Topology Control

Paulina Varshavskaya

1 Overview

Li, Halpern, Bahl, Wang, and Wattenhofer. Analysis of a Cone-based Distributed Topology Control Algorithm for Wireless Multi-hop Networks.

and

Bahramgiri, Hajiaghayi, and Mirrokni. Fault-tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks.

Start with network represented as a graph, where edges represent (some or all) nodes that can communicate with other in one bcast hop.

Nodes can transmit at some max power, can lower the power, which means they can communicate with fewer nodes. Thus, this makes the graph sparser.

Want nodes to reduce their power:
Save energy
Reduce interference

But should not reduce network connectivity "too much". "Too much" here means they should not disconnect a graph that was connected. The idea is that the graph is supposed to be used for some higher-level algorithms, e.g., message routing.

Issues with the "connectivity" criterion? Even though u and v are connected, the shortest path from u to v might become longer.

But that might be okay because it might be less costly, in terms of power, for a series of nodes to transmit (relay) between u and v, than for u to transmit directly to v, as the power required grows as a power of $n, n >= 2$ with distance.

Just maintaining connectivity doesn't give any redundancy, which might be useful for fault-tolerance, say in message routing applications.

Experimentally, their algorithm returns a graph that is still very redundant, at least in the experiments they show. The idea is to remove unnecessary interference and power while maintaining enough redundancy not to cause congestion and delays. It's a trade-off. But in theory there can be "pivot" nodes.

Their reduction should be by a distributed algorithm, using local information.
To save communication
To make it easier to adapt to changes.

They make strong theoretical assumptions about communication. Consider algorithms of a very particular kind: Examine angles, determine power to exactly the minimum needed to that they can reach SOME node in each angle of a certain size alpha. This assumption is used in proofs; but they discuss ways of increasing power that are heuristic.

Actually, once they have these reduced sets of neighbors, they modify the transmission power to produce symmetry in the neighbor graph. They do this in two different ways—we'll discuss.

Goal: get characterization results for the angle sizes that are needed to get connectivity, and k-connectivity.

2 Problem statement

2.1 Assumptions

Ad hoc network

Nodes: No location information (GPS) Have clocks (but this isn't completely essential—they talk about removing this assumption).

Broadcast assumptions: Can detect exact direction from which a transmission arrives. Nodes can transmit at different powers. All nodes have same max power P, which reaches exactly distance R. Function p describes EXACTLY how far they reach with power p (assumed to be a circle/sphere). Their notation suggests that this function p is the same for all nodes, but the way they discuss this suggests that it need not be.

Knowledge of the power: Node u knows the power at which it's transmitting. Another (rather unclearly stated) assumption: If node u gets a message from a node v, and knows the power p at which the message was sent and the power p' at which it was received, then u can estimate the exact power it needs to send to v. Justify this? Sounds like they would have to calculate the distance from a model of signal attenuation, then use this to infer power needed. Can that actually be done reliably?

This knowledge is important in the distributed algorithm parts of the paper. Most of the paper is actually just about the geometry, though. So this is a somewhat peripheral issue.

2.2 Problem requirements

Each node separately should determine a power to transmit at, in such a way as to maintain connectivity between any two nodes that were connected in the original graph.

This isn't exactly the same as "preserving connectivity" of the entire graph.

In fact, it's better than preserving global property of connectivity. It's preserving connected components in the original graph.

2.3 What the algorithms are allowed to do

They consider just a constrained type of algorithm. Parameterized by an angle alpha. Given UNDIRECTED graph G, they determine an UNDIRECTED subgraph G_α.

First, for each node, a neighbor set $N_\alpha(u)$ gets defined: Consider exactly the smallest radius d such that a circle around u with radius d contains (perhaps on the boundary) at least one node in each cone of angle alpha.

No need to talk of power levels here – equally well described with geometry.

Now let $N_\alpha(u)$ be the set of all the nodes in (or on) this circle.

Let N_α be the set of all (u,v) such that v is in $N_\alpha(u)$. Note that we can't use N_α as the set of edges of G_α, since we don't have symmetry.

(Do we only not have symmetry if $\alpha > 2\pi/3$? Their example seems to break down otherwise.)

So, they take one more step to get symmetry. This can be done in two ways:
1. Put in (u,v) if EITHER (u,v) or (v,u) is in N_α. In most of the LH paper, except for section 3.2.
2. Put in (u,v) if BOTH (u,v) and (v,u) are in N_α. In LH section 3.2, and all of the BHM paper.

LH defines G_α by means of 1 above, and $\overline{G}_\alpha$ by means of 2. BHM define G_α by means of 2. So let's use the LH notation here.

The actual distributed algorithms are supposed to determine the exact power level needed to just reach the neighbors in G_α.

The papers try to determine good choices of alpha: Upper bound results showing the choices preserve connectivity or k-connectivity. Lower bound results showing that bigger alphas don't work. And they give distributed algorithms to determine power levels to reach these neighbors.

We can try to separate the geometry aspects of the papers (what are the right choices of alpha?) from the distributed algorithms aspects.

2.4 Summary of what the papers do

Li et al.

They give exact characterization of alphas that preserve connectivity, given interpretation 1 (union of asymmetrical edges) above.
alpha = 5pi/6.

A sufficient condition for preserving connectivity, given interpretation 2, with a much simpler argument.
alpha = 2pi/3

Also describe how to determine the needed graphs with distributed algorithms.

BHM

Use interpretation 2 only. Consider k-connectivity.

Get characterizations that are "almost" exact: Exact for even k, a gap for odd k.
2pi/3k

Also 3D, similar results.

3 Li, Halpern

3.1 CBTC(α)

Given an original network represented by a graph G_R, CBTC (Cone-Based Topology Control is an algorithm for reducing the number of edges in that graph while preserving its connectivity.

Algorithm is distributed (executed at every node) and uses only local information (no GPS). But assumes that directions to neighbors can be computed.

The paper talks about power of transmission determining the radius of the neighborhood, but power issues can be kept separate. We will discuss the geometry of the network graph without talking about power of transmission.

$CBTC(\alpha)$
Initialize: empty set of neighbors and directions to neighbors, initial power p_0 (or initial neighborhood radius).
While power less than maximum possible power P and there's still a gap of size α between discovered neighbors, do
increase transmission power
broadcast a "hello" message with current power level and gather acks
add any discovered neighbors to the set of neighbors
add their directions to the set of directions.

Algorithm stops when either there are no gaps of size α or maximum power has been reached.

The final set of discovered neighbors ($N_\alpha(u)$) is not always symmetric. Node v can be in the final set of neighbors of u, but doesn't have u as its neighbor because there are closer nodes for all cones of size α.

Notation: $G_\alpha = (V, E_\alpha)$ is the new graph made by CBTC, where the set of edges E_α is constructed using interpretation 1 — (u,v) is in that set either if u discovered v as a neighbor, or if v discovered u.
$\overline{G}_\alpha$ is the new graph under interpretation 2 — (u,v) is in the set of edges $\overline{E}_\alpha$ if both u and v discovered each other.

How could an algorithm determine $\overline{G}_\alpha$? (p.269, last paragraph of section 3.2) After finishing CBTC(alpha), a node u must send a message to each node v to which it sent an ack that is not in $N_\alpha(u)$, telling v to remove u from $N_\alpha(v)$.

But we will talk about the geometry of graphs instead of powers of transmission.

3.2 Sufficient condition for interpretation 2:

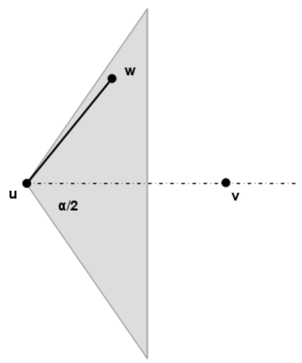
How large an angle α can we give CBTC and still preserve connectivity?

Theorem 3.2 (p. 269):

If $\alpha \leq 2\pi/3$, then $\overline{G}_\alpha$ preserves connectivity of original graph G_R.

Proof:
Preserves connectivity means if nodes were originally connected, they stay connected. Can be connected by an edge, or a path through intermediate nodes ($u_0 ... u_k$).
Two lemmas (3.4 is a strengthening of 3.3):
1. If (u,v) was an edge in G_R then either (u,v) is in N_α or there is a path ($u_0 ... u_k$) s.t. (u_i, u_{i+1}) is in N_α and $d(u_i, u_{i+1}) < d(u, v)$ for each i=0...k-1.

Proof by induction on the ordering of edges by length.
If (u,v) is in N_α, done.
Otherwise, (u,v) is not in N_α. That means that CBTC terminated with no α-gap; so there must be a closer neighbor w in the $cone(u, \alpha, v)$.



We know that d(u,w) < d(u,v), so by induction there's a path from u to w consisting of edges in N_α.
What about from w to v?
"Sweeping" up through the cone, choose w to be the node such that the angle (wvu) is smallest. The distance $d(wv)$ must be less than $d(uv)$ because of a triangle argument. Otherwise, since the angles in the triangle (wvu) are at most $\pi/3$, if the distances are equal, then v would be inside the cone, and the edge (u,v) would be in N_α, which it is not.
Therefore, the edge (w,v) is in E and has lower rank in the ordering of edges by length. Applying the induction hypothesis, the lemma holds for (u,v).

2. (Lemma 3.4).
If (u,v) was an edge in N_α then there is a path $u_0 ... u_k$ s.t. $u_0 = u, u_k = v, (u_i, u_{i+1})$ is in $\overline{E}_\alpha$, for i=0...k-1.
This means: for every edge in N_α there is a path from u to v consisting of symmetric edges (both directions are in N_α.

Proof by induction on the length of edges (as before).
Take edge (u,v) assume in N_α and suppose (v,u) is not in N_α.
We know that (v,u) is in E (length less than R), so Lemma 3.3. yields a path consisting of shorter edges in N_α. If any of those edges are in N_α but not in $\overline{E}_\alpha$, we can apply the inductive hypothesis to replace that edge with a path consisting of edges in $\overline{E}_\alpha$.
$\overline{E}_\alpha$ is symmetric by definition, so any path from v to u implies a path from u to v. Lemma holds.

Theorem 3.2. is now immediate.

3.3 Sufficient for interpretation 1

Of course, 2pi/3 would work here also, since G_α just adds extra edges to $\overline{G}_\alpha$. But the surprising thing is that anything up to 5pi/6 also works.

Theorem 2.1.
If $\alpha \leq 5\pi/6$ then u and v are connected in G_α iff they are connected in G_R.

Proof:
Lemma 2.2.
If $\alpha \leq 5\pi/6$ and (u,v) is an edge of original graph G_R, then either (u,v) is in E_α or there is a pair of nodes u', v' s.t. all of this holds:
d(u',v') < d(u,v)
either u'=u or (u,u') is in E_α
either v'=v or (v,v') is in E_α.

Start the same as for Theorem 3.2.
Same argument in the case that there are neighbors in the cone(u,2pi/3,v) and cone(v,2pi/3,u).
Case where those cones do not have neighbor nodes in them (neighbors are far apart):
By using the fact that every alpha-cone around both u and v is filled, get four points z,w,x, and y in a certain geometrical relationship to u, v and the cones (distances are within neighborhood, angles are large).
Then they prove by contradiction that one of the distances d(w,z) or d(x,y) must be less than neighborhood radius. This means that that edge can be taken as (u',v') and is in E_α.

Details of proof left out from lecture in the interest of time.
Insight: Assuming that the sides (w,z) and (x,y) are long, we can show that the final angle is big.

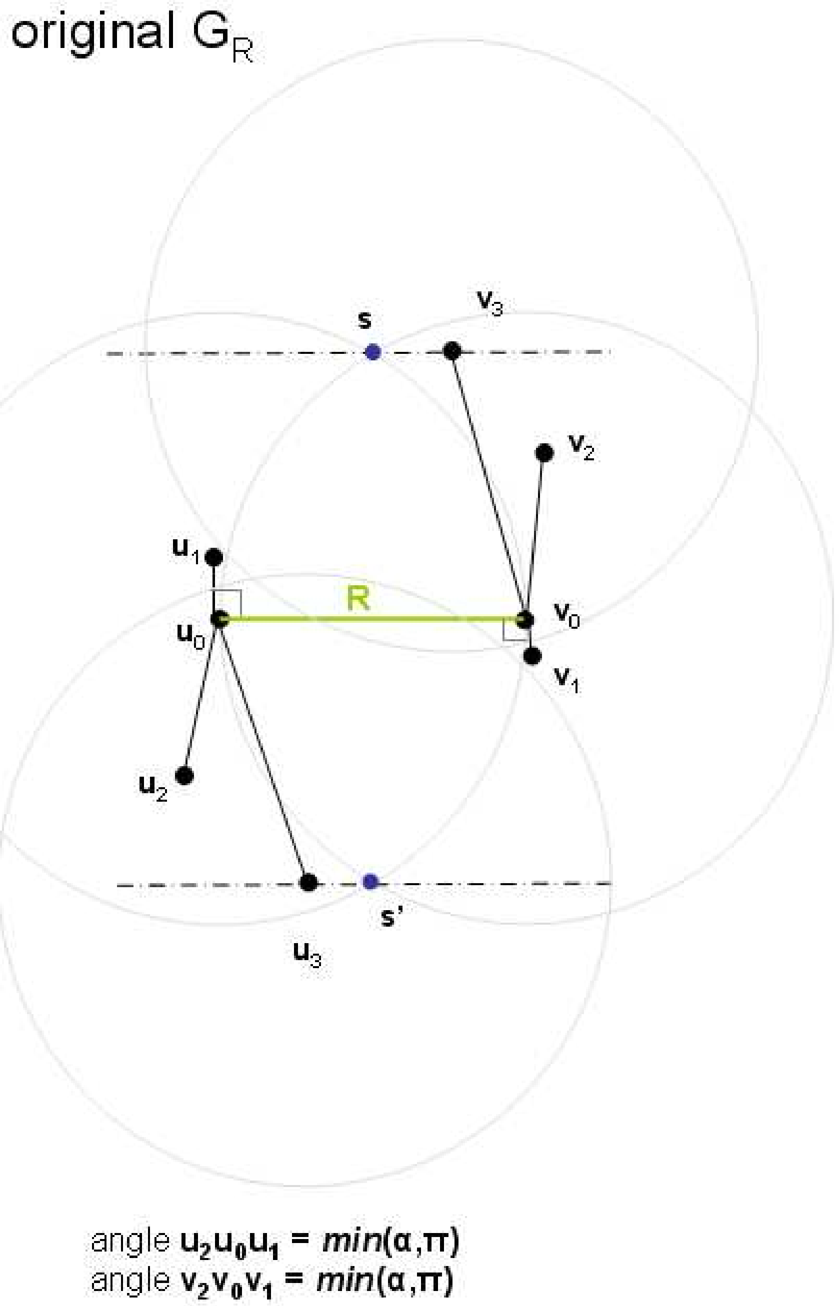
3.4 Necessity for interpretation 1

Theorem 2.4.
If $\alpha > 5\pi/6$ then CBTC(α) does not necessarily preserve connectivity.

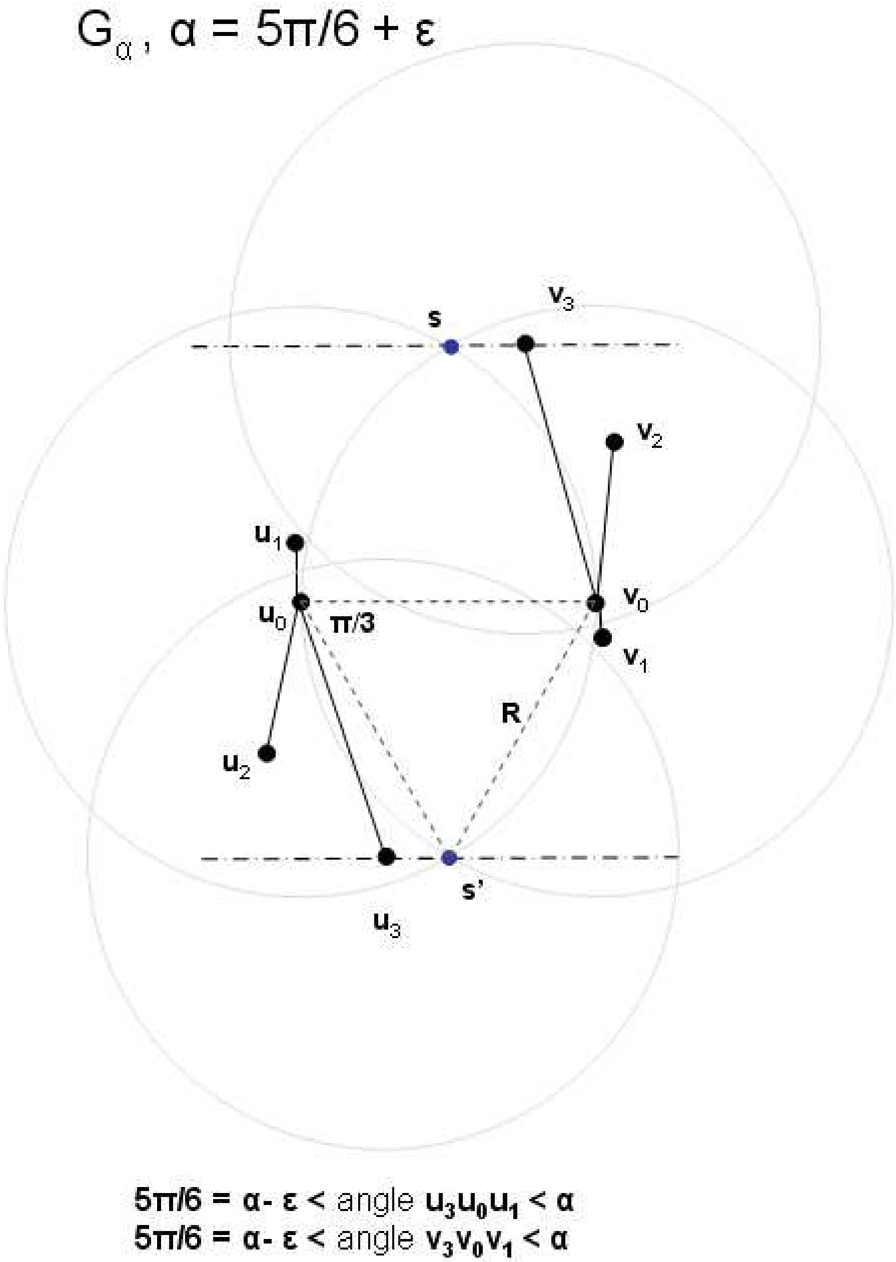
Proof by constructing an example network where it doesn't.

Step by step construction:
0. There will be 8 nodes in 2 clusters: the u cluster (u0...u3) and the v cluster (v0...v3).
1. $d(u_0, v_0) = R$ and it's the only edge between the clusters.
2. u_1 is placed so that the angle $u_1u_0v_0 = \pi/2$ — it's on a tangent to the circle with radius R originating at v_0.
3. u_2 is placed so that the angle $u_1u_0u_2 = min(\alpha, \pi)$ and d(u0,u1) = R/2.
4. Note that the angle s'u0u1 = 5pi/6 (=pi/3+pi/2). Let u_3 be a point on the line drawn through s' and parallel to (uv) s.t. the angle $u_3u_0u_1 < \alpha$. This is possible because $\alpha = 5\pi/6 + \epsilon, \epsilon > 0$.

v nodes are constructed similarly.



This is the original graph G_R with the only edge connecting u and v clusters.



This is the graph G_α after running CBTC(α) with $\alpha = 5\pi/6 + \epsilon$, $\epsilon > 0$.

The edge (u,v) disappears, since all inter-cluster distances are greater than R, but CBTC will terminate with a power $p_{u_0,\alpha} = max(d(u_0, u3), R/2) < R$. Similarly for v_0.

This lower bound result assumes that there is no power "overshooting" — the minimal power required for G_α is used.

3.5 Optimizations

Certain optimizations are possible and lead to even leaner graphs which still preserve connectivity.

3.5.1 Shrink-back

There will be "boundary nodes" in a network which simply do not have neighbors in some of their cones. Under non-optimized algorithm, those nodes will terminate with a gap and at maximum power R.

Instead, the nodes can keep track at which power each neighbor was discovered. When CBTC terminates, go back to the maximum level of power recorded.

3.6 Practical considerations

In networks, nodes can move away from the neighborhood, or die; new nodes can be added to the network.

CBTC can be used in conjunction with a Neighbor Discovery Protocol (NDP). Usually, a beaconing protocol for each node to tell neighbors that it is still alive.

What power should a node use for beaconing? With sufficient power to reach all of its neighbors in E_α.

Present a reconfiguration algorithm which will work if the network does not change too quickly. Can operate asynchronously.

Experimental results show average node degree and average radius of network after running CBTC(2pi/3) or CBTC(5pi/6). They are comparable, there are some trade-offs. Optimizations are applicable.

Potential problems: the resulting networks can be too sparse — not robust to failures of a "pivot" node, prone to congested traffic.

4 Bahramgiri et al.

In this paper, only symmetric edges are considered. We switch notation to G_α for interpretation 2 (the only interpretation here).

4.1 k-connectivity

K-connectivity means two things: 1) you can remove any number of nodes less than k and the net is still connected, and 2) there are k paths between any node u and v that don't go through the same nodes (node-disjoint paths).

These two definitions are equivalent by Menger's Theorem.

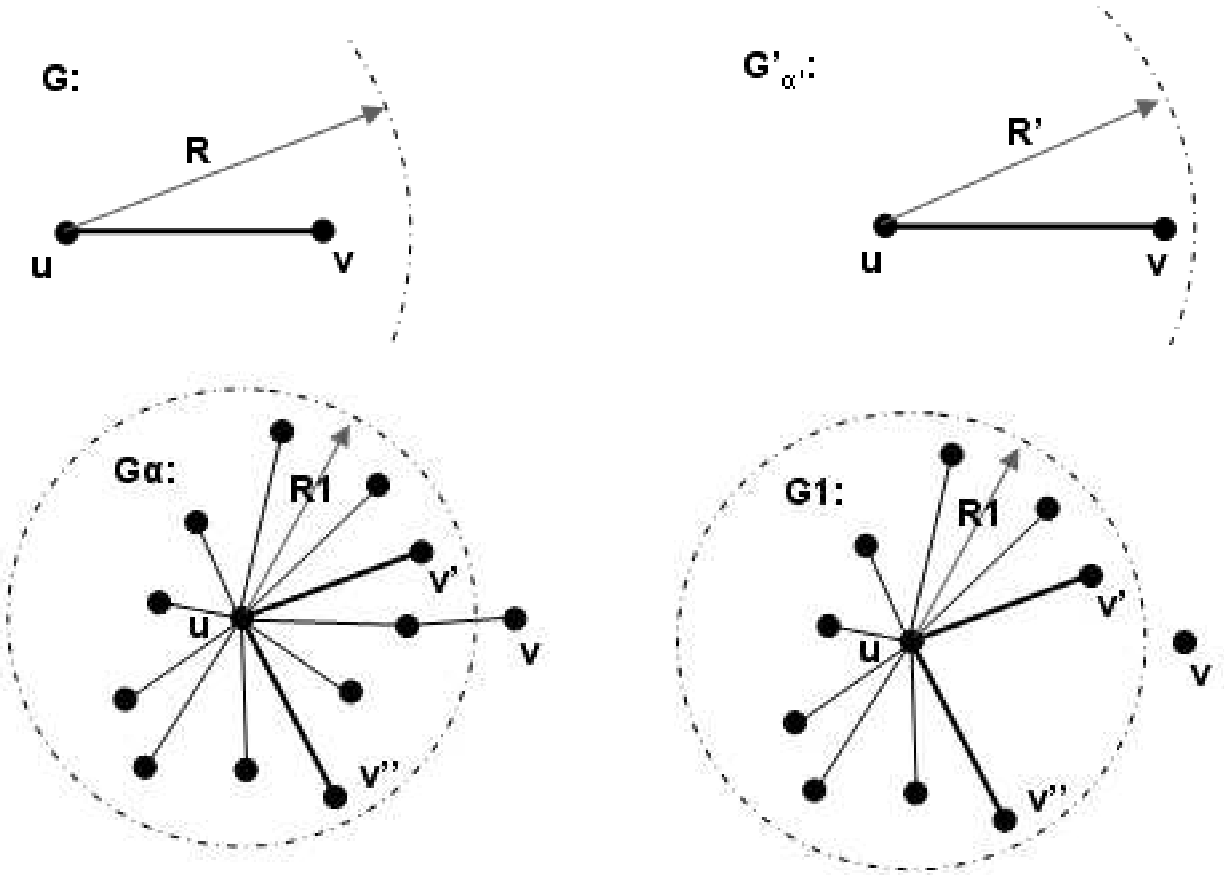
Obviously, a k-connected network is more fault-tolerant than only a 1-connected one.

4.2 Sufficient condition for k-connectivity

Running CBTC algorithm with a smaller angle α preserves k-connectivity under certain conditions.

Theorem 1.
If graph G is k-connected then so is $G_{2\pi/3k}$.
Uses Lemma 1 (result from Li et al): If G is connected, so is $G_{2\pi/3}$.

Proof of theorem:
k-connected G — CBTC(alpha) — G_α — remove k-1 nodes — G_1
k-connected G — remove k-1 nodes — G' —- CBTC(alpha') — $G'_{\alpha'}$
$\alpha = 2\pi/3k, \alpha' = 2\pi/3 = k\alpha$
Proof by contradiction.Suppose that there are k-1 nodes s.t. if they are removed from G_α the resulting graph G_1 is disconnected.



G is k-connected, therefore if we remove the same k-1 nodes from G, the resulting graph G' is

connected. From lemma 1, $G'_{\alpha'}$ is also connected. Prove that $G'_{\alpha'}$ is a subgraph of G1; this will contradict our assumption of G1 being disconnected.
Suppose there is an edge (uv) in $G'_{\alpha'}$, which is not an edge in G1. The distance (uv) must be less than or equal to R. And power to talk over that distance must be greater than the power set by CBTC(alpha).
Edge (uv) not an edge in G1, therefore not in G_α either. Therefore, there must be some vertices closer than v to u. The max. angle between those closer nodes must be at most 2pi/3k.
When k-1 nodes are removed, the the max. angle between the remaining closer nodes must be at most $2\pi/3k + (k-1) * 2\pi/3k = 2\pi/3 = \alpha'$ (see figure above).
These nodes must be also in $G'_{\alpha'}$, and therefore the power of u is less than the distance (uv) in $G'_{\alpha'}$.
Therefore (uv) is not in $G'_{\alpha'}$ (only symmetric edges allowed). Contradiction.

Note that this proof uses the minimum power idea, rather than increments.

4.3 Necessary conditions for k-connectivity

What is the maximum angle for preserving k-connectivity? Two results for odd or even k.

Theorem 3:
If k is odd, there exists a k-connected graph G s.t. if we run CBTC(alpha) on it with $\alpha\, 2\pi/3(k-1)$ the resulting graph is not k-connected.

Theorem 4:
If k is even, there exists a k-connected graph G s.t. if we run CBTC(alpha) on it with $\alpha\, 2\pi/3k$ the resulting graph is not k-connected.

Proofs use Harray graphs (figure 1. in paper). A theorem says that $\kappa(H_{k,n} = k$, a Harray graph with n vertices and k neighbors per vertex, is k-connected.
Another theorem: if you take a k-connected graph, add a node and edges to every other node from it, the resulting graph is (k+1)-connected.

Then construct a Harray graph on a circle of radius $R - \epsilon$, plus a center node c with edges to every other node, plus a node u that is a distance R away from c. The nodes are placed in such a way that if CBTC is run with $\alpha = 2\pi/3(k-1) + \delta$, then all cones starting at c have nodes in them on the circle of radius $R - \epsilon$. That is where the power will stop. And thus u will not have an edge to c, and the graph will not be k-connected (deleting all neighbors of u will disconnect it, and there are k-1 of them).

Note: for large k, the difference between the two bounds is very small.

Shrink-back optimization is possible.

4.4 3 dimensions

The same algorithm carries over to 3D, with modified procedure for checking if there is a gap of size α. The procedure works by checking intersections of a sphere around the node u with 3D cones emanating from u.

Li et al's results for simple connectivity and $\alpha \leq 2\pi/3$, carry over to 3D.
k-connectivity with $\alpha \leq 2\pi/3k$ carries over to 3D.

For tight upper bound results on α, prove that $\alpha > 2\pi/3$ results in disconnections sometimes (simple connectivity).
Specific results for 2- and 3-connectivity and $\alpha > \pi/3$.

4.5 Discussion

E.g., what other properties of the graph can cone-based topology control be applied to?