Sensor Network Configuration and the Curse of Dimensionality

Sameera Poduri[1], Sundeep Pattem[2], Bhaskar Krishnamachari[1,2], Gaurav S. Sukhatme[1,2]
[1] Department of Computer Science, [2] Department of Electrical Engineering,
University of Southern California,
Los Angeles, CA 90089-0781, USA
{sameera, pattem, bkrishna, gaurav}@usc.edu*

Abstract

Sensor network problems in three dimensions have not been adequately addressed - there is a tendency to either ignore the extension of algorithms from two dimensions (2D) to three dimensions (3D) for simplicity or believe that it is straightforward. We draw examples from well known problems in geometry and argue that this step needs special investigation - while some properties of networks in 2D directly generalize to 3D, many require additional computational complexity, and a few do not generalize at all. This paper focuses on the problem of deployment and configuration of sensor networks in 3D, draws attention to the fundamental difficulties involved, and presents a set of local geometric rules that can be used to construct efficient network topologies in 3D.

1. Introduction

A large number of sensor networks embedded in the physical world will be three dimensional (3D). Take for instance, networks deployed on multiple floors of a building, in a forest (on trees of different heights), or underwater. Current literature is heavily focussed on deployments in two dimensions (2D) and there is a tendency to believe that results will hold even in 3D. Consider the problem of deploying and configuring a network. Given a set of nodes, how should they be placed? How should their duty cycles and transmission powers be adjusted to save power? These questions are reasonably well understood in 2D. The network can be deployed randomly with high density and later pruned using sleep scheduling and/or power control techniques [13, 11, 14, 17]. Alternatively, the node placement can be controlled at deployment stage to construct efficient topologies [10, 12]. Extending these approaches to 3D raises many questions: Are random high density deployments practical for 3D scenarios? How does the computational complexity of various algorithms increase in 3D? Which of the existing techniques is better suited in 3D?

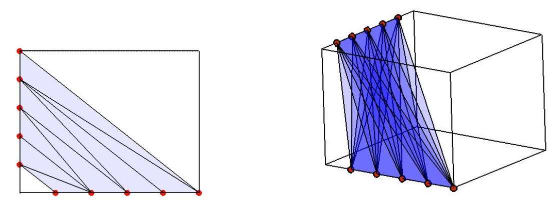


Figure 1. The curse of dimensionality: Delaunay triangulation of n points in 2D will have at most $O(n)$ triangles but in 3D it can have as many as $O(n^2)$ tetrahedra.

Network configuration is governed by the sensing coverage and connectivity requirements of the application. Both coverage and connectivity are functions of the relative positions of nodes and are inherently geometric in nature. There are several problems in geometry that are easily solved in 2D but are very complex in 3D. Examples include the art-gallery and sphere-packing problems that are closely tied to sensing coverage optimization. The art-gallery problem can be solved optimally in 2D but is NP-hard in 3D. In fact, the main tool used for solving the 2D case, triangulation, does not generalize to 3D (Fig. 1) - there exist polyhedra whose interior cannot be partitioned into tetrahedra whose vertices are selected from the polyhedra vertices. The sphere-packing problem i.e., finding the densest packing of spheres in 3D, remained an open problem for nearly 400 years. Its 2D counterpart, that of finding the densest packing of circles is easily solved.

In the initial euphoria of the sensor networks vision,

* This work is supported in part by the National Science Foundation (NSF) under grants CNS-0540420, CNS-0520305, CNS-0325875, CCR-0120778, CAREER CNS-0347621, NeTS-NOSS CNS-0435505, and ITR CNS-0325875.

of massively distributed and untethered nodes, the mythical beast of “uniform random deployment” was born and quickly became a well accepted basis for design and evaluation of many algorithms. Such a model is not representative of experimental testbeds currently being deployed or of medium size deployments of the future. While this assumption is debatable enough in 2D, it becomes even more so in 3D because the node density required to ensure connectivity seems to be prohibitively high in 3D (Fig. 2). Earlier designs have to be critically re-examined in this light.

In this paper, we review a number of deployment and configuration techniques in 2D to highlight the nature of the difficulty involved in extending them to 3D. These insights will be of value to stimulate research into new and appropriate techniques for 3D. As a first step in this direction, we have explored local geometric conditions that can guarantee global network properties in 3D. We present some initial results on these and conclude with a discussion on future directions.

2. Related work

There is a rich body of literature on deployment and configuration algorithms for networks in 2D. Most of them assume a dense and well connected initial deployment and prune the network using sleep-scheduling and/or power control techniques. Here, network configuration is a decision problem, of identifying links and/or nodes that are redundant and deleting them. This can be considered a ‘top-down’ approach. In sleep scheduling techniques like CCP [14] and OGDC [17], nodes that are redundant for sensing coverage as well as connectivity are turned off. The coverage-redundant nodes can be identified locally in $O(d^2)$ time at each node in 2D and $O(d^3)$ time in 3D [7], where d is the average node degree and n is the network size. Power control techniques [13, 11] involve adjusting the transmission power of nodes and thus varying the network topology. They cannot affect the sensing coverage. Many of the techniques use angular information coupled with a disc model for communication to guarantee connected networks [15] or proximity graphs like RNG [3]. Extensions to 3D have not been adequately addressed. In [1], an algorithm has been presented to extend CBTC [15]. The computational complexity at each node is $O(d^3 log(d))$ as against $O(dlog(d))$ in 2D, d being the average node degree. XTC [16] does not use the disc assumption or angular information; given an initially connected network in 3D, it can retain connectivity.

In contrast to the ‘top-down’ approach used by the above mentioned techniques, network topologies can also be controlled in a ‘bottom-up’ fashion where nodes positions are carefully chosen to ensure coverage and connectivity properties [10, 12]. *NET* graphs [12] are a single parameter family of graphs that can achieve different coverage and connectivity trade-offs based on the value of the sector angle θ. Deployment algorithms for networks of mobile nodes have been proposed. There are two popular approaches for maximizing sensing coverage - the potential field approach [6] and the voronoi diagram approach [4]. These approaches can be significantly more complex in 3D. For instance, computation of 3D voronoi diagrams takes $O(n^2)$ time. Moreover these algorithms are only useful for underwater and arial deployments where the motion of the mobile node is truly 3D.

There have been a few recent efforts to understand 3D sensor network problems like geographic routing [9] and localization [2].

3. Why 3D configuration is hard

In this section, we list some issues in extending existing designs for deployment and configuration to 3D.

1. Uniform random deployment: Random deployment with high density is a well accepted set-up for 2D deployments. For n nodes deployed randomly in a unit cube, $[0,1]^d$, in d dimensions, the critical transmission radius for connectivity is $O((\frac{log(n)}{n})^{\frac{1}{d}})$ [5]. It appears that the critical constant will have a much higher value in 3D compared to 2D (Fig. 2). For $n = 1000$, the critical radius in 2D is ≈ 0.07 while in 3D it is ≈ 0.2, resulting in a critical average node degree of ($\pi \cdot 0.07^2 \cdot 1000 \approx$) 15 in 2D and ($\frac{4\pi}{3} \cdot 0.2^3 \cdot 1000 \approx$) 34 in 3D. This implies that a uniform random deployment in 3D that is almost surely connected will have a highly dense topology.

If we assume each node in a uniform random deployment covers a sphere of radius R_s around it then the ratio of area covered by the network is $C = 1 - e^{-\lambda V_s}$ where λ is the density of deployment and $V_s = \frac{4}{3}\pi R_s^3$. For the domain to be covered with high probability, say $C \geq 0.99$ we must have $\lambda \geq \frac{4.6}{V_s}$. The sensing region of a node will intersect with the sensing regions of all nodes that are less than $2R_s$ away. The number of such nodes, on an average, will be $\lambda(\frac{4}{3}\pi(2R_s)^3) = \lambda(2^3 V_s) \geq 4.6 \cdot 2^3 \approx 37$ in 3D. The corresponding number in 2D is ≈ 18.

In several distributed algorithms the computation at each node depends on the number of communicating neighbors or the number of nodes with which its sensing range overlaps. In 3D, both these numbers are twice the corresponding numbers in 2D.

2. Angular information: Several network configuration algorithms involve using angular information of neighboring nodes ([15, 3, 14, 17] are a few examples). A basic primitive needed for this is an ordering of neighbors based on their orientations. For example in several power control techniques [15, 3], each node increases its transmission power till the angle between adjacent neighbors is less than

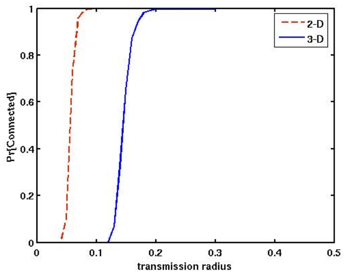


Figure 2. The number of neighbors needed for connectivity in a random 3D graph is high - around 30 for 1000 nodes with binary disc communication model

a threshold. Each time a new neighbor is added the node can insert it into a list of neighbors sorted according to their orientations in $O(log(d))$ time steps where d is the average node degree. This is a trivial operation in 2D. In 3D there is no natural ordering of neighbors based on angles. We propose an efficient algorithm to obtain an ordering and checking angular orientation of neighbors in Section 4.3. Its computation complexity is $O(dlog(d))$.

3. Coverage criteria: To check if every point in a domain is covered by k nodes, it is sufficient to check if the intersections of sensing regions are covered by $k+1$ nodes. In 2D the intersections of sensing regions are points and the checking step takes $O(d^2)$ time where d is the average number of neighboring sensing regions that a node's sensing region will intersect. In 3D the intersections will be circles on the surface of a sphere. To check for coverage this problem can be reduced to a 2D problem on the surface of the sphere. The time complexity will be $O(d^3)$ at each node. The number of intersections d might also be larger because a higher density of initial deployment. If this step must be repeated frequently for load balancing between the nodes the computation can become a significant overhead.

4. Symmetric placements: Maximizing sensing coverage implies minimizing overlap between the sensing regions of nodes. If we assume that the sensing regions are spheres, then maximum coverage is achieved when neighbors are located symmetrically on the communication range. Local symmetric arrangement of neighbors around a node is trivial in 2D and is possible for any number of neighbors - given k neighbors, they can be placed at $\frac{2\pi}{k}$ from each other. In 3D, such a placement is only possible for degrees 4, 6, 8, 12 and 20. This is because symmetric placement of k neighbors corresponds to a regular polyhedra of k vertices and it is well known that there exist only five regular convex polyhedra.

5. Structural restrictions: In many scenarios, the possible locations for nodes are restricted by the structure of the environment. For example in a building, nodes may only be deployed along the walls and not at arbitrary coordinates. This restriction is more severe in 3D. There are applications like environmental monitoring (eg. in a desert) where nodes can be deployed at any random location as long as they are on the ground. But there will rarely be a scenario where nodes can be deployed at any arbitrary height. Truly random 3D deployments will probably only happen in underwater or aerial settings. The algorithms designed must take this restriction into account and whenever possible exploit the knowledge of the structure imposed on the network topology.

From the above discussion, we learn that (i) random high density deployments are not suitable for 3D, (ii) the feasibility and complexity of techniques proposed for 2D must to be carefully studied for 3D. We hope that these insights will stimulate research to revisit and extend current 2D techniques and also investigate novel 3D specific techniques. The next section presents some early results from our efforts in this direction.

4. Initial results using local geometry

Key to achieving efficient network configuration in a distributed manner are local geometric conditions that can influence global coverage and connectivity properties. Proximity graphs (such as RNG, GG) and symmetric tiling structures are examples of topologies that have desirable global properties and can be constructed using purely local rules. We present these local rules and also an algorithm to integrate them with network configuration and deployment. For the purpose of analysis we assume a binary disc communication model and sensing model for the nodes. It must be emphasized here that such analysis only provides broad guidelines. Any algorithm design must account for irregular communication range, etc.

4.1. Proximity Graphs

Proximity graphs such as the Relative Neighborhood Graph (RNG) , Gabriel Graph (GG) and Delaunay Graph (DelG) are guaranteed to be connected and have other useful properties [11]. They have been extensively used for topology control, routing, etc.

We now present sufficient conditions for the communication graph to contain each of RNG, GG, and DelG. The analysis is non-trivial because the edge-lengths in communication graphs are restricted by the communication range of nodes, while in proximity graphs they depend only on the relative positions of nodes and very long edges are possible.

Let V be the set of nodes in a wireless communication network in $\mathbb{R}^2$. Let R_c^x be the communication radius of $x \in$

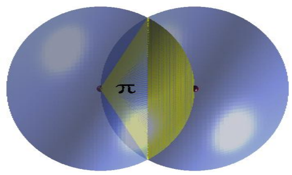


Figure 3. RNG cone angle condition

V. For $u, v \in V$, let $d(u, v)$ denote the Euclidean distance from u to v. Given positive $r \in \mathbb{R}$, let $S(p, r)$ be the sphere consisting of points whose distance from point p is strictly less than r. Define the lune, denoted $L(p, q)$, to be the intersection of two spheres, both of radius $d(p, q)$, centered at these points, that is, $L(p, q) = S(p, d(p, q)) \cap S(q, d(p, q))$.

Relative Neighborhood Graph (RNG(V)): The undirected graph containing an edge (u, v) if there is no point $w \in V$ that is simultaneously closer to both u and v. Equivalently, (p, q) is an edge if $L(p, q) \cap V = \emptyset$.

$$E = \{(u, v) | u, v \in V \text{ and } \exists \text{ no } w \in V \ni$$
$$d(u, w) < d(u, v) \text{ and } d(v, w) < d(u, v)\}$$

Theorem 1 *If each node $X \in V$ has at least one neighbor in every $\theta = \pi$ cone of $S(X, R_c^X)$, the communication graph is a supergraph of $RNG(V)$.*
Proof: *Consider any node $X \in V$. Suppose X has at least one neighbor in every π cone of $S(X, R_c^X)$. It is sufficient to show that for any node Y outside $S(X, R_c^X)$, the edge $(X, Y) \notin RNG(V)$. The lune $L(X, Y)$ will contain a cone whose apex angle α is at least $\frac{2\pi}{3}$ and a corresponding solid angle of at least $\theta = 2\pi(1 - cos(\alpha)) = \pi$ (Fig. 3). By premise, $\exists$ a node Z that lies in $L(X, Y)$.*

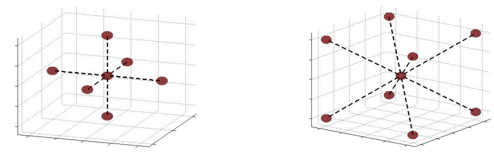
This implies that the RNG does not have any edges incident on X with length greater than R_c^X and therefore $RNG(V)$ is contained in the communication graph.$\square$

It can be shown that π is the largest angle that satisfies the above property.

Theorem 2 *If each node $X \in V$ has at least one neighbor in every $\theta = 2\pi(1 - \frac{r}{R_c})$ cone of $S(X, R_c^X)$, the communication graph is a supergraph of $GG(V)$ and $DelG(V)$. Moreover, $2\pi(1 - \frac{r}{R_c})$ is the largest angle that satisfies this property.*
The proof is similar to that of RNG.

4.2. 3D tiling structures

Tiling structures, because of their symmetry, possess many interesting properties: they are connected and sparse, and the symmetric arrangement maximizes sensing coverage for a given node degree [8, 12]. A local unit arrangement can be replicated globally to fill the space. Due to this property, the local geometric rules to construct these topologies are trivial and the coordination required between nodes is minimal. This coordination is more expensive in 3D compared to 2D (section 3) and therefore tiling structures are particularly desirable in 3D.



(a) 6 neighbors (b) 8 neighbors

Figure 4. Symmetric tiling structures

In 2D, there are three possible tiling structures for node degrees 3(hexagonal), 4(square) and 6(triangle). In 3D, tilings exist for 6 (cuboid with a neighbor on each of the faces) (Fig. 4a) , 8(cuboid with neighbors on each vertex) (Fig. 4b) and 12(cubic closed packing (CCP) and hexagonal close packing (HCP)). CCP and HCP minimize the uncovered volume for a given number of nodes. Tilings with limited symmetry are possible. These arise when the hexagonal and triangle tiling in 2D are replicated on parallel planes. Each node will have 5 and 8 neighbors respectively.

4.3. Integrating geometric conditions with configuration

The local conditions described above can be integrated with node placement to construct efficient topologies. A key requirement is an algorithm to check for empty cones larger than a given $< \theta$. For instance, to check for formation of RNG, $\theta = \pi$. As discussed in section 3, this step is non-trivial in 3D because there exists no natural "order" of neighbors. We propose the following algorithm for finding the largest empty cone around a given node.

Algorithm: largestCone($G = (V, E)$**,**$v \in V$**)**
let S be the unit sphere centered at v
for each $u \in Neighbor(v)$, let $\overrightarrow{vu}$ be the direction vector from v to u
let c_u be the intersection of $\overrightarrow{vu}$ with S
let DT be spherical delaunay triangulation $c_u \forall u$
find $a_{i,j,k}$ = area of circumcircle of triangle $(u_i, u_j, u_k) \in DT$
return $max(a_{i,j,k})$

The cone returned by *largestCone* is empty because in a spherical delaunay triangulation, the circumcircle of every (spherical) triangle is empty. Suppose there exists an empty cone (whose image on the unit sphere is the circle c) that is larger than the one returned by *largestCone*. Then the cen-

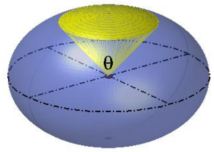


Figure 5. *NET* graph: each cone of angle θ must have at least 1 neighbor.

ter of c lies in some triangle t of the delaunay triangulation. The circumcircle of t will be larger than c which is a contradiction. Therefore *largestCone* correctly returns the largest empty cone. The computational complexity of *largestCone* is $O(dlog(d))$ where d is the number of neighbors of a node. This algorithm can be used as a primitive for extending several topology control algorithms that use directional information.

By varying the value of θ, a family of graphs can be defined. We call these *Neighbor-Every-Theta*(*NET*) [12] graphs that have the property that each node has at least one neighbor in every θ angle cone of its communication range (Fig. 5). We believe that *NET*graphs can provide a range of coverage and connectivity tradeoffs for different values of θ.

Controlled deployments are feasible when positions of individual nodes can be altered - either by the nodes themselves or by an external agent. In such scenarios, the control on position can be exploited to integrate the geometric conditions described above and also account for structural restrictions imposed by the environment.

5. Conclusion

Deployment and configuration of sensor networks to ensure desired levels of connectivity and sensing coverage is fundamentally more challenging in 3D as compared to 2D. In this paper we highlight some of the challenges in designing algorithms for 3D. We hope that these insights will foster interest in the research community to revisit and extend existing 2D algorithms and develop new techniques for 3D. We present initial results on local geometric conditions that can be used for effective network configuration in 3D.

References

[1] M. Bahramgiri, M. T. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Intl. Conf. in Computer Communications and Networks (ICCCN)*, pages 392–397, 2002.

[2] P. Biswas, T.-C. Liang, K.-C. Toh, and Y. Ye. An sdp based approach for anchor-free 3d graph realization. working paper, 2005.

[3] S. A. Borbash and E. H. Jennings. Distributed topology control algorithm for multihop wireless networks. In *Intl. Joint Conf. on Neural Networks (IJCNN)*, pages 355–360, 2002.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[5] A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. *Annals of Applied Probability*, 2006.

[6] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems*, pages 299–308, 2002.

[7] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo. The coverage problem in three-dimensional wireless sensor networks. In *IEEE Globecom*, 2004.

[8] R. Iyengar, K. Kar, and S. Banerjee. Low-coordination topologies for redundancy in sensor networks. In *ACM MobiHoc*, 2005.

[9] G. Kao, T. Fevens, and J. Opatrny. Position-based routing on 3-d geometric graphs in mobile ad hoc networks. In *Canadian Conf. on Computational Geometry*, 2005.

[10] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Intl. Conf. on Information Processing in Sensor Networks*, 2006.

[11] X. Li. Topology control in wireless ad hoc networks. Book Chapter in ”Ad Hoc Wireless Networking” Kluwer, edited by XiuZhen Cheng, Xiao Huang and Ding-Zhu Du, 2003.

[12] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme. A unifying framework for tunable topology control in sensor networks. University of Southern California, Center for Robotics and Embedded Systems Technical Report, CRES-05-004, 2005.

[13] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2):164–194, 2005.

[14] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM Conf. on Embedded Networked Sensor Systems, Sensys'03*, 2003.

[15] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Transactions on Networking*, 2005.

[16] R. Wattenhofer and A. Zollinger. Xtc: A practical topology control algorithm for ad-hoc networks. In *14th Intl. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2004.

[17] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. In *NSF Intl. Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004.