbors. Our algorithm is described in the simplest version, and we do not try to optimize the constant hidden in the O notation.

We start from the moment the virtual backbone is already constructed, and every node knows the ID and the position of its neighbors. The idea of the algorithm is for every node to efficiently announce its ID and position to a subset of nodes which includes its 2-hop neighbors.

The responsibility for announcing the ID and position of a node v is taken by the MIS nodes adjacent to v. Each such MIS node assembles a packet containing: $< ID, position, counter >$, with the ID and position of v, and a counter variable being set to 2. The MIS node then broadcasts the packet.

A connector node is used to establish a link in between several pairs of virtually-adjacent MIS nodes, and will not retransmit packets which do not travel in between these pairs of MIS nodes. The connector node will rebroadcast packets with nonzero *counter* originated by one of the nodes in a pair of virtually-adjacent MIS nodes, thus making sure the packet advances towards the other MIS node in the pair. Recall that the path in between a pair of virtually-adjacent MIS nodes has one or two connector nodes.

When receiving a packet of type $< ID, position, counter >$, an MIS node checks whether this is the first message with this ID, and if yes decreases the *counter* variable and rebroadcasts the packet.

A node listens to the packets broadcasted by all the adjacent MIS nodes (here it is convenient to assume a MIS is adjacent to itself), and, using its internal list of 1-hop neighbors, checks if the node announced in the packet is a 2-hop neighbor or not - thus constructing the list of 2-hop neighbors.

Theorem 1. *When finished, the algorithm described above correctly computes the 2-hop neighborhood for every node in the network, and uses $O(n)$ messages of size $O(\log n)$ each.*

Proof. The fact that the virtual backbone is a bounded-degree hop-spanner essentially implies the correctness of the algorithm. The precise argument is as follows. Assume nodes v and u share a neighbor x, and let $\bar{v}$, $\bar{u}$, and $\bar{x}$ be nodes in MIS which are adjacent to v, u, and x. Then $\bar{v}$ creates a packet with the ID and position of v, and with its counter set to 2. As $\bar{v}$ and $\bar{x}$ are virtually-adjacent, $\bar{x}$ will receive the packet and retransmit it with counter set to 1. As $\bar{x}$ and $\bar{u}$ are virtually-adjacent, $\bar{u}$ will also broadcast the packet, and therefore u finds out the ID and position of v.

Regarding the number of messages, we count the packets announcing the ID and position of x. Such packets are being sent by S_1, the MIS nodes adjacent to x, and we recall that $|S_1| \leq 5$. They are also sent by S_2, the MIS nodes virtually-adjacent to S_1, by S_3, the MIS nodes virtually-adjacent to S_2, and by the connector nodes in between pairs of virtually-adjacent MIS nodes inside $S_1 \cup S_2$, and by the connector nodes in between virtually-adjacent MIS nodes of S_2 and S_3. Thus the total number of nodes retransmitting packets announcing ID and position of x is $O(\Delta^2)$. As Δ, the maximum degree of the virtual backbone is constant, the total number of messages is $O(n)$. □

We remark that with the counter of a packet being initially set to k (and decreased by one whenever a MIS node retransmits), the same argument as above implies that with $O(\Delta^k)$ messages every node can compute its k-hop neighborhoods.

4 Pairwise Distances Available

In this section we assume that neighboring nodes can compute their pairwise distance, but are not aware of their precise geographical position.

Our approach is based on the virtual backbone used before and *rigid pieces*, which we define as subgraphs containing one MIS node and a subset of its neighbors such that a system of coordinates can be locally established and in which the position of every node of the rigid piece is completely defined. A theory of geometric rigidity is well established [23]. We need only simple properties which are easily proved below.

First we describe the distributed algorithm for computing the rigid pieces. Before the actual construction, every node announces all the MIS nodes to which it is adjacent, and records the information transmitted by all its neighbors.

Every MIS node v constructs one after the other the rigid pieces in which it participates, and ensures these pieces are disjoint with the exception of v. Each such piece will have an ID, composed of the ID of the unique MIS which is in the piece and an integer in between 1 and 18. Once a node is assigned to a piece together with v, it announces in a broadcast message the ID of the rigid piece and its coordinates with respect to the rigid piece.

Let us describe the construction of one such rigid piece. The MIS node v always has coordinates $(0, 0)$ with respect to the rigid piece. If all nodes adjacent to v are in a rigid piece with v, the procedure stops. Otherwise, v selects a neighbor x which is not in a rigid piece with v, and asks x to announce its participation in the rigid piece and its coordinates with respect to the rigid piece: $(||xv||, 0)$. Every node y adjacent to both v and x and not yet in some other rigid piece with v, computes its coordinates with respect to v and x based on the length of the sides of the triangle xyv. Actually, while the first coordinate of y is unique, the second one is not: only its absolute value can be computed exactly. If the angle $\widehat{yvx}$ is bigger than $\pi/3$, y will not participate in the rigid piece. If the second coordinate of y is 0, then y participates in the piece and announces its participation and its unique coordinates with respect to the rigid piece. If the angle $\widehat{yvx}$ is at most $\pi/3$ and the second coordinate of y is nonzero, y announces it is willing to participate in the piece. Node v will pick only one such y (assuming it exists), and announce that both of y's coordinates with respect to the rigid piece will be positive. See Figure 3 for intuition. At this moment y announces its participation in the rigid piece and its coordinates with respect to the rigid piece. Every node z adjacent to v, x, and y, and not yet in some other rigid piece with v, computes its unique coordinates with respect to the rigid piece, and announces its participation in the rigid piece and its coordinates.

The following theorem enumerates the important properties of the distributed algorithm described above.

Theorem 2. *Every non-MIS node is a member of at most five rigid pieces. Every MIS node is a member of at most 18 rigid pieces. Computing the nodes of a rigid piece and the coordinates with respect to the rigid piece of every node can be done with a number of messages bounded by a constant times the number of nodes adjacent to the MIS node in the piece. The total number of messages (each having $O(\log n)$ bits) until every node announces every piece in which it participates, together with its coordinates with respect to the rigid piece, is $O(n)$.*

Proof. Once we prove that a MIS node constructs at most 18 rigid pieces, the remaining assertions of the theorem follow from the description of the algorithm.

Let k be the number of rigid pieces constructed and let x_i be the first nodes selected by v when constructing the i^{th} piece. Let y_i be the node picked by v as the first node of the rigid piece with nonzero second coordinate with respect to the i^{th} rigid piece, if such a node exists. If y_i exists, define R_i be the sector of the unit disk centered at v consisting of the points z with angles $\widehat{zvx_i}$ and $\widehat{zvy_i}$ at most $\pi/3$. If y_i does not exists, let R_i be the sector of the unit disk centered at v

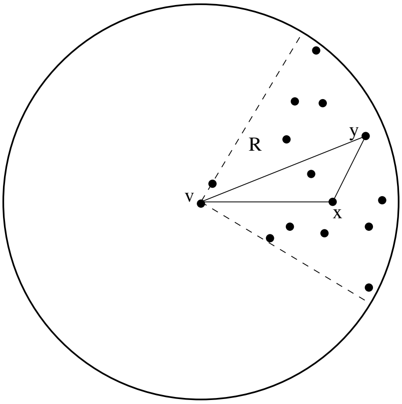

Fig. 3. The unnamed nodes in the figure can join the rigid piece started by v, x, and y. In the system of coordinates used, v is the origin, x has second coordinate 0, and y has the second coordinate positive. Notice that every node in the sector of the disk $R = R_i$ can join the rigid piece and that R covers at least 1/6 of the unit disk centered at v

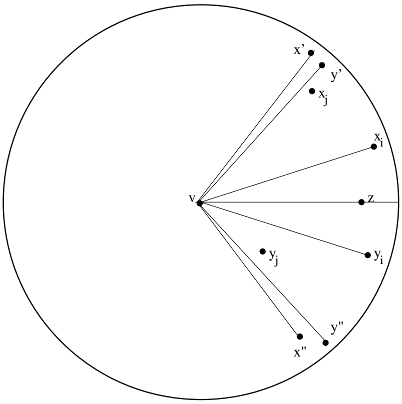

Fig. 4. There could be at most three sectors R_k which contain the point z: the first given by x_i and y_i (which are on opposite sides of the line vz) and then two sectors given by x' and y' (both above the line vz) and by x" and y" (both under the line vz). As shown in proof of Theorem 2, a fourth sector such as the one given by x_j and y_j cannot exists

consisting of the points z with angles $\widehat{zvx_i}$ at most $\pi/3$. Figure 3 again provides intuition.

We claim that any point z belongs to at most three sectors R_i, $1 \leq i \leq k$. See Figure 4 for intuition. Indeed, if there are $i < j$ with z in both R_i and R_j and x_i, y_i (if y_i exists), x_j and y_j (if y_j exists) are all on the same side of vz, then $x_j \in R_i$ as the angle $\widehat{x_ivx_j}$ is at most $\pi/3$ and $\widehat{y_ivx_j}$, if y_i exists, is also at most $\pi/3$. Therefore x_j should have entered rigid component i, a contradiction.

If both y_i and y_j exist, x_i and y_i are on different sides of vz, and x_j and y_j are also on different sides of vz, then we obtain a contradiction as follow. Assuming $i < j$ and x_i and x_j are on the same side of vz (the case when x_i and y_j are on the same side of vz is symmetric), we have that the angle $\widehat{y_i v x_j}$ is bigger than $\pi/3$, as otherwise x_j should have been taken in rigid piece i. Since the angle $\widehat{x_i v y_i}$ is at most $\pi/3$, we conclude that x_i is inside the angle $\widehat{zvx_j}$ and a symmetric argument yields that y_i is inside the angle $\widehat{zvy_j}$. Since $\widehat{y_i v x_j}$ is bigger than $\pi/3$, we conclude that $\widehat{x_j v y_j}$ is also bigger than $\pi/3$, a contradiction.

Thus only three sectors R_i can contain z: two with x_i on one side of vz and y_i nonexistent or on the same side as x_i, and one i with x_i and y_i on different sides of vz.

As any R_i covers at least 1/6 of the unit disk centered at v, and any point belongs to at most three sectors, we conclude that there are at most 18 such sectors. This finishes the proof of Theorem 2. □

At this moment the rigid pieces are constructed, every node has announced its participation in the rigid pieces together with its coordinates with respect to that piece. We now describe the second phase of the distributed algorithm, in which every node v gives enough information to every of its 2-hop neighbors y to determine the fact that y is a 2-hop neighbor of v and which is the set of common neighbors. More precisely, for every rigid piece which intersects the 1-hop neighborhood of v, y will have enough information to compute which of its neighbors from the rigid piece are adjacent to v.

Every node records all the information passed by its neighbors. In the 1-hop neighborhood of a node v, there can be at most a constant number of rigid pieces, as the number of MIS nodes in the 1-hop and 2-hop neighborhood of v is bounded by 25: if we draw a disk of radius 1/2 around every MIS node, we obtain disjoint disks of area $\pi/4$ which are included in the disk centered at v with radius 5/2, whose area is $25\pi/4$ (this argument is also used in [2]).

Separately for every rigid piece which intersects the 1-hop neighborhood of v (including the rigid pieces containing v), v determines how many neighbors it has in the rigid piece. If v has at most two neighbors in the rigid piece, it ask all its neighbors in the rigid piece to announce they are neighbors with v.

If v has three non-coliniar neighbors in common with the rigid piece, using the distance to these three points, v can compute its coordinates with respect to the rigid piece. Then v asks its neighbors in MIS to announce its position with respect to the rigid piece. This is done exactly as the announcements in Section 3, with packets containing $< nodeID, pieceID, coordinates, counter >$. Any node receiving such a message evaluates whether it has neighbors in the rigid piece, and if yes the node computes the set of its neighbors from the rigid piece which are adjacent to v, based on their coordinates with respect to the rigid piece.

If v has three or more neighbors in common with a rigid piece, but they are coliniar, v cannot exactly compute its coordinates with respect to the rigid piece, but has exactly two possible value for its coordinates. Then v asks its neighbors in MIS to announce both positions with a packet containing

$< nodeID, pieceID, coordinates_1, coordinates_2, counter >$. Any node y receiving such a message evaluates if it has neighbors in the rigid piece, and if yes, as above, it can compute two sets S_1 and S_2 of nodes in the rigid piece which could be the common neighbors with v - assuming v has coordinates $coordinates_1$ or $coordinates_2$ with respect to the rigid piece. At least one of S_1 and S_2 is a set of colinear points, and if both are, they coincide. That set of colinear points is the set of neighbors common to y and v in the rigid piece.

All cases are taken care of and we conclude:

Theorem 3. *There is a distributed algorithm which, under the assumption that every node can estimate the distance to every adjacent node, computes for every node v the set of its 2-hop neighbors $N_2(v)$ and the links in between $N_1(v)$ and $N_2(v)$ with a total of $O(n)$ messages each of size $O(\log n)$ bits.*

5 Updating the 2-Hop Neighborhoods

In this section we discuss the message complexity of updating the 2-hop neighborhoods due to changes in network topology. We do not address updating the virtual backbone as this was done in [2]. The proposed protocol is straightforward and does not use the virtual backbone. We assume geographical knowledge is available in this section.

Before leaving the network, a node u uses its knowledge to let its 2-hop neighborhs know the fact it is leaving as described below. First the node u computes a maximal independent set (MIS) in the graph induced by its 2-hop neighborhs. Then u computes at most one "connector" node for each MIS node. As before, MIS is a dominating set, and using an area argument, has constant size. Node u prepares an $< ID, position, leaving, relay >$ message, with its own ID and position, the fact that it is leaving the network, and the full list of relay nodes. Each node, after receiving such a message, make a note that u is leaving and updates its 2-hop neighborhood accordingly, and, if it finds itself in the list of relay nodes, rebroadcast the message once.

When a node v joins the network, it will broadcasts its ID and position. Every existing node which receives this message will rebroadcast the ID and position of v. Every node y receiving such a message, will update its stored 2-hop neighborhood to reflect the presence of v. If y is adjacent to v, it will broadcast its ID and position. If y is a 2-hop neighbor of v, it selects a common neighbor x and asks x to relay to v the position and ID of y. The total bit complexity of message is $O(q \log n)$, where q is the size of the 2-hop neighborhood of v, and it cannot be improved by more than a constant factor since v must find out the IDs of the nodes in its 2-hop neighborhood.

6 Conclusions

The virtual backbone of Alzoubi, Wan, and Frieder [2,21] can be constructed without any geographical knowledge: their algorithm "operates" directly on the unit-disk graph. We need at least the distance in between any pair of adjacent nodes. Same arguments using rigid pieces apply when a node is able to compute the angle in between the segments to adjacent nodes. However, without any geographical knowledge we do not know whether it is possible to compute 2-hop neighborhoods with $O(n)$ messages each having $O(\log n)$ bits. This observation raises the interesting question whether there are any (meaningful) problems which have higher communication complexity on unit-disk graphs than on embedded (nodes aware of their geographical position) unit-disk graphs. Note that it is NP-Hard to recognize unit-disk graphs [7].

However, it follows from standard algebraic geometry results (page 542 of [17] or improved bounds in [4]) that the number of labeled unit-disk graphs of n nodes is between $2^{c_1 n \log n}$ and $2^{c_2 n \log n}$, for constants c_1 and c_2 and therefore a protocol with a total $O(n \log n)$ bits communication complexity is possible. An $O(n \log n)$ bits communication complexity would follow from a solution to an open problem in algebraic geometry [5]. It is worth mentioning that algebraic geometry solutions seem to have huge running time and space complexity.

Our model does not account for messages lost because of interference. It would be desirable to design synchronous distributed algorithms with low message complexity and low time complexity in a model where messages are lost either due to signal interference or due to node overloading.

Acknowledgments

The author thanks Peng-Jun Wan and Xiang-Yang Li, who inspired this paper by presenting their results. The author thanks Sougata Basu, Adrian Dumitrescu, and Peter Sanders for insight in the issue of extending the results to case when geographical knowledge is not available.

References

1. Khaled M. Alzoubi, "Distributed Algorithms for Connected Dominating Set in Wireless Ad Hoc Networks", Illinois Institute of Technology, 2002.
2. Khaled M. Alzoubi, Peng-Jun Wan and Ophir Frieder, "Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks", in *ACM MOBIHOC '02*.
3. L. Bao and J. J. Garcia-Luna-Aceves, "Channel Access Scheduling in Ad Hoc Networks with Unidirectional Links", *5th International Workshop on Discrete Algorithms and Methods for Mobility*, 2001, Pages 9–18.
4. S. Basu "Different bounds on the different Betti numbers of semi-algebraic sets", to appear in *Discrete and Computational Geometry.* Available at http://www.math.gatech.edu/~saugata/.
5. S. Basu, R. Pollack, and M. F. Roy, Algorithms in Real Algebraic Geometry, Springer-Verlag, 2003.
6. V. Bharghavan and B. Das, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets", *International Conference on Communications'97*, Montreal, Canada. June 1997.

7. Heinz Breu and David G. Kirkpatrick. "Unit disk graph recognition is NP-hard", *Computational Geometry. Theory and Applications*, 9:3–24, 1998.
8. T. Calamoneri and R. Petreschi, "L(2,1)-Labeling of Planar Graphs", *5th International Workshop on Discrete Algorithms and Methods for Mobility*, 2001, Pages 28–33.
9. G. Calinescu, I. Mandoiu, P.-J. Wan, and A. Zelikovsky, "Selecting Forwarding Neighbors in Wireless Ad Hoc Networks", *5th International Workshop on Discrete Algorithms and Methods for Mobility*, 2001, Pages 34–43.
10. I. Chlamtac and S. Pinter, "Distributed Nodes Organization Algorithm for Channel Access in a Multihop Dynamic Radio Network", *IEEE Trans. on Computers*, 36(6):728–737, 1987.
11. B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit Disk Graphs", *Discrete Mathematics*, 86:165–177, 1990.
12. J. R. Griggs and R. K. Yeh, "Labeling Graphs with a Condition at Distance 2", *SIAM J. Disc. Math*, 5:586–595, 1992.
13. J. Hightower and G Borriello, "Location Systems for Ubiquitous Computing", *IEEE Computer*, vol. 34(8), 2001, pp 57–66.
14. P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models", in *R*SRCP, Special issue on Mobility and Internet, 2001.
15. K. Krizman, T. Bieda, and T. Rappaport, "Wireless position location: fundamentals, implementation strategies, and source of error", *Veh. Tech. Conf*, 1997, 919–923.
16. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz, "Simple Heuristics for Unit Disk Graphs", *Networks*, Vol. 25, 1995, pp. 59–68.
17. B. Mishra, "Computational Real Algebraic Geometry" in *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke (editors), CRC Press, 1997.
18. A. Nasipuri and K. Lim "A Directionality based Location Discovery Scheme for Wireless Sensor Networks", *WSNA 2002*.
19. S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communication networks", *ACM/Baltzer Mobile Networks and Applications*, 89–104, 1996.
20. I Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks", *IEEE Transactions on Parallel and Distributed Systems*, 12:1023–1032, 2001.
21. Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks", in *IEEE INFOCOM 2002*.
22. Yu Wang and Xiang-Yang Li, "Geometric Spanners for Wireless Ad Hoc Networks", in *ICDCS 2002*.
23. W. Whiteley "Rigidity and Scene Analysis", *Handbook of Discrete and Computational Geometry*, 893–916, ed. J. E. Goodman and J. O'Rourke, CRC Press, 1997.
24. J. Wu and H.L. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks", *Proceedings of the 3rd ACM international workshop on Discrete algorithms and methods for mobile computing and communications*, 1999, Pages 7–14.