Efficient Distributed Topology Control in 3-Dimensional Wireless Networks

Amitabha Ghosh, Yi Wang, Bhaskar Krishnamachari
Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90007
{amitabhg, wangyi, bkrishna}@usc.edu

***Abstract*—Distributed topology control mechanisms for 3-dimensional settings are of considerable interest for automated network configuration in diverse applications including structural monitoring networks and underwater networks. The 3-D CBTC technique proposed by Bahramgiri *et al.* [7] has a complexity of $O(d^3 \log d)$, where d represents the average number of neighbors per node. We present two efficient alternatives. The first is a heuristic based on 2-D orthographic projections that provides excellent performance in practice, but is theoretically not guaranteed to produce a connected network. The second is a more rigorous approach based on spherical Delaunay triangulation (SDT). Both have significantly better running times that scale as $O(d \log d)$. Our simulation results indicate that network topologies generated based on the SDT algorithm have substantially lower average node degree and average transmission power level compared to the original network for random deployments.**

I. Introduction

Multihop wireless networks deployed in 3-dimensional space have potentially diverse applications in structural health monitoring [1], underwater sensor networks [2] etc. Unlike wired networks, where each node has a fixed number of neighbors, a multihop wireless network consists of (possibly mobile) nodes that can change its neighbor-set based on transmission power levels, and thus, in turn, cause changes in the network topology. A fundamental criterion for such networks to efficiently process and route data is the preservation of network connectivity, either on a continuous basis or intermittently among clusters of nodes. However, it is undesirable for nodes to transmit at their maximum power level to preserve connectivity. This is because firstly, it increases the possibility of interference with neighboring transmissions, and secondly, relaying through intermediate nodes via multiple hops may consume less energy than transmitting at higher power. Therefore, topology control techniques that minimize the transmission power play an important role in optimizing throughput and prolonging lifetime. In this paper, we address the problem of efficient topology control by adjusting transmission power levels of individual nodes in 3-D wireless sensor networks. This problem falls under the broader class of problems that guarantee a global network property by satisfying certain local constraints. In our case, this global property is network connectivity, and the local constraints, as we will see, are the existence of certain number of neighbors within each node's communication range.

Although the problem of efficient topology control has been addressed well in literature for the 2-D case, its extension to 3-D brings several challenges [5] that have not been adequately addressed in the past. For instance, there is a natural ordering of nodes in 2-D in terms of directions (angles), based on which many topology control algorithms e.g, CBTC (Cone-Based Topology Control) have been developed; however, in 3-D no such ordering is possible. We only have the notion of solid angles in 3-D, which does not lend itself to any particular ordering of nodes. Secondly, uniform random deployment of nodes requires very high average node degrees for a network to be connected in 3-D, compared to its 2-D counterpart. Therefore, reduction in time complexity of algorithms that are a function of average node degree would be very useful in 3-D.

In this study, we present two efficient, and distributed algorithms that guarantee network connectivity in 3-D with much lower time complexity as compared to existing approaches [3], [4], [6], [7]. Our first approach is based on orthographic projections in 2-D that is simple to implement and runs in $O(d \log d)$ time, where d represents the average node degree. This approach borrows the 2-D CBTC technique [4] and performs

This work has been funded in part through NSF grants number CNS-0435505, CNS-0347621, CNS-0627028, CCF-0430061, and CNS-0325875.

1-4244-1268-4/07/$25.00 ©2007 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE SECON 2007 proceedings.

very well in practice, although theoretically, it does not guarantee a connected network. Our second approach is based on a computational geometry construct, called spherical Delaunay triangulation [5], [11] that also runs in $O(d \log d)$ time but is always guaranteed to produce a connected network.

The rest of the paper is organized as follows. Section II presents related works on topology control techniques. In Section III, we describe preliminaries and our solution approach. Section IV discusses the multi-dimensional scaling technique in 3-D that we use as a primitive. We present our heuristic based approach based on 2-D orthographic projections in Section V, and the more rigorous spherical Delaunay triangulation based approach in section VI. Section VII presents detailed simulation results and we conclude the paper in Section VIII.

II. Related Works

There are several works in the literature on topology and power control techniques that aim to minimize resource consumption and increase network lifetime. However, in most cases, the existing algorithms are applicable only in 2-D networks and are not readily extensible in 3-D. Moreover, they are based on several assumptions, such as, the availability of location information of all the nodes, and/or directional information of the signal sources using possibly directional antennas. Most notable among these is the improved version of the CBTC algorithm [3] proposed by Li *et al.* which guarantees global network connectivity using only local information. In 2-D, the CBTC results says that if every node adjusts its power level, such that there exists at least one neighbor in every sector of angle $\theta = 5\pi/6$ around it, then network connectivity can be guaranteed so long as the communication graph formed by all the nodes transmitting at their maximum power level is connected. Henceforth, we will refer to this communication graph as the maximum power graph $G_{R^{max}}$ (MPG). The original version of the CBTC algorithm [4] proposed the θ value to be equal to $2\pi/3$. In the improved version, several optimizations have also been proposed, such as, the shrink-back operation on boundary nodes, removal of asymmetric links and redundant edges to further cut down the number of links. However, it should be noted that there is a tradeoff between cutting down the number of links and network performance. Fewer number of links implies lesser interference, but higher congestion and lower robustness as there are fewer redundant paths between pairs of nodes.

Bahramgiri *et al.* [7] extended the original CBTC result of $\theta \leq 2\pi/3$ constraint for k-connectivity and proved that if every node adjusts its power level, such that, it can reach at least one node in every $\theta = 2\pi/3k$ angle around it, then the graph will remain k-connected as long as the MPG was k-connected. They also proposed an extension to their algorithm in 3-D by using the notion of 3-D cones instead of angles. However, they assume that the directions of the signal sources are also ordered in 3-D, which is not valid because there is no notion of angles in 3-D. Moreover, the computational complexity of the proposed algorithm is very high, $O(d^3 \log d)$, compared to its 2-D CBTC counterpart that runs in $O(d \log d)$ time.

The algorithms proposed by D'souza et al. [6] also use the local geometric θ constraints to guarantee network connectivity in 2-D. The novelty of their approach is that they do not assume a binary disc communication model and connectivity of the MPG. If the boundary nodes satisfy certain connectivity criteria, their proposed algorithm can still guarantee network connectivity in the presence of arbitrary wireless foot prints. In particular, if the boundary nodes can communicate with neighboring nodes, and all the interior nodes satisfy $\theta \leq \pi$ constraint, then the network is at least one connected under arbitrary wireless communication model. However, if the condition on the boundary nodes is relaxed to $\theta \leq 3\pi/2$, then network connectivity can still be achieved with the interior nodes satisfying $\theta \leq \pi$ constraint under a weak-monotonic model of wireless footprints. In [9], Wattenhofer *et al.* described a practical topology control algorithm that is simple and local. Their technique does not assume location information, neither does it require the network to be a Euclidean graph.

Our work builds on the spherical Delaunay triangulation (SDT) based technique first introduced by Sameera *et al.* [5] for identifying largest empty cones. As noted in that paper, "this algorithm can be used as a primitive for extending several topology control algorithms that use directional information". We extend their approach to find the surface area of the largest spherical cap generated from triangulation and bound that value under a certain threshold, such that there is no empty 3-D cones of apex angle $2\pi/3$ around any node. Then it follows from the results in [7] that the network is guaranteed to be at least one connected. We check for additional constraints on the boundary nodes so that, unlike in CBTC, they do not end up transmitting at maximum power levels. We also propose an orthographic projection based heuristic that uses the CBTC technique and produces connected networks most of the times in practice. Both our algorithms have time complexities $O(d \log d)$. Finally, we do not assume directional information, instead we use multi-dimensional scaling in 3-D to get relative neighbor locations for each node.

III. Preliminaries and Solution Approach

In this Section, we describe preliminaries and our solution approach. Given a set of nodes $S = \{u_1, u_2, ..., u_N\}$ distributed in 3-D, we associate a three tuple $(x_i, y_i, z_i) \in \Re^3$, as the location of node u_i. We denote the spherical ball with center at u_i and radius R_i as $B(u_i, R_i)$. Given any three non-collinear points p_i, p_j, and p_k on the surface of a sphere, we define a spherical cap as the smaller (in terms of volume) portion of the sphere that is cut off by a plane passing through these three points. Note that, a plane can be uniquely determined by a set of three non-collinear points in 3-D. We denote such a spherical cap as $Cap(p_i, p_j, p_k)$. The height of the cap is denoted by h and the radius of its base is denoted by r (Figure 1).

Each node u_i is assumed to have a maximum power level P^{max}. In general, the transmission range of a node is a monotonic function of its power level, i.e, $R_i = f(P)$. Given a particular power level P, we denote the set of neighbors of node u_i as $N_i(P)$. Similar to prior works, we also assume the connectivity of the MPG $G_{R^{max}} = (V, E)$, where V is the set of nodes, and an edge exists between two nodes u_i and u_j if they are within their communication range. Our goal is to produce a subgraph $G = (V, \bar{E}) \subseteq G_{R^{max}}$, such that, (1) it is connected, (2) it consists of fewer number of edges, and (3) every node transmits at a power level that is locally minimum, i.e, even if one node transmits at a power level lower than what G requires, and others remain the same, then the network will cease to be connected.

Our approach in developing a topology control algorithm that works in 3-D with substantially lower computational complexity consists of two phases. In the first phase, we use the multi-dimensional scaling (MDS) technique to find out the relative locations of *all* possible neighbors for each node, i.e, the set of neighbors $N_i(P^{max})$, when node u_i transmits at its maximum power level. In the second phase, we propose two alternative strategies to solve the problem. The first one is heuristic based and uses the well known CBTC algorithm. The heuristic performs extremely well in practice, although theoretically, sometimes it can produce disconnected networks for degenerate cases, as we will see later. The basic idea is that we *reduce* the problem from 3-D into multiple similar problems on 2-D using orthographic projections, and then solve the 2-D problems using CBTC technique. The second alternative, which is the more rigorous one and always guarantees a connected network, is to solve the problem directly in 3-D by using the properties of SDT.

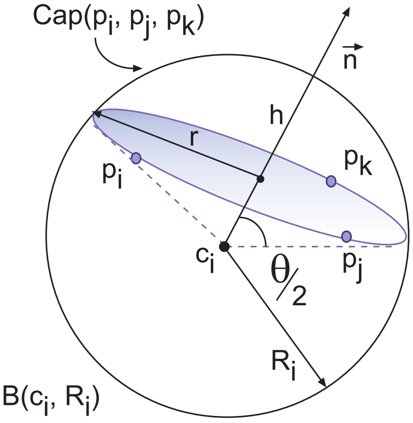


Fig. 1. Three non collinear points p_i, p_j, and p_k on the surface of a sphere uniquely determine a spherical cap. r is the radius of the base of the cap and h is the height of the cap. $\vec{n}$ denotes normal to the cap.

IV. Phase I: Multi-Dimensional Scaling in 3-D

Multi-dimensional scaling [14] is a statistical method that has been widely used to discover spatial structures and relationships among sets of objects from their observed similarity or dissimilarity data sets. The technique basically transforms a pairwise distance matrix among a set of objects into a set of coordinates, such that the pairwise Euclidean distances derived from these coordinates approximate the original distances as closely as possible. The distance matrix, however, cannot be analyzed directly using Eigen-decomposition because distance matrices are not positive semi-definite. But if it can be converted into an equivalent cross-product matrix then eigen-decomposition is possible, which gives a principal component analysis (PCA). MDS precisely does that. Each object is represented as a point in a multi-dimensional space, and the points are so arranged that their pairwise distances have the strongest possible relation to the similarities among the pairs of objects. That is, two similar objects are represented by two points that are closer to each other, and two dissimilar objects are represented by two points that are further apart. Finding out the appropriate dimension is also part of the problem in MDS. However, in our case, since we know that the space is 3-dimensional, we can get much better approximations of the relative location maps. Note that, the general MDS technique works in any dimension and even in non-Euclidean space.

MDS was applied for localization in 2-D [8] to calculate relative sensor locations based on their pairwise RSSI (Received Signal Strength Intensity) values. In our work, based on LQI (Link Quality Indicator) values, we use MDS in 3-D as a primitive for finding out relative location maps of all possible neighbors, $N_i(P^{max})$, for each node u_i. Note that, this is slightly different from the earlier approaches as a localization technique. Here, we are interested only in relative locations of each node's neighbors.

V. Phase II: 2-D Orthographic Projections

In the second phase of the algorithm, we simplify the original 3-D problem by reducing it into multiple similar problems on 2-D using orthographic projections, and then solve the 2-D problems borrowing techniques from CBTC. First, we state and prove the following lemma.

Lemma 1: Consider the projections of the locations of a node u_i and its set of neighbors $N_i(P)$ for some power level P ($P \leq P^{max}$) on each of the three orthogonal planes xy, yz, and zx, as illustrated in Figure 2. If there is an empty sector of angle θ around u's projection on *any* of the planes, then there exists an infinite number of empty 3-D cones of angle θ around u's location in the 3-D.

Proof: We show the proof by construction. In Figure 2 there exists an empty sector of angle θ around the projected location u_i' of node u_i on the xy plane for some communication range $R = f(P)$. Consider the two intersection points p' and q' of the circle and the sector. If we raise the plane of the triangle $\triangle u_i'p'q'$ vertically upwards and parallel to the xy plane, it will sweep a triangular shaped volume bounded by three vertical planes Π_1, Π_2, and Π_3. From solid geometry, we know that a plane in 3-D can be uniquely determined by two non-overlapping lines. Here, the three planes Π_1, Π_2, and Π_3 are uniquely determined by the three pairs of lines L_1, L_2; L_1, L_3; and L_2, L_3, respectively. By construction it is trivial to note that the 3-D cones of apex angle θ contained within the region formed by the intersection of this triangular shaped volume and the spherical ball $B(u_i, R)$ are all empty. ■

The lemma implies that if there exists an empty sector of angle $\theta = 2\pi/3$ on any of the three orthogonal projection planes, then there will exist an infinite number of empty 3-D cones with apex angle $2\pi/3$ around node u_i, which in turn will imply, by the results in [4], that the communication graph formed by the nodes in 3-D will not be connected if u_i chooses to transmit at this power level. Next, we describe the algorithm.

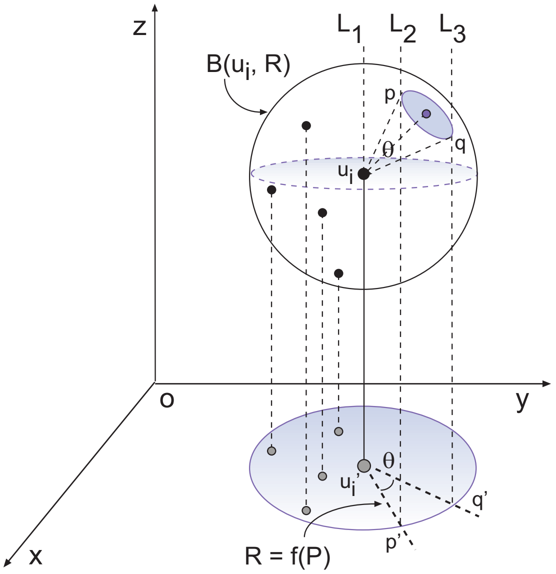


Fig. 2. An empty sector of angle θ around u_i's projected location on the xy plane implies the existence an infinite number of empty 3-D cones of apex angle θ around u_i's location.

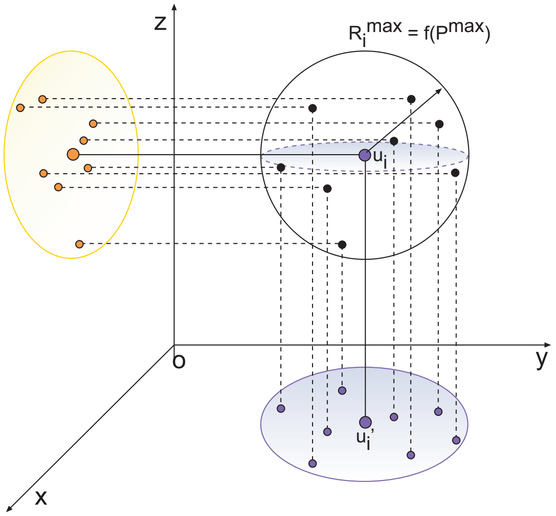


Fig. 3. Projected locations on xy and xz planes of node u_i and its neighbors $N_i(P^{max})$ when u_i transmits at maximum power P^{max}.

Each node u_i starts off by transmitting a "Hello" message at its minimum transmission power level. Neighboring nodes that hear the "Hello" message will acknowledge back with a "Reply" message. Node u_i then projects the locations of those neighbors from which it heard the "Reply" message on to the xy, yz, and zx planes, as illustrated in Figure 3. Then for each of the three planes u_i checks whether there is any empty sector of angle $\theta = 2\pi/3$ around it using the CBTC technique. If *all* the three planes satisfy the θ constraint, it stops and chooses the current power level. Otherwise, it increments its power to the next level, sends a "Hello" message and repeats the above process until there is no empty sector of angle $2\pi/3$ around it on *all* the three projection planes, or until the maximum power is reached. The minimum power that is required to guarantee the $\theta = 2\pi/3$ constraint on all the three planes is chosen as the transmission power for that node. As in CBTC, we assume the existence of the following functions: (1) $increment(P)$ that takes the current power level and increases it to the next level, (2) $dir_i(v)$ that takes the projected locations on a plane of a node's neighbors and sorts them with respect to some reference direction, and (3) $gap_\theta(D_i)$ that takes input as a set of directions and checks if there is an empty sector of angle θ around the projected location of u_i. Note that, we do not need special antennas to find out the directions of the neighbors because MDS in Phase I already yields the relative location maps of the neighbors. Formally, the steps are described in Figure 4.

```
$N_i(P) \leftarrow \phi$
$P \leftarrow P^{min}$
$D_i \leftarrow \phi$: directions of projected neighbors
while ($P \leq P^{max}$ and $gap_\theta(D_i)$) do
    Broadcast "Hello" message at power $P$ and gather
    "Reply" messages from neighbors;
    $N_i(P) \leftarrow N_i(P) \cup \{v \mid$ node $v$ replied$\}$;
    Project locations of $N_i(P)$ and itself on $xy$, $yz$,
    and $zx$ planes;
    for (each of the 3 orthographic planes) do
        $D_i \leftarrow D_i \cup \{dir_i(v)\}$
        if ($gap_\theta(D_i)$) then
            $P \leftarrow increment(P)$;
            break;
        end if
    end for
end while
```

Fig. 4. Second phase of algorithm based on orthographic projections and CBTC.

The novelty of the heuristic described above is that the algorithm runs in $O(d \log d)$ time and does not assume directional information. However, it should be noted that the network topologies thus generated with transmission power levels as dictated by the algorithm are not always guaranteed to be connected. This can be intuitively seen by the following argument. Consider a particular node u_i located at the origin and its set of neighbors that lies above the xy plane for a given power level, and project those neighbors on the three orthographic planes. Next, consider a particular 3-D cone of angle $\theta < \pi/2$ around node u_i contained within the first quadrant (i.e., positive x, y, z) and project the cone on the three planes as well. This will form three sectors of angle θ around u_i's projected locations on the three planes. Now, let there be a particular neighbor that lies just outside and above the surface of the 3-D cone at such a position, which when projected on the three planes, falls within the respective sectors formed by the cone's projection on two of the planes (say, xy and xz). Note that, unless a neighbor lies *inside* the cone, its projection will not fall inside all the three projected sectors. Now with little thought we can convince ourselves that there could be another node(s) that does not lie within the cone but falls within the projected cone's sector on the third plane (yz). Therefore, we observe that even though the projected sectors are not empty, that is, they satisfy the θ constraint on all the three planes, the 3-D cone can be empty. This implies that if we base our conclusion of network connectivity by satisfying θ constraint on the three planes, it might be incorrect at times for such degenerate cases. Here we restrained $\theta < \pi/2$ to illustrate one particular instance; however, the augument holds true for $\theta = 2\pi/3$ as well.

VI. Phase II: Spherical Delaunay Triangulation

The second approach in Phase II of our algorithm is based on the properties of spherical Delaunay triangulation. In computational geometry, Delaunay triangulation is the dual of Voronoi diagrams [12], which, for a set of N points tessellate the 2-D (3-D) region into a set of N convex polygons (polyhedra), such that any point lying within a polygon (polyhedron) is closest to the point $\in N$ that is inside the polygon (polyhedron). This is known as the nearest neighborhood property of Voronoi diagrams. Likewise, Delaunay triangulation follows the dual of the nearest neighborhood property, called the empty circle property as defined below.

Definition 1: Delaunay empty circle property: If N points lie on a plane in the general position, i.e., no four points are co-circular (affinely independent), then the circumcircle around each of the Delaunay triangles is empty, i.e., it does not contain any of the N points in its interior. The empty circle property generalizes in 3-D in the form of empty spheres for Delaunay tetrahedrization.

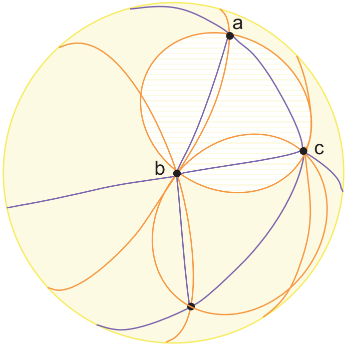


Fig. 5. Spherical Delaunay triangulation illustrating empty circle property: the spherical cap $Cap(a, b, c)$ is empty in its interior.

When the Delaunay triangulation is carried out on points that lie on the surface of a sphere, it produces spherical triangles, and the empty circle property still holds. That is, for any three points a, b, and c that form the vertices of a spherical triangle, the spherical cap $Cap(a, b, c)$ is empty. This is illustrated in Figure 5. We use this *empty spherical cap property* as a primitive in our algorithm.

Consider node u_i and its set of neighbors $N_i(P)$ for some power level P. We project the locations of those neighbors on the surface of a spherical ball centered at u_i and radius R, $(R = f(P))$. This construction basically means drawing radial lines connecting u_i and each of the neighbors until they intersect with the spherical surface. Then, if we construct a spherical Delaunay triangulation with the projected points, we can state the following lemma.

LEMMA 2: The 3-D cones that are formed with u_i as the apex and the spherical caps generated from Delaunay triangulation (or more rigorously, the base-circles of the spherical caps) as their bases are empty.

Proof: This can be easily seen from the construction in Figure 6. Consider a particular 3-D cone that has its base as the spherical cap formed by the vertices of a spherical triangle p, q, and r. Assume that the cone is not empty. This means that there is some point that lies inside the cone, whose projection, by construction, on the spherical surface will fall in the interior of $Cap(p, q, r)$. But this is a contradiction, because according to the Delaunay empty cap property this cap is empty. Hence, the cone is empty. ■

THEOREM 1: Let each node u_i construct spherical Delaunay triangulation of its projected neighbor locations on the spherical surface for some power level P_i. If the largest surface area Ω_i^{max} of the spherical cap for node u_i (except for boundary nodes, as discussed next) satisfy $\Omega_i^{max} \leq (2.72)R_i^2$, $(\forall i = 1, 2, ..., N)$, where $R_i = f(P_i)$, then the network topology formed by the nodes with transmission power levels P_i is guaranteed to be at least one connected.

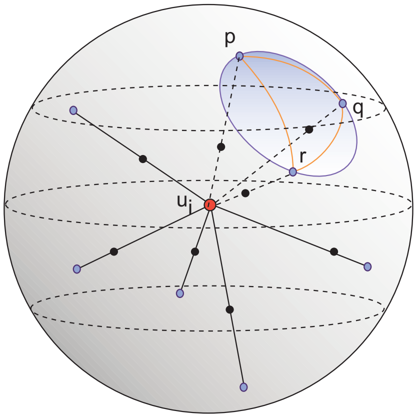


Fig. 6. The 3-D cone formed by the spherical cap $Cap(p, q, r)$ as the base and u_i as the apex is empty. Black dots show the actual locations of the neighbors, and blue dots show the projected locations on the surface of the spherical ball. $\triangle pqr$ is one of the spherical Delaunay triangles.

Proof: Let the 3-D cone formed with apex at node u_i and base as the spherical cap with the largest surface area Ω_i^{max} has an apex angle θ. From simple trigonometry, the surface area of the spherical cap is $2\pi rh$, where r is the radius of the base of the cap and h is the height of the cap, and are given by (Figure 1):

$$r = R_i \sin\frac{\theta}{2}, \qquad h = R_i\left(1 - \cos\frac{\theta}{2}\right). \qquad (1)$$

Therefore,

$$\Omega_i^{max} = 2\pi rh = 2\pi R_i^2 \sin\frac{\theta}{2}\left(1 - \cos\frac{\theta}{2}\right) \qquad (2)$$

Now it is easy to verify that if $\Omega_i^{max} \leq (2.72)R_i^2$, then $\theta \leq 2\pi/3$. This implies that if the surface area of the largest cap is not greater than $(2.72)R_i^2$, then there will be no empty 3-D cone of apex angle greater than $2\pi/3$ around u_i. Since this is true for all u_i's, $(i = 1, ..., N)$, it implies that there is no empty 3-D cone with apex angle greater than $2\pi/3$ around any of the nodes for power level P_i. Hence, from the result in [7], the network is at least one connected. ■

The implication of the above theorem is that if every node adjusts its power level to have enough neighbors, such that, none of the caps of the spherical Delaunay triangulation has a surface area greater than the threshold mentioned above, then the network will be at least one connected so long as the maximum power graph is connected.

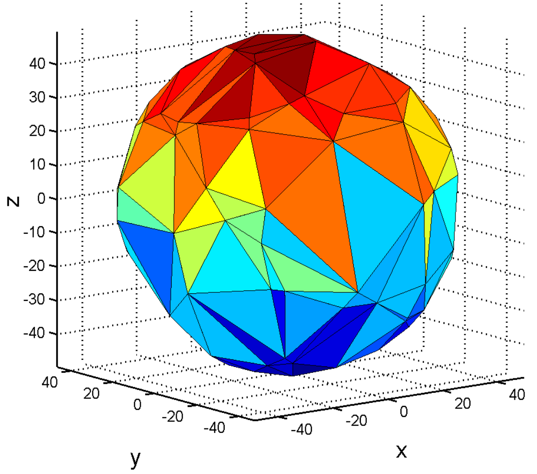


Fig. 7. Spherical Delaunay triangulation using the Quickhull algorithm of a set of 100 points randomly distributed on the surface of a sphere of radius 50.

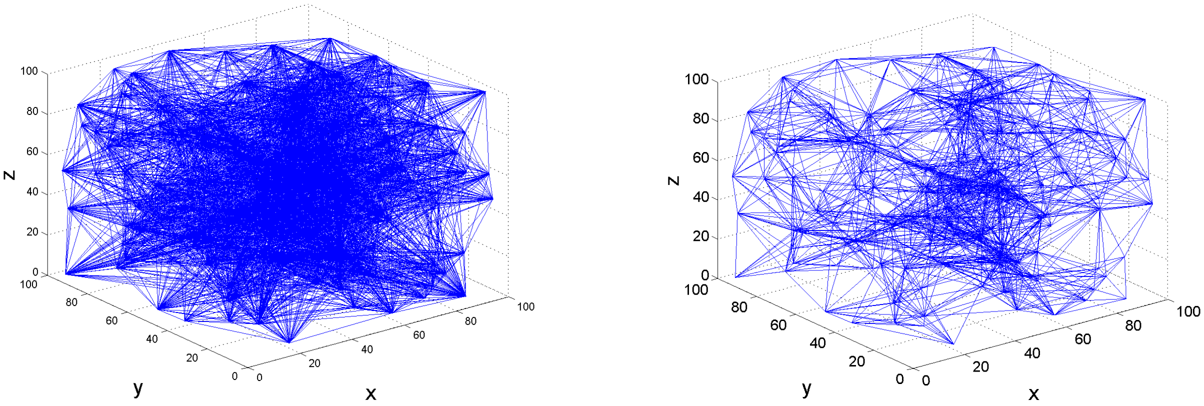
```
1:  $N_i(P) \leftarrow \phi$
2:  $P \leftarrow P^{min}$
3:  while $(P \leq P^{max})$ do
4:      Broadcast "Hello" message at power $P$ and gather "Reply" messages from neighbors;
5:      $N_i(P) \leftarrow N_i(P) \cup \{v \mid \text{node } v \text{ replied}\}$;
6:      Project the locations of $N_i(P)$ on the surface of the spherical ball $B(u_i, R)$;
7:      Construct SDT with the projected points;
8:      Find the largest spherical cap surface area, $\Omega_i^{max}$;
9:      if $(\Omega_i^{max} > (2.72)R^2)$ then
10:         $P \leftarrow increment(P)$;
11:     else
12:         break;
13:     end if
14: end while
```

Fig. 8. Second phase of algorithm based on spherical Delaunay triangulation.

There is a subtlety with boundary nodes while checking for the largest spherical cap. We define a boundary node as one that lies *outside* the 3-D convex hull formed with all the neighbors when it transmits at maximum power. A node can identify itself as a boundary node in Phase I of the algorithm after running MDS and constructing a 3-D convex hull with all the neighbors that lie within its communication range. Each node can perform this in $O(d \log d)$ time using the Quickhuhll algorithm [13], where d is its degree. Since a boundary node does not have neighbors in all directions around itself, its spherical Delaunay triangulation might form caps that are smaller than the threshold area but still have empty 3-D cones pointing outwards. In such cases, a boundary node calculates the difference in surface area of the spherical ball and the sum of the spherical triangles. If this difference is greater than the threshold then it further increases its power level. One advantage of this SDT based approach compared to the CBTC technique is that the boundary nodes do not end up with maximum power levels. Formally, the following steps shown in Figure 8 will be executed on each internal node u_i.

Constructing a spherical Delaunay triangulation in Step 7 of the algorithm is equivalent of finding the 3-D convex hull for the set of projected points on the sphere. This can be done in $O(d \log d)$ time using the Quickhull algorithm [10], [13], where d is number of neighbors (see Figure 7 for an illustration). Note that, the number of spherical caps thus generated is of $O(d)$. Hence, the time complexity of the above algorithm is $O(d \log d)$. This is a substantial improvement over the existing algorithms that run in $O(d^3 \log d)$ time. Since the average node degree in 3-D is very high compared to that in 2-D for a network to be connected with high probability under random deployment of nodes, an improvement of d^2 implies a much faster algorithm. For instance, when the number of nodes is $n = 1000$, the critical transmission radius for connectivity in 2-D, according to percolation theory, is 0.07, while in 3-D it is 0.02, resulting in an average node degree of 15 in 2-D and 34 in 3-D [5].

As we explained earlier, the orthographic projection based approach (Section V), although works very well in practice, does not theoretically guarantee a connected 3-D network even when the θ constraint is satisfied on all the three planes. The degenerate cases can be identified by combining the two approaches in the following way. Before projecting the neighbors on the spherical surface in Step 6 of Figure 8, each node first projects them on the xy, yz, and zx planes. Then it increments its power level until all the three planes satisfy $\theta = 2\pi/3$ constraint, i.e., until the function $gap_\theta(D_i)$ returns *false* for all the three planes. Let this power level be P_{proj}. Only after this, the neighbors are projected on the surface of a sphere of radius $R = f(P_{proj})$ and Steps 7, 8, 9 and so on are executed. If the algorithm terminates with the node choosing a power level that is smaller than its maximum power level, then this is a degenrate case. This is because had the node settled down with its maximum power level P^{max}, it would mean that either the largest spherical cap area is *greater* than or just *equal to* the threshold value of $\Omega_{max} = (2.72)R^2$. In the first case, the 3-D network



(a) Maximum power graph.

(b) Final connected network topology based on SDT.

Fig. 9. Network topology before and after the SDT algorithm, $n = 200$, $P^{max} = 40$. Note that, node degrees have drastically reduced.

will not be connected, whereas in the second case the network finally formed is, in fact, the original maximum power graph, which was assumed to be connected.

VII. Simulation Results

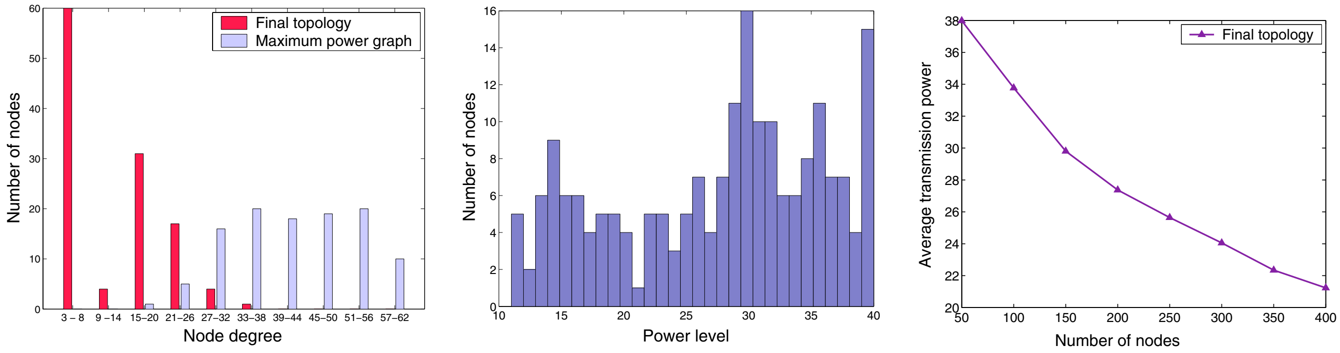
In order to understand the efficiency of our algorithms, we conducted simulations by generating random networks in 3-D of different sizes inside a cuboid of dimensions $100 \times 100 \times 100$. Before running the projection based or the spherical Delaunay triangulation based algorithm, we run MDS to get relative neighbor locations for each node at their maximum transmission power level.

For the SDT based algorithm, we generated random network topologies for $n = 200$ nodes under different maximum power levels. The performance of our algorithm is measured in terms of average node degree and average transmission power level. As an example, we show one specific instance of the initial network topology (the connected maximum power graph) when $P^{max} = 40$ in Figure 9(a). In Figure 9(b) we show the same network after nodes have settled down with minimal transmission power levels according to the SDT algorithm. For the same problem instance, we plotted the initial and final node degrees in Figure 10(a). Finally, in Figure 10(b) we show the assigned minimal power levels for all the nodes. Note that, node degrees have drastically reduced. We also observe that only 7.5% of the nodes transmit at their maximum power level, and more than 25% of the nodes transmit at less than half the maximum power level (i.e., below 20). Next, we describe the general trends.

We measured the dependencies between (1) average node degrees, and (2) average transmission power levels with different network sizes. In Figure 11(a), which shows the plot of average node degree with network sizes for $P^{max} = 40$, we observe that as the network size (density) increases, the average node degree increases sharply for maximum power graphs whereas increases very slowly for network topologies formed based on the SDT algorithm. Figure 10(c) shows the plot of average transmission power levels with increasing network density for topologies generated based on SDT. We observe that as the network gets denser the average transmission power decreases. This is expected because as the density increases, nodes can use lower power levels to form more number of multihop paths to guarantee network connectivity. Note that, in Figure 11(a) when the number of nodes increases to 400, the average node degree reaches ≈ 15, which is in accordance with the percolation theory of critical average node degree for network connectivity (average node degree = $\frac{4\pi}{3}(0.2)^3(400) \approx 13.4$).

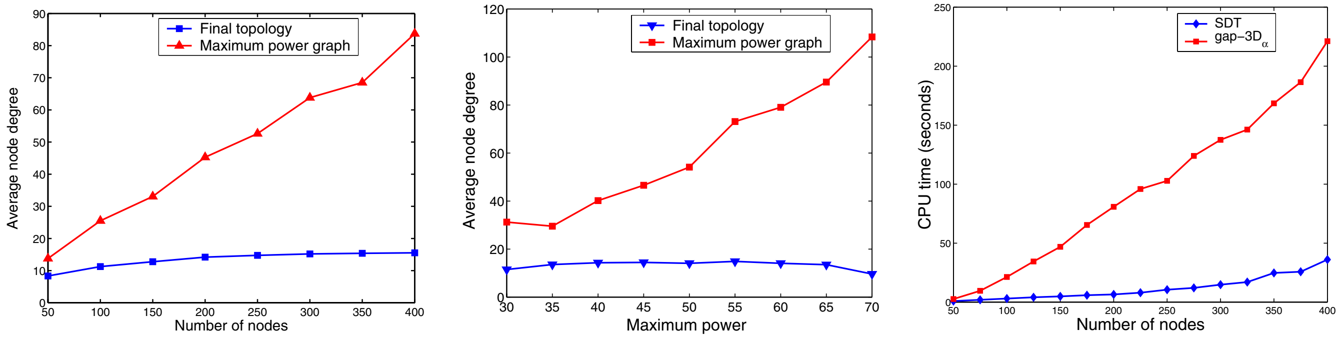
In Figure 11(b) we plot average node degrees for random topologies of 200 nodes with increasing maximum power levels. We observe that average node degrees almost remain flat in the final topology generated based on the SDT algorithm as compared to that of the maximum power graph. This is because of the fact that as the maximum power varies, only the *few* boundary nodes get affected, as they are more likely to transmit at or close to the maximum power compared to the internal nodes in order to guarantee a minimal number of neighbors. The average transmission power level chosen by the nodes in the final topology is observed to vary from 26 to 21 as the maximum power increased from 30 to 70 (not plotted). Thus, the effectiveness of the algorithm is predominant at higher maximum power levels.

To compare the SDT based algorithm with the one



(a) (b) (c)

Fig. 10. (a) Node degrees of the maximum power graph and that of the final topology for $n = 200$, $P^{max} = 40$. (b) Final assigned minimal transmission power levels of nodes, for $n = 200$, $P^{max} = 40$. (c) Dependency of average transmission power with network size.



(a) (b) (c)

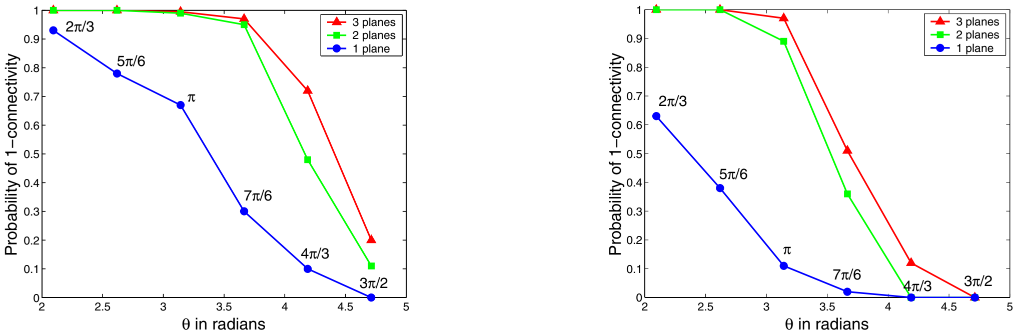
Fig. 11. (a) Dependency of average node degree with network size. (b) Dependency of average node degree with maximum power for $n = 200$; node degree remains almost flat in the final topology based on the SDT algorithm. (c) CPU execution time of the SDT based algorithm and the one [7] based on the procedure $gap - 3D_\alpha$ for different random topologies with $P^{max} = 40$.

described in [7] based on the procedure $gap - 3D_\alpha$, both of which check for empty 3-D cones, we simulated the algorithms and measured CPU running times using the Matlab *profile* tool. Our measurements show that the respective procedures $sdtcheck()$ (part of our implementation) and $gap - 3D_\alpha()$ take most percentages $(80 - 90\%)$ of total execution times. In Figure 11(c) we show the plot of CPU execution times of these two functions on random network topologies of different sizes with maximum power set to 40. We observed that SDT performs much faster than the other one and with increasing network size (or equivalently with increasing average node degree) the difference between execution times becomes more predominant.

Finally, to measure the practicality of the orthographic projection based algorithm we generated two random networks with $n = 50, 100$ nodes and $P^{max} = 100$. Then we ran the projection based algorithm and generated different network topologies where the θ constraint is satisfied on one, two or all the three planes. For each of these cases we checked the connectivity of the original 3-D graph. In Figure 12 we plot the probability of connectivity with θ. Each point in the plot is an average of 200 runs. We observe that when the θ constraint is satisfied on only one plane the original 3-D graph is actually never connected. However, when each node guaranteed $\theta = 2\pi/3$ on all the three planes, the 3-D topology generated with that power level is found to be connected at *all* times. Moreover, even when the constraint is satisfied on only two planes, we found that the 3-D graph is disconnected only a very small number $(< 1\%)$ of times. This shows that the heuristic based approach works very well in practice.

VIII. Conclusions

We presented two efficient distributed topology control algorithms for 3-D multihop wireless networks. Our first approach was based on the idea of 2-D orthographic projections, by which we reduced and simplified the original 3-D problem into a set of three similar prob-



(a) For $n = 50$ nodes

(b) For $n = 100$ nodes

Fig. 12. Probability of network connectivity as the θ constraint is satisfied on 1, 2, or all 3 orthogonal planes.

lems on 2-D, and used existing techniques from CBTC to show that network connectivity can be guaranteed almost at all times. In the second approach we used the properties of Delaunay triangulation on the surface of a sphere to determine the existence of empty 3-D cones in an efficient way and showed that the network topology generated based on this is always connected. Both the algorithms are computationally very efficient and scale as $O(d \log d)$ in time, where d represents average node degree. The d^2 improvement in running time implies a *much faster* algorithm especially in 3-D, as d is typically very high in 3-D compared to that in 2-D, and this was verified by measuring CPU execution times on topologies with increasing node degree. To substantiate our claims we conducted simulations on network topologies in 3-D for both the SDT based algorithm and the heuristic based approach. Doing a probabilistic analysis of network connectivity for the orthographic projection based approach and implementation of the SDT algorithm on real motes are part of our future work.

References

[1] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin, “A wireless sensor network For structural monitoring”, In *Proceedings of the second International Conference on Embedded Networked Sensor Systems (SenSys ’04)*, Baltimore, MD, USA, pp. 13-24, November 2004.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater Acoustic Sensor Networks: Research Challenges”, *Elsevier’s Journal of Ad Hoc Networks*, Vol. 3, Issue 3, pp. 257-279, March 2005.

[3] Li Li, Joseph Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer, “A cone-Based distributed Topology-Control Algorithm for Wireless Multi-Hop Networks”, *IEEE/ACM Transactions on Networking*, Vol. 13, No. 1, pp. 147-159, February 2005.

[4] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang, “Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks”, In *Proceedings of the twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom ’01)*, Vol. 3, pp. 1388-1397, Anchorage, AK, USA, April 2001.

[5] Sameera Poduri, Sundeep Pattem, Bhaskar Krishnamachari, and Gaurav Sukhatme, “Sensor Network Configuration and the Curse of Dimensionality”, In *Proceedings of the third IEEE Workshop on Embedded Networked Sensors (EmNets ’06)*, Cambridge, MA, USA, May 2006.

[6] Raissa d’Souza, David Galvin, Cristopher Moore, and Dana Randall, “Global Connectivity from Local Geometric Constraints for Sensor Networks with Various Wireless Footprints”, In *Proceedings of the fifth International Conference on Information Processing in Sensor Networks (IPSN ’06)*, pp. 19-26, Nashville, TN, USA, April 2006.

[7] Mohsen Bahramgiri, Mohammadtaghi Hajiaghayi, and Vahab Mirrokni, “Fault-Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks”, *Wireless Networks*, Vol. 12, Issue. 2, pp. 179-188, March 2006.

[8] Xiang Ji, Hongyuan Zha, “Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling”, In *Proceedings of the twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom ’04)*, Vol. 4, pp. 2652-2661, March 2004.

[9] R. Wattenhofer and A. Zollinger, “XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks”, In *Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS ’04)*, April 2004.

[10] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa, “The Quickhull Algorithm for Convex Hulls”, In *ACM Transactions on Mathematical Software (TOMS ’96)*, Vol. 22, Number 4, pp. 469-483, December 1996.

[11] Robert J. Renka, “Algorithm 772: STRIPACK: Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere”, In *ACM Transactions on Mathematical Software (TOMS ’97)*, Vol. 23, Issue 3, pp. 416-434, September 1997.

[12] Franz Aurenhammer, “Voronoi diagrams - A Survey of a Fundamental Geometric Data Structure”, In *ACM Computing Surveys (CSUR ’91)*, Vol. 23, Issue 3, pp. 345-405, September 1991.

[13] “Qhull - A General Dimension Code for Computing Convex Hulls, Delaunay triangulations, and Voronoi Diagrams”, *http://www.qhull.org*.

[14] Herve Adbi, “Metric Multidimensional Scaling (MDS): Analyzing Distance Matrices”, In *N.J. Salkind (Ed.): Encyclopedia of Measurement and Statistics*, Thousand Oaks (CA): Sage.