FLSS: A Fault-Tolerant Topology Control Algorithm for Wireless Networks

Ning Li and Jennifer C. Hou
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
nli, jhou@cs.uiuc.edu

***Abstract*— The development of wireless communication in recent years has posed new challenges in system design and analysis of wireless networks, among which energy efficiency and network capacity are perhaps the most important issues. As such, topology control algorithms have been proposed to maintain network connectivity while reducing energy consumption and improving network. However, by reducing the number of links in the network, topology control algorithms actually decrease the degree of routing redundancy, and hence the topology thus derived is more susceptible to node failures/departures. In this paper, we consider k-vertex connectivity of a wireless network. We first present a centralized greedy algorithm, called *Fault-tolerant Global Spanning Subgraph* (FGSS$_k$), which preserves k-vertex connectivity. FGSS$_k$ is min-max optimal, i.e., FGSS$_k$ minimizes the maximum transmission power used in the network, among all algorithms that preserve the k-vertex connectivity. Based on FGSS$_k$, we then propose a localized algorithm, called *Fault-tolerant Local Spanning Subgraph* (FLSS$_k$). We formally prove that FLSS$_k$ preserves k-vertex connectivity while maintaining bi-directionality of the network. We also prove FLSS$_k$ is min-max optimal among all strictly localized algorithms. Finally, we relax several widely used assumptions for topology control, in FGSS$_k$ and FLSS$_k$ so as to enhance their practicality. Simulation results show that FLSS$_k$ is more power-efficient than other existing distributed/localized topology control algorithms.**

***Index Terms*— Topology control, fault tolerance, k-vertex connectivity.**

I. Introduction

The development of wireless communication in recent years has posed new challenges in system design and analysis of wireless networks, among which energy efficiency and network capacity are perhaps the most important issues. As such, topology control algorithms have been proposed to maintain network connectivity while reducing energy consumption and improving network capacity [1]–[9]. Instead of transmitting with the maximal power, nodes in a wireless multi-hop network collaboratively determine their transmission power and derive the network topology by forming proper neighbor relation under a specific topology control algorithm. By enabling wireless nodes to use adequate transmission power, topology control not only saves energy and prolongs network lifetime, but also improves spatial reuse (and hence the network capacity) and mitigate the MAC-level medium contention.

On the other hand, by reducing the number of links in the network, topology control algorithms actually decrease the degree of routing redundancy. As a result, the topology thus derived is more susceptible to node failures/departures. This problem can be mitigated if an adequate level of routing redundancy can be properly figured into topology control. In particular, a k-vertex connected network is $k-1$ fault-tolerant, i.e., it can survive the failure of at most $k-1$ nodes.

In this paper, we first present a centralized greedy algorithm, called *Fault-tolerant Global Spanning Subgraph* (FGSS$_k$), that preserves k-vertex connectivity and is min-max optimal (as will be elaborated on, the property of min-max optimality is critical to extend the network lifetime). Based on this algorithm, we then propose a fully localized algorithm, called *Fault-tolerant Local Spanning Subgraph* (FLSS$_k$), for topology control in wireless networks. By fully localized we mean each node operates on the information locally collected. This feature enables FLSS$_k$ to adapt to topology changes more easily. It can be proved that FLSS$_k$ preserves k-vertex connectivity and maintains bi-directionality for all the links in the topology, while reducing the power consumption and improving the network capacity. We also prove that FLSS$_k$ is min-max optimal among all strictly localized algorithms.

After the theoretical base is formally laid and the algorithms devised, we also examine several widely used assumptions in topology control, e.g., use of a common maximal transmission power among all the nodes, obstacle-free communication channel, capability of obtaining position information, and seek solutions to relax these assumptions, thus improving the practicality of FGSS$_k$ and FLSS$_k$. Finally we have shown via simulation that as compared with the topologies derived under other distributed/localized fault-tolerance centric topology control algorithms, the topology derived under FLSS$_k$ has smaller average node degree, smaller average link length, and smaller average transmission power. The former property reduces MAC-level contention, while the latter two properties implies that only small transmission power is required.

The rest of the paper is organized as follows. We first define the network model in Section II, and summarize related work in Section III. We then elaborate on FGSS$_k$ and FLSS$_k$, and their properties in Section IV. Following that, we discuss in Section V how to relax several assumptions made in topology control so as to promote the practicality of FGSS$_k$ and FLSS$_k$. Finally, we present a simulation study of FLSS$_k$ in Section VI, and conclude the paper in Section VII.

II. Network Model

In this section, we define the network model[1]. Consider a homogeneous wireless network where each node has the same maximal transmission power, which corresponds to the common transmission range r_{max}. Let the network topology be represented by an undirected simple graph $G = (V(G), E(G))$ in the 2-D plane, where $V(G) = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (vertices) in the network and $E(G)$ is the set of links (edges). A unique *id* (such as an IP/MAC address) is assigned to each node. Here we let $id(v_i) = i$ for simplicity. For ease of presentation, we assume for now that G is geometric, i.e., $E(G) = \{(u,v) : d(u,v) \leq r_{max}, u, v \in V\}$, and $d(u,v)$ is the Euclidean distance between u and v. Note, however, that our algorithms can function correctly for general graphs.

We assume that the wireless channel is symmetric and obstacle-free, and each node has the capability to gather its own location information via, for example, several lightweight localization techniques for wireless networks (the interested reader is referred to, for example, [10] for a summary).

We will further discuss how to relax the above assumptions in Section V.

Definition 1 (Visible Neighborhood). *The visible neighborhood N_u^V is the set of nodes that node u can reach by using the maximum transmission power, i.e., $N_u^V = \{v \in V(G) : d(u,v) \leq r_{max}\}$. For each node $u \in V(G)$, let $G_u^V = (V(G_u^V), E(G_u^V))$ be the induced subgraph of G such that $V(G_u^V) = N_u^V$.*

Definition 2 (Weight Function). *Given two edges $(u_1, v_1), (u_2, v_2) \in E(G)$ and the Euclidean distance function $d(\cdot,\cdot)$, the weight function $w : E \mapsto R$ satisfies:*

$$
\begin{aligned}
& w(u_1,v_1) > w(u_2,v_2) \\
\Leftrightarrow\ & d(u_1,v_1) > d(u_2,v_2) \\
\textit{or}\ & (d(u_1,v_1) = d(u_2,v_2) \\
& \&\& \max\{id(u_1), id(v_1)\} > \max\{id(u_2), id(v_2)\}) \\
\textit{or}\ & (d(u_1,v_1) = d(u_2,v_2) \\
& \&\& \max\{id(u_1), id(v_1)\} = \max\{id(u_2), id(v_2)\} \\
& \&\& \min\{id(u_1), id(v_1)\} > \min\{id(u_2), id(v_2)\}).
\end{aligned}
$$

The weight function w ensures that two edges with different end-nodes have different weights, which can guarantee the unique outcome of the greedy algorithms that will be proposed in Section IV. Also note that $w(u,v) = w(v,u)$.

Definition 3 (Neighbor Set). *Node v is a* neighbor *of node u's under an algorithm ALG (denoted $u \xrightarrow{ALG} v$), if and only if there exists an edge (u,v) in the topology generated by the algorithm. In particular, we use $u \rightarrow v$ to denote the neighbor relation in G. $u \xleftrightarrow{ALG} v$ if and only if $u \xrightarrow{ALG} v$ and $v \xrightarrow{ALG} u$. The* neighbor set *of node u is $N_{ALG}(u) = \{v \in V(G) : u \xrightarrow{ALG} v\}$.*

Definition 4 (Topology). *The topology generated by an algorithm ALG is a directed graph $G_{ALG} = (E(G_{ALG}), V(G_{ALG}))$, where $V(G_{ALG}) = V(G)$, $E(G_{ALG}) = \{(u,v) : u \xrightarrow{ALG} v, u, v \in V(G_{ALG})\}$.*

Definition 5 (Radius). *The radius, R_u, of node u is defined as the distance between node u and its farthest neighbor (in terms of Euclidean distance), i.e, $R_u = \max_{v \in N_{ALG}(u)}\{w(u,v)\}$.*

Definition 6 (Connectivity). *For any topology generated by an algorithm ALG, node u is said to be* connected to *node v (denoted $u \Rightarrow v$) if there exists a path $(p_0 = u, p_1, \ldots, p_{m-1}, p_m = v)$ such that $p_i \xrightarrow{ALG} p_{i+1}, i = 0, 1, \ldots, m-1$, where $p_k \in V(G_{ALG}), k = 0, 1, \ldots, m$. It follows that $u \Rightarrow v$ if $u \Rightarrow p$ and $p \Rightarrow v$ for some $p \in V(G_{ALG})$.*

Definition 7 (Bi-Directionality). *A topology generated by an algorithm ALG is* bi-directional, *if for any two nodes $u, v \in V(G_{ALG})$, $u \in N_{ALG}(v)$ implies $v \in N_{ALG}(u)$.*

[1]Although the model used in this paper is similar to those used in [8] and [9], there exist several subtle differences.

Definition 8 (Bi-Directional Connectivity). *For any topology generated by an algorithm ALG, node u is said to be* bi-directionally connected to *node v (denoted $u \Leftrightarrow v$) if there exists a path $(p_0 = u, p_1, \ldots, p_{m-1}, p_m = v)$ such that $p_i \overset{ALG}{\longleftrightarrow} p_{i+1}, i = 0, 1, \ldots, m-1$, where $p_k \in V(G_{ALG}), k = 0, 1, \ldots, m$. It follows that $u \Leftrightarrow v$ if $u \Leftrightarrow p$ and $p \Leftrightarrow v$ for some $p \in V(G_{ALG})$.*

Deriving network topology consisting of only bi-directional links facilitates link level acknowledgment, which is a critical operation for packet transmissions and retransmissions over unreliable wireless media. Bi-directionality is also an important property for floor acquisition mechanisms such as the RTS/CTS mechanism in IEEE 802.11.

Definition 9 (Addition and Removal). *The* Addition *operation is to add an extra edge (v, u) into G_{ALG} if $(u, v) \in E(G_{ALG})$ and $(v, u) \notin E(G_{ALG})$. The* Removal *operation is to delete any edge $(u, v) \in E(G_{ALG})$ if $(v, u) \notin E(G_{ALG})$.*

Both the *Addition* and *Removal* operations attempt to create a bi-directional topology by removing uni-directional edges or converting uni-directional edges into bi-directional. The resulting topology after *Addition* or *Removal* is alway bi-directional, since the transmission range r_{max} for each node is the same. If the transmission range for each node is not the same, the result of *Removal* is still bi-directional, while the result of *Addition* may not be bi-directional (see [9] for more discussions).

Definition 10 (k-vertex connectivity). *A graph G is k-vertex connected if for any two vertices $v_1, v_2 \in V(G)$, there are k pairwise-internally-vertex-disjoint paths from v_1 to v_2. Or equivalently, a graph is k-vertex connected if the removal of any $k-1$ nodes (and all the related links) does not partition the network.*

k-edge connectivity can be defined accordingly. In wireless networks, we are more concerned with k-vertex connectivity since a k-vertex connected network can survive failure of $k-1$ nodes. In this paper, we will concentrate on k-vertex connectivity and use k-connectivity to refer to k-vertex connectivity for simplicity.

III. Related Work

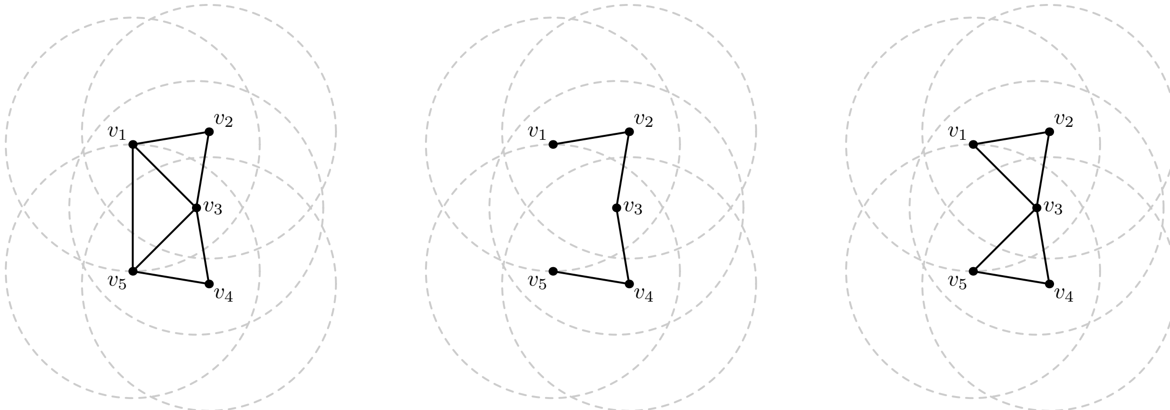
Since the problem of finding a minimum-cost k-connected subgraph is proved to be NP-hard, many approximation algorithms have been proposed (see, for example, [11] and [12] for a summary). Although most topology control algorithms [1]–[9] (see [9] for a summary) do not take fault tolerance into consideration, there have been several research efforts recently on studying the properties of k-connected topologies [13], [14], devising algorithms to construct such topologies [11], [15], or both [16].

Work that studies the properties of fault-tolerant topologies:: Penrose [13] studied k-connectivity in a geometric random graph of n nodes derived by adding an edge between each pair of nodes at most r apart. He proved that the minimum value of r at which the graph is k-connected is equal to the minimum value of r at which the graph has the minimum degree of k, with probability 1 as n goes to infinity. The significance of this result is that it links k-connectivity, a global property of the graph, to node degree, a local parameter. However, the minimum value of r is not given in the paper. Bettstetter [14] also investigated the relation between the minimum node degree and k-connectivity for geometric random graphs. The analytical expression of the required range r_0 for the almost surely k-connected network is derived and verified by simulation.

Li *et al.* [16] extended Penrose's work and gave the lower bound and the upper bound on the minimum value of r at which the graph is k-connected with high probability. The analysis shows that, for a unit-area square region, the probability that the network of n nodes is k-connected is at least $e^{e^{\alpha}}$, if the common transmission radius r_n satisfies $\pi r_n^2 \geq \ln n + (2k-3)\ln\ln n - 2\ln(k-1)! + 2\alpha$, for $k > 0$ and n sufficiently large. Under the homogeneous network assumption (i.e., the maximal transmission power of each node is the same), they also proposed a localized topology control algorithm that preserves k-connectivity. The proposed structure, Yao$_{p,k}$, is based on the Yao structure, and is constructed by having every node u choose k closest neighbors in each of the $p \geq 6$ equal cones around u. Yao$_{p,k}$ is proved to preserve k-connectivity and is a length spanner.

Work that devises algorithms to construct fault-tolerant topologies:: Bahramgiri *et al.* [15] augmented the CBTC algorithm [3] to provide fault tolerance. Specifically, let the directed subgraph of G, $D(\alpha)$, be the output of $CBTC(\alpha)$ algorithm. Note that in $CBTC(\alpha)$, every vertex u increases its transmission power until either the maximum angle between its two consecutive neighbors is at most α or its maximal power is reached. Let $G(\alpha)$ be the result of applying *Removal* on $D(\alpha)$. It is proved in [15] that $G(\frac{2\pi}{3k})$ preserves k-connectivity of G. As the work is extended from the CBTC algorithm, it shares the same assumption of a homogeneous network where the maximal transmission power of each node is the same. However, the assumption of homogeneity may not always hold in practice [9].

Hajiaghayi *et al.* [11] presented three approximation



(a) The original topology is 2-connected.

(b) The minimum spanning tree.

(c) The topology by distributed 2-UPVCS is not 2-connected since the removal of v_3 will result in a disconnected graph.

Fig. 1. An example that shows that the distributed 2-UPVCS algorithm [11] cannot preserve 2-connectivity. When node v_3 executes the distributed 2-UPVCS algorithm, it first finds all the neighbors on the minimum spanning tree, v_2 and v_4. Then it attempts to add an edge between v_2 and v_4 but fails due to their limited transmission power.

algorithms to find the minimum power k-connected subgraph. Two global algorithms are based on existing approaches. The first gives an $O(k\alpha)$-approximation, where α is the best approximation factor for the k-UPVCS problem defined in the paper. The other improves the approximation factor to $O(k)$ for general graphs. The third is a distributed algorithm that gives an $k^{O(c)}$-approximation, where c is the exponent in the propagation model. It first computes the minimum spanning tree (MST) of the input graph by using a distributed algorithm, then it adds a path amongst the neighbors of each node in the returned tree. Since this distributed algorithm is based on the distributed MST algorithm, it is not localized, i.e., it relies on information that is multiple hops away to construct the MST. This implies more maintenance overhead and delay when the topology has to be changed in response to node mobility or failure. Moreover, a closer investigation of the distributed algorithm reveals that the neighbors of a node on the minimum spanning tree may not be able to communicate with each other due to the limited transmission power. As a result, the “arbitrary path connecting neighbors” in the algorithm may not exist in a network of low density. A counter-example in Fig. 1 shows that the 2-UPVCS algorithm does not always preserve 2-connectivity.

Ramanathan and Rosales-Hain [2] presented two centralized algorithms, *CONNECT* and *BICONN-AUGMENT*, to minimize the maximal power used per node while maintaining the (bi)connectivity of the network. Both are simple greedy algorithms that iteratively merge different components until only one remains. Although FGSS_k and FLSS_k bear some similarity to *CONNECT* and *BICONN-AUGMENT* in the way the topology is derived (i.e., different components are iteratively merged until one remains), they differ from the latter in that (1) FGSS_k is more general, i.e., FGSS_k preserves the k-connectivity, while *BICONN-AUGMENT* only preserves 2-connectivity; (2) the correctness of *BICONN-AUGMENT* is only mentioned but not formally proved in [2], while a formal treatment of the correctness of FGSS_k is given in this paper; (3) *CONNECT* and *BICONN-AUGMENT* are both centralized algorithms that require collection and distribution of global information, while FLSS_k is fully decentralized and localized; and (4) *CONNECT* and *BICONN-AUGMENT* operate under the assumption of homogeneous networks, while as will be formally proved in Section V, FGSS_k and FLSS_k can be applied to heterogeneous networks where the maximal transmission power of each node may be different.

IV. Fault-Tolerant Spanning Subgraph

In this section, we first describe a centralized greedy algorithm, *Fault-tolerant Global Spanning Subgraph* (FGSS_k), for fault-tolerant topology control. Then we present its localized version, *Fault-tolerant Local Spanning Subgraph* (FLSS_k).

A. FGSS_k: Fault-tolerant Global Spanning Subgraph

We first present a centralized greedy algorithm, FGSS_k, that builds k-connected spanning subgraphs. Kruskal’s algorithm [17] is a well-known algorithm to construct the minimum spanning tree (1-connected spanning subgraph) of a given graph. FGSS_k is a generalized version of Kruskal's algorithm for $k \geq 2$. The algorithm is given in Algorithm 1.

Algorithm 1 FGSS$_k$

INPUT: $G(V, E)$, a k-connected simple graph;
OUTPUT: $G_k(V_k, E_k)$, a k-connected spanning subgraph of G;

$V_k := V$, $E_k := \emptyset$;
sort all edges in E in an ascending order of weight (as defined in *Definition 2*);
for each edge (u_0, v_0) in the order **do**
  if u_0 is not k-connected to v_0 in G_k **then**
    $E_k := E_k \cup \{(u_0, v_0)\}$;
  else if all nodes are in the same k-connected component **then**
    exit;
  end if
end for

By using network flow techniques [18], a query on whether two vertices are k-connected can be answered in $O(n+m)$ time for any fixed k, where n is the number of vertices and m is the number of edges in the graph. For $k \leq 3$, there also exists $O(1)$ time algorithms [19]. Therefore, the time complexity of FGSS$_k$ is $O(m(n+m))$, and can be improved to $O(m)$ for $k \leq 3$.

Let the path $(u, w_1, w_2, \ldots, w_l, v)$ from u to v be represented by an ordered set p of vertices on the path, i.e., $p = \{u, w_1, w_2, \ldots, w_l, v\}$. Let $S_{uv}(F)$ be a maximal set of pairwise-internally-vertex-disjoint paths from u to v in F. Thus for $\forall p_1, p_2 \in S_{uv}(F)$, we have $p_1 \cap p_2 = \{u, v\}$.

Lemma 1. *Let u_1 and u_2 be two vertices in a k-connected undirected graph F. If u_1 and u_2 are k-connected after the removal of edge (u_1, u_2), then $F - (u_1, u_2)$ is still k-connected.*

Proof: Equivalently, we prove that $F' = F - (u_1, u_2)$ is connected after the removal of any $k-1$ vertices in F'. Consider any two vertices v_1 and v_2 in F'. Without loss of generality, we assume $\{u_1, u_2\} \cap \{v_1, v_2\} = \emptyset$ (other cases can be proved using a similar approach). We now prove that v_1 is still connected to v_2 after removal of the set of any $k-1$ vertices $W = \{w_1, w_2, \ldots, w_{k-1}\}$, where $w_i \in V(F') - \{v_1, v_2\}$. Since F is k-connected, $|S_{v_1v_2}(F)| \geq k$. This is obvious true if (v_1, v_2) is an edge in F. Therefore, we only consider the case where there is no edge from v_1 to v_2 in F.

Let F'' be the resulting graph after (u_1, u_2) and W (and related edges) are removed from F, and let s_1 be the number of paths in $S_{v_1v_2}(F')$ that are broken due to the removal of vertices in W, i.e., $s_1 = |\{p \in S_{v_1v_2}(F') : \exists w \in W, w \in p\}|$. Since the paths in $S_{v_1v_2}(F')$ are pairwise-internally-vertex-disjoint, the removal of any one vertex in W breaks at most one path in the set. Given $|W| = k-1$, we have $s_1 \leq k-1$.

If $|S_{v_1v_2}(F')| \geq k$, then $|S_{v_1v_2}(F'')| \geq |S_{v_1v_2}(F')| - s_1 \geq 1$, i.e., v_1 is still connected to v_2 in F''. Now we consider the case where $|S_{v_1v_2}(F')| < k$. This occurs only when the removal of (u_1, u_2) breaks one path $p^0 \in S_{v_1v_2}(F)$. Without loss of generality, let the order of vertices on p^0 be v_1, u_1, u_2, v_2. Since the removal of (u_1, u_2) reduces the number of pairwise-internally-vertex-disjoint paths between v_1 and v_2 by at most one, $|S_{v_1v_2}(F) - \{p^0\}| \geq k-1$. Hence $|S_{v_1v_2}(F')| = k-1$. Now we consider two cases:

1) $s_1 < k-1$: $|S_{v_1v_2}(F'')| \geq |S_{v_1v_2}(F')| - s_1 \geq 1$, i.e., v_1 is still connected to v_2 in F''.
2) $s_1 = k-1$: hence every vertex in W belongs to some path in $S_{v_1v_2}(F')$. Since p^0 is internally-disjoint with all paths in $S_{v_1v_2}(F')$, we have $p^0 \cap W = \emptyset$. Thus v_1 is connected to u_1 and u_2 is connected to v_2 in F''. Let s_2 be the number of paths in $S_{u_1u_2}(F')$ that are broken due to the removal of vertices in W, i.e., $s_2 = |\{p \in S_{u_1u_2}(F') : \exists w \in W, w \in p\}|$. Since $|S_{u_1u_2}(F')| \geq k$ and $s_2 \leq k-1$, $|S_{u_1u_2}(F'')| \geq 1$, i.e., u_1 is still connected to u_2 in F''. Therefore, v_1 is still connected to v_2 in F''.

We have proved that for any two vertices $v_1, v_2 \in F'$, v_1 is connected to v_2 after the removal of any $k-1$ vertices from $F' - \{v_1, v_2\}$. Therefore, F' is k-connected. □

Lemma 2. *Let G and G' be two undirected simple graphs such that $V(G) = V(G')$. If G is k-connected, and every edge $(u, v) \in E(G) - E(G')$ satisfies that u is k-connected to v in $G - \{(u_0, v_0) \in E(G) : w(u_0, v_0) \geq w(u, v)\}$, then G' is also k-connected.*

Proof: Let $E = E(G) - E(G') = \{(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m)\}$ be an set of edges in an descending order of weight, i.e., $w(u_1, v_1) \geq w(u_2, v_2) \geq \ldots \geq w(u_m, v_m)$. We define a series of graphs that are subgraphs of G: $G^0 = G$, and $G^i = G^{i-1} - (u_i, v_i)$, $i = 1, 2, \ldots, m$. Now we prove by induction.

1) *Base*: $G^0 = G$ is k-connected.
2) *Induction*: If G^{i-1} is k-connected, we prove that G^i is k-connected, where $i = 1, 2, \ldots, m$. Since $G - \{(u, v) \in E(G) : w(u, v) \geq w(u_i, v_i)\} \subseteq G^{i-1} - (u_i, v_i)$, u_i is k-connected to v_i in $G^{i-1} - (u_i, v_i)$. Applying Lemma 1 to G^{i-1}, we can prove that G^i is still k-connected.

Now we have proved that G^m is k-connected. Since

$E(G^m) \subseteq E(G')$, G' is also k-connected. □

Theorem 1. *FGSS$_k$ can preserve the k-connectivity of G, i.e., G_k is k-connected if G is k-connected.*

Proof: Since edges are inserted into G_k in an ascending order, whether u is k-connected to v at the moment before (u, v) is inserted depends only on the edges of smaller weight. Therefore, every edge $(u, v) \in E_0 = E(G) - E(G_k)$ satisfies that u is k-connected to v in $G - \{(u, v) \in E(G) : w(u, v) > w(u_0, v_0)\}$. we can prove that G_k preserves the k-connectivity of G by applying Lemma 2 to G_k. □

Let $\rho(F)$ be the largest radius of all nodes in F, i.e., $\rho(F) = \max_{u \in V(F)}\{R_u\}$. Now we prove that FGSS$_k$ achieves the min-max optimality, i.e., let $SS_k(G)$ be the set of all k-connected spanning subgraphs of G, then $\rho(G_k) = \min\{\rho(F) : F \in SS_k(G)\}$. This optimality is proved in [2] for $k = 2$, we extend the result to arbitrary k.

Theorem 2. *The maximum transmission radius (or equivalently, power) among all nodes is minimized by FGSS$_k$, i.e., $\rho(G_k) = \min\{\rho(F) : F \in SS_k(G)\}$.*

Proof: Suppose G is k-connected. By Theorem 1 G_k is also k-connected. Let (u, v) be the last edge that is inserted into G_k, we have $w(u, v) = \max_{(u_0,v_0) \in E(G_k)}\{w(u_0, v_0)\}$ and $R_u = R_v = \rho(G_k)$. Let $G'_k = G_k - (u, v)$, we have $|S_{uv}(G'_k)| < k$; otherwise according to Algorithm 1, (u, v) should not be included in G_k. Now consider a graph $H = (V(H), E(H))$, where $V(H) = V(G)$ and $E(H) = \{(u_0, v_0) \in E(G) : w(u_0, v_0) < w(u, v)\}$. If we can prove that H is not k-connected, we will be able to conclude that any $F \in SS_k(G)$ must have at least one edge equal to or longer than (u, v), which means $\rho(G_k) = \min\{\rho(F) : F \in SS_k(G)\}$.

Now we prove by contradiction that H is not k-connected. Assume H is k-connected and hence $|S_{uv}(H)| \geq k$. We have $E(H) \nsubseteq E(G'_k)$; otherwise, $|S_{uv}(G'_k)| \geq |S_{uv}(H)| \geq k$. Therefore, $E_0 = E(H) - E(G'_k) \neq \emptyset$. Since edges are inserted into G'_k in an ascending order, $\forall(u_1, v_1) \in E_0$ satisfies that u_1 is k-connected to v_1 in $H - \{(u_0, v_0) \in E(H) : w(u_0, v_0) \geq w(u_1, v_1)\}$. By Lemma 2, we can prove that u is still k-connected to v after the removal of all edges in E_0. This means $|S_{uv}(G'_k)| \geq k$, which is a contradiction. □

The min-max optimality of FGSS$_k$ is an important feature. Let the network lifetime be defined as the time it takes for the first node to deplete its energy. If we assume a static network in which each node has the same energy and may send data to any other node, then the network lifetime is approximately the same as the lifetime of the node that uses the maximum radius among all nodes. By minimizing the maximum radius (and transmission power), FGSS$_k$ achieves the maximum network lifetime.

FGSS$_k$ is a centralized algorithm that requires the knowledge of global information. Since there is, in general, no central authority in a wireless multi-hop network, it is very difficult to collect and distribute global information, and by doing so, the major objective of topology control — power saving — may be defeated. It is more desirable to devise distributed algorithms where each node makes its decision based on the information collected. To be less susceptible to mobility, it is also desirable that the algorithm depends only on the information locally collected, e.g., within one hop, and thus incurs less message overhead/delay in collecting information. In the next section, we will devise a localized algorithm based on FGSS$_k$.

B. FLSS$_k$: Fault-tolerant Local Spanning Subgraph

In this section, we present a localized, fault tolerant topology control algorithm, called *Fault-tolerant Local Spanning Subgraph* (FLSS$_k$). The topology is derived by having each node build its neighbor set and adjust its transmission power based on locally collected information. The algorithm consists of three phases [8]:

- I *Information Collection*: each node u collects local information of neighbors, such as their positions and *ids*, and identifies the *Visible Neighborhood* N_u^V.
- II *Topology Construction*: each node defines, based on the information in N_u^V, the proper list of neighbors for the final topology.
- III *Construction of Topology with Only Bi-Directional Links* (Optional): each node adjusts its list of neighbors to make sure that all the edges are bi-directional.

In what follows we elaborate on each of the three phases.

1) Information Collection: The information needed by each node u is its visible neighborhood N_u^V. This can be obtained locally by having each node broadcast periodically a Hello message using the maximal transmission power. The information contained in a Hello message includes at least the node *id* and the position of the node. These periodic messages can be sent either in the data channel or in a separate low-bandwidth control channel. The Hello messages can also be combined with those that are already employed in most ad hoc routing protocols. In addition, each node can piggy-back its location information in data packets to reduce the number of Hello exchanges.

2) Topology Construction: Given the visible neighborhood N_u^V, each node u builds its local spanning subgraph $S_u = (V(S_u), E(S_u))$ over N_u^V using algorithm FGSS$_k$ given in Section IV-A, with one modification on line 6–7 that the algorithm stops if u is k-connected to every other node in N_u^V.

Definition 11 (Neighbor Relation in FLSS$_k$). *In Fault-tolerant Local Spanning Subgraph (FLSS$_k$), node v is a neighbor of node u's, denoted $u \xrightarrow{FLSS} v$, if and only if $(u,v) \in E(S_u)$. That is, v is a neighbor of u's if and only if v is on u's local spanning subgraph S_u, and is one hop away from u.*

The network topology under FLSS$_k$ is all the nodes in $V(G)$ and their individually perceived neighbor relations. Note that the topology is *not* a simple superposition of all local spanning subgraphs. In addition, the neighbor relation defined above is not symmetric, i.e., $u \xrightarrow{FLSS} v$ does not necessarily imply $v \xrightarrow{FLSS} u$.

Definition 12 (Topology G_{FLSS}). *The topology, G_{FLSS}, derived under FLSS$_k$ is a directed graph $G_{FLSS} = (V_{G_{FLSS}}, E_{G_{FLSS}})$, where $V_{G_{FLSS}} = V(G)$, $E_{G_{FLSS}} = \{(u,v) : u \xrightarrow{FLSS} v, u, v \in V(G)\}$.*

3) Construction of Topology with Only Bi-Directional Edges: As mentioned previously, some links in G_{FLSS} may be uni-directional. We can apply *Addition* or *Removal* to enforce every edge to be bi-directional. The new topologies G^+_{FLSS} and G^-_{FLSS} can be defined respectively.

Definition 13 (Topology G^+_{FLSS}). *The topology, G^+_{FLSS}, is a undirected graph $G^+_{FLSS} = (V(G^+_{FLSS}), E(G^+_{FLSS}))$, where $V(G^+_{FLSS}) = V(G_{FLSS})$, $E^+_{FLSS} = \{(u,v) : (u,v) \in E(G_{FLSS})$ or $(v,u) \in E(G_{FLSS})\}$.*

Definition 14 (Topology G^-_{FLSS}). *The topology, G^-_{FLSS}, is a undirected graph $G^-_{FLSS} = (V(G^-_{FLSS}), E(G^-_{FLSS}))$, where $V(G^-_{FLSS}) = V(G_{FLSS})$, $E^-_{FLSS} = \{(u,v) : (u,v) \in E(G_{FLSS})$ and $(v,u) \in E(G_{FLSS})\}$.*

4) Properties of FLSS$_k$: We are now in a position to state and formally prove several properties of FLSS$_k$.

Theorem 3 (Connectivity of FLSS$_k$). *If G is k-connected, then G_{FLSS}, G^+_{FLSS} and G^-_{FLSS} are all k-connected.*

Proof: We only need to prove that G^-_{FLSS} preserves the k-connectivity of G, for $E(G^-_{FLSS}) \subseteq E(G_{FLSS}) \subseteq E(G^+_{FLSS})$. Since G^-_{FLSS} is bi-directional, we can treat it as an undirected graph. Let $E = E(G) - E(G^-_{FLSS})$. For any edge $e = (u,v) \in E$, at least one of (u,v) and (v,u) was not in G_{FLSS}, since $e \notin E(G^-_{FLSS})$. Without loss of generality assume (u,v) was not in G_{FLSS}. Thus in the process of local topology construction of node u, u was already k-connected to v before (u,v) was inspected. Since edges are inserted in an ascending order, whether u is k-connected to v at the moment before (u,v) is inspected depends only on the edges of smaller weights. Therefore, u is k-connected to v in $G - \{(u_0,v_0) \in E(G) : w(u_0,v_0) > w(u,v)\}$. Let $G' = G^-_{FLSS}$, we can conclude that G^-_{FLSS} is k-connected by Lemma 2. □

Definition 15 (Strictly Localized Algorithms). *An algorithm is strictly localized if its operation on any node u is based only on the information that is originated from the nodes in N_u^V.*

For any node u running a strictly localized algorithm, the information which u may rely on is quite limited. For instance, u and $v \in N_u^V$ may be not k-connected in G_u^V, but k-connected in G (which is impossible for u to know). Therefore, u has to keep local "connectedness" as much as possible, i.e., if u and v are not k-connected before edge (u,v) is considered, (u,v) has to be in the final topology constructed by u. Fig. 1 gives a good example. For node v_5, $N_{v_5}^V = \{v_1, v_3, v_4, v_5\}$. Before edge (v_5, v_1) is considered, v_5 is not yet 2-connected to v_1. Therefore, Node v_5 has to choose v_1 as its neighbor in the final topology, to preserve the 2-connectivity of $G_{v_5}^V$; otherwise, the resulting topology is not 2-connected as shown in Fig 1(c).

Let $LSS_k(G)$ be the set of all k-connected spanning subgraphs of G that are constructed by strictly localized algorithms. Now we prove that FLSS achieves the min-max optimality among all strictly localized algorithms, i.e., $\rho(G_{FLSS}) = \min\{\rho(F) : F \in LSS_k(G)\}$.

Theorem 4. *Among all strictly localized algorithms, FLSS$_k$ minimizes the maximum transmission radius (or power) of nodes in the network, i.e., $\rho(G_{FLSS}) = \min\{\rho(F) : F \in LSS_k(G)\}$.*

Proof: Suppose G is k-connected. Let (u,v) be the last edge inserted into G_{FLSS}. We have $w(u,v) = \max_{(u_0,v_0)\in E(G_{FLSS})}\{w(u_0,v_0)\}$ and $R_u = R_v = \rho(G_{FLSS})$. Let G_0 be the induced subgraph of G_{FLSS} where $V(G_0) = N_u^V$, and let $G'_0 = G_0 - \{(u,v)\}$. We have $|S_{uv}(G'_0)| < k$; otherwise (u,v) should not be included in G_0. Also define $H_0 = (V(H_0), E(H_0))$, where $V(H_0) = V(G_u^V)$ and $E(H_0) = \{(u_0,v_0) \in E(G_u^V) : w(u_0,v_0) < w(u,v)\}$.

To prove that H_0 is not k-connected, we replace G, G_k, G'_k, and H with G_u^V, G_0, G'_0, and H_0 respectively, and follow the corresponding proof in Theorem 2. After proving H_0 is not k-connected, we consider the following cases:

1) u is k-connected to v in G_u^V: since H_0 is not k-connected, any $F \in LSS_k(G)$ should have had at least one edge equal to or longer than (u, v),
2) u is not k-connected to v in G_u^V: to preserve the connectedness as much as possible, any $F \in LSS_k(G)$ should have included (u, v);

In both cases, $\rho(F) \geq \rho(G_u^V) = \rho(G_{FLSS})$, which means $\rho(G_{FLSS}) = \min\{\rho(F) : F \in LSS_k(G)\}$. □

V. Relaxation of Several Assumptions

Although the assumptions stated in Section II are widely used in existing topology control work, some of them are made for ease of analysis and may not be practical. In this section, we discuss how to relax these assumptions in FGSS$_k$ and FLSS$_k$ so as to promote their use in realistic situations.

A. Relaxing the Homogeneous Network Assumption

As mentioned in [9], the assumption of homogeneous nodes does not always hold in practice, since even devices of the same type may have slightly different maximal transmission power, let alone the fact that devices of different types possess dramatically different capabilities. The original topology of a heterogeneous network, instead of being defined as an undirected graph in Section II, should be defined as a directed graph $G = (V(G), E(G))$ in the 2-D plane. Here G is not necessarily a geometric graph. Let r_u denote the maximal transmission range of u.

First we prove that FGSS$_k$ preserves k-connectivity and is min-max optimal even in heterogeneous networks. The following results correspond to Lemma 1, Lemma 2, Theorem 1 and Theorem 2, respectively. The proof is literally the same as that in Section IV-A, except that now we consider directed graphs consisting of directed edges. This resemblance is by no means a coincidence, since we actually consider more general cases when we proved them in Section IV-A.

Lemma 1*. *Let u_1 and u_2 be two vertices in a k-connected* directed *graph F. If u_1 are k-connected to u_2 after the removal of edge (u_1, u_2), then $F - (u_1, u_2)$ is still k-connected.*

Lemma 2*. *Let G and G' be two* directed *simple graphs such that $V(G) = V(G')$. If G is k-connected, and every edge $(u, v) \in E(G) - E(G')$ satisfies that u is k-connected to v in $G - \{(u_0, v_0) \in E(G) : w(u_0, v_0) \geq w(u, v)\}$, then G' is also k-connected.*

Theorem 1*. *FGSS$_k$ can preserve the k-connectivity in heterogeneous networks, i.e., G_k is k-connected if G is k-connected, where G is a directed graph.*

Theorem 2*. *The maximum transmission radius (or power) among all nodes is minimized by FGSS$_k$, i.e., $\rho(G_k) = \min\{\rho(F) : F \in SS_k(G)\}$, where G is a directed graph and $SS_k(G)$ is the set of all k-connected spanning subgraphs of G.*

To show that FLSS$_k$ can be applied to heterogeneous networks, we first define the counterpart of Definition 1 in heterogeneous networks:

Definition 16 (Reachable Neighborhood). *The reachable neighborhood N_u^R is the set of nodes that node u can reach by using the maximum transmission power, i.e., $N_u^R = \{v \in V(G) : (u, v) \in E(G)\}$. For each node $u \in V(G)$, let $G_u^V = (V(G_u^V), E(G_u^V))$ be the induced subgraph of G such that $V(G_u^V) = N_u^V$.*

We now prove the k-connectivity of FLSS$_k$. Note that G_{FLSS}^- can no longer preserve k-connectivity for heterogeneous networks.

Theorem 3* (Connectivity of FLSS$_k$). *If G is k-connected, then G_{FLSS}, G_{FLSS}^+ are both k-connected.*

Proof: We only need to prove that G_{FLSS} preserves the k-connectivity of G, since $E(G_{FLSS}) \subseteq E(G_{FLSS}^+)$. Let $E = E(G) - E(G_{FLSS})$. For any edge $e = (u, v) \in E$, it is not in $E(G_{FLSS})$ because in the process of local topology construction of node u, u was already k-connected to v before (u, v) was inserted. Since edges are inserted in an ascending order, whether u is k-connected to v at the moment before (u, v) is inserted depends only on the edges of smaller weights. Therefore, u is k-connected to v in $G - \{(u_0, v_0) \in E(G) : w(u_0, v_0) > w(u, v)\}$. Let $G' = G_{FLSS}$, we can conclude that G_{FLSS} is k-connected by Lemma 2*. □

The min-max optimality of FLSS$_k$ can be proved in a straightforward manner:

Theorem 4*. *Among all strictly localized algorithms, FLSS$_k$ minimizes the maximum transmission radius (or power) of nodes in the network, i.e., $\rho(G_{FLSS}) = \min\{\rho(F) : F \in LSS_k(G)\}$, where G is a directed graph.*

B. Relaxing the Obstacle Free Propagation Model

We assume in Section II an obstacle free propagation model. In this section, we state this assumption can be readily dismissed.

We have previously assumed that the original network topology, G, is a general directed or undirected graph.

The information needed by $\text{FGSS}_k/\text{FLSS}_k$ is the edges that exist in G. An edge that was not formed in the network, whether because the two endpoints of the edge are not within the transmission range of each other or because there exists a obstacle in between, does not have any impact on the results of $\text{FGSS}_k/\text{FLSS}_k$. Also from the point of view of a node u, it only knows whether or not there exists a link between itself and another node v, but has no way to infer the exact reason (either (a) the two nodes are not within transmission range of each other; or (b) the obstacle between the two nodes blocks the communication). As long as the original topology (which has taken into consideration of obstacles in the network) is k-connected, $\text{FGSS}_k/\text{FLSS}_k$ can be applied to provide a min-max optimal solution to preserve k-connectivity. Therefore, the assumption of obstacle-free wireless channel can be relaxed without any modification to FLSS_k.

C. Relaxing the Requirement on Position Information

It is assumed in Section II that each node is equipped with the capability of gathering its own location information. In this subsection, we relax this requirement.

As mentioned in Section V-B, what is required by $\text{FGSS}_k/\text{FLSS}_k$ is the information of all the existing edges in the network. $\text{FGSS}_k/\text{FLSS}_k$ can operate without the knowledge of specific positions of nodes in the network as follows. If each node knows it own position, either by special hardware or localization service provided by the network, it will be fairly easy to gather the knowledge of existing edges. Otherwise, we augment FLSS_k with an extra run of information dissemination. First, each node periodically broadcasts, using its maximal transmission power, a very short Hi message which includes only its node *id* and its transmission power. Upon receiving such a message from a neighbor node v, each node u estimates the length of the edge (v, u) based on the attenuation incurred in the transmission. Denote the set of edges incident at u as $E_u^T = \{(v, u) : v \in N_u^W\}$). After u has collected the information on E_u^T, u can then broadcast this information in an Edge message. Each node will be able to construct the edge set $E(N_u^W)$ based on the Edge messages received from all of its neighbors.

Although this solution may incur more communication and computation overhead, and make FLSS_k less "localized", it eliminates the need for the position information, and thus is better suited for wireless sensor networks where the cost and the energy consumption should be kept as low as possible.

D. Relaxing the Assumption of Perfect Omni-directional Antenna Patterns

Many topology control algorithms assume a Unit Disk Graph (UDG) model, i.e., the antenna pattern of a wireless device is a perfect disk. This is also the underlying assumption for algorithms that use explicit channel propagation models. Since the same models are applied to all directions, the antenna patterns have to be isotropic, which in turn implies that the transmission area is a perfect disk.

The antenna model does not affect FGSS_k, since FGSS_k is a centralized algorithm that can be given perfect, global information. For FLSS_k, the antenna pattern model influences the manner in which the information on N_u^W can be collected. Given an arbitrary antenna pattern, we can simply employ the information dissemination technique in Section V-C. It is obvious that the information dissemination technique does not rely on any specific antenna pattern, except that the estimation of edge length becomes quite difficult. This is due to the fact that the antenna pattern is not necessarily isotropic, i.e., the power attenuation may vary in different directions. We are currently investigating how to address this problem.

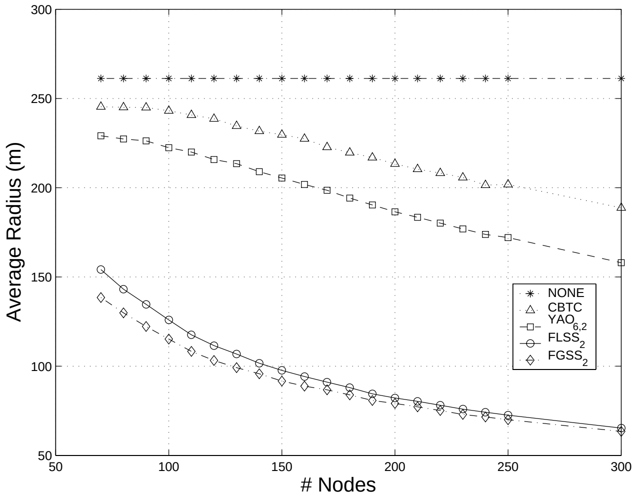
VI. Performance Evaluation

In this section, we evaluate the performance of FLSS_k against two distributed/localized algorithms, CBTC [15], $\text{Yao}_{p,k}$ [16], and Hajiaghayi's algorithm [11] with respect to several metrics via simulation. The parameter p in $\text{Yao}_{p,k}$ is set to 6 in order to minimize the average power [16].

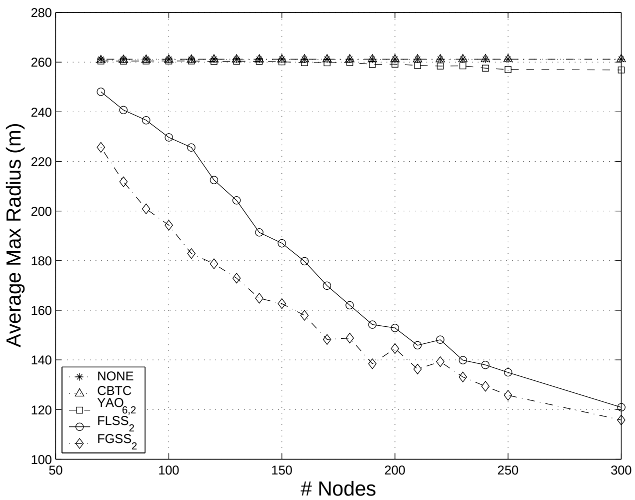
For the sake of fair comparison, we have to use several common assumptions among all the algorithms, i.e., the UDG model. The performance of the centralized algorithm FGSS_k is also shown as a baseline. As will be shown in the following discussion, the performance of FLSS_k is only slightly worse than that of FGSS_k.

In the first set of simulation, we assume that nodes are uniformly distributed in a $1000m \times 1000m$ region. The transmission range for all nodes is $261.195m$. We vary the number of nodes in the region from 70 to 300. Each data point is the average of 50 simulation runs.

Radius and average link length*:* Fig. 2 shows the average radius and the average maximum radius for the topologies derived under NONE (with no topology control), CBTC, $\text{YAO}_{6,2}$, FLSS_2 and FGSS_2. The average radius of FLSS_2 is much smaller than that of other algorithms. This implies nodes in FLSS_2 use much less average power to transmit. The average maximum radius of CBTC or $\text{YAO}_{6,2}$ comes very close



(a) Average radius.



(b) Average maximum radius.

Fig. 2. Comparison of NONE, CBTC, $YAO_{6,2}$ $FLSS_2$ and $FGSS_2$ with respect to radius ($k = 2$).

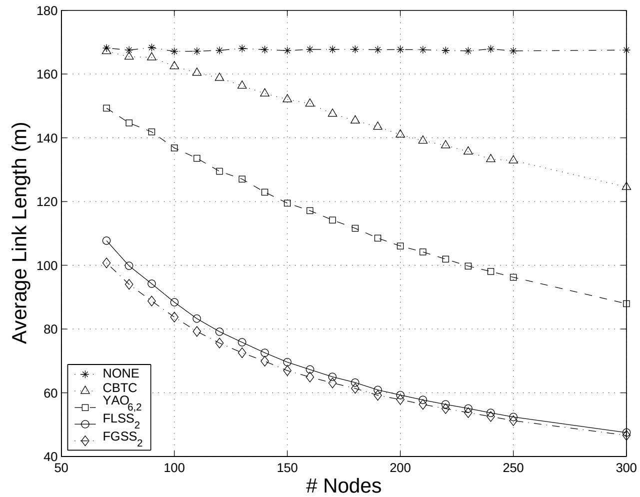


Fig. 3. Comparison of NONE, CBTC, $YAO_{6,2}$ $FLSS_2$ and $FGSS_2$ with respect to average link length ($k = 2$).

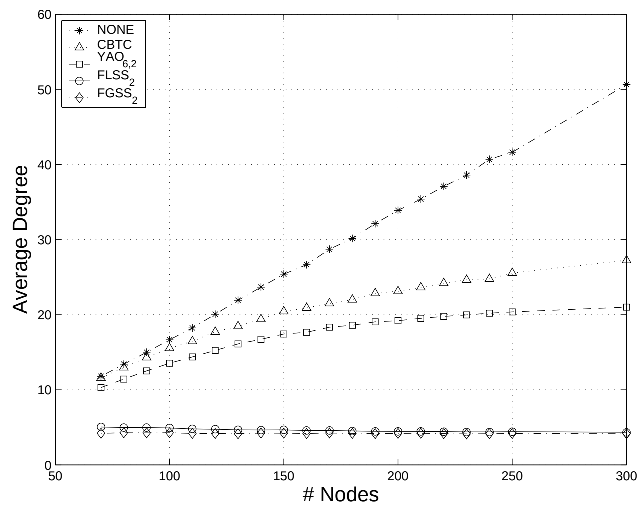
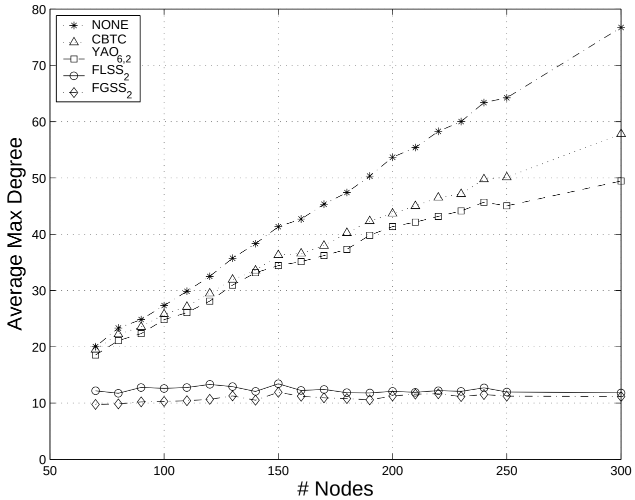


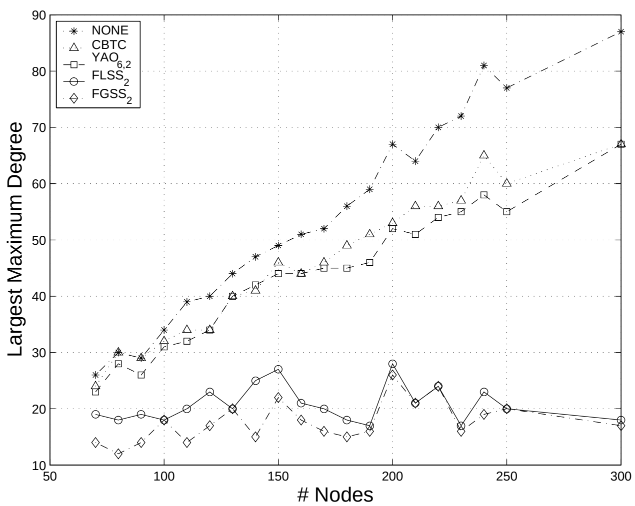
Fig. 4. Comparison of NONE, CBTC, $YAO_{6,2}$ $FLSS_2$ and $FGSS_2$ with respect to average degree ($k = 2$).

to that of NONE. This means the CBTC and $YAO_{6,2}$ cannot really improve the network lifetime. In contrast, the average maximum radius of $FLSS_2$ is significantly smaller, which implies that FLSS can greatly prolong the network lifetime. The average link length of the topologies derived under different algorithms is shown in Fig. 3. $FLSS_2$ outperforms the others, and moreover its performance is very close to that of the centralized algorithm $FGSS_2$.

Node degree*:* We also compare the average node degree of the topologies derived under different algorithms, where the node degree is defined as the number of nodes within the transmission radius of a node. The node degree is an indication of the level of MAC interference (and hence the extent of spatial reuse), i.e., the smaller the node degree of a node, the less number of nodes its transmission may interfere with, and potentially affect. Fig. 4 shows the average node degree of the topologies derived under CBTC, $YAO_{6,2}$, $FLSS_2$ and $FGSS_2$. The average degree under NONE increases almost linearly with the number of nodes. The average degree under CBTC and $YAO_{6,2}$ also increases as the number of nodes increases. In contrast, the average degree under $FGSS_2$ and $FLSS_2$ actually decreases. Fig. 5 gives the average maximum node degree and the largest value of the maximum node degrees among all the nodes in the topologies derived under CBTC, $YAO_{6,2}$, $FLSS_2$, and $FGSS_2$. Both values under $FGSS_2$/$FLSS_2$ are significantly smaller than those under NONE/CBTC/$YAO_{6,2}$. All the results show that $FLSS_2$ can achieve better spatial reuse, and the performance improvement becomes even more prominent when the network density becomes higher.

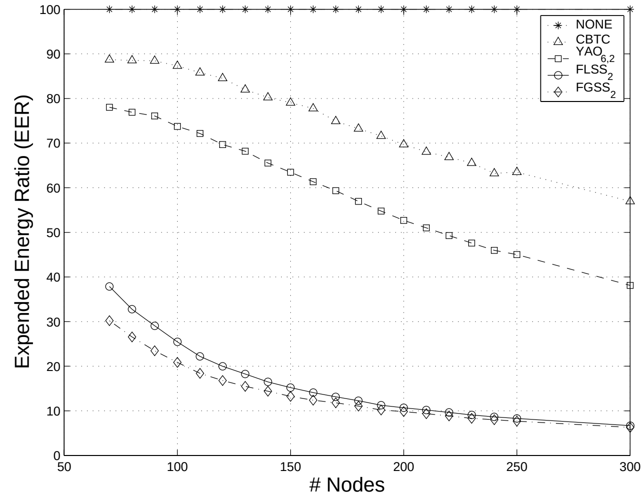


(a) Average maximum degree.

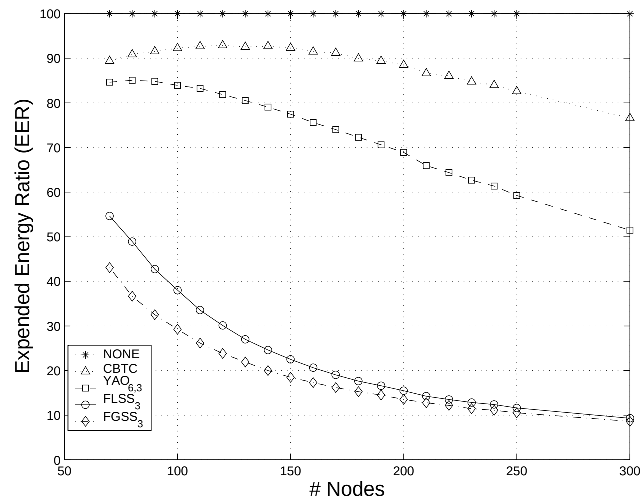


(b) Largest maximum degree.

Fig. 5. Comparison of NONE, CBTC, $\text{YAO}_{6,2}$ FLSS_2 and FGSS_2 with respect to the maximum degree ($k = 2$).



(a) $k = 2$.



(b) $k = 3$.

Fig. 6. Comparison of NONE, CBTC, YAO, FLSS and FGSS with respect to expended energy ratio (EER).

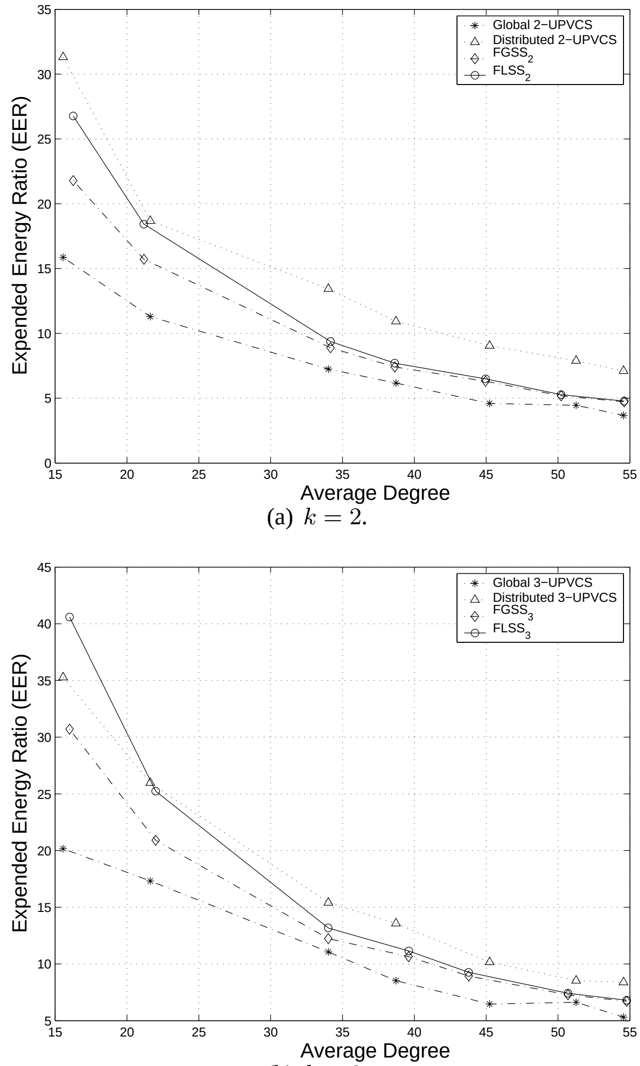
Energy saving*:* We compare the various algorithms with respect to the average expended energy ratio (EER) defined in [11] as

$$EER = \frac{E_{ave}}{E_{max}} \times 100,$$

where E_{ave} is the average transmission power over all the nodes in the network, and E_{max} is the maximal transmission power that can reach the transmission range of $261.195m$. Here we use the free-space propagation model to calculate the transmission power. Fig. 6 gives the comparison results for both $k = 2$ and $k = 3$. FLSS_k clearly has the advantage.

In the second set of simulations, we compare FGSS_k and FLSS_k with both the distributed and centralized versions of k-UPVCS [11] in terms of EER, for both $k = 2$ and $k = 3$. The simulation is conducted in a similar setting to that in [11]. Note that we are unable to accurately control the density of the original graph (with the maximal transmission range), thus we compare the algorithms under the topology of roughly the same average degree. Fig. 7 gives the comparison results, where FGSS and FLSS are shown to perform better than the distributed version of k-UPVCS in almost every setting, and worse than the global version of k-UPVCS.

Tradeoff between topology robustness and performance*:* In the third set of simulations, we compare FLSS_2 and FLSS_3 against a localized topology control algorithm, LMST [8], that renders 1-connected subgraphs. As shown in Fig. 8, FLSS_k renders topologies that have larger average degrees, longer average radii, and longer average maximum radii, and consume more power than LMST. However, the topologies are also more robust and are resilient to $k-1$ failures (the same



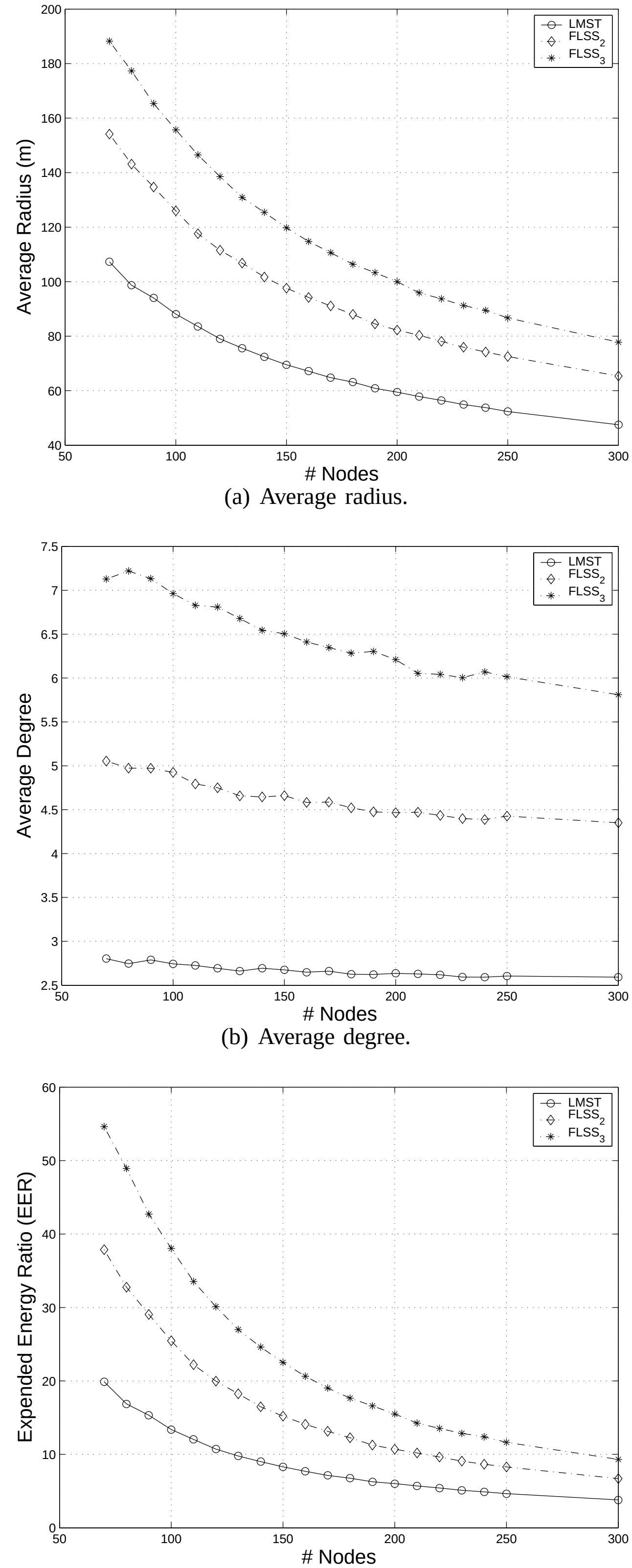
(a) $k = 2$.

(b) $k = 3$.

Fig. 7. Comparison of FGSS, FLSS, and the global and distributed versions of k-UPVCS ([11]) with respect to EER.

conclusion can be drawn as k increases). This shows the tradeoff between the robustness of the topology and the other performance metrics (e.g., power consumption, network lifetime, spatial reuse, and MAC level interference).

Finally we compare LMST, FLSS_2 and FLSS_3 with respect to network capacity and energy efficiency. In this set of simulation, n nodes are randomly distributed in a $1500m \times 200m$ region, with half of them being sources and the other half being destinations. To observe the effect of spatial reuse, the deployment region should be large enough as compared to the transmission/interference range. To reduce the number of nodes and to expedite simulation, we use a rectangular region, rather than a square region. In the simulation, the propagation model is the two-ray ground model, the

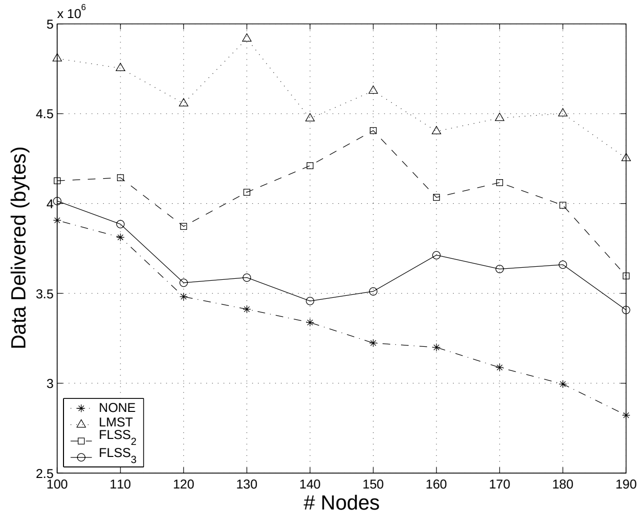


(a) Average radius.

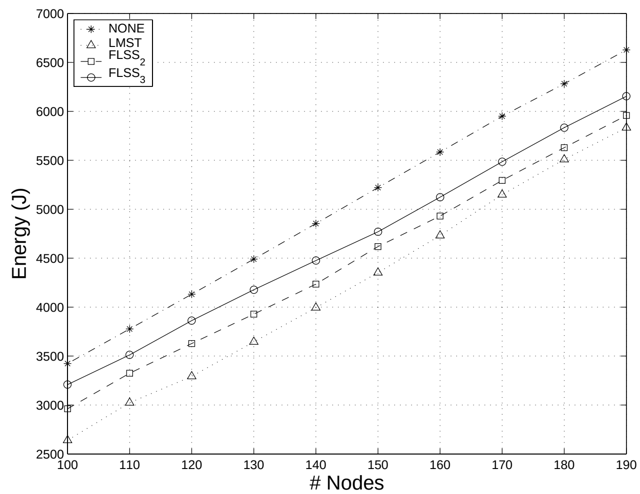
(b) Average degree.

(c) EER.

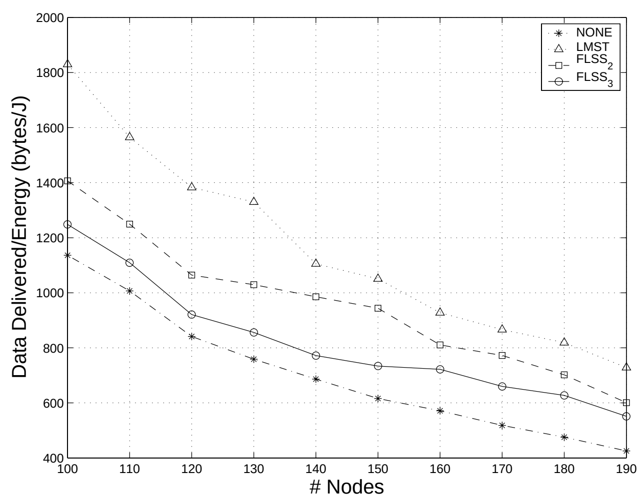
Fig. 8. Comparison of LMST, FLSS_2 and FLSS_3 with respect to the average radius, average node degree and EER.



(a) Total throughput (bytes).



(b) Total energy consumption (Joule).



(c) Energy efficiency (bytes/J).

Fig. 9. Comparison of LMST, FLSS_2 and FLSS_3 with respect to the network capacity and the energy efficiency under CBR traffic.

MAC protocol is IEEE 802.11, the routing protocol is AODV, and the traffic sources are CBR (note that results obtained by using TCP traffic with bulk FTP sources exhibit similar trends, and hence are not reported here). The start time of each connection is chosen randomly from $[0s, 10s]$. Each simulation run lasts for 100 seconds.

We compare the total amount of data delivered (in bytes, Fig. 9(a)), the total energy consumption (in Joule, Fig. 9(b)), and the energy efficiency (in bytes/J, Fig. 9(c)). It can be observed that with the increase in the level of network connectivity (in the order of LMST, FLSS_2, FLSS_3, NONE), the total throughput decreases, the total energy consumption increases, and the energy efficiency decreases. This result again demonstrates the trade-off between the robustness (or routing redundancy) and the network capacity/energy efficiency.

VII. Conclusions

In this paper, we have taken into account of fault tolerance in topology control in wireless ad-hoc networks and sensor networks. We first present a centralized greedy algorithm, FGSS_k, to find a k-connected spanning subgraph of the topology. We prove that FGSS_k preserve k-connectivity and is min-max optimal among all centralized algorithms. By min-max optimality we mean that the maximum transmission power (radius) used among all the nodes is minimized. The min-max optimality is critical in prolonging the network lifetime. Since localized algorithms rely only on the information that can be locally collected and are hence more power-efficient when the overhead incurred in information collection is considered, we propose, based on FGSS_k, a localized topology control algorithm FLSS_k. We prove FLSS_k preserves k-connectivity and bi-directionality, and is min-max optimal among all strictly localized algorithms.

After the theoretical base is laid and FLSS_k devised, we proceed to examine several widely used assumptions in topology control, e.g., uniform maximal transmission power, obstacle-free communication channel, capability of obtaining position information, and perfect antenna pattern, relax these assumptions for FGSS_k and FLSS_k so as to promote their practicality.

Although FLSS_k outperforms other localized algorithms in random networks in terms of power consumption, it does not give any performance bound on power consumption as many centralized algorithms do [12] (in contrast, the distributed version of Hajiaghayi's algorithm [11] is shown to give a performance bound, but does not preserve k-connectivity as shown in Fig. 1). The dominating reason for the lack of a performance guarantee is that FLSS_k is greedy and highly localized.

Although it performs very well in most cases, we highly suspect that the information available within each node's transmission range is too limited to upper-bound the performance under some rare, extreme cases. As part of our future research, we will extend FLSS_k to utilize more information in the network so as to provide some performance bound.

REFERENCES

[1] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[2] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000, pp. 404–413.

[3] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. ACM Symposium on Principles of Distributed Computing (PODC)*, Newport, RI, USA, Aug. 2001, pp. 264–273.

[4] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," in *Proc. of European Wireless 2002, Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florence, Italy, Feb. 2002, pp. 156–162.

[5] V. Kawadia and P. Kumar, "Power control and clustering in ad hoc networks," in *Proc. IEEE INFOCOM 2003*, San Francisco, CA, USA, Apr. 2003.

[6] S. A. Borbash and E. H. Jennings, "Distributed topology control algorithm for multihop wireless networks," in *Proc. 2002 World Congress on Computational Intelligence (WCCI)*, Honolulu, HI, USA, May 2002.

[7] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed construction of planar spanner and routing for ad hoc networks," in *Proc. IEEE INFOCOM 2002*, New York, NY, USA, June 2002.

[8] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *Proc. IEEE INFOCOM 2003*, San Francisco, CA, USA, Apr. 2003.

[9] N. Li and J. C. Hou, "Topology control in heterogeneous wireless networks: problems and solutions," in *Proc. IEEE INFOCOM 2004*, Hong Kong, Mar. 2004 (the version in the conference proceedings has a technical error, and an update version of the paper appears as a technical report at Department of Computer Science, University of Illinois at Urbana Champaign, Technical Report No. UIUCDCS-R-2004-2412, Mar 2004).

[10] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. ACM International Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, CA, USA, Sept. 2003, pp. 81–95.

[11] M. Hajiaghayi, N. Immorlica, and V. S. Mirrokni, "Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks," in *Proc. ACM International Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, CA, USA, Sept. 2003, pp. 300–312.

[12] S. Khuller, "Approximation algorithms for finding highly connected subgraphs," in *Approximation algorithms for NP-hard problems*, D. S. Hochbaum, Ed. Boston, MA: PWS Publishing Company, 1996.

[13] M. D. Penrose, "On k-connecitivity for a geometric random graph," *Random Structures and Algorithms*, vol. 15, no. 2, pp. 145–164, 1999.

[14] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Lausanne, Switzerland, June 2002.

[15] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni, "Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks," in *Proc. Eleventh International Conference on Computer Communications and Networks (ICCCN)*, Oct. 2002, pp. 392–397.

[16] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi, "Fault tolerant deployment and topology control in wireless networks," in *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Annapolis, MD, USA, June 2003.

[17] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.

[18] S. Even and R. E. Tarjan, "Network flow and testing graph connectivity," *SIAM Journal on Computing*, vol. 4, pp. 507–518, 1975.

[19] G. D. Battista, R. Tamassia, and L. Vismara, "Output-sensitive reporting of disjoint paths," *Algorithmica*, vol. 23, no. 4, pp. 302–340, 1999.