World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:3, 2012

Local algorithm for establishing a virtual backbone in 3D ad hoc network

Alaa E. Abdallah, M. Bsoul, Emad E. Abdallah, Ahmad Al-Khasawneh, and Muath Alzghool

Abstract—Due to the limited lifetime of the nodes in ad hoc and sensor networks, energy efficiency needs to be an important design consideration in any routing algorithm. It is known that by employing a virtual backbone in a wireless network, the efficiency of any routing scheme for the network can be improved. One common design for routing protocols in mobile ad hoc networks is to use positioning information; we use the node's geometric locations to introduce an algorithm that can construct the virtual backbone structure locally in 3D environment. The algorithm construction has a constant time.

Keywords—virtual backbone, dominating set, UDG.

I. Introduction

THE recent advances in technologies have enabled a new kind of networks, so called wireless ad hoc networks, which do not require any pre-existing infrastructure for establishing connectivity and routing messages. Two nodes can communicate in a bidirectional manner if and only if the distance between them is at most the minimum of their transmission ranges. If one node wishes to communicate with another node outside its transmission range, the set of nodes between the two endpoints should forward their packets so they can communicate. This means that in ad hoc network, any node must be able to play the role of a router in a conventional network.

A crucial problem in multihop routing is to find an efficient and correct route between a source and a destination; however for many networks, a more important problem is to provide an energy efficient routing protocol because of the limited battery life of the wireless nodes. One way to decrease the power consumption (communication overhead) of the routing protocols is to narrow down the search space for a route to the node in the virtual backbone. A connected dominating set (CDS) [4], [9], [17], [18] can form an interesting virtual backbone. A connected dominating set of a graph is a connected subset of nodes such that each node in the graph is either in the subset or adjacent to at least one node in that subset.

The routing algorithms that use virtual backbone only allow nodes of the connected dominating set (dominators) act as routers; all other nodes communicate via a neighbor in the dominating set. Clearly, the efficiency of this approach depends largely on the process of finding the dominating sets and the size of the corresponding virtual backbone.

Alaa E. Abdallah, M. Bsoul, Emad E. Abdallah, Ahmad Al-Khasawneh are with prince al Hussein bin Abdullah II faculty of information technology ,Hashemite University, Zarqa 13115, Jordan Email:{aabdallah, mbsoul, emad, akhasawneh}@hu.edu.jo

Muath Alzghool is with Department of Computer Science, Al-Balqa Applied University, Alsalt, Jordan Email:alzghool@huson.edu.jo

Due to the nature of ad hoc network, Algorithms to construct a virtual backbone should be local, where each node of the network only uses information obtained uniquely from the nodes located no more than a constant (independent of the size of the network) number of hops from it. In this paper we propose a local algorithm to construct a virtual backbone for 3d wireless ad hoc network. The new algorithm has a constant time complexity.

The rest of the paper is organized as follows: In Section II, we briefly present the network model and some related work. A Space Partition System of the 3D space needed in our algorithms is described in Section III. In Section IV, we introduce our local algorithm to construct a virtual backbone. We conclude our paper in Section V.

II. Preliminaries

A. The Network Model

We assume that the set of n wireless nodes is represented as a point set S in 3D space; each mobile host knows the coordinates of its position. All network nodes have the same communication range r. Two nodes are connected by an edge if the Euclidean distance between them is at most r. The resulting graph is called a unit disk graph (UDG)[5], [15]. A dominating set for a graph is a set of vertices whose neighbors, along with themselves, constitute all the vertices in the graph.

B. Related Work

The general problem of finding the smallest virtual backbone (smallest connected dominating set) for a graph is known to be NP-Hard [7], [11], [12], [14]. Several algorithms for finding an approximation for a small size virtual backbone have been proposed. The Greedy algorithm [6], [13], [16] for constructing a virtual backbone is a global algorithm where the run time depends on the number of nodes. The greedy algorithm grows a tree rooted at the node that has the maximum number of neighbors. The root is colored black and all its neighbors are colored gray. Then, the algorithm scans the gray nodes and their white neighbors iteratively, and selects the gray node or the pair of gray and white node with the maximum number of white neighbors. The selected node(s) are marked black and their neighbors are marked gray. The algorithm terminates when all the nodes have been marked either black or gray.

Alzoubi *et al.* [3], [4] proposed a distributed algorithm to construct virtual backbone; in this algorithm if the node unique ID is minimum among its neighbors, it adds itself to the dominating set and removes all its neighbors from the

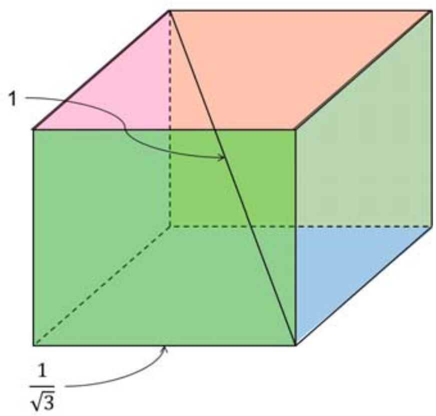

Fig. 1. Unit diameter cube. top face is orange, right side face is light green

consideration of the set members. This process is repeated at each node, such that the resulting set is a non-connected dominating set. The nodes in the resulting set use local topology information for a node, up to 3 hops away, to add gateway nodes to the set until the set becomes a connected dominating set. The main disadvantage of this algorithm is the construction time of the independent set which can be proportional to the number of nodes, thus it is a non-local algorithm. It has been proved in [1] that this algorithm is not a local algorithm.

None of the algorithms mentioned above has both constant approximation bound and constant worst case time bound. One approach to achieve these bounds is to use the underlying geographic information. The first algorithm to determine a virtual backbone in 2D within a constant approximation of the optimal dominating set in a constant time was proposed by Czyzowicz *et al.* [8]. A 3D version of the algorithm has been proposed in [1].

III. Space Partition System

Our space partition system uses identically shaped tiles that fill the entire space. Each tile used in our tiling system consists of 27 cube of diameter equal to 1 unit, see Fig. 1. Each cube in the tile represents one class which has a unique integer. Assume that the first cube is centered at the coordinates (x_1, y_1, z_1), i.e. the z-axis passes through the center of face 1 (top face), the x-axis passes through the center of face 2 (right side face) and y-axis passes through the center of face 3 (front side face). We will call this orientation as the centering orientation; the coordinates of the centers of the classes from 2 to 27 are shown in table I. They all have the same orientation as class 1. See Fig. 2 for an example of the tile used, showing the placement of the cubes in the tile with the associated classes labels.

Let the tiling starts by placing the center of one tile, TL_{CN} , at the coordinate (x_1, y_1, z_1), with orientation equal to the centering orientation. To cover all the faces of TL_{CN} we need 26 other adjacent tiles that are in contact with TL_{CN} in the positions summarized on Table II. Each tile has the same orientation as TL_{CN} . Fig. 3 shows the space tiling process used in our algorithm. It is clear that any point can

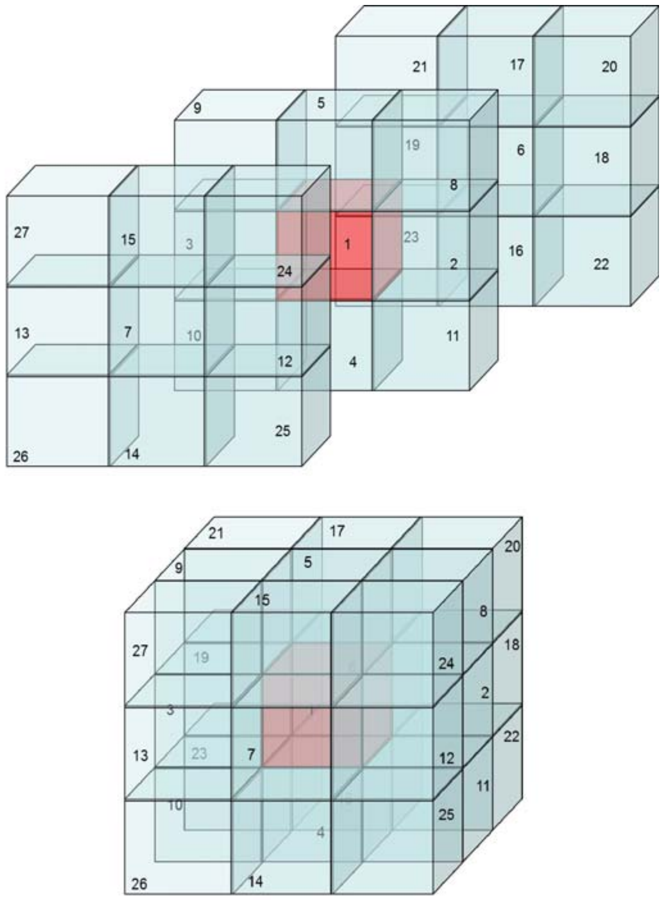

Fig. 2. The tile used in the space partition system divided into 27 cube of diameter 1 and the class numbering associated with the cube (a) shows the cubes class number. (b) shows the whole tile.

calculate locally its class number by determining to which tile and corresponding cube it belongs. If a node located exactly on the shared face between two cubes C1 and C2, the node is considered of class 1 (the class with the smaller ID).

IV. A Local Algorithm for 3D Virtual Backbone (3DVBP)

Our local algorithm to construct a virtual backbone consists of two phases. In the first phase, a dominating set 3DDOM is constructed using Algorithm 1. In the second phase, each node from 3DDOM creates paths connecting dominators that are at most three hops apart. Using the space partition described in the previous section, each node can determine its class number locally using a constant number of arithmetic operations. The nodes are aware of the locations of all their neighbors, using the periodic hello messages, so they can also calculate the class number of each neighbor. It is clear that the nodes that are in same cube are neighbors because the diameter of the cube is 1 Our local construction of the dominating sets is based on a similar algorithm proposed by Czyzowicz *et al.* [8] for 2D. To calculate a dominating set of a unit disk graph G, each node of G executes Algorithm 1.

TABLE I
COORDINATES OF THE 27 CUBES THAT FORMS THE CENTER TILE, FIRST CUBE IS CENTERED AT (x_1, y_1, z_1).

C 1	(x_1, y_1, z_1)	C 2	$(x_1 + 1/\sqrt{3}, y_1, z_1)$
C 3	$(x_1 - 1/\sqrt{3}, y_1, z_1)$	C 4	$(x_1, y_1 - 1/\sqrt{3}, z_1)$
C 5	$(x_1, y_1, z_1 + 1/\sqrt{3})$	C 6	$(x_1, y_1 + 1/\sqrt{3}, z_1)$
C 7	$(x_1, y_1 - 1/\sqrt{3}, z_1)$	C 8	$(x_1 + 1/\sqrt{3}, y_1, z_1 + 1/\sqrt{3})$
C 9	$(x_1 - 1/\sqrt{3}, y_1, z_1 + 1/\sqrt{3})$	C 10	$(x_1 - 1/\sqrt{3}, y_1, z_1 - 1/\sqrt{3})$
C 11	$(x_1 + 1/\sqrt{3}, y_1, z_1 - 1/\sqrt{3})$	C 12	$(x_1 + 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1)$
C 13	$(x_1 - 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1)$	C 14	$(x_1, y_1 - 1/\sqrt{3}, z_1 - 1/\sqrt{3})$
C 15	$(x_1, y_1 - 1/\sqrt{3}, z_1 + 1/\sqrt{3})$	C 16	$(x_1, y_1 + 1/\sqrt{3}, z_1 - 1/\sqrt{3})$
C 17	$(x_1, y_1 + 1/\sqrt{3}, z_1 + 1/\sqrt{3})$	C 18	$(x_1 + 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1)$
C 19	$(x_1 - 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1)$	C 20	$(x_1 + 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1 + 1/\sqrt{3})$
C 21	$(x_1 - 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1 + 1/\sqrt{3})$	C 22	$(x_1 + 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1 - 1/\sqrt{3})$
C 23	$(x_1 - 1/\sqrt{3}, y_1 + 1/\sqrt{3}, z_1 - 1/\sqrt{3})$	C 24	$(x_1 + 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1 + 1/\sqrt{3})$
C 25	$(x_1 + 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1 - 1/\sqrt{3})$	C 26	$(x_1 - 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1 - 1/\sqrt{3})$
C 27	$(x_1 - 1/\sqrt{3}, y_1 - 1/\sqrt{3}, z_1 + 1/\sqrt{3})$		

TABLE II
COORDINATES OF THE CENTERS OF THE 26 NEIGHBORS.

center tile 1	(x_1, y_1, z_1)	neighbor 1	$(x_1 + 3/\sqrt{3}, y_1, z_1)$
neighbor 2	$(x_1 - 3/\sqrt{3}, y_1, z_1)$	neighbor 3	$(x_1, y_1 - 3/\sqrt{3}, z_1)$
neighbor 4	$(x_1, y_1, z_1 + 3/\sqrt{3})$	neighbor 5	$(x_1, y_1 + 3/\sqrt{3}, z_1)$
neighbor 6	$(x_1, y_1 - 3/\sqrt{3}, z_1)$	neighbor 7	$(x_1 + 3/\sqrt{3}, y_1, z_1 + 3/\sqrt{3})$
neighbor 8	$(x_1 - 3/\sqrt{3}, y_1, z_1 + 3/\sqrt{3})$	neighbor 9	$(x_1 - 3/\sqrt{3}, y_1, z_1 - 3/\sqrt{3})$
neighbor 10	$(x_1 + 3/\sqrt{3}, y_1, z_1 - 3/\sqrt{3})$	neighbor 11	$(x_1 + 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1)$
neighbor 12	$(x_1 - 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1)$	neighbor 13	$(x_1, y_1 - 3/\sqrt{3}, z_1 - 3/\sqrt{3})$
neighbor 14	$(x_1, y_1 - 3/\sqrt{3}, z_1 + 3/\sqrt{3})$	neighbor 15	$(x_1, y_1 + 3/\sqrt{3}, z_1 - 3/\sqrt{3})$
neighbor 16	$(x_1, y_1 + 3/\sqrt{3}, z_1 + 3/\sqrt{3})$	neighbor 17	$(x_1 + 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1)$
neighbor 18	$(x_1 - 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1)$	neighbor 19	$(x_1 + 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1 + 3/\sqrt{3})$
neighbor 20	$(x_1 - 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1 + 3/\sqrt{3})$	neighbor 21	$(x_1 + 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1 - 3/\sqrt{3})$
neighbor 22	$(x_1 - 3/\sqrt{3}, y_1 + 3/\sqrt{3}, z_1 - 3/\sqrt{3})$	neighbor 23	$(x_1 + 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1 + 3/\sqrt{3})$
neighbor 24	$(x_1 + 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1 - 3/\sqrt{3})$	neighbor 25	$(x_1 - 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1 - 3/\sqrt{3})$
neighbor 26	$(x_1 - 3/\sqrt{3}, y_1 - 3/\sqrt{3}, z_1 + 3/\sqrt{3})$		

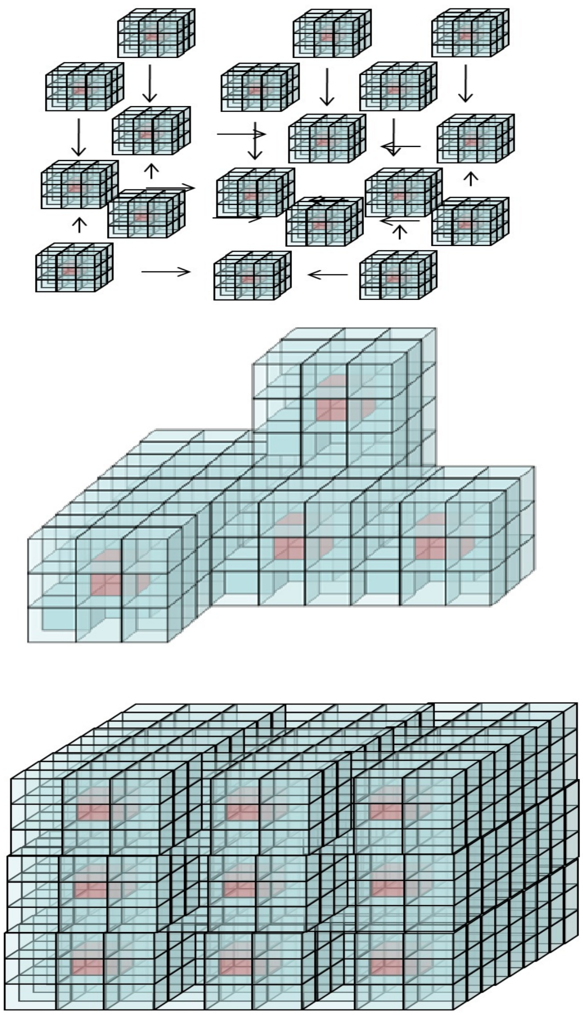

Fig. 3. Tilling system used.

In the following we will discuss some properties of Algorithm 1. To begin with, we show that Algorithm1 is local algorithm by showing that it will terminate in a constant number of steps. Let 3DDOM be the set of dominator nodes that results from applying Algorithm1 on each node in from UDG.

A. *Algorithm 1 is a local algorithm*

Proof: If the selection of a dominator in a cube depends only on the nodes that are at most constant hops away from the nodes in the given cube, then the algorithm is local. The selection of a dominator of a node in a cube of class 1 is done by checking only the nodes inside that cube, which is not more than 1 hop away. If the node of class $i <> 1$, here the algorithm waits the results that comes from neighbors nodes of lower classes, so eventually it will reaches nodes of class 1 after at most $i - 1$ steps. ■

Algorithm 1: A LOCAL ALGORITHM FOR 3D DOMINATING SETS

// Algorithm is executed independently by each node. Execution starts either when a node X needs to find its dominator, or if it receives a request to find a dominator in its cube.

begin
1 X determines its class number using its coordinates and the tiling information.
2 X finds all its neighbors and obtains their coordinates and class numbers.
3 Let CX be the cube that contains X
4 **if** X *of class number* 1 **then**
5   The node N closest to the center of CX is designated as a dominator.
6 **else**
7   X finds the set $S1(X)$ that contains all nodes in CX that has no neighbor of lower class.
8   **if** $S1(X)$ *is not empty* **then**
9   One of nodes in $S1(X)$ closest to the center of CX is designated as a dominator.
10   **else**
11   X requests from every neighbor of lower class number to run the algorithm if not already running.
12   When all nodes in CX finish their calculations, node M from CX that is not dominated and closest to the center of CX becomes a dominator.
13 X informs all its neighbors that a dominator selection in its cube is completed and gives them the results.
end

B. *Every vertex of UDG is either in 3DDOM or adjacent to a vertex in 3DDOM. Thus, set 3DDOM is a dominating set of the UDG*

Proof: For a node v in UDG that is not in 3DDOM. If v is of class 1, then one of the nodes in the cube containing v is designated as a dominator in line 5. Else if v is of class $i > 1$ and at least one node of its cube is not dominated by a node of an adjacent lower class, one of the nodes in the cube is designated as a dominator in line 12. Since the diameter of the cube is 1, node v is dominated by the designated node. ■

C. *The hop distance between a node $u \in 3DDOM$ and it closest node $v \in 3DDOM$ is at most 3*

Proof: Assume the shortest path between a node $u \in 3DDOM$ and it closest node $v \in 3DDOM$ has at least 4 hops, i. e. u, x_1, x_2, x_3, v. But x_1 and x_3 are not in 3DDOM because they are neighbors to other in 3DDOM. If $x_2 \in 3DDOM$, it makes the hop distance between u and x_2 equal to 2, a contradiction. If x_2 is a neighbor to a node $w \in 3DDOM$, it makes the hop distance between u and w equal to 3, a contradiction. ■

There are many algorithms proposed to connect a set of dominators [2], [3], [4], [10], most of them depend on using three hops node information. Our algorithm for finding the connectors can be described as follows: each node X from 3DDOM independently runs Algorithm 2.

Algorithm 2: A LOCAL ALGORITHM FOR 3D VIRTUAL BACKBONE

// Let $3DDOM$ be the set of dominating set calculated by applying Algorithm 1.
// Each node X from 3DDOM run the rest of the algorithm.
// Let $H1(X)$ is the set of one hop neighbor of X, $H2(X)$ is the set of the two hops neighbors of X, and $H3(X)$ is the set of the three hops neighbors of X

begin
  for *every node Y in $H2(X)$* **do**
    if *Y in $3DDOM$ and class number of Y is less than the class number of X* **then**
      Node X chooses from $H1(X)$ a node U with the highest degree that creates a path (X, U, Y)
  for *every node Y in $H3(X)$* **do**
    if *Y in $3DDOM$ and class number of Y is less than the class number of X* **then**
      Node X chooses two nodes U from $H1(X)$ and V from $H2(X)$ that creates the path (X, U, V, Y). Where U and V have the highest degree.
  The union of the selected nodes along with the nodes from $3DDOM$ describes the virtual backbone for UDG.
end

V. Conclusion

In this paper, we proposed a fully local algorithms that construct a virtual backbone of UDG in 3D environment in a constat time. The algorithm does not rely on the spanning tree construction, which makes it practical for situations, where the topology changes are frequent and unpredictable. We showed the importance of the constructed dominating set on the construction of the virtual backbone, but the approximation ratio of the virtual backbone is still not clear, we leave this for future work.

References

[1] A.E. Abdallah, T. Fevens, and J. Opatrny. 3d local algorithm for dominating sets of unit disk graphs. In *Proc. of the Canadian Conference on Computational Geometry (CCCG2010)*, pages 35–38, Winnipeg - Canada, 2010.

[2] K. Alzoubi, X. Li, Y. Wang, P. Wan, and O. Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(4):408–421, 2003.

[3] K. Alzoubi, P. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications Networks*, 4(1):141–149, 2002.

[4] K. Alzoubi, P. Wan, and O. Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *Proc. of the Third ACM international Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 02)*, pages 157–164, Lausanne, June 2002.

[5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks journal*, 7(6):609–616, 2001.

[6] V. Chvatal. A greedy heuristic for the set covering problem. *Math. Oper. Res. 4*, pages 233–235, 1979.

[7] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Math*, 86:165–177, 1990.

[8] J. Czyzowicz, S. Dobrev, T. Fevens, H. Gonzalez-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia. Local algorithms for dominating and connected dominating sets of unit disk graphs. In *Proc. of the 8th Latin American Theoretical Informatics Symposium LATIN08*, April 2008.

[9] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proc. of the IEEE International Conference on Communications (ICC 97), Vol. 1*, pages 376–380, Monral, June 1997.

[10] B. Gao, Y. Yang, and H. Ma. An efficient approximation scheme for minimum connected dominating set in wireless ad hoc networks. In *Proc. of the IEEE Vehicular Technology Conference*, pages 2744–2748, Los Angeles, September 2004.

[11] M. Garey and D. Johnson. *Computers and Intractability-A Guide to the Theory of NP-Completeness*. CA:Freeman, San Francisco, 1979.

[12] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWA Publishing Company, Boston, MA, 1995.

[13] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, pages 256–278, 1974.

[14] R. Karp. Reducibility among combinatorial problems. In *Proceedings of the Symposium on Complexity of Computer Computations*, pages 85–103, 1972.

[15] M. Mauve, J. Widmer, and H. Hartenstein. A survey of position-based routing in mobile ad-hoc networks. *IEEE Network Magazine*, 15(6):30–39, 2001.

[16] P. Slavik. A tight analysis of the greedy algorithm for set cover. In *Proc. of the 28th ACM Symposium on Theory of Computing (STOC)*, pages 435–441, 1996.

[17] 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 Communications and Networks*, 4(1):59–70, 2002.

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