Fault-tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks ⋆

Mohsen Bahramgiri[1], MohammadTaghi Hajiaghayi[2], Vahab S. Mirrokni[2]

[1] Department of Mathematics, Massachusetts Institute of Technology
[2] Laboratory for Computer Science, Massachusetts Institute of Technology

Abstract. We can control the topology of a multi-hop wireless network by varying the transmission power at each node. The life-time of such networks depends on battery power at each node. This paper presents a distributed fault-tolerant topology control algorithm for minimum energy consumption in these networks. More precisely, we present algorithms which preserve the connectivity of a network upon failing of, at most, k nodes (k is constant) and simultaneously minimize the transmission power at each node to some extent. In addition, we present simulations to support the effectiveness of our algorithm. We also demonstrate some optimizations to further minimize the power at each node. Finally, we show how our algorithms can be extended to 3-dimensions.

1 Introduction

Multi-hop wireless networks, in which communication between two nodes can go through multiple nodes, can be deployed in various civil and military applications. Unlike wired networks, in these networks, each node can move and thus change the topology of the network. In this case, we need to adjust the transmission power to keep some properties of the network such as connectivity. The lifetime of a wireless network, which depending on battery power, usually is restricted because of limited capacity and resources. Thus a main goal of topology control is to increase the longevity of such networks which can be obtained by designing power-efficient algorithms. Indeed, minimizing energy consumption in topology control is a key factor in the optimal usage of wireless sensor networks [1]. We also note that because of limited capacity, we need to have as few as possible facilities such as GPS.

One property of the network that has been considered by Li et al.[15, 10] is connectivity. They assume that nodes do not have any kind of GPS and their algorithm works using only directional information. They demonstrate a simple distributed algorithm in which each node uses only local decisions about its transmission power to guarantee the global connectivity of the network. More precisely, using only directional information, each node increases its transmission power until it detects a neighbor in all directions. Using this algorithm, the authors simultaneously reduce both transmission power and traffic interference. The algorithm tends to minimize the power consumption in each node, but there are some issues that may make the network unreliable. The first one is that the algorithm makes the network very sparse and thus any failure in the network might fail a routing process. Another issue is that for some nodes in the network, we might have a lot of congestion since those nodes are the only ones on the paths from some nodes to others in the network. Thus, the algorithm might result in hot spots and congestion, which in turn might drain battery power and lead to a network partition, as pointed out by Li et al. [10]. These motivations lead us to search for more powerful properties in the network by which we can tolerate failures and avoid network partition. In this paper, we consider k-connectivity of networks in which we satisfy the following two properties simultaneously: First for each $p < k$, failures of (or eliminating) p nodes in the graph does not disconnect it. Second, there are k node-disjoint paths between any two nodes in the network. One can observe that the former property solves the first issue and the latter property solves the second issue.

Another assumption made by Li et al, [10] is that our set of nodes is deployed in a 2-dimensional area. In this paper, we also consider the 3-dimensional model. In fact, suppose that our nodes are some wireless sensors in a multi-floor building. In this network, each node independently explores its surrounding region and establishes connections with other neighbors that are within its transmission and reception range; i.e., they are in the sphere of some radius r centered at the node. One can observe that if the number of floors is more than two or three, then we cannot model this network by the aforementioned 2-dimensional model and thus we need a 3-dimensional model.

⋆ Emails: m_bahram@mit.edu and {hajiagha,mirrokni}@theory.lcs.mit.edu

There are some other results on topology control and network design for increasing network longevity. Hu [6] presents a topology control based on Delaunay triangulation. He uses some heuristics to choose some links making a regular and uniform graph; however, he does not use the adaptive transmission power control. Ramanathan and Rosales-Hain [11] consider the connectivity and biconnectivity (or 2-connectivity), using a centralized spanning tree. However their work is based on some heuristics and unfortunately there is no guarantee of connectivity in all cases. Rodoplu and Meng [12] and Li and Halpern [15] present a distributed topology control which preserves connectivity. Li et al. [10] present a better description of their previous algorithm in which the genesis of our paper lies. Furthermore, these problem s implicitly have been considered in other more graph-theoretic papers [3, 7, 14]. Other approaches also have been presented in the field of packet radio networks, sensor networks and wireless ad-hoc networks for power minimization and network longevity. The reader is referred to the papers due to Takagi and Kleinrock [13], Hou and Li [5] and Henizelman et al. [4] for further information.

The rest of this paper is organized as follows. First, Section 2 introduces the terminology used throughout the paper, and formally define k-connectivity and our model in a plane and in 3-dimensional space. We introduce our cone-based topology control (CBTC) algorithm in Section 3. In Section 4, we present the bounds on the angles to preserve k-connectivity in the CBTC algorithm. Section 5 is devoted to the generalization of k-connectivity algorithms from 2-dimensions to 3-dimensions. We describe very briefly how we can handle reconfiguration due to mobility in Section 6. Finally, in Section 7, we conclude with a list of potential extensions for future work.

2 The Model

Our model is very similar to the model introduced by Li et al. [10]. We assume our sensor wireless network consists of set V of n *nodes* (or *vertices*) located in plane (space). Each node v is denoted by its coordinated $(x(v), y(v))$ $((x(v), y(v), z(v)))$ in 2-dimensions (3-dimensions). Each node v has a power supply function $p(d)$ where $p(d)$ is the minimum power needed to communicate with a node u of distance d away from v. We suppose that the maximum power for all nodes is equal to P and this power provides enough supply to communicate within distance R, that is $p(R) = P$. Since in practice function p depends on the nth power $(n \geq 2)$ of distance d, sending a message through a series of intermediate nodes might take less power than sending it directly. If each node transmits with power P, then we have an induced graph $G_R = (V, E)$ such that $E = \{(u, v)|d(u, v) \leq R\}$ where $d(u, v)$ is the Euclidean distance between u and v in a plane (space). Our antennas in the model are omni-directional ones and hence a node can *broadcast* a message to all nodes within some distance r with power $p \leq P$.

Here we suppose the radio communication unit is able to determine the direction of the sender when it receives a message. As mentioned in the introduction, nodes have no GPS. The reader is referred to Krizman et al. [8] for further information on estimating direction without position information.

Our primitives are the same as primitives mentioned by Li et al. [10]. More precisely, we have *send(u, p, m, v)* by which a node u sends message m with power p to v; *recv(u, m, v)* used by u to receive message m from v; and finally *bcast(u, p, m)* by which a node u broadcasts message m to all nodes v for which $p(d(u, v)) \leq p$. In addition, we assume that if a node u can reach node v with power p then node v can also send a message to node u with any power $p' \geq p$. If a node u tags the message with sending power p, node v can figure out how much power was used to communicate with node v but cannot deduce the distance of u.

We assume our model is an asynchronous setting, and the communication channels are reliable. Nodes can be mobile, i.e., nodes can change their positions, new nodes may be added to the network or some nodes may even die because of the lack of energy. Our goal is to preserve a global property P in the network, e.g., property P is connectivity in the paper due to Li et al.[10] or k-connectivity, defined below, in this paper. Since our topology may change frequently, the reconfiguration algorithm, described later, may not ever catch up with the changes, and thus in some time, we may lose property P. However, as we will show, we guarantee that our topology stabilizes after some time depending on the speed of processes and communication links.

The generalization of connectivity, namely *k-connectivity*, is an important fault-tolerant property of graphs.

Definition 1. *A* separating set *or* vertex cut *of a graph $G = (V, E)$ is a set $S \subseteq V(G)$ such that $G - S$ has more than one component. A graph G is k*-connected *if every vertex cut has at least k vertices. The connectivity of G, denoted by $\kappa(G)$, is the maximum k such that G is k-connected.*

The k-connectivity property can be considered from another viewpoint. Two paths are *internally-disjoint* if neither contains a non-endpoint vertex of the other. It is well known that a graph is k-connected if and only if for each pair x and y of vertices, there exist k pairwise internally-disjoint paths whose endpoints are x and y (see Menger's Theorem [16]). The reader is referred to the book by West on graph theory [16] for further information about k-connectivity. Preserving k-connectivity using local decisions is one of our major goals in this paper, which guarantees a network to be fault-tolerant (see the introduction). In the rest of this paper, we assume that graph G_R itself is k-connected and we try to keep the property and simultaneously minimize the power at each node locally.

3 The cone-based topology control algorithm

Our algorithm for topology control is very similar to the algorithm used by Li et al. [15, 10]. Here we briefly present the algorithm; and the reader is referred to the original papers for more discussion. The algorithm called *cone-based topology control (CBTC)* is as follows. A **Hello** message is originally broadcast from a node u using minimum power p_0. Each message also contains the power used to broadcast. The power is then increased at each step using some function *Increase* (see the details in [9, 10]). Upon receiving such a message from a node u, node v replies with an **Ack** message. Node v encloses both the original power that was used by u to send the message to v (which it has received from the **Hello** message) and the power that it uses to send the message. Upon receiving the **Ack** message from v, node u adds v to its set N_u of neighbors, tags the message with the power with which it originally sent the **Hello** message to v and finally adds v's direction $dir_u(v)$ to its set D_u of directions. Here $dir_u(v)$, which is measured as an angle relative to some fixed angle, can be obtained by our earlier assumptions. Then using procedure $gap_\alpha(D_u)$, we test whether there exists a gap greater than α between the successive directions in D_u (directions are sorted according to their angles).

CBTC(α)
$N_u = \emptyset; D_u = \emptyset; p_u = p_0;$
while ($p_u < P$ and $gap_\alpha(D_u)$) do
    p_u = Increase(p_u)
    bcast(u, p_u, **Hello**) and gather acknowledgments.
    $N_u = N_u + \{v : v \text{ discovered}\}$
    $D_u = D_u + \{dir_u(v) : v \text{ discovered}\}$

We define set $N_\alpha = \{(u, v) \in V \times V : v \in N_\alpha(u)\}$ where $N_\alpha(u)$ is the final set of discovered neighbors computed by a node u at the end of the running of CBTC(α). Also, let $p_{u,\alpha}$ be the corresponding final power. First we note that the N_α relation is not symmetric. Now we consider graph $G_\alpha = (V, E_\alpha)$, where V consists of all nodes and $E_\alpha = \{\{u, v\} | (u, v) \in N_\alpha, (v, u) \in N_\alpha\}$. In other words, in our new graph, each node only talks to its neighbors in E_α with power at most $p_{u,\alpha}$ (the new graph can be constructed by some message passing between each vertex u and vertices in $N_\alpha(u)$). Li et al. [15, 10] have shown that for $\alpha \leq \frac{2\pi}{3}$, G_α is connected if and only if G is connected and the theorem does not work necessarily for $\alpha > \frac{2\pi}{3}$. In the next section, we generalize this property to k-connectivity.

For more detailed timing issues, we refer the reader to [15, 10]. Also the details of implementing and simulating of this model can be found in [9]. In section 6, you can see some other ideas for dealing with mobility. In the rest of this paper, we assume we have this model with the aforementioned properties and prove new properties for the output of this model with different angles α.

4 Bounds on the angles for preserving k-connectivity

The **CBTC(α)** algorithm first constructs a directed graph. Then, it eliminates one-directional edges from this graph and keeps bidirectional edges. The following definition formalizes this process:

Definition 2. *Let G be an undirected graph. Let D_α be the directed subgraph of G that is the output of* **CBTC(α)** *algorithm i.e., each vertex increases its power (edge length) until it reaches its maximum power or the maximum angle between two consecutive neighbors of G is at most α. Let G_α be the undirected subgraph of G attained by keeping the bidirectional edges of D_α and removing other edges.*

We know the following lemma from [10]:

Lemma 1. *If graph G is connected, then $G_{\frac{2\pi}{3}}$ is connected.*

Now, we have the following result for preserving $k-$connectivity.

Theorem 1. *If a graph G is k-connected then $G_{\frac{2\pi}{3k}}$ is also k-connected. It means that if the* **CBTC(α)** *is applied with $\alpha = \frac{2\pi}{3k}$ for a k-connected graph, then the resulting graph $G_{\frac{2\pi}{3k}}$ is also k-connected.*

Proof. Suppose we run the algorithm with $\alpha = \frac{2\pi}{3k}$. After execution of the algorithm, we want to prove that the graph G_α is k-connected. We prove it by contradiction. Suppose that there are $k-1$ nodes $\{v_1, v_2, \dots, v_{k-1}\}$ whose removal makes G_α a disconnected graph, called graph G_1. Thus G_1 is a disconnected graph resulting from removing $k-1$ vertices from G_α.

Since G is k-connected, the graph G' resulting from removing $\{v_1, v_2, \dots, v_{k-1}\}$ from G is also connected. From Lemma 1, $G'_{\frac{2\pi}{3}}$ is also connected. Now we prove that $G'_{\frac{2\pi}{3}}$ is a subgraph of G_1, and it contradicts non-connectivity of G_1. Suppose that there exists an edge uv in $G'_{\frac{2\pi}{3}}$ which is not in G_1. Edge uv is an edge in $G'_{\frac{2\pi}{3}}$, thus the distance of u and v is less than or equal to R. Furthermore, the maximum required power of either u or v, say u, in $G_{\frac{2\pi}{3k}}$ is less than the distance of uv. Edge uv is not an edge of G_1, thus it is not in $G_{\frac{2\pi}{3k}}$ as well. Therefore, there exist some vertices closer than v to u for which the maximum angle between two consecutive ones is at most $\frac{2\pi}{3k}$. After removing $k-1$ vertices, there exist some vertices closer than v to u for which the maximum angle between two consecutive ones is at most $\frac{2\pi}{3}$. Hence, these vertices are also in $G'_{\frac{2\pi}{3}}$ and therefore, the power of u is less than uv in $G'_{\frac{2\pi}{3}}$. It contradicts the fact that uv is an edge of $G'_{\frac{2\pi}{3}}$, because we just put bidirectional edges in $G'_{\frac{2\pi}{3}}$. □

For the other side, i.e., the maximum α for preserving k-connectivity, we need the following definition and theorem.

Definition 3. *A k-connected* Harary graph *$H_{k,n}$, for $k = 2r < n$, can be constructed by placing n vertices in circular order and making each vertex adjacent to the nearest r vertices in each direction around the circle.*

$H_{4,8}$ is depicted in Figure 1(aa).

Theorem 2. *(Harary(1992)[16]) $\kappa(H_{k,n}) = k$, i.e., $H_{k,n}$ is k-connected.*

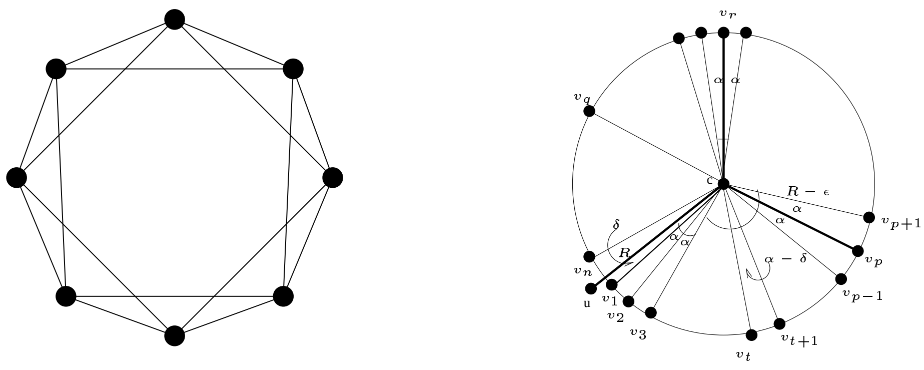
Remark 1. For $k = 2r$, graph $H_{n,k}$ is k-regular. It means that after deleting an edge from G, the degree of each of its adjacent vertices is $k-1$. Thus, the new graph is $k-1$-connected and not k-connected.

Using Harary graphs, we construct examples to show an upper bound for the angle to preserve k-connectivity.

Lemma 2. *If graph G is k-connected, then graph G', resulting from G by adding a vertex v to G and adding edges between v and all other vertices, is $k+1$-connected.*

Proof. The proof is by contradiction. If we remove a subset of k vertices of G' and it becomes disconnected, then this subset, say S, contains the node v, because otherwise v is adjacent to the rest of the graph. Thus the graph with v is connected. Therefore, this subset of size k contains v, and in graph G, if we delete subset $S - \{v\}$, it becomes disconnected. Now, $|S - \{v\}| = k - 1$ contradicts the k-connectivity of G. □

Theorem 3. *For odd $k = 2s+1$, there exists a k-connected graph G on which if we run* **CBTC(α)** *algorithm with $\alpha > \frac{2\pi}{3(k-1)}$ the resulting graph is not $k-$connected. In other words, it is necessary to have $\alpha \le \frac{2\pi}{3(k-1)}$ to preserve k-connectivity in* **CBTC(α)** *algorithm.*



(aa) **(bb)**

Fig. 1. (aa) The graph $H_{4,8}$. (bb) The counterexample for $\alpha = \frac{2\pi}{3(k-1)} + \delta$

Proof. We construct a $2s+1$-connected network on which if we use the algorithm with $\alpha = \frac{2\pi}{3(k-1)} + \delta$ for $\delta > 0$, the resulting graph is not $2s+1$-connected. This graph is shown in Figure 1(bb). There is a node c in the center of the circle. There is also a node u in the distance R of c and the other nodes are on the circle of radius $R-\epsilon$; each of them has the angle $a = \alpha = \frac{2\pi}{3(k-1)}$ from each of its neighbors except v_t, v_{t+1}, u and v_1. Points v_q, v_r, u, v_p are placed around the circle in such a way that $\widehat{v_p c v_r} = \widehat{u c v_r} = \widehat{v_p c u} = \frac{2\pi}{3}$ and $\widehat{v_q c v_r} = \frac{\pi}{3}$. Here, $\widehat{u c v_1} = \delta$, $\widehat{v_t c v_{t+1}} = a - \delta$ and for all other i's, $\widehat{v_i c v_{i+1}} = a$. Also we have $\widehat{u c v_t} = \frac{\pi}{3} + \delta$, therefore uv_t can be the largest edge in the triangle cuv_t i.e., $uv_t > R$. In the first graph, there is an edge between each two nodes of distance at most R. This graph without node c is $H_{2s,n}$, since each node is connected to at least s other nodes in each side. Thus the first graph without node c is $2s$-connected. Lemma 2 shows that after adding node c to the graph the resulting graph is $2s+1$-connected. Notice that $uv_t > R$ implies that $uv_t \notin E(G)$, hence $\deg(u) = 2s+1$.

Now, by running the algorithm with $\alpha = \frac{2\pi}{3(k-1)} + \delta$, when the center point reaches to power $R-\epsilon$, the largest angle between two consecutive vertices around c will be $\frac{2\pi}{3(k-1)} + \delta$, and it will not increase its power. Thus, in the resulting graph from the algorithm, we do not have the edge cu. Now, the degree of u in the new graph is $2s$, and deleting all neighbors of u will make this graph disconnected. Thus, the new graph is not $2s+1$-connected. Therefore, angle $\frac{2\pi}{3(k-1)} + \delta$, for $\delta > 0$, is not enough for preserving k-connectivity for odd k. □

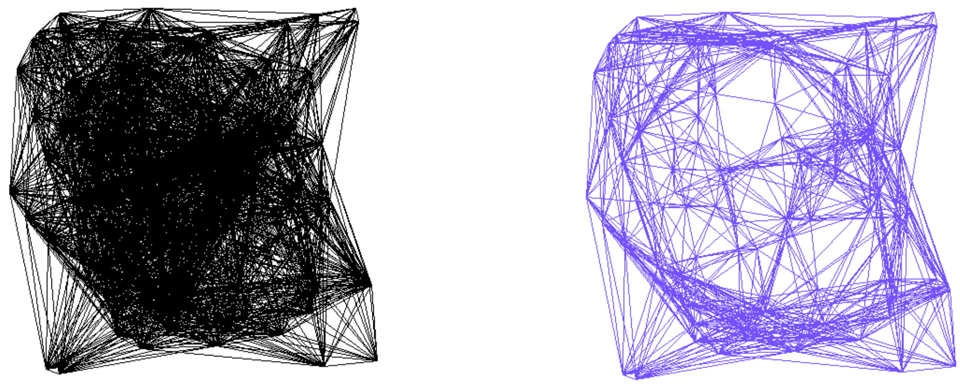
Theorem 4. *For even $k = 2s$, there exists a k-connected graph G on which if we run* **CBTC(α)** *algorithm with $\alpha > \frac{2\pi}{3k}$ the resulting graph is not $k-$connected. In other words, it is necessary (and sufficient) to have $\alpha \le \frac{2\pi}{3k}$ to preserve k-connectivity in* **CBTC(α)** *algorithm.*

Proof. We construct this network very similar to the network for odd k. This graph differs from the previous one only for the place of node v_q. Its place is slightly changed such that $\widehat{u c v_q} = \frac{\pi}{3} + \frac{\delta}{2}$. Thus, the edge between u and v_q is eliminated in the first graph, and in this graph $\deg(u) = 2s$. One can observe that by choosing appropriate δ and ϵ it is possible to just delete the edge uv_q and no other edges. Therefore, the first graph in Figure 1(bb) with this change will be a $2s$-connected graph, and if we run the algorithm with the same $\alpha = \frac{2\pi}{3k} + \delta$, then the resulting graph will have similar properties that were mentioned in the proof of Theorem 3. In other words, the edge between u and c will not be in the resulting graph. The new graph is not $2s$-connected, because it has a node of degree $2s-1$. Hence, this example shows that the angle $\alpha = \frac{2\pi}{3k} + \delta$ for any $\delta > 0$ is not sufficient for preserving k connectivity, where k is even. □

So far, we have shown that for odd k, $\frac{2\pi}{3k} \le \alpha \le \frac{2\pi}{3(k-1)}$ and for even k, $\alpha = \frac{2\pi}{3k}$ is sufficient and necessary for preserving k-connectivity. For large k's the difference between lower bound and upper bound is small. For example, for $k > 10^o$, this difference is at most 1^o.

4.1 Experimental Results

In order to understand the effectiveness of our algorithm, we generated random networks on which we see its effect. You can see an example in Figures 2. In that example, there are 100 nodes randomly placed in a 400×400 rectangular grid. Each node has a maximum transmission radius of 200.



(aa) **(bb)**

Fig. 2. (aa) First Network. (bb) Network after $\mathbf{CBTC}(\frac{2\pi}{6})$

In table 4.1, we also list the results for $k = 1, 2, 3$ in terms of the average node degree and the average radius. These graphs have 200 nodes randomly placed in a grid of 400×400 with maximum power 260. The effectiveness of this algorithm can be easily seen from the average radius of new networks compared with the first network.

	Max Power	$\alpha = \frac{2\pi}{3}(k=1)$	$\alpha = \frac{2\pi}{6}(k=2)$	$\alpha = \frac{2\pi}{9}(k=3)$
Average Degree	67.400	4.055	14.600	21.925
Average Radius	260	92.510	158.388	184.025

Table 1. Average degree and radius of the cone-based topology control algorithm for 1-, 2-, and 3-connectivity.

4.2 An optimization

Similar to the *shrink-back* operation in [1] for preserving connectivity, we can design an optimization operation to the basic algorithm to further reduce the maximum power at each node while still preserving k-connectivity. Here the same shrink-back optimization in [1] does not work for preserving k-connectivity; however it can be modified as follows. Given a set dir of directions (angles), define $\text{cover}_k(dir) = \{\theta |$ for at least k members $\theta' \in dir, |\theta - \theta'| \ mod \ 2\pi \leq \frac{\pi}{3}\}$. We modify $\mathbf{CBTC}(\alpha)$ such that at each iteration, a node in N_u is tagged with the power used the first time it was discovered. Suppose that the power levels used by node u during the algorithm are $p_1, \ldots, p_q$. If u is a boundary node, p_q is the maximum power P. A boundary node successively removes nodes tagged with power p_q, then p_{q-1}, and so on, as long as their removal does not change the coverage. In other words, let dir_i, $i = 1, \ldots, q$, be the set of directions found with all power levels p_i or less, then the minimum i such that $cover_k(dir_i) = cover_k(dir_q)$ is found. Let $N^s_{\alpha,k}(u)$ consist of all the nodes in $N_{\alpha,k}(u)$ tagged with power p_i or less. Suppose $N^s_{\alpha,k} = \{\{u,v\} | v \in N^s_{\alpha,k}(u)\}$. Also, similar to G_α, $G_{\alpha,k}$ is the undirected (symmetric) graph constructed from $N_{\alpha,k}$ and $G^s_{\alpha,k}$ is the undirected (symmetric) graph constructed from $N^s_{\alpha,k}$.

In order to prove correctness of this optimization operation for preserving k-connectivity, we need the following definition and theorem.

Definition 4. *We say a subgraph G' of G_R has* $\frac{2\pi}{3}-$property, *if for every edge $uv \in E(G)$ such that $uv \notin E(G')$, there exists a vertex w in $V(G)$ such that either $\widehat{wuv} \leq \frac{\pi}{3}$ and $|wu| < |uv|$ or $\widehat{wvu} \leq \frac{\pi}{3}$ and $|wv| < |uv|$.*

From [10], it is not hard to see that graph $G_{\frac{2\pi}{3}}$ has $\frac{2\pi}{3}-$property. The following theorem states a stronger fact:

Theorem 5. *If G_R is a connected graph, then every subgraph G' of G with $\frac{2\pi}{3}-$property is connected.*

Proof. We prove it by contradiction. Suppose G' is not connected and suppose u and v are the closest pair of vertices that are in different connected components and $uv \in E(G)$. Since $uv \notin E(G')$, there exists a vertex w for which either $\widehat{wuv} \leq \frac{\pi}{3}$ and $|wu| < |uv|$ or $\widehat{wvu} \leq \frac{\pi}{3}$ and $|vw| < |uv|$. Without loss of generality, assume the former case. $uv \in E(G)$ means $|uv| \leq R$, thus $|wu| < |uv| \leq R$, and $uw \in E(G)$. Angle $\widehat{wvu} \leq \frac{\pi}{3}$ means that edge wv cannot be the largest edge in triangle uvw, thus $|wv| < |uv| \leq R$, thus $wv \in E(G)$. Since u and v are not in the same connected component, $wv \notin E(G')$ or $wu \notin E(G')$ which contradicts the minimality of uv. This contradiction shows the connectivity of G'. □

The next theorem shows that shrink-back is a proper optimization operation.

Theorem 6. *The shrink-back optimization described above preserves k-connectivity. In other words, if $\alpha \leq \frac{2\pi}{3k}$, then according to the above definition, $G^s_{\alpha,k}$ is k-connected if the graph G_R is k-connected.*

Proof. We know that if G is k-connected then $G_{\alpha,k}$ is k-connected. Now, we should prove that in this case $G^s_{\alpha,k}$ is k-connected as well. We can prove this fact, by proving that even after removing $k-1$ vertices the shrink-back operation does not violate the $\frac{2\pi}{3}-$property of the graph. Notice that the first graph, $G_{\alpha,k}$ after removing $k-1$ vertices has $\frac{2\pi}{3}-$property. Now, removing edge uv in the shrink-back operation means this edge does not change the set $\text{cover}_k(dir)$, and it implies that there are at least k other vertices in the original graph with angle less than $\frac{\pi}{3}$ to u and closer than v to u. After removing $k-1$ vertices there are at least one such vertex and it shows that removing uv keeps the $\frac{2\pi}{3}-$property of the graph. From Theorem 5, we know that $G^s_{\alpha,k}$ is connected after removing $k-1$ vertices, and thus $G^s_{\alpha,k}$ is $k-$connected. □

5 Preserving connectivity in 3-dimensions

In this section, we generalize the connectivity algorithm from 2-dimensions to 3-dimensions. The main idea here is that each node increases its power until there is no 3D-cone of degree α in which there is no other node. Algorithm **CBTC-3D** is exactly the same as **CBTC**, except in which procedure $gap_\alpha(D_u)$ is replaced by procedure $gap-3D_\alpha$. The procedure $gap-3D_\alpha$ for a node u tests whether there exists a node in each 3D-cone of degree α centered at u.

Procedure $gap-3D_\alpha(D_u)$
let S be a sphere centered at u with arbitrary radius r
for each direction $dir_v \in D_u$
    let c_v to be the intersection of sphere S with the cone of degree α which is symmetric around dir_v
    let o_v be the center of circle c_v on sphere S
for each two circles $c_v = c_w$ **do** keep c_v and eliminate c_w
if there exists a circle c_v not intersected to any other circle
    return **false**
for intersection(s) x of any two circles c_v and c_w $(v \neq w)$ **do**
    if there is no other circle c_p such that x is inside c_p
        sort all centers o_i's according to their angles relative to x with respect to a fixed direction
        where o_i is the center of circle c_i containing x on its boundary
        if there exists an angle $\widehat{o_i x o_{i+1}} \geq \pi$ (consider i's circularly)
            return **false**
return **true**

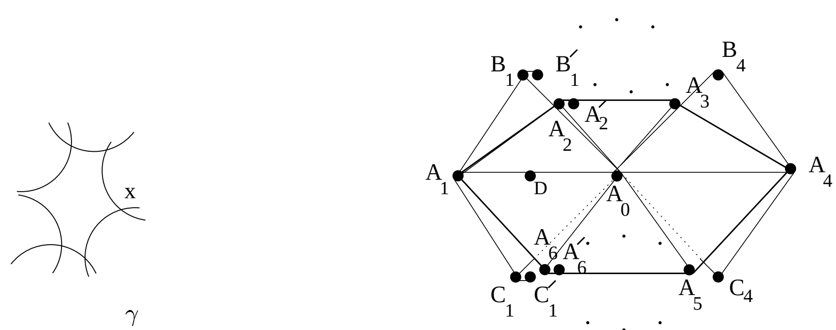
Intuitively, in procedure $gap-3D_\alpha$, first we find the intersection of sphere S with each cone of degree α symmetric around dir_v, and then we check whether these intersections cover all the sphere.

Theorem 7. *Procedure $gap3D_\alpha(D_u)$ detects correctly whether there exists a cone of degree α centered at u without any other node.*

Proof. First, we note that there is no cone of degree α, centered at a node u, without any other node if and only if the set U, the intersection of sphere S with all cones of degree α symmetric around dir_v's covers the sphere S. We show that in the case $U \neq S$ one of the following three situations happens (here by S we mean all points on sphere S).

1. U is disconnected.
2. There exists a point x which is the intersection of boundaries of exactly two circles (these circles are intersections of cones with sphere S) and belongs to (the interior of) none of the other circles.
3. There exists a point which is the intersection of k circles ($k \geq 3$) and the angle of two consecutive centers of these circles is greater than or equal to $\frac{\pi}{2}$.

Suppose $U \neq S$ and let the curve γ be (one of the connected components of) the boundary of U. Curve γ is either a circle or the union of several concave arcs (see Figure 3(aa)). In the former case, since $\alpha < \pi$, the set U is disconnected and the claim is proved. In the latter case, we consider $x \in \gamma$ one of the intersections of two boundary arcs (see Figure 3(aa)). If it is the intersection of exactly two circles then there is a point close to x which belongs to none of the other circles and the proof is complete. Suppose x is the intersection of k circles where $k \geq 3$. Let $o_1, \ldots, o_k$ be the centers of the circles passing through x. Since the radii of the circles are equal, $o_1, \ldots, o_k$ belongs to a circle centered at x. If $\widehat{o_i x o_{i+1}} < \frac{\pi}{2}$ for each $i = 1, \ldots, k$ ($o_{k+1} = o_1$) then the circles around $o_1, \ldots, o_k$ (and thus the set U) cover all neighborhood close enough to x. Since x in on the boundary of U, it is impossible. Hence the claim is proved. □



(aa) **(bb)**

Fig. 3. (aa) Concave curves (bb) The 3D counterexample.

Below we prove that procedure **CBTC**(α) with $\alpha \leq \frac{2\pi}{3}$ works correctly.

Theorem 8. *If $\alpha \leq \frac{2\pi}{3}$, procedure* **CBTC**($\alpha$) *preserves connectivity.*

Proof. We prove by obtaining a contradiction. Consider two nodes u and v of minimum distance which are connected in G_R, but there is no path from u to v in G_α. Since $\{u, v\} \notin E_\alpha$, w.l.o.g. we can assume $v \notin N_\alpha(u)$. Thus there exists a node w such that $\widehat{wuv} \leq \frac{\pi}{3}$. Because of procedure **CBTC-3D**, we know $|wu| < |uv|$. Since $\widehat{wuv} \leq \frac{\pi}{3}$ and $|wu| < |uv|$, uv is the largest edge of the triangle wuv. Thus $|wv| < |uv| \leq R$ and $|wu| < |uv| \leq R$. By the assumption that $\{u, v\}$ is a counterexample with minimum distance, w and v are connected in G_α and also w and u are connected in G_α. Thus v and u are connected in G_α and it is a contradiction. □

It is worth mentioning that using a very similar approach of k-connectivity in 2-dimensions, we can also prove procedure **CBTC**(α) for $\alpha \leq \frac{2\pi}{3k}$ preserves k-connectivity. Now we show that these bounds are tight upper bounds in some cases.

Theorem 9. *For $\alpha > \frac{2\pi}{3}$, procedure* **CBTC**(α) *does not necessarily preserve connectivity and for $\alpha > \frac{\pi}{3}$ does not necessarily preserve 2- or 3-connectivity.*

Proof. In the 3-dimensional space with axes x, y and z consider the hexagon $A_1, \cdots, A_6$ in the xy-plane centered at $A_0 = 0$ (i.e., A_0 is the origin) with radius R (see Figure 3(bb)). For $i = 1, \cdots, 6$, let B_i and C_i be two points in the space such that the triangles $\triangle A_0 A_i B_i$ and $\triangle A_0 A_i C_i$ are equilateral whose planes are orthogonal to the xy-plane (B_i above and C_i under the xy-plane). We define vector $v = \delta \overrightarrow{A_1 A_0}$ and $B'_1 = B_1 + v$, $C'_1 = C_1 + v$, $A'_2 = A_2 + v$, $A'_6 = A_6 + v$ (here we consider points and their vectors from origin equivalent). Figure 3(bb) depicts these points. If δ is sufficiently small,

$$\widehat{B'_1 A_0 C'_1} < \frac{2\pi}{3} + \epsilon \ , \ |A_1 B'_1| = |A_1 C'_1| > R \ and \ |A_1 A'_6| = |A_1 A'_2| > R. \tag{1}$$

For an arbitrary point P define $\overline{P} = (1 - \delta_1)P$ (i.e. $\overline{P}$ is slightly closer to the origin than P). For sufficiently small $\delta_1 > 0$, because of (1),

$$\widehat{\overline{B'_1} A_0 \overline{C'_1}} < \frac{2\pi}{3} + \epsilon \ , \ |A_1 \overline{B'_1}| = |A_1 \overline{C'_1}| > R \ and \ |A_1 \overline{A'_6}| = |A_1 \overline{A'_2}| > R. \tag{2}$$

We consider tight examples for k-connectivity for different k's:

Case $k = 1$**:** We consider the following set of points $S_1 = \{A_0 = \overline{A_0}, A_1, \overline{A_3}, \overline{A_5}, \overline{B'_1}, \overline{B_3}, \overline{B_5}, \overline{C'_1}, \overline{C_3}, \overline{C_5}\}$. Before the process, A_1 is connected only to A_0. After the process with $\alpha = \frac{2\pi}{3} + \epsilon$, by radius $(1 - \delta_1)R$ the vertex A_0 can observe all of the other points except A_1. Here in each cone of degree $\frac{2\pi}{3} + \epsilon$, node A_0 can see a vertex, which makes A_1 isolated.

Case $k = 2$**:** Let set $S_2 = \{A_0, A_1, \overline{A'_2}, \overline{A_3}, \cdots, \overline{A_5}, \overline{A'_6}, \overline{B'_1}, \overline{B_2}, \cdots, \overline{B_6}, \overline{C'_1}, \overline{C_2}, \cdots, \overline{C_6}, D\}$, where $D = \frac{1}{2}A_1$. Before the process, A_1 is connected only to A_0 and D and the graph is 2-connected. After the process, with $\alpha = \frac{\pi}{3} + \frac{\epsilon}{2}$ the connection between A_0 and A_1 will be removed and the graph loses its 2-connectivity.

Case $k = 3$**:** Assume set $S_3 = (S_2 - \{\overline{A'_2}\}) \cup \{\overline{A_2}\}$. Since triangle $A_0 A_1 A_2$ is equilateral for small $\delta_1 > 0$, $|A_1 \overline{A_2}| < |A_1 A_0| = R$. Thus before the process, A_1 is connected to A_0, $\overline{A_2}$ and D, and the graph is 3-connected. After the process with $\alpha = \frac{\pi}{3} + \frac{\epsilon}{2}$, the connection between A_0 and A_1 is removed and the degree of node A_1 is less than 3. Thus the graph loses its 3-connectivity.

It shows that $\alpha = \frac{2\pi}{3}$ and $\alpha = \frac{\pi}{3}$ is tight for cases $k = 1$ and $k = 2$ respectively and $\alpha = \frac{\pi}{3}$ is an upper bound for $k = 3$. □

6 Dealing with mobility

In a wireless multi-hop network, nodes may be added to the system, may change their positions, or even may die due to lack of power supply. To deal with these situations, Li et al. [10] presented the following *Neighbor Discovery Protocol (NDP)*. Each node uses a beaconing protocol to inform its neighbors that it is still alive. The beacon includes the transmission power and the sending node's ID. This beacon is sent with the power obtained from the **CBTC** algorithm. A node u considers a node v failed if it does not receive any beacon within a time interval T from v ($leave_u(v)$ event). A node v considers a node u joined to the system, if it receives a beacon from v within the current time interval T and it has not received any beacon from v within the previous time interval ($join_u(v)$ event). Finally, a node u considers changing in position of node v, if its angle with respect to u has changed ($change_u(v)$ event).

Using aforementioned events, the reconfiguration algorithm is simple. If a $leave_u(v)$ event happens, and there exists an α-gap after dropping $dir_u(v)$ from D_u, node u reruns **CBTC**(α) with the current power as the initial power instead of p_0 (see **CBTC** algorithm). If a $join_u(v)$ event happens, node u do the shrink-back operation, removing farthest neighbor as long as their removal does not change the coverage. Finally, if a $change_u(v)$ event happens, first node u treats as it treats for $leave_u(v)$ event, and then if there is no α-gap it treats as it treats in $join_u(v)$ event. The implementation and timing issues and difficulties with asynchrony have been discussed in Li's et al. paper [10]. The reader is referred to the paper for further details.

7 Conclusions and future work

In this paper, we considered fault-tolerant distributed topology control algorithm. Our algorithm was based on the the cone-based algorithm introduced by Li et al. We showed that running the algorithm with $\alpha = \frac{2\pi}{3k}$ is sufficient for preserving k-connectivity. In addition, if k is even this upper bound is tight and if k is odd this upper bound is very near to the optimal α. We also considered the extension of the cone-based algorithm to 3-dimensions, and showed that again running the algorithm with $\alpha = \frac{2\pi}{3k}$ is an upper bound for preserving k-connectivity and for $k = 1, 2(3)$ this bound is (nearly) tight. Here, we present several open problems that are possible extensions of this paper.

In this paper, we designed a fault-tolerant topology control algorithm by finding a $k-$connected subnetwork. The problem of finding $k-$connected subgraph with minimum cost is known to be NP-Hard. There are constant-factor approximation algorithms for this problem [2]. In order to use them in a distributed mobile topology control algorithm, one issue is to implement these algorithms on distributed networks (without global available information), and another issue is to find the solution using the solution for the previous network after some topology changes without recomputing everything. Adapting algorithms in [2] on distributed and mobile networks might be a future work in this area.

One topic of interest is finding other properties of the graph for which the cone-based distributed topology control can be applied. Finding algorithms similar to the cone-based algorithm which only uses some local decisions to preserve a global property is an open area of research. Finding further optimizations that can be applied after the cone-based algorithm are interesting too.

We suspect that $\alpha = \frac{2\pi}{3k}$ is also a tight upper bound for preserving k-connectivity where k is odd. Constructing an example for which the algorithm with $\alpha = \frac{2\pi}{3k} + \epsilon$ does not work would be instructive. In addition, we believe that the upper bounds for 2-dimensions also works for 3-dimensions, as we showed for k-connectivity for $k \leq 3$. Exploring general examples for these bounds would be a nice theoretical result.

Acknowledgements We are grateful to professor Nancy Lynch for her encouragement, reading, and helpful comments on this paper.

References

1. A. Chandrakasan, R. Amirtharajah, S.H. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and Wang A. Design considerations for distributed microsensor systems. In *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pages 279–286. May 1999.
2. M.X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.
3. Y. Hassin and D. Peleg. Sparse communication networks and efficient routing. In *Symp. on Principles of Distributed Computing*, pages 41–50. 2000.
4. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *IEEE Hawaii Int. Conf. on Systems Sciences*, pages 4–7. January 2000.
5. T.C. Hou and V.O.K. Li. Transmission range control in multi-hop radio networks. *IEEE Trans. on Communications*, 34(1):38–44, 1986.
6. L. Hu. Topology control for multihop packet radio networks. *IEEE Trans. on Communications*, 41(10):1482–1493, 1993.
7. M. Keil and C.A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete and Computational Geometry*, 7:13–28, 1992.
8. K. Krizman, T.E. Biedka, and T.S. Rappaport. Wireless position location: fundamentals, implementation strategies, and source of error. In *IEEE 47th Vehicular Technology Conference*, pages 919–923. 1997.
9. L. Li and J. Halpern. Minimum energy mobile wireless networks revisited. In *IEEE International Conference on Communications (ICC)*. June 2001.
10. L. Li, J. Halpern, V. Bahl, Y.M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks. In *ACM Symposium on Principle of Distributed Computing (PODC)*. August 2001.
11. R. Ramanathan and R. Rosales-HainRodoplu. Topology control of multihop radio networks using transmit power adjustment. In *Proc. IEEE INFOCOM*, pages 404–413. March 2000.
12. Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications*, 17(8):1633–1639, 1999.
13. H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. on Communications*, 32(1):38–44, 1984.
14. Godfried T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern recognition*, 12(4):261–268, 1980.
15. R. Wattenhofer, L. Li, V. Bahl, and Y.M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *IEEE INFOCOM*. April 2001.
16. Douglas B. West. *Introduction to Graph Theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.