3D Local Algorithm for Dominating Sets of Unit Disk Graphs

A.E. Abdallah and T. Fevens and J. Opatrny*

Abstract

A dominating set of a graph $G = (V, E)$ is a subset $V' \in V$ of the nodes such that for all nodes $v \in V$, either $v \in V'$ or a neighbor u of v is in V'. Several routing protocols in ad hoc networks use a dominating set of the nodes for routing. None of the existing algorithms has a constant approximation factor in a constant number of rounds in 3D. In this paper, we use the nodes' geometric locations to propose the first local, constant time algorithm that constructs a Dominating Set and Connected Dominating Set of a Unit Disk Graph (UDG) in a 3D environment. The approximation ratios of our algorithms are given.

1 Introduction

Wireless ad hoc and sensor networks are most often modeled by Unit Disk Graphs [8], abbreviated by UDGs, a geometric graph $G = (V, E)$ in which the vertex set V is a set of n points in $\mathbb{R}^d$, where d is the dimensions, and the edge set E consists of m pairs from V. Let $dist(u, v)$ be the Euclidean distance between the nodes u and v: $dist(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}$. Two nodes u and v are considered adjacent if the Euclidean distance between them is less than or equal to 1 unit. The *neighborhood* $N1(v)$ of a vertex $v \in V$ is the set of all vertices adjacent to v, i.e., $N1(v) = \{u | uv \in E\}$. A path from the node u_1 to the node u_2 is a sequence of nodes $u_1 = v_1, v_2, ..., v_k = u_2$, such that v_i and v_{i+1} are neighbors. The length of the path is the sum of the number of edges (i.e., number of hops) along the path. For a node v, we define $N2(v)$ and $N3(v)$ to be the set of nodes that are 2 and 3 hops away from v respectively.

A subset V' of V is called *dominating* if every vertex from $V - V'$ is adjacent to some node in V'. A dominating set is called a *connected dominating set* (CDS) if the subgraph $P(G)$ induced by V' is connected. The smallest subset of vertices that is both connected and dominating is called a *minimum connected dominating set* (MIN-CDS). A subset of vertices in a graph G is an independent set if no two vertices are connected by an edge. An independent set is maximal (MIS) if it cannot be extended by the addition of any other vertex from the graph without violating the independence property [7].

A subgraph of G, $P(G)$ is called a t-spanner of G if the length of the shortest path between any two nodes in $P(G)$ is not more than t times longer than the shortest path connecting them in G, where t is known as the stretch factor.

In general, the main characteristics of mobile computing are low bandwidth, mobility, and low power. Due to these characteristics, routing in ad hoc wireless networks requires fast convergence and low communication overhead. Hence, routing information has to be localized to adapt quickly the network topological changes. Localized CDS-based routing can be a solution to this kind of network environment. The main advantage of CDS-based routing is that it centralizes the whole network into a relatively small connected dominating set sub network, which means only gateway hosts maintain routing information, so that as long as network topological changes do not affect this sub network and there is no need to recalculate routing tables. Clearly, the efficiency of this approach depends largely on the process of finding the dominating sets and the size of the corresponding sub-networks [13]. Finding a MIN-CDS is NP-complete in general even for UDG's [11].

One of the major assumptions made by many routing algorithms is assuming that nodes are deployed in a 2D plane. Such an assumption is invalid in real life scenarios and hence these algorithms cannot be applied in many situations. Some applications of 3D networks arise: (1) Building networks where nodes are located on different floors. (2) Irregular terrains like mountains and hills leaves the nodes lying on surfaces embedded in 3D. (3) Underwater networks that perform ocean column monitoring. (4) Space exploration where wireless sensor networks will play an important role in planetary explorations. A rover functioning as a base station collects measurements and relays aggregated results to an orbiter. The transition from 2D to 3D is not always easy, since many problems in 3D are significantly harder than their 2D counterparts.

A distributed algorithm is called *local* if 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. Thus, during the algorithm, no node is ever aware of the existence of the nodes of the network further away than this constant number of hops.

*Department of Computer Science and Software Engineering Concordia University Montréal, QC, Canada, H3G 1M8,{ae_abdal,fevens,opatrny}@cs.concordia.ca

Several algorithms have been previously proposed to construct an independent dominating set and a CDS. But none of these algorithms [3, 4, 5, 6, 9, 12] has the following 3 characteristics: (1) can be constructed in a constant time (local according to the definition above), (2) has a constant approximation bound and (3) works in a 3D environment.

In this paper we propose the first local algorithms that construct an independent dominating set (IDS) and a CDS of a Unit Disk Graph (UDG) in a 3D environment (also termed a Unit Ball Graph). The new algorithms have a constant time complexity and approximation bounds that are completely independent of the size of the network. In this paper we assume the following: (1) Nodes are static or can be viewed as static during a reasonable period of time. (2) Each node knows the geometric location of all its neighbors. (3) In the communication model, a node can send a message to all its neighbors and can receive a message from one of them at a time.

The rest of the paper is organized as follows. We present a tiling system of the 3D space needed in our algorithms in Section 2. In Section 3, we introduce our local algorithm for IDS based on this tiling system and give an upper bound on its' size. In Section 4, we describe a local algorithm for generating a connected dominating set. We conclude our paper in Section 5.

2 Tiling System

In this section we present a truncated octahedron tiling system of the 3D space to assign to each node a class number depending on the position of the node within the tiling system. Then, based on this tiling classification system, we present generalizations of the algorithms from [10] for constructing dominating sets and CDSs in 3D.

Consider a truncated octahedron of unit diameter as shown in Figure 1 which consists of 14 faces, 6 squares of edge length equal to α, and 8 hexagons of edge length equal to α. Since the truncated octahedron has a diameter equal to 1, then $\alpha = \frac{1}{\sqrt{10}}$.

The tiling system is based on a two level subdivision of the three dimensional space. At the highest level, the space is tiled with identically shaped tiles that fill the entire space, with no gaps or overlaps. Each tile used in our tiling system consists of 65 truncated octahedra which occupy the entire volume of the tile with no gaps or overlaps. Each truncated octahedron in the tile represents one class which has a unique integer. For any truncated octahedron only faces $1, 2, 3, 4, 5, 6$ and 7 belong to it, see Figure 1. In other words, if a node located exactly on the shared face between two truncated octahedra T_1 and T_2, the node is considered of class 1 if according to T_1 this face is $1, 2, 3, 4, 5, 6$ or 7 otherwise it will be considered to be of class 2. Assuming that the first truncated octahedron, class 1, we will call this class as the *center class*, is centered at the coordinates (x_1, y_1, z_1), i.e. the z-axis passes through the center of face 1, the x-axis passes through the center of the edge between face 5 and the face opposite to face 2, and y-axis passes through the center of the edge between face 4 and the face opposite to face 3. We will call this orientation the *centering orientation*, the coordinates of the centers of the classes from 2 to 65 are not shown here because of the space limitation. They all have the same orientation as class 1. See Figure 2 for an example of the tile used, showing the placement of the truncated octahedra in the tile with the associated classes labels.

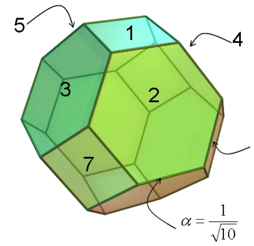

Figure 1: Unit diameter truncated octahedron, the faces labeled $1, 2, 3, 4, 5, 6, 7$ belong to the class represented by this truncated octahedron.

Assume that the tiling starts by placing the center of one tile, T_j, at the coordinate (x_1, y_1, z_1), with orientation equal to the centering orientation. To cover all the faces of T_j we need 14 other adjacent tiles that are in contact with T_j in the positions summarized on Table 1. Each tile has the same orientation as T_j. It is clear that any point can calculate locally its class number by determining to which tile and corresponding truncated octahedron it belongs. In the following we prove some properties of our space tiling system.

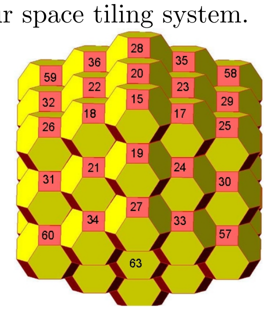

Figure 2: The tile used in the tiling system divided into 65 truncated octahedra of diameter 1 and the class numbering associated with the truncated octahedra.

Lemma 1 *In the 3D space tiling system above, any two points that are of the same class number, but belong to two different truncated octahedra, are at Euclidean distance greater than 2.*

Table 1: Coordinates of the 14 tiles around T_j, with $\alpha = \frac{1}{\sqrt{10}}$.

T_j	(x_1, y_1, z_1)
T_{j1}	$(x_1, y_1, z_1 - 10\alpha\sqrt{2})$
T_{j2}	$(x_1, y_1, z_1 + 10\alpha\sqrt{2})$
T_{j3}	$(x_1 + 6\alpha, y_1 + 4\alpha, z_1 + 5\alpha\sqrt{2})$
T_{j4}	$(x_1 + 6\alpha, y_1 + 4\alpha, z_1 - 5\alpha\sqrt{2})$
T_{j5}	$(x_1 - 4\alpha, y_1 + 6\alpha, z_1 + 5\alpha\sqrt{2})$
T_{j6}	$(x_1 - 4\alpha, y_1 + 6\alpha, z_1 - 5\alpha\sqrt{2})$
T_{j7}	$(x_1 - 6\alpha, y_1 - 4\alpha, z_1 + 5\alpha\sqrt{2})$
T_{j8}	$(x_1 - 6\alpha, y_1 - 4\alpha, z_1 - 5\alpha\sqrt{2})$
T_{j9}	$(x_1 + 4\alpha, y_1 - 6\alpha, z_1 + 5\alpha\sqrt{2})$
T_{j10}	$(x_1 + 4\alpha, y_1 - 6\alpha, z_1 - 5\alpha\sqrt{2})$
T_{j11}	$(x_1 + 10\alpha, y_1 + 2\alpha, z_1)$
T_{j12}	$(x_1 + 10\alpha, y_1 - 2\alpha, z_1)$
T_{j13}	$(x_1 - 10\alpha, y_1 + 2\alpha, z_1)$
T_{j14}	$(x_1 - 10\alpha, y_1 - 2\alpha, z_1)$

Proof. (see [1] for all lemmas and theorems proofs). □

In our initial study of tiling the 3D space, we used a cube of unit diameter as a cell (class) instead of a truncated octahedron. We found that each tile would need at least 125 cubes to guarantee that the points with the same class number in different tiles are separated by a distance greater than 2. Compared to using a truncated octahedron, this number of cubes would substantially increase the constant number of rounds required for the algorithms in Sections 3 and 4.

3 A Local Algorithm For 3D Independent Dominating Sets (3D-LIDS)

Using the space partition described in the previous section, each node can determine its class number locally (using a constant number of arithmetic operations). Because the nodes are aware of the locations of all their neighbors, using the *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 truncated octahedron are neighbors because the diameter of the truncated octahedron is 1.

Our local construction of the dominating sets is based on a similar algorithm proposed by Czyzowicz *et al.* [10] for $2D$. In this algorithm the dominator node m can be chosen according to different heuristics. I.e., the node with the highest degree; the node with the maximum power-level, if the power saving is an important issue for the algorithm; or the node closest to the center of the truncated octahedron. (This latter heuristic is used in our algorithm description).

Let T_x be the truncated octahedron that contains the node x. Each node x independently does one of the following depending on its class number and two hop information:

1. If x is of class 1, then a node m closest to the center of T_x in the same truncated octahedron (T_x) will be designated as a dominator.
2. If x is of class other than 1, using the information about its' neighbors, x defines a set $S_1(x)$ of all nodes in the same truncated octahedron that have no neighbor of lower class, and then chooses from $S_1(x)$ a node m closest to center of T_x to be a dominator.
3. If x is of class other than 1, and the set $S_1(x)$ is empty, then x requests from every neighbor i of lower class number to run the algorithm if not already running. When all nodes in T_x finish their calculations, node m from T_x that is not dominated and closest to the center of T_x becomes a dominator.

When the node finishes its calculation, it informs all its neighbors that the dominator selection is complete in its truncated octahedron. This algorithmns is local since the number of classes is finite.

Before we bound the size of the independent dominating set that resulted from 3D-LIDS, we will describe some of its properties. Let Dom be the set of dominator nodes that results from applying 3D-LIDS on each node in V.

Lemma 2 *The selection of a dominator in a truncated octahedron of Class i by 3D-LIDS depends only on the nodes that are at most $i-1$ hops away from the nodes in the given truncated octahedron.*

Lemma 2 proves that the 3D-LIDS is a local algorithm because it terminates in a constant number of steps. The $2D$ versions of the following two lemmas are proved in [10], and the 3D versions of these lemmas are included here for completeness.

Lemma 3 *Let G be a connected UDG. Dom is a dominating set of G.*

Lemma 4 *The Euclidean distance between any two nodes of Dom is more than one. Thus Dom is an independent set of G.*

Lemma 5 *For any dominator node $u \in Dom$, there is at least another dominator $v \in Dom$ such that the hop distance between them is at most 3.*

Lemma 6 *For any node u, the number of dominators inside the sphere centered at u with a radius of k units is bounded by a constant η_k, where $\eta_k = \frac{\frac{4}{3}\pi(k+0.5)^3}{\frac{4}{3}\pi(0.5)^3}$.*

Theorem 7 *Let $G = UDG(V, E)$ be a unit disk graph and Dom be a set of dominators for G calculated by 3D-LIDS. For any optimal dominating set Dom^* of G, we have $|Dom|/|Dom^*| < 24$. Thus the approximation ratio of 3D-LIDS is* 24.

4 Connected Dominating Set

Our local algorithm to construct a CDS consists of two phases. In the first phase, an IDS is found using 3D-LIDS. In the second phase, each dominator creates paths connecting dominators that are at most three hops apart. There are many algorithms proposed to connect a set of dominators [2, 3], most of them depend on using three hops node information. In our algorithm the connector node can be chosen between different candidates according to different heuristics. I.e., the node with the maximum power level, if the power saving is an important issue for the algorithm; the node closest to a dominator; or the node with the highest degree; (This latter heuristic is used in our algorithm description). Our algorithm for finding the connectors can be described as follows: each node x independently does the following:

1. For every dominator node y in $N2(x)$ with a lower class number than x. Node x chooses from $N1(x)$ a node u with the highest degree that creates a path (x, u, y).
2. For every dominator node y in $N3(x)$, with a lower class number than x, or has the same class number as x but x is closer to the central class than y. Node x chooses two nodes u from $N1(x)$ and v from $N2(x)$ that creates the path (x, u, v, y). Where u, v can be chosen according to the same heuristic above.

Lemma 8 *If UDG is connected, then the set of dominators and connectors constructed by 3D-LIDS and Algorithm 3D-LCDS is a connected dominating set.*

Lemma 9 *In the worst case, the number of connectors added by 3D-LCDS is* 280 *for every dominator.*

It has been proved in [2] that the stretch factor of their connected dominating set is 3. It can be observed that this proof is applicable to our 3D-LCDS. The new algorithms were implemented. Experiments show that their performance is very good on randomly generated 3D UDG graphs, details can be found in [1].

5 Conclusion

In this paper, we proposed the first fully local algorithms that construct a dominating set and a connected dominating set of a UDG in 3D in constant time. The algorithms do not rely on the construction of a spanning tree, which makes them practical for situations where topology changes are frequent and unpredictable. We proved that the size of the constructed dominating set is at most 24 times the optimal.

References

[1] A. Abdallah. Position-based routing algorithms for three-dimensional ad hoc networks. In *PhD thesis, Concordia University*, Montreal, January 2009.

[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] D. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694–1701, 1981.

[6] D. Baker, A. Ephremides, and J. Flynn. The design and simulation of a mobile radio network with distributed control. *IEEE Journal on Selected Areas in Communications*, 2(1):226–237, 1984.

[7] J. Bondy and U. Murty. Graph theory with application. Elsevier Science Publishing Co., 1976.

[8] 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.

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

[10] 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.

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

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

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