This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

Bi-Criteria Approximation Algorithms for Power-Efficient and Low-Interference Topology Control in Unreliable Ad hoc Networks

Maleq Khan*, V. S. Anil Kumar‡, Madhav V. Marathe‡, Gopal Pandurangan§, and S.S. Ravi¶
*Virginia Bioinformatics Institute, Virginia Tech, Email: maleq@vbi.vt.edu
‡Department of Computer Science and Virginia Bioinformatics Institute, Virginia Tech, Email: {akumar, mmarathe}@vbi.vt.edu
§Department of Computer Science, Purdue University, Email: gopal@cs.purdue.edu
¶Dept. of Computer Science, University at Albany - State University of New York, Email: ravi@cs.albany.edu

***Abstract*—Topology control in ad hoc networks is a multi-criteria optimization problem involving (contradictory) objectives of connectivity, interference, and power minimization. Additionally, nodes can be unreliable, which adds another dimension to an already challenging problem. In this paper, we study topology control problems in ad hoc networks under node failures for *arbitrary* node distributions. We consider a simple and natural stochastic failure model, in which each node can fail independently with a given probability. The *Topology Control Problem Under Stochastic Failures* is to choose a power level for each node and a subset of edges such that the residual graph (i.e., the graph formed by the nodes which have not failed) is connected and can be scheduled efficiently, with high probability. We develop provably efficient bi-criteria approximation algorithms for this problem that simultaneously minimize power, reduce interference, and ensure that the surviving graph is connected with high probability. Our algorithms can be implemented efficiently in a distributed manner.**

I. Introduction

A. Overview

Topology control is a fundamental problem in multi-hop ad hoc wireless networks where it is undesirable for nodes to transmit at the maximum power levels — in addition to wasting energy, which is an important constraint, it leads to high interference. However, reducing power levels can impact connectivity, and so topology control involves a trade-off between the (contradictory) objectives of connectivity, low interference, and energy minimization [1], [2]. The focus of this paper is on topology control in *unreliable* sensor networks, in which nodes can fail — this is especially important in applications involving large scale deployment of such networks, e.g., in habitat monitoring [3].

There has been a lot of work on topologies which remain connected even when some nodes fail [4], [5], [6], [7]. Two classes of failure models have been studied: *adversarial* and *stochastic*. In the adversarial failures model, the topology is required to be well-connected no matter which k nodes fail; in other words, the topology should be a $(k+1)$-connected graph. The cost of such solutions (relative to the optimum) typically depends on k, and can become weak when k is large. The adversarial model has been studied in both random and arbitrary node distributions. In the stochastic failures model, nodes are assumed to fail independently with some probability. Under this model, the connectivity properties have been studied as a function of the transmission range in the case of random node distributions [8].

In this paper, we study topology control problems in the stochastic node failures model, for arbitrary node distributions. Formally, given a set of n nodes (transceivers) placed arbitrarily in the plane, wherein each node v fails independently with probability $1 - p(v) \in [0, 1]$, the *Topology Control Problem under Stochastic Failures (*TCPSF*)* is to choose a topology (i.e., a subset of edges) and power levels for all nodes, so that the random subset of surviving nodes form a "well connected" graph with low interference (formally defined later), with probability at least $1 - \epsilon$, where ϵ is a parameter. We consider a specific time horizon, and each node can fail and can come back again at any time; the power levels for nodes are determined at the beginning, and when a nodes recovers, it uses the power level specified initially. In contrast to random node distributions, in which many connectivity properties can be mathematically analyzed very accurately, stochastic failures in arbitrary node distributions leads to new challenges. For instance, we prove that even determining the probability that there is a component containing a μ-fraction of the surviving nodes for a given constant μ (or approximating it within a small constant factor) is #P-complete. In contrast, the results of [8] yield tight bounds for this quantity for random node distributions. To the best of our knowledge, this is the first work that addresses the topology control problem under stochastic node failures for arbitrary node distributions.

Similar issues arise in the design of sleep scheduling protocols, which try to minimize energy consumption by cycling nodes between sleep and active modes, and scheduling transmissions to achieve convergecast [9], [10]. While many references present protocols that involve a careful choice of schedules for nodes to sleep, transmit, and receive [10], others such as Chiasserini et al. [9] study the performance of protocols in which nodes randomly go to sleep for some

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

(randomly chosen) duration and then wake up. Solving the TCPSF problem will also yield topologies that would work well under such sleep scheduling models.

Early work on the theoretical foundations of topology control, e.g., [11], [1], [6], [12] involved optimization of transmission power levels so that the resulting topology is well connected. However, it has been observed that focusing on the transmission power alone is not very insightful [13], [2], and in recent years there has been interest in designing *low-interference* topologies, in addition to optimizing the transmission powers [2], [14], [15]. We focus on both these metrics, and develop algorithms to construct a topology formed by a set E' of edges that can be scheduled efficiently, while minimizing different functions of the transmission power levels needed to realize the set E' of links.

B. Preliminaries and Network Model

Let V be the set of n nodes on the plane, and let $\ell(u,v)$ denote the distance between u and v, i.e., the length of edge (u,v), for all $u,v \in V$. We will assume throughout that $\min\{\ell(u,v) : u,v \in V\} = 1$ and $\max\{\ell(u,v) : u,v \in V\} = O(n^{O(1)})$. For node $v \in V$, let $B(v,r) = \{w \in V : \ell(v,w) \leq r\}$ be the set of nodes within distance r from v. Following [11], [12], we assume that transmission on a link (u,v) requires a power level $P(u) \geq \phi(u,v) = c \cdot \ell(u,v)^\gamma$, where c and γ are constants. We will consider graphs induced by either a given set E of links on V, or by a vector $\vec{P}$ of power levels for nodes. Let $G = (V,E)$ denote the graph formed by a set E of links. For a given set E of links, let $P(u,E) = \max\{\phi(u,v) : (u,v) \in E\}$ denote power level needed for node u to realize the edges incident on u, and $\vec{P}(E)$ denote the resulting vector of power levels. Designing a topology corresponds to constructing such a set E of edges. For a given transmission power vector $\vec{P}$, let $G = (V,\vec{P})$ denote the graph formed by the set $E = E(G) = \{(u,v) \in V \times V : P(u) \geq \phi(u,v)\}$ of all possible edges realized by $\vec{P}$. If $P(u) = \phi$ for all nodes u, we also refer to the graph $G(V,\vec{P})$ by $G(V,\phi)$. Let $G(V)$ denote the complete weighted graph on V with each edge (u,v) having weight $\ell(u,v)$. We let $MST(V)$ denote a minimum spanning tree of $G(V)$.

We assume that each node v can fail with probability $1 - p(v)$; for simplicity, we assume a uniform probability $p(v) = p$, though all the results in this paper extend to the more general case. Let $V(p)$ denote a random subset of V, containing each $v \in V$ with probability p. Let $G(V(p),\vec{P})$ denote a random subgraph of $G(V,\vec{P})$ induced by the random subset $V(p)$; similarly, let $G(V(p),E)$ denote a random subgraph of $G = (V,E)$.

The TCPSF problem involves constructing a set E of edges so that the subgraph $G(V(p),E)$ induced by the surviving nodes has some desired property $\mathbb{P}$, which could be, for instance, the property that $G(V(p),E)$ is strongly connected, has a giant component or has low diameter. In most of this paper, $\mathbb{P}$ will be the property of strong connectivity. We are interested in two objectives related to E: the *interference* (which is related to how efficiently the edges in E can be scheduled) and the *energy cost*.

We assume adaptive power control, meaning that to transmit on edge $e = (u,v)$, node u uses the minimum power $P(u) = \phi(u,v)$ required for transmission on this link. We use the Tx-Rx model for interference (see [16] for the definitions), which roughly means that edges e and e' can be used simultaneously, provided they do not have a common end point, and there is no edge connecting their end points. We seek to construct edge sets E with low "scheduling complexity" or interference cost, i.e., they can be scheduled in the Tx-Rx model (or other similar models) efficiently. There have been several proposal for combinatorial measures for the interference cost, e.g., the "coverage" measure of [2] and the slightly different congestion measure of [16], [17]. Our interference cost is based on [16], [17], since it seems to be better measure of the scheduling complexity if adaptive power levels are allowed (i.e., nodes can vary the transmission power level on different links). Our algorithms produce topologies with low interference. Given a set E of edges, for edge $e \in E$, define interference of $e = (u,v)$, denoted by $\mathbb{I}(e,E)$, as

$$\mathbb{I}(e,E) = \{e' = (u',v') : e' \in E,\ \ell(e) \leq \ell(e'),\ \text{and}\ \min\{\ell(u,u'),\ell(u,v'),\ell(v,u'),\ell(v,v')\} \leq \ell(e')\},$$

and $\mathbb{I}(E) = \max_{e \in E} |\mathbb{I}(e,E)|$. When the set E is clear from the context, we will refer to the set $\mathbb{I}(e,E)$ simply by $\mathbb{I}(e)$. Intuitively, $\mathbb{I}(e)$ consists of edges that are located "close" to e, and interfere with it, and are longer in length; this measure is motivated by the following result from [16], [17]:

Lemma 1: Given a set E of edges, it is possible to construct an interference-free schedule of length $O(\mathbb{I}(E))$ in the Tx-Rx model (see, e.g., [16] for definitions).

It has also been shown in [16], [17] that the above lemma holds for many different interference models, and this interference metric is stronger than the one in [2] if adaptive power levels are used; the "$\ell(e') \geq \ell(e)$" restriction in the definition is crucial in this case, as illustrated in Figure 1. The second measure we consider is the energy cost, $cost(\vec{P}(E))$, and we are interested in two objectives: the maximum power, $max(\vec{P}) = \max_{u \in V} P(u)$ and the total power, $sum(\vec{P}) = \sum_{u \in V} P(u)$.

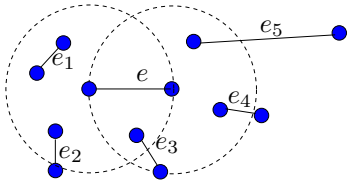

Fig. 1. An example illustrating the interference measure: Consider the set $E = \{e, e_1, e_2, e_3, e_4, e_5\}$ and let $\vec{P}(E)$ denote the power level vector. Then, $\mathbb{I}(e) = \{e_5\}$, $\mathbb{I}(e_4) = \{e, e_5\}$, and $\mathbb{I}(E) = 3$, so that all the edges shown can be scheduled in three time slots: $e_1, \ldots, e_4$ can be scheduled simultaneously, while e and e_5 can be scheduled in two separate slots. In contrast, the coverage defined in [2] is higher, and leads to a longer schedule.

For a given parameter ϵ, let $E = E_{opt}$ denote a set of edges with the minimum possible $\mathbb{I}(E)$, so that $G(V(p),E)$

is strongly connected with high probability[1]; among multiple such E_{opt} which may be possible, we consider one that minimizes the cost of power, i.e., $cost(\vec{P}(E_{opt}))$. The TCPSF problem is a bi-criteria optimization problem that involves constructing a set E of edges so that $G(V(p), E)$ is strongly connected with high probability such that $\mathbb{I}(E)$ and $cost(\vec{P}(E))$ are both small, relative to $\mathbb{I}(E_{opt})$ and $cost(\vec{P}(E_{opt}))$, respectively. The algorithms we study will be randomized bi-criteria approximations: we say our algorithm produces an (α_1, α_2)-approximation to the TCPSF problem if it produces a solution E such that $\mathbb{I}(E) \leq \alpha_1 \mathbb{I}(E_{opt})$ and $cost(\vec{P}(E)) \leq \alpha_2 cost(\vec{P}(E_{opt}))$, with high probability.

C. Summary of Results

1. Computational hardness of TCPSF problem. Various versions of this problem are computationally hard to solve exactly. We prove in Section III that even computing $\Pr[G(V(p), \phi)$ has a component of size at least $\mu|V|]$ is #P-complete, for a given constant μ - this is in contrast with results on stochastic failures in random node distributions [18], [8], where this quantity can be estimated analytically.

Therefore, we focus on approximation algorithms with provable approximation guarantees of the form described in Section I-B. In all our results, the interference complexity will be polylogarithmic, and so we classify the results based on the energy objective. To avoid trivialities, we assume that $p > 1/n^{3/2}$ i.e., the failure probability is not very close to 1.

2. The Sum of Powers Objective. We present an algorithm that gives an $(O(\log^2 n), O(\log^{\gamma+2} n))$-approximation[2] for the sum objective for energy; in other words, our algorithm chooses a set E of edges such that $\mathbb{I}(E) = O(\log^2 n)\mathbb{I}(E_{opt})$, and $sum(\vec{P}(E))/sum(\vec{P}(E_{opt})) = O(\log^{\gamma+2} n)$. Our algorithm is based on a local labeling algorithm, in which each node chooses a set E of edges to some number of higher labeled nodes. We prove that for *any* choice of distinct label numbers, the resulting set E of edges guarantees that $\Pr[G(V(p), E)$ is strongly connected$] \geq 1 - 1/n^3$, and $\mathbb{I}(E) = O(\log^2 n)\mathbb{I}(E_{opt})$. However, the energy cost of the solution, relative to the optimum, i.e., the ratio $sum(\vec{P}(E))/sum(\vec{P}(E_{opt}))$, can be bounded only by choosing the labels carefully. We show that this algorithm can be efficiently implemented in a distributed fashion. As mentioned earlier, the above bounds hold for any node distribution. If the nodes are distributed randomly on a plane, it can be shown that the labels can be chosen randomly by the nodes, with a little worse approximation guarantees.

3. The Maximum Power Objective. We develop an $(O(\log^2 n), O(\log n))$-approximation for the max power objective. This algorithm is randomized and is based on Monte-Carlo sampling. However, a crucial aspect of our algorithm is that the sampling is done with probability not p, but a different value $q < p$. The overall running time of this algorithm is $O(n \log^3 n)$. This algorithm is a non-trivial extension of a result by Goemans and Vondrak [19], who develop an elegant technique for covering minimum spanning trees (MSTs) of random subgraphs of a given graph G. While their result implies that the solution computed by the algorithm is strongly connected with probability at least $1 - 1/n^3$, it does not give any bounds on the cost of the resulting solution. Our proof of this bound crucially uses the geometric structure.

Next, we show that at the expense of larger running time, the approximation factor of the above algorithm can be improved to $(O(\log^2 n), 1)$ for the max objective. This algorithm is a simpler Monte-Carlo algorithm than the one above, and has a running time of $O(n^3 \log n)$. For any "testable" monotone property $\mathbb{P}$ (such as strong connectedness, existence of giant component or low diameter; this is defined formally in Section V-B), this algorithm can be used to find a good power level vector $\vec{P}$ so that $G(V(p), \vec{P})$ has property $\mathbb{P}$.

This algorithm works with the same guarantees, even in a setting where the failure probabilities are non-uniform. Both of these algorithms are based on MST computations, and can be efficiently implemented in a distributed manner following techniques given in [20].

4. Empirical Results. We analyze the empirical performance of our algorithms for both the objective functions. We observe a sharp threshold in the variation of the max power level as a function of ϵ, the probability that $G(V(p), \vec{P})$ is not strongly connected; a similar, though less sharp, threshold is observed for the sum objective also. We also implement our algorithm for the sum objective, and find that the approximation guarantee in practice is significantly better than what we are able to prove analytically in Section IV.

The main focus of this paper is a theoretical analysis of this class of problems. The approximation guarantees we prove here are *worst case* guarantees, which hold for *any* instance. In practice, the approximation guarantees are much better, as we find in our empirical results. We believe that the framework of stochastic failures can help model many applications, and the proof techniques we develop here are likely to be useful in more general settings, such as sleep scheduling protocols.

II. Related Work

Early work on the theoretical foundations of topology control, e.g., [11], [1], [6], [12] involved optimization of transmission power levels so that the resulting topology is well connected. The maximum power objective can be minimized efficiently for any monotone graph property that can be tested efficiently [12]. Under the total power minimization objective, topology control problems for many graph properties (e.g. connectedness, bounded diameter) are known to be NP-hard and approximation algorithms for many such problems have been developed, e.g., [6], [21], [4].

There has been a lot of work on finding k-connected networks, to deal with deterministic failure models, e.g., [4], [5], [7]. These algorithms provide performance guarantees which are polynomial functions of k. Bredin et al. [22] consider the problem of adding a minimum number of sensor nodes to a

[1] usually we take this to mean a probability of at least $1 - \frac{1}{n^3}$, but this can be any parameter

[2] We use $\log^r n$ to mean $(\log n)^r$.

given sensor network so that the augmented network is k-node-connected. Since this problem is NP-hard, they present an approximation algorithm with a performance guarantee which is a polynomial in k.

To our knowledge, under the stochastic failure model, the topology control problem has not received much attention. As mentioned earlier, some results when nodes are arranged on a grid, or placed randomly in the plane have been reported in the literature. For example, Li et al. [18] consider a set of n points placed randomly in the unit square and establish bounds on the transmission radius to be used for each node so that the resulting graph is k connected with high probability. Shakkottai et al. [8] analyze a configuration of nodes on a grid with stochastic failures. They establish relationships between the failure probability and the transmission radius to be used for each node to ensure coverage, connectivity and low diameter. Kumar et al. [23] obtain similar results for other random distributions of nodes.

In recent years, some papers such as [13] have pointed out several issues arising out of focusing on transmission power alone. Therefore, there has been interest in designing *low-interference* topologies, in addition to optimizing the transmission powers [2], [14], [15]. These papers have developed interference measures, which are related to the complexity of scheduling the set of chosen edges, and have designed approximation algorithms for optimizing the interference. However, these papers do not consider unreliable ad hoc networks.

III. Challenges of Analyzing Stochastic Failure Models

Most algorithms for topology control either allow no failures, or consider worst case models for failures, in which any k nodes can fail, as mentioned earlier. We now discuss to what extent these algorithms can be used for the stochastic failure models.

Consider an example of n nodes $v_1, \ldots, v_n$ arranged on a line, with uniform spacing of ℓ, and a power level $\phi(v_i, v_{i+1}) = \phi = c \cdot \ell^\gamma$ for all i. Suppose each node fails with probability $1/2$. Then, it can be shown that the random surviving subgraph is strongly connected with high probability if and only if every node has a power level of $\phi' = \Omega(\phi \log^\gamma n)$. Therefore, in order to be robust to stochastic failures, the power levels have to be much higher than what is chosen in a "failure-free" setting. The right power levels depend crucially on the probability with which we want to ensure the random surviving subgraph be connected, making this a non-trivial problem.

A number of papers on topology control [5], [7], [4], [18] have considered a "worst-case" notion of failures for arbitrary node distributions - they give algorithms for power choice to ensure connectivity when *any* set of k nodes fail. These results do not directly lead to efficient solutions for the TCPSF problem. For the instance described above, with high probability, $\Theta(n)$ nodes will fail. If we run the algorithms from [5] or [7] with $k = \Theta(n)$, the resulting power level choice will indeed guarantee that the surviving subgraph will be connected, but would have a total cost of $\Theta(n^\gamma / \log^\gamma n)$ times the optimum described above. Thus, algorithms that ensure k-connectivity do not necessarily give good solutions for stochastic failures.

However, from a theoretical perspective, the stochastic failures add a different kind of complexity to the TCPSF problem. For instance, even determining the probability that $G(V(p), \vec{P})$ has a large connected component (which seems to be much simpler than the TCPSF problem, and would be needed to "verify" a solution) is hard, as shown in the following lemma; its proof is a simple extension of a result in [24] and is omitted.

Lemma 2: Let $\mu > 1/2$ be a constant. For an arbitrary set V of nodes and power level ϕ, determining $\Pr[G(V(p), \phi)$ has a component with at least $\mu|V|$ nodes$]$ is #P-complete.

IV. The Total Power Objective

We describe Algorithm MinSum-TCPSF for approximating the TCPSF problem with the sum objective. The algorithm consists of two steps. A distinct label is chosen for each node in the first step. The second step uses these labels to choose the set E of edges, and it is interesting to note that *any* choice of distinct labels ensures that $\Pr[G(V(p), E)$ is strongly connected$] \geq 1 - 1/n^2$, and $\mathbb{I}(E) = O(\log^3 n)$. However, the cost, $sum(\vec{P}(E))$ depends on how the labels are chosen. The simplest method would be to choose the labels randomly for each node - this, in fact, works well if the nodes are distributed uniformly at random in the unit square, as we discuss in the full version of this paper. However, for arbitrary distributions of nodes, choosing labels randomly could lead to a high power cost (though, in fact, the interference is small). In Section IV-A, we describe a distributed implementation of Algorithm MinSum-TCPSF.

Algorithm MinSum-TCPSF:

1) Run algorithm ChooseLabelNumbers to choose distinct labels for all nodes.
2) Run algorithm ChooseEdges to find a set E of edges, and the corresponding power levels $\vec{P}(E)$.

Algorithm ChooseLabelNumbers:

1) Construct an Euclidean minimum spanning tree T on the set V of nodes.
2) Root the tree T at some arbitrary node r.
3) Traverse T in depth-first order starting at r. Let $v_1, \ldots, v_n$ be the nodes in this order, with $v_1 = r$.
4) For each node v_i, we define $L(v_i) = n + 1 - i$.

Algorithm ChooseEdges:

1) Let $k = c_1 \log_{1/(1-p)} n$ for some constant c_1. Let the nodes in V be ordered $v_1, \ldots, v_n$ such that $L(v_1) < L(v_2) < \ldots < L(v_n)$.
2) For each node v_i, find the smallest radius $r(v_i)$ such that ball $B(v_i, r(v_i))$ contains $\min\{k, n-i\}$ nodes with labels larger than $L(v_i)$. For each such larger-labeled node v_j, add the edges (v_i, v_j) and (v_j, v_i) to set Q.
3) Set $r'(v_i) = \max_{e=(v_i,v_j)\in Q}\{\ell(e)\}$ and $P(v_i) = c \cdot r'(v_i)^\gamma$.

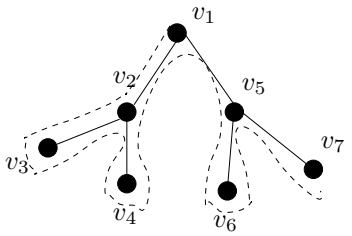

Fig. 2. An illustration of Algorithm Choose-Label-Numbers and some of the terms used in Lemma 4 on an instance with 7 nodes $v_1, \ldots, v_7$. The dashed line shows the traversal order, leading to the sequence $\pi = v_1, v_2, v_3, v_2, v_4, v_2, v_1, v_5, v_6, v_5, v_7$ and the order of the nodes $v_1, v_2, v_3, v_4, v_5, v_6, v_7$. Thus the resulting labels of the nodes are $L(v_1) = 7$, $L(v_2) = 6$, $L(v_3) = 5$, $L(v_4) = 4$, $L(v_5) = 3$, $L(v_6) = 2$, and $L(v_7) = 1$. The reverse sequence of π is $\pi^R = v_7, v_5, v_6, u_5, v_1, v_2, v_4, v_2, v_3, v_2, v_1$.

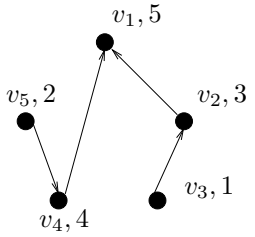

Fig. 3. An illustration of Algorithm Choose-Edges with $k = 1$. The number next to each node v_i indicates its label number $L(v_i)$. The directed edge (v, w) implies that w is the closest node with label larger than $L(v)$. In this example, we have radius $r(v_2) = \ell(v_2, v_1)$, $r(v_3) = \ell(v_3, v_2)$, $r(v_4) = \ell(v_4, v_1)$ and $r(v_5) = \ell(v_5, v_4)$.

Recall the notation defined in Section I-B. We first describe the two steps of our algorithm at an intuitive level here. Let $L(v)$ denote the label chosen using Algorithm CHOOSELABELNUMBERS. Algorithm CHOOSEEDGES is really simple - for each node v, it chooses edges to certain number of closest nodes of higher labels; let $r(v)$ be the length of the longest edge chosen by v. The power level for node v is then $P(v) = c \cdot r(v)^\gamma$, following the model described in Section I-B. We show in Theorem 1 that for this set E, $G(V(p), E)$ is strongly connected with high probability. To construct a "good" labeling of the nodes in Algorithm CHOOSELABELNUMBERS, we first build a minimum spanning tree (MST) and order the nodes based on a depth-first traversal order on this MST. This type of labeling ensures that for each node, the radius $r(v)$ chosen for it is small on average.

The two steps of Algorithm MINSUM-TCPSF are illustrated in Figures 2 and 3. The running time of this algorithm is $O(n^2 \log n)$ if implemented sequentially. The following theorem shows that the power levels chosen by this algorithm ensure that the surviving subgraph is strongly connected, with high probability.

Theorem 1: Let Q be the set of edges chosen by Algorithm CHOOSEEDGES. Then, $\Pr[G(V(p), Q)$ is connected$] \geq 1 - 1/n^3$.

Proof: Let $v_1, v_2, \ldots, v_n$ be the nodes in increasing order or theirs label numbers, i.e., $L(v_i) = i$. For each node v_i, let $H(v_i) = \{v_j \in B(v_i, r(v_i)) : L(v_j) > L(v_i)\}$. Let $S = \{v_1, \ldots, v_{n-k}\}$ and $T = V \setminus S$. Consider any node $v_i \in S$. By construction, we have $|H(v_i)| = k$, and so $\Pr[$all nodes in $H(v_i)$ fail to be in $V(p)] = (1-p)^k = \frac{1}{n^{c_1}}$. By the union bound, the probability that there is a node $v_i \in S$ such that all nodes in $H(v_i)$ fail is $\frac{1}{n^{c_1-1}}$, and so with probability $1 - \frac{1}{n^{c_1-1}}$, corresponding to each node $v_i \in S$, some node $u \in H(v_i)$ survives in a random subset $V(p)$.

By a similar argument, since $|T| = k$, at least one node from T survives in $V(p)$ with probability at least $1 - \frac{1}{n^{c_1}}$. Let w be the node of the largest label that survives from set T in the random set $V(p)$. By construction, we have $H(v_i) = \{v_{i+1}, \ldots, v_n\}$ for each $v_i \in T$. Therefore, with probability $1 - \frac{1}{n^{c_1-1}} - \frac{1}{n^{c_1}}$, for each node $v \in V(p)$, $v \neq w$, some node $u \in H(v)$ survives. By construction of the topology in Algorithm CHOOSEEDGES, we have the edges (v, u) and (u, v) in $G(V(p), Q)$, which implies that $G(V(p), Q)$ is connected, since there is a path from any surviving node $v \in V(p)$ to node w (and back), passing through nodes of increasing label numbers. By choosing c_1 to be greater than 4, we have the result stated in the theorem. ∎

We now bound the interference of the set Q chosen by the algorithm.

Lemma 3: $\mathbb{I}(Q) = O(\log^2 n)\mathbb{I}(E_{opt})$.

Proof: (**Sketch**) Consider an edge $e = (u, v)$. We will show that $\mathbb{I}(e) = O(\log^3 n)$. Let $Q' = \{e' = (u', v') \in \mathbb{I}(e) : L(u') < L(v')\}$, where $L()$ denotes the labels chosen by Algorithm CHOOSELABELNUMBERS; then $|Q'| = \mathbb{I}(e)/2$. Let $A_1 = \{e' = (u', v') \in Q' : u' \in B(u, \ell(e)) \cup B(v, \ell(e))\}$. Let $A_2(i) = \{e' \in Q' - A_1 : \ell(e') \in [2^i, 2^{i+1})\}$, for $i \leq c \log n$ for some constant c. We will show that the sets A_1 and $A_2(i)$ all have size at most $O(\log^2 n)$. First, consider the set A_1. Let V_1 denote the set of end points of edges in A_1 that lie in $B(u, \ell(e)) \cup B(v, \ell(e)$, and let u_1 denote the node of smallest label in V_1. There is a constant c_2 such that if $|V_1| \geq c_2 \log n$, then $B(u_1, \ell(e)/2)$ would contain k nodes of higher label, and therefore, Algorithm CHOOSEEDGES would choose $r(u_1) < \ell(e)$. This contradicts the fact that there is an edge $e_1 = (u_1, v_1) \in Q'$ with $\ell(e_1) \geq \ell(e)$. Therefore, $|V_1| = O(\log n)$. By construction, each node has an outdegree of $O(\log n)$ in the set Q chosen by Algorithm CHOOSEEDGES, and so $|A_1| = O(\log^2 n)$. Next, consider the set $A_2(i)$ and let $V_2(i)$ denote the set of all end points u' such that $(u', v') \in A_2(i)$ for some node v'. Let u_2 denote the node of the smallest label in $V_2(i)$, and let $(u_2, v_2) \in A_2(i)$. As before, if $|V_2(i)| > c_3 \log n$ for some constant c_3, $B(u_2, 2^{i-1})$ would contain k nodes of higher label than u_2, which contradicts the assumption that Algorithm CHOOSEEDGES added the edge (u_2, v_2) to Q'. Therefore, $|V_2(i)| = O(\log n)$. Again, since the outdegree of each node in Q' is $O(\log n)$, we have $|A_2(i)| = O(\log^2 n)$. Therefore, $\mathbb{I}(e) = O(\log^3 n)$. Next, observe that $\mathbb{I}(E_{opt}) = \Omega(\log n)$. This is because the optimum solution E_{opt} must have at least one node v with degree $\Omega(\log n)$, in order for E_{opt} to be robust to failures. It then follows that for the shortest edge e incident on v, $|\mathbb{I}(e)| = \Omega(\log n)$. ∎

Next, we bound the cost $sum(\vec{P})$ of the solution produced

by Algorithm CHOOSEEDGES, relative to the optimum in the following lemma. The specific labeling assigned by Algorithm CHOOSELABELNUMBERS turns out to be crucial.

Lemma 4: Let Q be the set of edges computed in Algorithm CHOOSEEDGES. We have $\sum_{e\in Q}\ell(e)^\gamma \leq (2k)^{\gamma+2}\sum_{e\in T}\ell(e)^\gamma$, where T is the Euclidean minimum spanning tree on the set V of nodes.

Proof: Let π be the in-order traversal on T, which contains the exact sequence of nodes visited - this includes nodes repeated during the traversal (see Figure 2). Let π^R denote the reverse sequence of π.

Consider any node v_i and the minimal subsequence π^R_j, π^R_{j+1}, ..., $\pi^R_{j'}$ such that: (i) $\pi^R_j = v_i$ and π^R_j is the last occurrence of v_i in π^R (ii) there are $k' = \min\{k, n-i\}$ distinct nodes in the sequence $\pi^R_{j+1}, \ldots, \pi^R_{j'}$. For simplicity of notation, define $w_1 = \pi^R_{j+1}, w_2 = \pi^R_{j+2}, \ldots, w_{j'-j} = \pi^R_{j'}$. As the degree of any node in an Euclidean MST can be at most 5, some node may occur as much as 5 times in this subsequence. However, a subsequence of size $2k'$ is sufficient to have k' distinct nodes; because each edge is traversed at most twice, and each traversed edge introduce a new node to the sequence. Thus $j' - j \leq 2k' \leq 2k$. Let A denote the set of distinct nodes that occur in this subsequence. By construction, all these nodes have label numbers larger than $L(v_i)$, since they appear before the first occurrence of v_i in sequence π. Consider the radius $r(v_i)$ computed in Algorithm CHOOSEEDGES. Radius $r(v_i)$ denotes the smallest radius such that ball $B(v_i, r(v_i))$ contains k' nodes with label numbers larger than $L(v_i)$. That is, $r(v_i) \leq \max\{\ell(v,w) : \ w \in A\}$.

Next, observe that

$$\left(\sum_{s=1}^{j'-j-1}\ell(w_s, w_{s+1})\right)^\gamma \leq (2k)^\gamma \sum_{s=1}^{j'-j-1}\ell(w_s, w_{s+1})^\gamma$$

because of the fact that $j'-j \leq 2k$. By the triangle inequality, we have $\max\{\ell(v,w) : \ w \in A\} \leq \sum_{s=1}^{j'-j-1}\ell(w_s, w_{s+1})$, which implies $r(v_i)^\gamma \leq (2k)^\gamma \sum_{s=1}^{j'-j-1}\ell(w_s, w_{s+1})^\gamma$. Let $H(v_i)$ denote the set of nodes with label numbers larger than $L(v_i)$ in the set $B(v_i, r(v_i))$. By definition, for each $w \in H(v_i)$, we have $\ell(v_i, w) \leq r(v_i)$, which implies $\sum_{w\in H(v_i)}\ell(v_i,w)^\gamma \leq k(2k)^\gamma \sum_{s=1}^{j'-j-1}\ell(w_s, w_{s+1})^\gamma$. In this case, we say that node v_i places a charge of $k(2k)^\gamma$ on each of the edges (w_s, w_{s+1}) along this subsequence.

By construction, we have $Q = \cup_{v_i}\{(v_i, w) : \ w \in H(v_i)\}$, and therefore, $\sum_{e\in Q}\ell(e)^\gamma$ can be expressed in terms of the costs of the edges in the subsequence π^R. The only problem is that edges in this subsequence could appear in the summations of a number of nodes, and we need to bound this charge. When we consider a node v_i, we only consider the subsequence of π^R starting at v_i of length at most $2k$. Therefore, an edge (w_s, w_{s+1}) could get charged by at most $2k$ such nodes v_i. This implies

$$\begin{aligned}\sum_{e\in Q}\ell(e)^\gamma &= \sum_{v_i}\sum_{w\in H(v_i)}\ell(v_i,w)^\gamma \\ &\leq \sum_{v_i} k(2k)^\gamma \sum_{s=1}^{j'-j-1}\ell(w_s(v_i), w_{s+1}(v_i))^\gamma \\ &\leq 2k^2(2k)^\gamma \sum_{s\geq 1}\ell(\pi^R_s, \pi^R_{s+1})^\gamma \\ &\leq 2^2k^2(2k)^\gamma \sum_{e\in T}\ell(e)^\gamma\end{aligned}$$

■

Theorem 2: Algorithm MinSum-TCPSF is an $(O(\log^2 n), O(\log^{\gamma+2} n))$-approximation algorithm for the sum objective.

Proof: Since any edge $e \in Q$ can contribute to the radii of at most two nodes, the two end points of e, we have $sum(\vec{P}(Q)) \leq 2\sum_{e\in Q} c\ell(e)^\gamma$.

It is easy to see that if we construct an MST using the weights $c\ell(e)^\gamma$ for all edge e, it would be the same MST using the weights $\ell(e)$. Along with this fact, using Theorem 3.2 (Claim 1) in [25], we have, for any ϵ, $\sum_{e\in MST(V)} c\ell(e)^\gamma \leq sum(\vec{P^{opt}_\epsilon})$. Thus using Lemma 4, we have $sum(\vec{P}) \leq 2(2k)^{\gamma+2}\sum_{e\in MST} c\ell(e)^\gamma \leq 2(2k)^{\gamma+2} sum(\vec{P^{opt}_\epsilon})$. With $k = O(\log n)$ the result now follows from Lemma 3. ■

A. Distributed Implementation

Algorithm CHOOSEEDGES and CHOOSELABELNUMBERS leads to an efficient implementation in a distributed setting as described below.

1. Constructing MST and choosing root. Construct an MST using the distributed algorithm due to Gallager, Humblet and Spira (GHS) [26]. This algorithm takes $O(n \log n)$ time and $O(|E| + n\log n)$ messages. (Reference [27] discusses how this algorithm can be adapted to run in a wireless network setting.) The GHS algorithm also elects a leader, which serves as a root of the MST.

2. Node counting. The root broadcasts a *count* message using the tree edges to all other nodes. A leaf-node after receiving the *count* message, immediately sends back a *count-reply* message to its parent with count = 1. Any intermediate node waits until it receives *count-reply* from all of its children, then aggregates the count and sends a *count-reply* message to its parent with count equal to the aggregated count. Thus, the root can determine the total number of nodes n. Each intermediate node (and the root) also stores the counts received from all of its children; thus it knows the number of nodes in the subtree rooted at each of its children.

3. Label number selection. The root picks the label number n and divides the range $[a..b] = [1..n-1]$ as follows: let the root have t children and the node counts received from its children are $C_1, C_2, \ldots, C_t$; the order of the children is determined by the reverse depth-first order. Then the root sends the range $[L_i, L_i + C_i - 1]$ to its i^{th} child, where $L_i = a + \sum_{j=1}^{i-1} C_j$. Similarly, an intermediate node, after receiving the range $[a..b]$ from its parent, picks the label number b and distributes the range $[a..b-1]$ to its descendants.

Example. In Figure 2, the root $r = v_1$ has total count 7. In reverse depth-first order, count C_1 is the number nodes in the subtree rooted at v_5 and C_2 the number of nodes in the subtree rooted at v_2. Root v_1 picks the label number 7 and

sends the range [1..3] to v_5 and [4..6] to v_2. Node v_5 selects 3 and sends [1..1] to v_7 and [2..2] to v_6, and so on.

4. Finding the k the Nearest Nodes with Larger Labels. Each node executes the following algorithm to find the closest nodes with larger label numbers. Let d be the largest possible distance between any two nodes.

1) Initialize R to be the distance to the closest neighbor.
2) Each node v repeats the following until $R/2 \geq d$ or k nodes with larger label numbers are found.
   a) Set transmission radius r to R and broadcast a message containing $L(v)$.
   b) Any node u, on receiving the message from v, sends back $L(u)$ to v iff $L(u) > L(v)$.
   c) Set $R \leftarrow 2R$.

V. The Maximum Power Level

We now discuss algorithms for the max objective. In Section V-A, we describe a randomized algorithm that gives an $(O(\log^2 n), O(\log n))$-approximation for the max objective, with a running time of $O(n \log^3 n)$. In Section V-B, we describe another algorithm which gives an improved approximation of $(O(\log^2 n), 1)$, but with a higher running time of $O(n^4 \log n)$, thus illustrating the tradeoff between the quality of approximation and running time. Both algorithms are simple, and involve Monte Carlo sampling. However, the analysis, is non-trivial.

A. An $(O(\log^2 n), O(\log n))$-approximation algorithm

Our algorithm MinMax-TCPSF builds on an interesting result by [19], who show that a small subset of $O(n \log_b n)$ edges can "cover" the MST of the surviving subgraph, with high probability. However, we need a non-trivial extension of their analysis to bound the approximation guarantee for the algorithm, which crucially uses the geometric properties.

Recall the notation from Section I-B. We assume that the power threshold $\phi(u, v)$ for any pair of nodes $u, v \in V$ is given by $\phi(u, v) = c \cdot \ell(u, v)^\gamma$ for some constants c and γ.

Algorithm: MinMax-TCPSF

I. Let $b = 1/(1-p)$ and $k = \lceil 5 \log_b n \rceil + 1$.
II. For $i = 1$ to $32ek^2 \ln n$ do
   1) Generate a random subset $V_i = V(q)$ by choosing each vertex independently from V with probability $q = 1/k$.
   2) Find a Euclidean minimum spanning tree T_i of the complete graph on V_i, with the length of each edge (u, v) equal to $\ell(u, v)$.
   3) For each edge e, include e in Q if it appears in at least $16 \ln n$ different T_i's.
III. Set $r = \max_{e \in Q} \ell(e)$ for all nodes $v \in V$.

We need the following result due to [19], which shows that the set Q constructed in the above algorithm contains an MST of the random subgraph $V(q)$, with high probability.

Theorem 3 ([19]): Let Q be the set computed in the above algorithm. Then, we have $|Q| \leq 10en \log_b n + O(n)$, and $\Pr\{MST(V(p)) \subseteq Q\} > 1 - \frac{1}{n^3}$.

We first show that the interference complexity of Q is low.

Lemma 5: $\mathbb{I}(Q) = O(\log^2 n)\mathbb{I}(E_{opt})$.

Proof: (**Sketch**) For any edge $e = (u, v) \in Q$, we prove that $\mathbb{I}(e) = O(\log^3 n)$. Let $A_i(e) = \{e' \in \mathbb{I}(e) : \ell(e') \in [2^i, 2^{i+1})\}$, for each i, and let $V_i(e) = \{u' : \exists v'$ such that $(u', v') \in \mathbb{I}(e)$, and $v' \in B(u, \ell(e)) \cup B(v, \ell(e))\}$ be the set of end points of edges in $A_i(e)$ that do not lie in $B(u, \ell(e)) \cup B(v, \ell(e))$. If $|V_i(e)| > c_4 \log^2 n$ for some constant c_4, there would exist a node $u_4 \in V_i(e)$ such that $|B(u_4, \ell(e)/2) \cap V_i(e)| \geq c_5 \log n$. A direct application of Chernoff's bound implies that $|B(u_4, \ell(e)/2) \cap V_i(e) \cap V(q)| = \Omega(\log n)$ with high probability, and therefore, many nodes in $B(u_4, \ell(e)/2) \cap V_i(e)$ will survive in step II(1) of Algorithm MinMax-TCPSF. This implies that the Euclidean MST in Step II(2) will not end up choosing many edges between nodes in $B(u_4, \ell(e)/2) \cap V_i(e) \cap V(q)$ and nodes in $B(u, \ell(e)) \cup B(v, \ell(e))$, which contradicts the assumption that each node in $V_i(e)$ has an edge to some node in $B(u, \ell(e)) \cup B(v, \ell(e))$. Since there are $O(\log n)$ values of i, we have $\mathbb{I}(e) = O(\log^3 n)$. By the same argument as in the proof of Lemma 3, the proof now follows. ■

Theorem 4: The solution Q produced by Algorithm MinMax-TCPSF is an $(O(\log^2 n), O(\log n))$-approximation for the max power objective.

Proof: By Theorem 3, it follows that $\Pr[G(V(p), Q)$ is strongly connected$] \geq 1 - \frac{1}{n^3}$. Let r_{opt} be the transmission radius corresponding to the optimum solution. We show that with probability at least $1 - \frac{1}{n^3}$, every edge e chosen in Q satisfies $\ell(e) \leq c_1 r_{opt} \log_{b'} n$, where c_1 is a constant to be specified later, and $b' = 1/(1-q)$. In order to do this, we first bound the probability that an edge $e = (u, v)$ with $\ell(e) > c_1 r_{opt} \log_{b'} n$ is added to the set Q in the algorithm. Recall our assumption in Section I-B that $p^2 > 1/n^3$. We claim that $G(V, r_{opt})$ contains a path connecting u and v - if there is no such path connecting u and v, $G(V(p), r_{opt})$ would be disconnected whenever both u and v survive, which happens with probability $p^2 > 1/n^3$, implying that $\Pr[G(V(p), r_{opt})$ is connected $] < 1 - 1/n^3$, which contradicts the definition of r_{opt}. Let $P = \langle w_1, \ldots, w_k \rangle$ be any such path between $w_1 = u$ and $w_k = v$ in $G(V, r_{opt})$. By definition of $G(V, r_{opt})$, every edge $e' = (w_i, w_{i+1}) \in P$ must have $\ell(e') \leq r_{opt} < \frac{\ell(e)}{c_1 \log_{b'} n}$. By the triangle inequality, we have $|P| > c_1 \log_{b'} n$. Let c_2 be a constant such that $2c_2 < c_1$. We partition P into $k/k' = \lceil c_1/c_2 \rceil$ blocks $B_1, \ldots, B_{k/k'}$, each of size roughly $k' = c_2 \log_{b'} n$. We do the analysis assuming that k' is an integer; if this is not true, the analysis can be modified easily to deal with the slight non-uniformity in sizes. Therefore, $B_i = \{w_{(i-1)k'+1}, \ldots, w_{ik'}\}$. These are shown in Figure 4.

The probability that all the nodes in any block B_i fail in a random sample $V(q)$ is $(1-q)^{|B_i|} = (1-q)^{c_2 \log_{1/(1-q)} n} = \frac{1}{n^{c_2}}$. By the union bound, the probability that there is a block B_i in which all nodes fail in $V(q)$ is therefore at most $\frac{c_1}{c_2 n^{c_2}}$,

and so with probability $1-\frac{c_1/c_2}{n^{c_2}}$, at least one node survives in each block B_i in $V(q)$. Consider one such sample $V(q)$, and let $w_{g(i)}$ denote the node that survives in B_i, $i = 1, \ldots, k'$. By the triangle inequality, we have

$$\begin{aligned}\ell(w_{g(i)}, w_{g(i+1)}) &\leq \sum_{j=g(i)}^{g(i+1)-1} \ell(w_j, w_{j+1}) \leq 2k' r_{opt} \\ &\leq 2k'\ell(e)/(c_1 \log n) \leq \frac{2c_2}{c_1}\ell(e).\end{aligned}$$

Since we have $2c_2 < c_1$, we would have $\ell(w_{g(i)}, w_{g(i+1)}) < \ell(e)$, for each $i = 1, \ldots, k/k'$. This means we have an alternate path $u, w_{g(1)}, \ldots, w_{g(k')}, v$ from u to v in $V(q)$, each of whose edges are shorter than e, with probability at least $1 - \frac{c_1/c_2}{n^{c_2}}$. We set $c_1 = 5$ and $c_2 = 2$. Then, in our algorithm V-A, $\Pr[e \in T_i] \leq 1/n$, where T_i is the MST on the random subset $V(q)$. By a Chernoff bound, it follows that for any specific edge e with $\ell(e) > c_1 r_{opt} \log_{b'} n$, we have $\Pr[e$ appears in at least $16 \ln n$ different T_i's$] \leq \frac{1}{n^5}$. By the union bound, it follows that the probability Q contains any edge e' with $\ell(e') > c_1 r_{opt} \log_{b'} n$ is at most $\frac{n^2}{n^5} = \frac{1}{n^3}$. Therefore, the radius r chosen by our algorithm satisfies $r \leq c_1 r_{opt} \log_{b'} n$ with probability at least $1 - 1/n^3$. By definition of r_{opt} and by Lemma 5, the theorem follows. ■

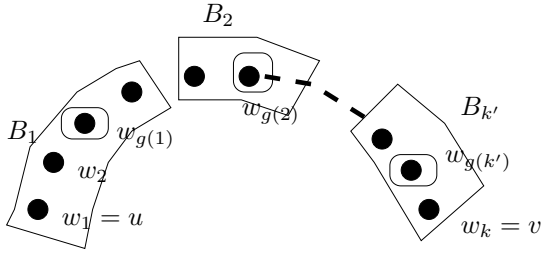

Fig. 4. Edge (u, v) with $\ell(e) > c_1 r_{opt} \log_{b'} n$. The black circles denote the nodes $w_1, \ldots, w_k$, and $B_1, \ldots, B_{k'}$ are the blocks on path P, referred to in the proof. Node $w_{g(i)}$ is the survivor in block B_i.

B. A $(O(\log^2 n), 1)$-approximation algorithm

In this section, we describe a simpler Monte-Carlo sampling based algorithm for the min objective, that works not just for connectivity, but any "testable" monotone property, and gives an $(O(\log^2 n), 1)$-approximation, at the cost of a higher running time, and a lower probability of reliability. Let $\mathbb{P}$ denote a monotone property that can be tested in polynomial time, and let $\mathcal{A}_{\mathbb{P}}$ denote such an algorithm to test for property $\mathbb{P}$; examples of such properties are strong connectivity, existence of a giant component, low diameter, etc. For graph $G(V, \vec{P})$, let $\mathcal{A}_{\mathbb{P}}(G(V, \vec{P}))$ be 1 if $G(V, \vec{P})$ has property $\mathbb{P}$, and 0 otherwise. For a given power level ϕ, the following algorithm, Algorithm Test(ϕ), determines if a given candidate power level ϕ is adequate for all nodes. Using Algorithm Test(ϕ) as a query function, a binary search on the set of power thresholds for all edges, $\{\phi(u, v) \mid u, v \in V\}$, can find the required minimum ϕ by calling Test(ϕ) at most $O(\log n)$ times, since there are at most n^2 distinct power thresholds. Once the correct power threshold ϕ is found, we use the algorithm of [19] to find the set Q of edges.

Algorithm: Test(ϕ)

Input: Set V of nodes on the plane, survival probability p for each node, and a common power level ϕ.
Output: YES, if $G(V(p), \phi)$ has property $\mathbb{P}$ with probability at least $1 - 1/n$.
I. For $i = 1$ to n^2 do

1) Generate a random subset $V_i = V(p)$ of V by choosing each vertex from V with probability p.
2) Compute $X_i = 1 - \mathcal{A}_{\mathbb{P}}(G(V_i, \phi))$, which is 1 if $G(V_i, \phi)$ does not have property $\mathbb{P}$.

II. If $\sum_{i=1}^{k} X_i \leq 3n$, output YES.

Theorem 5: If $\Pr[G(V(p), \phi)$ has property $\mathbb{P}] \geq 1 - \frac{1}{n}$, then $\Pr[$Test(ϕ) does not return YES$] \leq e^{-n}$. The solution Q computed by algorithm $Test()$ and the result of [19] is an $(O(\log^2 n), 1)$-approximation to the max objective. It requires $O(n^2)$ invocations of algorithm $\mathcal{A}_{\mathbb{P}}$; for strong connectivity, this takes time $O(n^3 \log n)$.

VI. Empirical Results

We present empirical results for the TCPSF problem. For most of our experiments, we use an instance consisting of a set of nodes distributed along the streets of the downtown Portland, OR, generated by the TRANSIMS mobility model [28]. We discuss the results for the max and sum objectives separately. In these experiments, we used $\gamma = 2$.

1. The max objective function. Figure 5 shows $\Pr[G(V(p), r)$ is connected$]$ versus radius r, for different values of the surviving probability p for the node distribution in Portland. Observe that there is a sharp threshold in the connectivity radius (or equivalently, the max power level needed) even in this setting. This threshold is known in the case of nodes distributed randomly on the plane [18], [29], or arranged on a grid [8], but it is surprising to observe this threshold even in this non-random setting.

2. The sum objective function. Figure 6 shows the variation of an approximation to the total power objective with ϵ for an instance of the Portland data. Observe that this has a similar threshold behavior as that for the max objective, though it seems to be a little less sharp.

For $\gamma = 2$, Theorem 2 gives an worst case bound of $O(\log^4 n)$ on the approximation factor of $sum(\vec{P})$ given by our algorithm MinSum-TCPSF to the optimum. In the average case, this bound can be much better. For randomly chosen 50-5000 nodes in a unit square, we computed this approximation factor for $\gamma = 2$. Figure 7 shows this experimental result. To understand its growth rate, we also plotted the function $\log n$ and $\log^2 n$. We observe that the growth rate of the approximation factor is even smaller than the growth rate of $\log^2 n$. In fact, the growth rate is pretty much close to that of $\log n$.

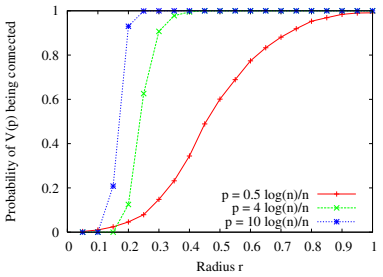

Fig. 5. Connectivity of $G(V(p), r)$ using a structured topology with $n = 1749$ nodes.

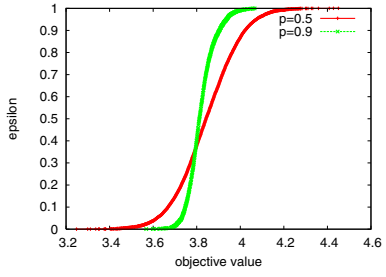

Fig. 6. The variation of (an approximation to) the sum objective with ϵ. Note that the threshold is not as sharp as for the max objective.

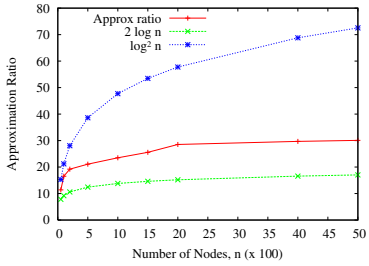

Fig. 7. Approximation ratio $\sum(\vec{P})/\sum(P_\epsilon^{opt})$ for random distributions of nodes computed by the Algorithm MinSum-TCPSF.

VII. Conclusion

In this paper, we designed and analyzed algorithms for topology control for unreliable ad hoc networks that simultaneously achieves provably good approximations to multiple objective criteria, namely, connectivity, power efficiency and interference reduction. Empirically, we find that our algorithm has significantly better performance guarantees in practice, than what we show analytically. Improving the performance guarantees, and extending our results to other topological properties (e.g., spanner) are interesting open problems.

Acknowledgment

We would like to thank the referees for their valuable comments. M. Khan, V.S. Anil Kumar, and M. Marathe was partially supported by NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, CDC Center of Excellence in Public Health Informatics Grant 2506055-01, NIH-NIGMS MIDAS project 5 U01 GM070694-05, DTRA CNIMS Grant HDTRA1-07-C-0113 and NSF NETS CNS-0831633. G. Pandurangan was supported in part by NSF Award CCF-0830476.

References

[1] P. Santi, *Topology Control in Wireless Ad hoc and Sensor Networks*. New York, NY: John Wiley & Sons, Inc., 2005.

[2] M. Burkhart, P. Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?" in *MobiHoc*, 2004.

[3] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," in *Proc. EWSN*, 2004.

[4] X. Jia, D. Kim, S. Makki, P. Wan, and C. Yi, "Power assignment for k-connectivity in wireless ad hoc networks," *J. Combinatorial Optimization*, vol. 9, no. 2, pp. 213–222, March 2005.

[5] N. Li and J. C. Hou, "Flss: A fault-tolerant topology control algorith for wireless networks," *MobiCom'04*, pp. 275–286, September-October 2004.

[6] G. Calinescu and P.-J. Wan, "Range assignment for high connectivity in wireless ad hoc networks," *Proc. International Conference on Ad hoc and Wireless Networks*, pp. 235–246, 2003.

[7] M. T. Hajiaghayi, N. Immorlica, and V. Mirrokni, "Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks," *MobiCom'03*, pp. 300–312, September 2003.

[8] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," *Ad Hoc Networks*, vol. 3, no. 6, pp. 702–716, 2005.

[9] C. Chiasserini and M. Garetto, "Modeling the performance of wireless sensor networks," in *INFOCOM*, 2004.

[10] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *INFOCOM*, 2005.

[11] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," *Proc. IEEE INFOCOM 2000*, pp. 404–413, March 2000.

[12] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," *Mobile Networks and Applications*, vol. 10, pp. 19–34, 2005.

[13] R. Min and A. Chandrakasan, "Top five myths about the energy consumption of wireless communication," in *Mobile Computing and Communications Review*, 2003.

[14] T. Moscibroda and R. Wattenhofer, "Minimizing interference in ad hoc and sensor networks," in *Proc. DIAL M-POMC*, 2005, pp. 24–33.

[15] K. Moaveni-Nejad and X.-Y. Li, "Low-interference topology control for wireless ad hoc networks," in *Proc. of IEEE SECON*, 2005.

[16] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "Provable algorithms for joint optimization of transport, routing and mac layers in wireless ad hoc networks," in *Proc. DialM-POMC Workshop on Foundations of Mobile Computing*, 2007.

[17] ——, "Algorithmic aspects of capacity in wireless networks," in *ACM SIGMETRICS*, 2005, pp. 133–144.

[18] X. Li, P.-J. Wan, Y. Wang, and C. Yi, "Fault tolerant deployment and topology control in wireless networks," *Proc. ACM MobiHoc*, pp. 117–128, 2003.

[19] M. Goemans and J. Vondrak, "Covering minimum spanning trees of random subgraphs," *Random Structures and Algorithms*, vol. 29, pp. 257–276, 2006.

[20] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2001.

[21] S. O. Krumke, R. Liu, E. L. Lloyd, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Topology control problems under symmetric and asymmetric power thresholds," *Proc. International Conference on Ad hoc and Wireless Networks*, vol. 2865, pp. 187–198, October 2003.

[22] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guranteed capacity and fault tolerance," *Proc. ACM MobiHoc*, pp. 309–319, 2005.

[23] S. Kumar, T. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," *MobiCom'04*, pp. 144–158, September-October 2004.

[24] H. AboElFotoh and C. Colbourn, "Computing 2-terminal reliability for radio-broadcast networks," *IEEE Transactions on Reliability*, vol. 38(5), pp. 538–555, 1989.

[25] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power consumption in packet radio networks," *Theoretical Computer Science*, vol. 243, no. 1-2, pp. 289–305, July 2000.

[26] R. Gallager, P. Humblet, and P. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 1, pp. 66–77, January 1983.

[27] M. Khan, G. Pandurangan, and V. S. A. Kumar, "Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 20, no. 1, pp. 124–139, 2009.

[28] *TRANSIMS: Transportation Analysis Simulation System. http://transims.tsasa.lanl.gov/*, Los Alamos National Lab.

[29] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications*, vol. A Volume in Honor of W.H. Fleming. Eds. W. McEneany, G. Yin, and Q. Zhang, pp. 547–566, 1998.