A Fast Distributed Approximation Algorithm for Minimum Spanning Trees in the SINR Model

Maleq Khan[1], V.S. Anil Kumar[1], Gopal Pandurangan[2], and Guanhong Pei[1]

[1]Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University,
Blacksburg, VA, USA
[2]Division of Mathematical Sciences, Nanyang Technological University, Singapore and
Department of Computer Science, Brown University, Providence, RI, USA

Abstract

A fundamental problem in wireless networks is the *minimum spanning tree* (MST) problem: given a set V of wireless nodes, compute a spanning tree T, so that the total cost of T is minimized. In recent years, there has been a lot of interest in the physical interference model based on SINR constraints. Distributed algorithms are especially challenging in the SINR model, because of the non-locality of the model.

In this paper, we develop a fast distributed approximation algorithm for MST construction in an SINR based distributed computing model. For an n-node network, our algorithm's running time is $O(D \log n + \mu \log n)$ and produces a spanning tree whose cost is within $O(\log n)$ times the optimal (MST cost), where D denotes the diameter of the disk graph obtained by using the maximum possible transmission range, and $\mu = \log \frac{d_{max}}{d_{min}}$ denotes the "distance diversity" w.r.t. the largest and smallest distances between two nodes. (When $\frac{d_{max}}{d_{min}}$ is n-polynomial, $\mu = O(\log n)$.) Our algorithm's running time is essentially optimal (upto a logarithmic factor), since computing *any* spanning tree takes $\Omega(D)$ time; thus our algorithm produces a low cost spanning tree in time only a logarithmic factor more than the time to compute a spanning tree. The distributed scheduling complexity of the spanning tree resulted from our algorithm is $O(\mu \log n)$. Our algorithmic design techniques can be useful in designing efficient distributed algorithms for related "global" problems in wireless networks in the SINR model.

1 Introduction

Emerging networking technologies such as ad hoc wireless and sensor networks operate under inherent resource constraints such as power, bandwidth etc. A distributed algorithm which exchanges a large number of messages and takes a lot of time can consume a relatively large amount of resources, and is not very suitable in a resource-constrained network. Also, the topology of these networks can change dynamically. Communication cost and running time is especially crucial in a dynamic setting. Hence it becomes necessary to design efficient distributed algorithms for various network optimization problems that have low communication and time complexity, even possibly at the cost of a reduced quality of solution. (For example, there is not much point in having an optimal algorithm if it takes too much time, since the topology could have changed by that time.) For this reason, in the distributed context, such algorithms are motivated even for network optimization problems that are not NP-hard, e.g., minimum spanning tree (MST), shortest paths (see e.g., [8]). However, much of the theory of distributed approximation for various fundamental problems such as MST have been developed in the context of wired networks, and not for wireless networks under a realistic interference model.

In this paper we focus on one of the most fundamental distributed computing problems in wireless networks, namely the *Minimum Spanning Tree (MST)* problem. It is a recurring sub-problem in many network and protocol design problems, and there has been a lot of work on distributed algorithms for computing the MST. Computing an MST by a distributed algorithm is a fundamental task, as the following distributed computation can be carried over the best backbone of the communication graph. Two important applications of an MST in wireless networks are broadcasting and data aggregation. An MST can be used as broadcast tree to minimize energy consumption since it minimizes $\sum_{(u,v)\in T} d^{\alpha}(u,v)$. It was shown in [2, 7, 34] that broadcasting based on MST consumes energy within a constant factor of the optimum.

Much of the traditional work on distributed algorithms [25, 31] has focused on the CONGEST model of message passing, in which a node can communicate with all its neighbors in one time step. This model is more suited for wired networks, and the complexity of many fundamental problems (e.g., MST [10], shortest paths [4], etc.) is very well understood in the wired model. This model does not capture *interference*, an inherent aspect of wireless networks, which causes collisions when "close-by" nodes transmit. The Radio Broadcast Network (RBN) Model (see, e.g., [1,12]) was developed specifically to address such issues. This model is defined on a unit disk communication graph $G = (V, E)$, and the transmission from a node u to its neighbor v is successful, provided no other neighbor $w \neq u, w \in N(v)$ transmits at the same time. Therefore, nodes that can transmit simultaneously form some kind of independent sets, and this model has been studied extensively over the last two decades. It is known that the RBN model is significantly different from the CONGEST model [12, 31], and its "local" structure has been used in designing efficient algorithms for many fundamental graph problems.

Though the RBN model is much closer model of wireless transmission than the CONGEST model, it still does not capture several crucial features [13,28]: even if a node v receives packets from two neighbors u and w, the transmissions are not necessarily lost — this depends on the ratio of the strength of the u's signal to all other noise being above a threshold. On the other hand, simultaneous transmissions by a number of "well-separated" nodes (which is possible in the RBN model) might be infeasible in reality if the above-mentioned ratio of signal strengths is violated at some receiver. Therefore, the complexity of wireless transmission is not captured by the simple "local" constraints of the RBN model. A different model, called the *Physical Interference model based on SINR (signal-to-interference & noise ratio) constraints* (henceforth, referred to as the SINR model) has been proposed to rectify the weaknesses of the RBN model [13, 28, 29]. The SINR model more accurately captures the physical nature of wireless networks and has been used in a number of recent studies (e.g., see the recent survey of Lotker and Peleg [24] and the references therein). However, unlike traditional (wired and RBN) models, distributed algorithms are especially challenging to design and analyze in the SINR model, because of the non-local nature of the model. In particular, to the best of our knowledge no prior distributed algorithms have been designed for "global" problems in the SINR model. Algorithms in the RBN model do not translate to efficient algorithms in the SINR model, whose non-locality makes it much harder.

Centralized algorithms for various fundamental problems, such as independent sets, coloring, dominating set, spanning tree have been developed in this model in recent years, e.g., [14, 18, 35]. However, distributed algorithms which satisfy the SINR constraints at each step are only known for very few problems (e.g., [9, 17, 20]). Furthermore, the distributed algorithms known are for "local" problems such as independent set and coloring and not for "global" problems. Global problem are those that require an algorithm to "traverse" the entire network. Classical "global" problems include spanning tree, minimum spanning tree, shortest path etc. Network diameter is an inherent lower bound for such problems. We note that the known algorithms in the SINR model for spanning tree problem and the connectivity problem [3,27,30] are in the centralized setting.

In this paper, we describe the first distributed algorithm in the SINR model for approximate MST construction. The SINR constraints are satisfied at all steps of the algorithm, which gives a spanning tree with cost $O(\log n)$, relative to the MST cost. The running time is within a polylogarithmic time of the lower bounds in the RBN model. We discuss our results in more detail below.

(1) The running time of our algorithm is $O(D \log n + \mu \log n)$, with high probability, where D is the diameter of the graph restricted by the

maximum range. This is optimal up to a polylogarithmic factor, since computing *any* spanning tree takes $\Omega(D)$ time. In particular, if the maximum power level is unconstrained, the running time is $O(\mu \log n)$. The spanning tree produced by our algorithm has a low distributed scheduling complexity (defined in Section 3.4) of $O(\mu \log n)$, *i.e.*, the transmission requests on the edges can be scheduled distributedly in $O(\mu \log n)$ time steps, for any orientation of the edges.

(2) Our main technical contribution is the adaptation of the technique of "Nearest Neighbor Trees" (NNT) [21–23] to the SINR model. This technique results in spatial separation at each step, which helps in ensuring the SINR constraints. Our algorithmic design technique can be useful in designing efficient distributed algorithms for related "global" problems in wireless networks in the SINR model.

All prior distributed algorithms in the SINR model have been restricted to scheduling kind of problems (e.g., independent sets, coloring and broadcasting). Our result is the first distributed algorithm for a problem not in the above class, and might be useful in other network design problems.

2 Preliminaries and distributed primitives

Table 1: Notation.

V	set of nodes	n	#nodes
D	network diameter	$d(u,v)$	dist. between u, v
α	path-loss exponent	β	SINR threshold
N	background noise	P	transmission power
μ	distance diversity	r	transmission range

Let V denote a set of tranceivers (henceforth, referred to as nodes) in the Euclidean plane, and $d(u,v)$ the Euclidean distance between nodes u, v. Let d_{min} and d_{max} denote the smallest and the largest distances between any two nodes respectively. d_{min} is a constant as a result of the dimension of a node as a wireless device. We normalize the distances, such that $d_{min} = 1$.We use D to denote the diameter of the disk graph obtained based on V by using the maximum possible transmission range. We say a node u is in the ball $B(v,r)$ of node v if and only if $d(u,v) \leq r$. Let $P(u)$ denote the transmission power chosen by node u, with a maximum power level possible, denoted by P_{max}. We also assume that the nodes are capable of "adaptive power control", which means that they can transmit at any power level in the range $[0, P_{max}]$. We assume the commonly used path loss models [6, 16], in which the transmission from u to v is possible only if: $\frac{P(u)}{d^\alpha(u,v)}/N \geq \beta$, where $\alpha > 2$ is the "path-loss exponent", $\beta > 1$ is the minimum SINR required for successful reception, and N is the background noise (note that α, β and N are all constants). This inequality also defines under a certain transmission power P the *transmission range*, which is the threshold distance beyond which two nodes cannot communicate with each other, and which equals $\sqrt[\alpha]{P/(N\beta)}$. To reduce notational clutter, we will say that the transmission range r associated with a transmission power P is $r = (P/c)^{1/\alpha}$, for a constant c. Let $r_{max} = (P_{max}/c)^{1/\alpha}$ denote the maximum transmission range of any node at the maximum power level. W.l.o.g., we assume $r_{max} \leq d_{max}$ in our algorithms.

We say that a set S of nodes in the plane form a "constant density set" w.r.t. range r if there are $O(1)$ nodes within the ball $B(v,r)$ of any node v, where v is the center and r is the radius. A set $S' \subset S$ is said to be a *constant density dominating set* for S w.r.t. range r, if S' is a constant density set, and for each $v \in S$, there exists $u \in S'$ such that $v \in B(u,r)$. Given a set S of nodes and range r, we define $G_r(S) = \big(S, E = \{(u,v) : u,v \in S, d(u,v) \leq r\}\big)$ to be the graph induced by S with range r.

Wireless interference. We use physical interference model based on geometric SINR constraints (henceforth referred to as the SINR model), where a set L of links can make successful transmissions simultaneously if and only if the following condition holds for each $l = (u,v) \in L$:

$$\frac{\frac{P(u)}{d^\alpha(u,v)}}{\sum_{u' \in V' \setminus \{u\}} \frac{P(u')}{d^\alpha(u',v)} + N} \geq \beta, \tag{1}$$

where V' is the set of transmitting nodes. Such a set L is said to be *feasible* in the context.

MST-SINR: The minimum spanning tree problem under the SINR model. Given a set V of wireless nodes, the goal is to find a spanning tree T, such that the total cost of T is minimized, w.r.t. a cost function $cost(u,v) = d(u,v)$ for any pair of nodes (u,v); for a set E of edges, we define $cost(T) = \sum_{(u,v)\in E} cost(u,v)$. We focus on developing distributed approximation algorithms. We say an algorithm gives a γ-approximation factor if it constructs a spanning tree T, with $cost(T) \leq \gamma \cdot cost(MST)$, where MST represents an optimum solution. In the problem we study here, we only require the tree to be constructed implicitly — we assume we have a sink node s, and we define a *parent*, $par(v)$ for all nodes $v \neq s$, such that the set of edges $\{(v, par(v)) : v \in V, v \neq s\}$ form a spanning tree. Each node only needs to know the identity of its parent. The goal of this paper is to design an algorithm for computing an approximate MST in the distributed computing model based on SINR constraints (described below); we do not require the transmissions (in either direction) on all the edges in the tree to be simultaneously feasible in the SINR model.

Distributed computing model under the SINR model. Traditionally, distributed algorithms for wireless networks have been studied in the radio broadcast model [5, 26, 33] and its variants. The SINR based computing model is relatively recent, and has not been studied that extensively. Therefore, we summarize the main aspects and assumptions underlying this model below.

(1) The network is synchronized and for simplicity we assume all time slots have the same length.

(2) The graph $G_{r_{max}/c}(V)$ induced by V with range r_{max}/c is connected, i.e., the graph can be connected by using a transmission power level of P_{max}/c' which is slightly less than the maximum but within a constant factor.

(3) All nodes have a common estimate of n within a n^c for some constant c.

(4) All nodes share a common estimate of $d_{min} = 1$ and d_{max}, the minimum and maximum distances between nodes. We use $\mu = \log \frac{d_{max}}{d_{min}}$ to denote the "distance diversity" (similar to the "link diversity" in [16]), which is the number of classes of similar length edges into which the set of all edges can be partitioned. It is common to assume that $\mu = O(\log n)$.

(5) As mentioned earlier, we assume nodes are equipped to be able to do adaptive power control, i.e., each node v can transmit at any power level $P \in [0, P_{max}]$.

2.1 Distributed primitives

We discuss two primitives that are needed in our MST algorithms for local broadcasting and dominating sets. The results of [15, 32] directly provide us efficient implementations for these two problems.

1. **Local broadcasting**: The *local broadcasting range* r_b is the distance up to which nodes intend to broadcast their messages. We say the local broadcasting from S to S' is successful if and only if for each node $u \in S$ with transmission power P_b, all the nodes in S' within u's local broadcasting range r_b receives the message from u. We assume we have an algorithm, $\texttt{LocalBroadcast}(S, S', r_b, P_b)$, which takes sets S, S' of nodes, a distance r_b and a power level P_b as input, and ensures that the local broadcasting from S to S' is successful. A small modification of the local broadcasting scheme of [15] achieves this step, which is described below.

Lemma 2.1. *Given two sets S, S' of nodes, a local broadcasting range r_b and a power level $P_b = c' r_b^\alpha$, for a constant c', there is a distributed algorithm $\texttt{LocalBroadcast}(S, S', r_b, P_b)$ that runs in $O(N(S, \gamma r_b) \log |S|)$ time where $N(S, \gamma r_b)$ is the maximum number of nodes in S within a distance γr_b of any node in S and γ is a constant, such that: (i) each node $v \in S'$ receives the message from all nodes in S within distance r_b* w.h.p.*, and (ii) each node $v \in S'$ is able to selectively ignore messages from any node $u \in S'$ which is beyond distance γr_b.*

Proof(sketch). We describe the protocol as follows. Following [15] we use random access, in which each node in S transmits at Power P_b with prob. $\frac{1}{N(S,\gamma r_b)}$ (known to each node in S), and each node in S' senses the channel to receive messages. Each time a node $v \in S'$ receives a message, v checks the total power received; if that exceeds $\frac{P_b}{(\gamma' r_b)^\alpha} + N$, v discards the received message.

Property (i) in Lemma 2.1 directly follows from the proofs in [15], by partitioning the space into rings and uppering bounding the stochastic interference. Then Property (ii) follows from the condition we put on accepting a message based on $SP(v)$, such that any message sent from a node beyond distance γr_b from v will be ignored, where γ is a constant. □

2. **Constant density dominating set**: We assume an algorithm $\texttt{ConstDominatingSet}(S, r_c)$ that takes as input a set S of nodes and a range $r_c \leq r_{max}$, and produces a constant density dominating set $S' \subset S$ corresponding to this range, such that for each node in S there is a node in S' which is with distance r_c, and the density of the nodes in S' is at most a constant. By density, we mean the number of nodes in S' within a range r_c of any node in S'. The algorithm in [32] serves this purpose, with the performance summarized below.

Lemma 2.2 ([32]). *Given a set S of nodes and a range $r_c \leq r_{max}$, a constant density dominating set for S can be constructed in time $O(\log n)$* w.h.p. *under the SINR model.*

2.2 Nearest Neighbor Tree Scheme

The algorithmic paradigm underlying our algorithm is the Nearest Neighbor Tree (NNT) scheme [21–23]. The NNT scheme can be used to construct a spanning tree, called the *Nearest Neighbor Tree (NNT)*, efficiently in a distributed fashion. The cost of the NNT can be shown to be within an $O(\log n)$ factor of the cost of the MST. The scheme used to construct an NNT (henceforth called *NNT scheme*) consists of the following two steps:

(1) each node first chooses a unique *rank* from a totally-ordered set; a ranking of the nodes corresponds to a permutation of the nodes;

(2) each node (except the one with the highest rank) connects (via the *shortest path*) to the *nearest* node of higher rank.

It can be shown that the NNT scheme constructs a spanning subgraph in any weighted graph whose cost is at most $O(\log n)$ times that of the MST, irrespective of how the ranks are selected (as long as they are distinct) [22, 23]. Note that some cycles can be introduced in step 2, and hence to get a spanning tree we need to remove some edges to break the cycles. The main advantage of the NNT scheme is that each node, individually, has the task of finding its nearest node of higher rank to connect to, and hence no explicit coordination is needed among the nodes.

However, despite the simplicity of the NNT scheme, it is non-trivial to efficiently implement the scheme in an arbitrary weighted graph. The work of [21] showed how to efficiently implement the scheme in an arbitrary weighted graph in the CONGEST (wired) model. This implementation was shown to have a running time of $\tilde{O}(D(G)+L(G,w))$ where $L(G,w)$ is a parameter called the *local shortest path diameter* and $D(G)$ is the (unweighted) diameter of the graph. This distributed implementation cannot be directly used here, due to the (additional) complication of having to obey SINR constraints when each node tries to search for the nearest node of higher rank to connect to. Second, the local shortest path diameter can be significantly larger than the diameter of the underlying graph. In particular, it can be as large as n, in which case, the above implementation does not give the time bound that we would like to show in this paper, i.e., close to the diameter of the underlying graph. Note that this is essentially the best possible, since computing a spanning tree on an arbitrary graph takes diameter time. To show this stronger bound, we exploit the fact that the underlying graph has a geometric structure. The main idea is to choose ranks in a particular way (in step (1) of the NNT scheme) that guarantees the each node can find its nearest node of higher rank fast. This idea was first implemented in the UDG-NNT (Unit disk graph-NNT) algorithm [23]. However, again this implementation does not directly work in the SINR model. We show that one has to implement both step (1) and step (2) of the NNT scheme in a incremental and staged fashion, so that SINR constraints are obeyed and still many nodes can progress simultaneously. We show that NNT technique results in spatial separation of "active" nodes (i.e., nodes that need to communicate) in each step, and hence is amenable to obey SINR constraints. The details are in the next Section.

3 Approximating the MST in the SINR model

We now discuss the algorithm $\texttt{MST-SINR}$ for finding a low cost spanning tree. We start with the main intuition for the algorithm. The basic idea for satisfying SINR constraints at each step is to ensure that the senders are "spatially separated". In many distributed MST algorithms, e.g., the algorithms of [10, 11], there is no guarantee that senders are well separated. However, the *Nearest Neighbor Tree (NNT)* scheme [21–23] has a nice property that nodes gradually increase their range, and the nodes which are active in any round are sufficiently well separated. However, this approach has only been studied in CONGEST model in [21–23], and we adapt it to the SINR model. For illustration, first imagine that the maximum power P_{max} is "unbounded" so that the $d_{max} \leq r_{max}$. The algorithm in this special case involves the following steps.

(1) We have $\log r_{max} \leq \mu$ phases, ranging from $i = 1, \ldots, \log r_{max}$.

(2) In the ith phase, a subset S_i of nodes participate, and the edges chosen so far form a forest rooted at nodes in S_i. The nodes in S_i transmit at power level of $c \cdot d_i^{\alpha}$ for a constant c, where $d_i = 2^i$.

(3) In the ith phase, each node $v \in S_i$ runs the NNT scheme: v connects to a "close-by" node in S_i within distance $c' \cdot d_i$ of higher rank, if one exists, for a constant c'. The nodes which are not able to connect continue into phase $i + 1$.

We need to prove that the above steps can be implemented in the SINR model, and the resulting implicit tree is a low cost tree, relative to the MST. One complication, in contrast to the original NNT scheme, is that there is no way to ensure that a node $v \in S_i$ connects to the "closest" node in S_i of higher rank, within distance d_i. Because of the SINR model, it is possible that a transmission from some far away node could be received by v. We show that the probability of this event is very low, so that with high probability, we get a tree that is "close to" the NNT, leading to the logarithmic approximation bound for the cost of the tree. We describe the above algorithm as subroutine NNT-SINR-BP in Section 3.1, with the modification that P_{max} is bounded, and instead of μ phases, we only have $\log r_{max}$ phases. The result of NNT-SINR-BP is a forest with the highest rank nodes forming the roots.

In the general case, P_{max} is bounded, so that $d_{max} > r_{max}$, and all we have is that the graph induced by range r_{max} is connected[*]. If we run the above bottom-up algorithm with some maximum range r_1, we would get a forest in which the roots are at least $c \cdot r_1$ apart, where c is a constant. In order to connect up the roots, we first use a "top-down" approach where we choose a set Dom of nodes so that (i) each node $v \notin Dom$ is within distance r_1 of some node in Dom, and (ii) for each node $u \in Dom$, there are a "small" number of nodes within distance r_1. The idea is that if we can construct a spanning tree on Dom (in the SINR model), and can ensure that each node in $V \setminus Dom$ has a higher rank neighbor in Dom, within distance r_1, then the above bottom-up phase restricted to $V \setminus Dom$ will allow the trees to be combined. Such a set Dom is precisely a "constant density dominating set", as defined and computed efficiently in [32]. The spanning tree on Dom is constructed by adapting the UDG-NNT algorithm of [23], which starts with a node s and spreads the ranks from it. This adaptation is discussed below as subroutine NNT-SINR-CD, which also works in the SINR model because the nodes in Dom are spatially well separated.

We first describe the two subroutines, and then discuss the main algorithm.

3.1 Constructing forests with power constraints

For subroutine NNT-SINR-BP, we are given a set S, a maximum power level P_{max} (corresponding to a range r_{max}), and a rank function $rank(\cdot)$, which assigns unique ranks for all nodes in S. The goal is to construct a forest, in which each node connects to a parent within range r_{max}. As discussed earlier, NNT-SINR-BP uses the NNT approach of [21–23], in which each node connects to the nearest node of higher rank, which leads to a forest. However, in order to be feasible in the SINR model, we need to do this in a careful and staged manner, so that the set of transmitting nodes in each round form a constant density set.

Lemma 3.1. *At the beginning of each phase i, for each node $v \in S_i$, there are at most a constant number of active nodes in the ball $B(v, d_i)$,* i.e., $\left|B(v,d_i) \bigcap S_i\right| = O(1)$. *Further, for each node $v \in S_i$,there are no other active nodes in the ball $B(v, d_i/2)$,* i.e., $B(v, d_i/2) \bigcap S_i = \{v\}$.

Proof by induction. Recall that S_i is the set of active nodes at phase i. For phase 1, $S_1 = S$ and $d_0 = 1$; obviously there can be at most 16 nodes in the

[*] Recall that in Section 2, we in fact assume something stronger: the graph induced by a range of r_{max}/c is connected for a constant c.

ball. Assume that for phase i, the statement is true. Then, at the beginning of phase $i+1$, if v is active, according to the algorithm, all nodes in the range of d_i of v should all have a lower rank and thus are all inactive. Therefore, the distance between any two nodes in S_{i+1} is at least d_i. By packing property, there can be at most 16 nodes in the ball $B(v, d_{i+1})$. Therefore, the statement is true for every phase. □

Algorithm 1: `NNT-SINR-BP`$(S, P_{max}, rank(\cdot))$

```
Input : node set $S$, max power $P_{max}$, and
        value $rank(v)$ for each $v \in S$
Output: set $F$ of edges
Each node $v \in S$ does the following:
$r_{max} \leftarrow (P_{max}/c)^{1/\alpha}$, $i_{max} \leftarrow \lfloor \log_2 r_{max} \rfloor$;
initially $S_i \leftarrow \emptyset, \forall i > 1$; # $S_i$ denotes the set
    of active nodes in phase $i$
$S_1 \leftarrow S$;
for $i = 1$ to $i_{max}$ do # in each phase
    # $v$ broadcasts its rank to all nodes
      in $S_i$ within range $d_i$
    $d_i \leftarrow 2^i$, $P_i \leftarrow c' d_i^{\alpha}$; # $c'$ is a constant
    if $v \in S_i$ then # broadcast rank value
        $v$ broadcasts its rank by participating in
        LocalBroadcast$(S_i, S_i, d_i, P_i)$;
    end
    $S_v \leftarrow$ set of nodes with ranks received by $v$;
    $v' \leftarrow$ the node with highest rank in $S_v$;
    if $v = v'$ then # $v$ has highest rank
    locally: add $v$ to the active set $S_{i+1}$
    for next phase
        add $v$ to $S_{i+1}$;
    else $par(v) \leftarrow v'$; # $v$ is done
end
$F \leftarrow \{(v, par(v)) : par(v) \neq \emptyset, \forall v \in S\}$;
```

Lemma 3.2. *In the end of the algorithm, the set F forms a forest. Further, the set $S_{i_{max}}$ forms the set of roots of F, and for any two nodes $u, v \in S_{i_{max}}$, we have $d(u, v) \geq r_{max}/2$.*

Proof. Because of the NNT property, in which each node only connects to a higher rank parent, there are no cycles. Next, by design, the nodes in $S_{i_{max}}$ do not connect to any other node, and form the roots of the forest F. By Lemma 3.1, it follows that for any $u, v \in S_{i_{max}}$, we have $d(u, v) \geq r_{max}/2$. □

Lemma 3.3. *The cost of the forest F is at most $O(\mu)$ times that of of the minimum spanning forest of the nodes in F. Further,* `NNT-SINR-BP` *is feasible in the SINR model, with running time of $O(\mu \log n)$,* w.h.p.

Proof. For $i < i_{max}$, let $F_i = \{(v, w) \in F : v \in S_i, w = par(v) \in S_{i+1}\}$ denote the set of edges in F between nodes in sets S_i and S_{i+1}. Recall that S_i denotes the set of nodes which remain active in the beginning of phase i. Let $MST(S_i)$ denote a minimum spanning tree of only the nodes in S_i. W.l.o.g., we assume $|S_i| > 1$. For any pair of nodes in S_i, the mutual distance is at least $d_i/2$ (from Lemma 3.1), it follows that $cost(MST(S_i)) \geq (|S_i| - 1)d_i/2 \geq |S_i|d_i/4$. Next, we have $cost(F_i) \leq d_i|F_i| \leq d_i|S_i|$, implying $cost(F_i) \leq 4cost(MST(S_i))$. Since there are $O(\log r_{max})$ phases and $cost(MST(S_i)) \leq cost(MST)$, we have $cost(F) = \sum_i cost(F_i) \leq 4(\log r_{max}) \sum_i cost(MST(S_i))$. We have assumed $r_{max} \leq d_{max}$; therefore, $\log r_{max} \leq \mu$, implying the statement of approximation ratio. By Lemmas 2.1 and 3.1, each run of the local broadcast primitive finishes in $O(\log n)$ time, so does each phase of the algorithm. The statement of running time follows. □

Algorithm 2: `NNT-SINR-CD`(S, s, r)

```
   Input  : constant density node set $S$, sink $s$,
            range $r$
   Output: spanning tree $T$ on $S$
1  Each node $v \in S$ does the following:
   # in phase 0:
2  if $v = s$ then # $s$ broadcasts to neighbors
3      $v$ generates a large $rank(v)$, and broadcasts
       it by LocalBroadcast$(\{s\}, S, r, cr^{\alpha})$;
4  else $v$ listens;
   # in each phase $i$:
5  for $i = 1$ to $2D(G_r(S))$ do
6      $S_i \leftarrow \emptyset$;
7      if $v$ receives a message $m = (id', rank')$
       then
8          if $m$ is the first message $v$ ever receives
           then
9              $v$ generates $rank(v)$ randomly such
               that $rank(v) < rank'$;
10             $S_i \leftarrow S_i \cup \{v\}$;
11             $v$ participates in
               LocalBroadcast$(S_i, S, r, cr^{\alpha})$;
12         end
13         add $id'$ to list $L(v)$, if $rank' > rank(v)$;
14     end
15 end
16 $par(v) \leftarrow$ one of the nodes in $L(v)$;
17 return the set $\{(v, par(v)) : v \neq s\}$ of edges
```

3.2 Spanning trees for instances with constant density

We now consider the problem of constructing a low cost spanning tree for a constant density instance. Subroutine `NNT-SINR-CD` takes such a set S, a sink s and a range r as input. As defined in Section 2, $G_r(S)$ denotes the S-induced graph based on range r. We adapt the UDG-NNT algorithm of [23] for producing a spanning tree. Let $D(G_r(S))$ denote the diameter of $G_r(S)$; we assume (an estimate of) $D(G_r(S))$ is known to all the nodes. We note that in Line 11 of the algorithm, the local broadcasts are run simultaneously for all nodes v, which get the rank message for the first time. Further, each iteration of the **for** loop in Lines 6-14 involves $O(\log n)$ time steps, in which the local broadcast is run; nodes v which do not have to send their ranks in some iteration remain silent during these steps.

Lemma 3.4. *If $G_r(S)$ is connected, and S is a constant density set with range r, Algorithm* `NNT-SINR-CD` *produces under the SINR model a spanning tree T_1 on S with $cost(T_1) = O(OPT(G_r(S)))$ in time $O(D(G_r(S)) \log n)$ with high probability, where $OPT(G_r(S)) = cost(MST(G_r(S)))$.*

Proof. Our proof mimics the proof of algorithm UDG-NNT from [23]. Since $G_r(S)$ is connected, each node $v \neq s$ is able to run line 12, and the tree T_1 is constructed. The simultaneous calls to `LocalBroadcast` all take time $O(\log n)$ in each round (with high probability), so that the overall running time is $O(D(G_r(S)) \log n)$, *w.h.p.*. Since the `LocalBroadcast` algorithm is feasible in the SINR model, `NNT-SINR-CD` is also feasible.

The constant density property of S implies that any ball $B(v, r')$ with $r' \leq r$ has $O(1)$ nodes in S. Therefore, $cost(MST(G_r(S))) = \Omega(|S|r)$. Next, the local broadcasts (Lemma 2.1) ensure that each node v receives messages from nodes within distance $c'r$ for a constant c'. Therefore, $cost(T_1) \leq |S|c'r$, and the lemma follows. □

3.3 Putting everything together: Algorithm **MST-SINR**

Our high level idea is to start with a constant density dominating set Dom, and run subroutine

NNT-SINR-CD to construct a tree T_1 on Dom. We then choose and disseminate ranks suitably, and run NNT-SINR-BP to form a forest, which get connected to the nodes in Dom, and together form a spanning tree.

Our analysis, at a high level, involves the following steps: we first show that T_1 is a spanning tree on Dom with range $2r_{max}/c$, and has low cost. Next, we show that Algorithm NNT-SINR-BP with the range of r_{max} results in a forest on $V \setminus Dom$, each of whose components gets connected to some node in Dom, because of the way the ranks are chosen. Finally, we show that the combined tree produced in this manner has low cost.

Lemma 3.5. *The graph $G_{2r_{max}/c}(Dom)$ induced by Dom w.r.t. a range of $2r_{max}/c$ is connected, and has a diameter of at most $2D$.*

Proof. Suppose $G_{2r_{max}/c}(Dom)$ is not connected. Then there exists a partition $Dom = Dom_1 \cup Dom_2$, which induce disconnected components, and $\min_{(u,v)\in Dom_1\times Dom_2}\{d(u,v)\} > 2r_{max}/c$, since the dominating set Dom constructed uses range r_{max}/c. That implies that G is connected with r_{max}/c, which is a contradiction. Therefore, $G_{2r_{max}/c}(Dom)$ is connected. □

Lemma 3.6. *Algorithm* MST-SINR *produces a spanning tree of cost $O(\mu)$ times the optimal in time $O(D\log n + \mu \log n)$, with high probability, in the SINR model.*

Proof. From Lemma 3.4, the call to NNT-SINR-CD produces a spanning tree T_1 on set Dom. Next, consider the call to NNT-SINR-BP in Line 14 of the algorithm. Let $V' = \{v \in V \setminus Dom : par(v) = \phi\}$ be the set of nodes for whom the parent is not defined during this call. Each nodes in V' corresponds to a tree root of the forest constructed in this call, *i.e.*, the edges $F_1 = \{(v, par(v)) : v \in V \setminus Dom, par(v) \neq \phi\}$ (constructed during this call) form a forest, rooted at nodes in V'. In Line 12, each node in $V \setminus Dom$ registers its dominating node in T_1; in Line 17, each node $v \in V'$ uses this information to connect to T_1. Therefore, $T_1 \cup F_1 \cup \{(v, par(v)) : \forall v \in V'\}$ is a spanning tree.

From Lemma 3.4, we have $cost(T_1) = O(OPT)$. Next, from Lemma 3.3, it follows that

Algorithm 3: MST-SINR(V)

```
   Input  : node set $V$
   Output: spanning tree $T$
   # use the algorithm of [32] to construct
     a constant density dominating set $Dom$
     w.r.t.  range $r_{max}/c$, for a constant $c$
1  $Dom \leftarrow$ ConstDominatingSet($V, r_{max}/c$) for a constant $c$;
   # we adapt the UDG-NNT algorithm of [23]
     for constructing a spanning tree on
     $Dom$
2  $s \leftarrow$ a node in $Dom$; # sink node
3  $T_1 \leftarrow$ NNT-SINR-CD($Dom, s, 2r_{max}/c$);
4  for each $v \in Dom$ do
5      $rank(v) \leftarrow$ the rank chosen in this process;
6  end
   # each $v \in Dom$ broadcasts its rank to
     nodes within its range
7  run LocalBroadcast($Dom, V \setminus Dom, r_{max}/c$)
   such that all $v \in Dom$ to broadcast $rank(v)$;
8  for each $v \in V \setminus Dom$ do
9      $b(v) \leftarrow$ largest rank received by $v$;
10     $q(v) \leftarrow$ id of the node that sent $v$ rank $b(v)$;
11     $v$ chooses $rank(v)$ uniformly at random
       such that $rank(v) < b(v)$;
12     $dom(v) \leftarrow q(v)$; # to connect to tree
         roots of forest $F_1$ constructed below
13 end
   # use the ranks chosen above to build a
     forest with tree roots in $Dom$ to
     connect up with the tree $T_1$
14 $F_1 \leftarrow$ NNT-SINR-BP($V \setminus Dom, P_{max}, rank(\cdot)$);
15 for each $v \in V \setminus Dom$ do
16     if $par(v) = \emptyset$ then # $v$ is a tree root in
       forest $F_1$
17         $par(v) \leftarrow dom(v)$; # connect to $T_1$
18     end
19 end
20 return set $\{(v, par(v)) : par(v) \neq \emptyset\}$ of edges.
```

$cost(F_1) = O(\mu \cdot OPT)$. Finally, for each $v \in V'$, $d(v, par(v)) \leq c' r_{max}/c$. Therefore, $\sum_{v \in V'} d(v, par(v)) = O(OPT)$, which implies the cost of the tree produced by `NNT-SINR-BP` is $O(\mu \cdot OPT)$.

Finally, we analyze the running time. The call to `ConstDominatingSet` in Line 1 takes $O(\log n)$ time. From Lemma 3.4, the call to `NNT-SINR-CD` takes time $O(D \log n)$, with high probability. Next, the local broadcast takes time $O(\log n)$, and the call to `NNT-SINR-BP` takes time $O(\mu \log n)$. Putting all of these together, the total running time is $O(D \log n + \mu \log n)$, with high probability. From [32], and Lemmas 3.4, 3.3 and 2.1, all the computations of the algorithm are feasible in the SINR model. □

3.4 Scheduling complexity of the spanning tree

The *scheduling complexity* (as defined in [30]) of a set of communication requests is the minimal number t of time slots during which each link request has made at least one successful transmission. Due to the constant factor approximation algorithm by Kesselheim [19] for finding a maximum feasible set of links, any spanning tree has a scheduling complexity of $O(\log n)$ under the SINR model. Here, we use the term *distributed scheduling complexity* to describe the scheduling complexity for a set of links that needs to make transmission decisions in a distributed fashion. The spanning tree T produced by our algorithm has the following property.

Lemma 3.7. *The spanning tree T produced by Algorithm* `MST-SINR` *has a distributed scheduling complexity of $O(\mu \log n)$, regardless of the direction of the transmission requests on the edges.*

Proof by construction. Let the edges in T be oriented in arbitrary directions, so that we obtain a set L of links, which represent a set of transmission requests.

We construct a feasible schedule which can be implemented in a distributed fashion. We partition the links into length classes $1, 2, \ldots, \log r_{max}$, such that all links in length class i have a length between 2^{i-1} and 2^i. In each phase $i \in [1, \log r_{max}/c]$, links in length class i transmit with probability $1/K$ in each time step, where K is an upper-bound of the number of edges/links in length class i with an end node within range 2^{i+1} of each node. With the same proof approach as that in [15], we can prove that in $O(K \log n)$ time steps, all the links in length class i have made successful transmissions with high probability. This way, in $\log r_{max}/c]$ phases, all the links have made successful transmissions, *w.h.p.*

Now, we argue that K is $O(1)$ in each phase, and thus the total schedule length is $O(\mu \log n)$. Due to the construction of T in Algorithm `MST-SINR`, since Dom is already a constant density node set, we only need to analyze the edges formed during the call to `NNT-SINR-BP`$(V \setminus Dom, P_{max}, rank(\cdot))$ in Line 14 of Algorithm `MST-SINR`. In each phase of the `NNT-SINR-BP`, the lengths of edges increases upon the previous phase. In one phase during `NNT-SINR-BP`, the newly formed edges corresponds to a length class, and the nodes involved in that phase form a constant density node set, according to Lemma 3.1. That implies that the number of edges in length class i with an end node within range 2^{i+1} of each node is $O(1)$.

In terms of the distributed implementation of scheduling, each edge may remember the phase number in `NNT-SINR-BP` when they are connected up by a new edge, and participate in the phase of the same phase number of our scheduling construction. □

4 Conclusion

In this paper, we describe the first distributed algorithm in the SINR model for approximate minimum spanning tree construction. This is the first such result for solving a "global" problem in the emerging SINR based distributed computing model— this is in contrast to "local" problems, such as independent sets and scheduling, for which distributed algorithms are known in this model. Our algorithm produces a logarithmic approximation to the MST, and takes time $O(D \log n + \mu \log n)$, featuring a distributed scheduling complexity of $O(\mu \log n)$. Our main technical contribution is the use of *nearest neighbor trees*, which naturally ensure spatial separation at each step, thereby allowing SINR constraints to be satisfied.

References

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43:290–298, 1991.

[2] C. Ambuhl. An optimal bound for the MST algorithms to compute energy efficient broadcast trees in wireless networks. In *Proceedings of the 32th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1139–1150, November 2005.

[3] Chen Avin, Zvi Lotker, Francesco Pasquale, and Yvonne Anne Pignolet. A note on uniform power connectivity in the sinr model. In *ALGOSENSORS*, pages 116–127, 2009.

[4] B. Awerbuch, A. Bar-Noy, and M. Gopal. Approximate distributed bellman-ford algorithms. *IEEE Transactions on Communications*, 42(8):2515–2517, 1994.

[5] H. Balakrishnan, C. Barrett, V. S. Anil Kumar, M. Marathe, and S. Thite. The distance 2-matching problem and its relationship to the MAC layer capacity of adhoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1069–1079, 2004.

[6] Deepti Chafekar, V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy 0002, and Aravind Srinivasan. Cross-layer latency minimization in wireless networks with sinr constraints. In *ACM MobiHoc*, pages 110–119, 2007.

[7] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraph. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 121–131, June 2001.

[8] M. Elkin. An overview of distributed approximation. *ACM SIGACT News Distributed Computing Column*, 35(4):40–57, December 2004.

[9] Alexander Fanghänel, Thomas Kesselheim, and Berthold Vöcking. Improved algorithms for latency minimization in wireless networks. In *Internatilonal Collogquium on Automata, Languages and Programming (ICALP)*, pages 447–458, 2009.

[10] R. Gallager, P. Humblet, and P. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1983.

[11] J. Garay, S. Kutten, and D. Peleg. A sublinear time distributed algorithm for minimum-weight spanning trees. *SIAM Journal of Computing*, 27:302–316, 1998.

[12] L. Gasieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 129–137, 2005.

[13] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *Proceedings of MOBIHOC*, pages 100–109, 2007.

[14] Olga Goussevskaia, Magnús Halldórsson, Roger Wattenhofer, and Emo Welzl. Capacity of arbitrary wireless networks. In *IEEE INFOCOM*, 2009.

[15] Olga Goussevskaia, Thomas Moscibroda, and Roger Wattenhofer. Local Broadcasting in the Physical Interference Model. In *ACM SIGACT-SIGOPT DialM-POMC*, 2008.

[16] Olga Goussevskaia, Yvonne Anne Oswald, and Roger Wattenhofer. Complexity in geometric sinr. In *ACM MobiHoc*, 2007.

[17] Magnús M. Halldórsson and Pradipta Mitra. Nearly optimal bounds for distributed wireless scheduling in the sinr model. In *ICALP*, 2011.

[18] Magnús M. Halldórsson and Roger Wattenhofer. Wireless communication is in apx. In *ICALP*, 2009. Corrected version: `http://www.hr.is/faculty/mmh/papers/juniform.pdf`.

[19] Thomas Kesselheim. A constant-factor approximation for wireless capacity maximization with power control in the sinr model. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[20] Thomas Kesselheim and Berthold Vöcking. Distributed contention resolution in wireless networks. In *DISC*, 2010.

[21] M. Khan and G. Pandurangan. A fast distributed approximation algorithm for minimum spanning trees. *Distributed Computing*, 20(6):391–402, Apr 2008. **Invited paper**, Best Student Paper Award in DISC 2006.

[22] M. Khan, G. Pandurangan, and V.S.A. Kumar. A simple randomized scheme for constructing low-weight k-connected spanning subgraphs with applications to distributed algorithms. *Theoretical Computer Science*, 385(1-3):101–114, Oct 2007.

[23] 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*, 20(1):124–139, Jan 2009.

[24] Zvi Lotker and David Peleg. Structure and algorithms in the sinr wireless model. *ACM SIGACT News*, 41(2), 2010.

[25] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996.

[26] Thomas Moscibroda and Roger Wattenhofer. Maximal independent sets in radio networks. In *ACM symposium on Principles of distributed computing (PODC)*, 2005.

[27] Thomas Moscibroda and Roger Wattenhofer. The complexity of connectivity in wireless networks. In *IEEE INFOCOM*, 2006.

[28] Thomas Moscibroda and Roger Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Barcelona, Spain*, April 2006.

[29] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger. Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Florence, Italy*, May 2006.

[30] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger. Topology control meets sinr: the scheduling complexity of arbitrary topologies. In *ACM MobiHoc*, pages 310–321, 2006.

[31] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[32] C. Scheideler, A. Richa, and P. Santi. An $o(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In *ACM MobiHoc*, pages 91–100, 2008.

[33] Johnnes Schneider and Roger Wattenhofer. A log* distributed maximal independent set algorithm for growth-bounded graphs. In *ACM symposium on Principles of distributed computing (PODC)*, 2005.

[34] P. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6):607–617, November 2002.

[35] Peng-Jun Wan, Xiaohua Jia, and Frances Yao. Maximum independent set of links under physical interference model. In *Proceedings of WASA*, pages 169–178, 2009.