Nearly Optimal Local Broadcasting in the SINR Model with Feedback

Leonid Barenboim[1,⋆] and David Peleg[2,⋆⋆]

[1] Department of Mathematics and Computer Science,
The Open University of Israel, Raanana, Israel
leonidb@openu.ac.il

[2] Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Rehovot, Israel
david.peleg@weizmann.ac.il

Abstract. We consider the SINR wireless model with uniform power. In this model the success of a transmission is determined by the ratio between the strength of the transmission signal and the noise produced by other transmitting processors plus ambient noise. The *local broadcasting* problem is a fundamental problem in this setting. Its goal is producing a schedule in which each processor successfully transmits a message to all its neighbors. This problem has been studied in various variants of the setting, where the best currently-known algorithm has running time $O(\bar{\Delta} + \log^2 n)$ in n-node networks with feedback, where $\bar{\Delta}$ is the maximum neighborhood size [9]. In the latter setting processors receive free feedback on a successful transmission. We improve this result by devising a local broadcasting algorithm with time $O(\bar{\Delta} + \log n \log\log n)$ in networks with feedback. Our result is nearly tight in view of the lower bounds $\Omega(\bar{\Delta})$ and $\Omega(\log n)$ [13]. Our results also show that the conjecture that $\Omega(\bar{\Delta} + \log^2 n)$ time is required for local broadcasting [9] is not true in some settings.

We also consider a closely related problem of *distant-k coloring*. This problem requires each pair of vertices at geometrical distance of at most k transmission ranges to obtain distinct colors. Although this problem cannot be always solved in the SINR setting, we are able to compute a solution using an optimal number of *Steiner points* (up to constant factors). We employ this result to devise a local broadcasting algorithm that after a preprocessing stage of $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ time obtains a local-broadcasting schedule of an optimal (up to constant factors) length $O(\bar{\Delta})$. This improves upon previous local-broadcasting algorithms in various settings whose preprocessing time was at least $O(\bar{\Delta} \log n)$ [3,10,5]. Finally, we prove a surprising phenomenon regarding the influence of the path-loss exponent α on performance of algorithms. Specifically, we show that in vacuum ($\alpha = 2$) any local broadcasting algorithm requires $\Omega(\bar{\Delta} \log n)$ time, while on earth ($\alpha > 2$) better results are possible as illustrated by our $O(\bar{\Delta} + \log n \log\log n)$-time algorithm.

⋆ Part of this work has been performed while the author was a postdoctoral fellow at a joint program of the Simons Institute at UC Berkeley and I-CORE at Weizmann Institute.

⋆⋆ Supported in part by the Israel Science Foundation (grant 1549/13) and the I-CORE program of the Israel PBC and ISF (grant 4/11).

C. Scheideler (Ed.): SIROCCO 2015, LNCS 9439, pp. 164–178, 2015.
DOI: 10.1007/978-3-319-25258-2_12

1 Introduction

Setting and Problems. We consider the SINR (Signal-to-Interference-plus-Noise-Ratio) wireless setting with uniform power. In this setting a set V of n processors (also called *vertices*) is placed on the plane in an arbitrary manner. The vertices perform local computations and send messages. A message sent from a vertex $x \in V$ to a vertex $y \in V$ successfully arrives if the transmitting signal is sufficiently strong with respect to the noise produced by other processors plus ambient noise. We assume all vertices transmit with the same fixed transmission power P, so the signal of x experienced at y depends only on the distance between x and y (denoted by d_{xy}), and on the path-loss exponent (denoted by α). The signal strength decreases as an inverse polynomial of the distance, where the polynomial degree is α. Specifically, the signal strength of x experienced at y is P/d_{xy}^{α}. Similarly, the noise level of another transmitting vertex $v \in V$ experienced at y is P/d_{vy}^{α}. Let $U \subseteq V$ be a set of processors that transmit in parallel, and $V \setminus U$ be the rest of the processors (the receivers). Whether a receiver $y \in V \setminus U$ succeeds in hearing the sender $x \in U$ is determined by the *SINR formula:*

$$\frac{P/d_{xy}^{\alpha}}{N + \sum_{v \in U \setminus \{x\}} P/d_{vy}^{\alpha}} \geq \beta.$$

Here N is the ambient noise, and $\beta \geq 1$ is the threshold for successful reception. The parameters α, β and N are constants whose values are defined by the environment. We assume that $\alpha \geq 2$, which is the case in practice, unless it is reduced artificially. Specifically, in vacuum it holds that $\alpha = 2$, and on earth it holds that $\alpha > 2$. The value of α usually ranges between 2 and 6.

The maximum transmission range R of a vertex is the maximum range to which a vertex can transmit if it is the only transmitter in the network (i.e., the only noise is N). Note that when the power level is the same for all vertices, R is the same for all vertices as well. Let Δ denote the maximum number of vertices in any disk of radius R centered at a vertex $v \in V$. Let $\rho < 1$ be a positive constant that is arbitrarily close to 1. We define $\bar{R} = \rho \cdot R$, and $\bar{\Delta}$ to be the maximum number of vertices in any disk of radius $\bar{R}$ centered at a vertex $v \in V$. We note that for a pair of vertices u, v that are exactly at distance R one from another, a successful transmission requires all other vertices in the network to be silent. Therefore, in order to allow parallel transmissions, we define a successful *local broadcasting of a vertex* as a transmission that is successfully received by all vertices within radius $\bar{R}$ rather than R.

The SINR setting has attracted considerable attention due to its more realistic assumptions comparing to other models, such as the radio network model or the unit disk graph model, which do not take into account the cumulative nature of interference. One of the most fundamental problems in the SINR setting is *local broadcasting.* The goal in this problem is to establish a schedule in which each vertex $u \in V$ successfully transmits to all vertices at distance at most $\bar{R}$ from u. We will henceforth refer to vertices at distance at most $\bar{R}$ from u as *neighbors* of u. Note that the problem requirement is that for each vertex there exists a transmission that is successfully received by all its neighbors. This is stronger than requiring that each vertex succeeds to deliver a message to all its neighbors, since such a delivery could be achieved by several transmissions, each covering a subset of neighbors. The latter requirement is sometimes referred to as *weak local broadcasting*. Although weak local broadcasting can be often used instead of (strong) local broadcasting, the disadvantage of weak local broadcasting is in the greater power consumption resulting from multiple transmissions. Therefore, the strong variant of local broadcasting is preferred, and this is the variant considered in the current paper.

The local broadcasting problem serves as a building block for many network tasks, and has numerous applications. One of the most notable applications is *Single Round Simulation*. Specifically, if we are given an algorithm that is designed for wired networks or networks with no interference, it can be simulated in the SINR setting using local broadcasting as follows. Each round of the original algorithm is simulated by performing local broadcasting in the SINR setting. Consequently, each vertex succeeds do communicate with all its neighbors, which make it possible to execute a single round of the original algorithm. If the original algorithm requires $T(n)$ time and local broadcasting requires $S(n)$ time, then the overall simulation time is $T(n) \cdot S(n)$.

Since the running time of local broadcasting affects significantly the time of tasks that employ it, designing efficient local broadcasting algorithms is crucial. There has been an intensive thread of research in this direction. The problem was introduced by Goussevskaia, Moscibroda and Wattenhofer [7] who studied several scenarios. In the harshest scenario the vertices are unaware of neighborhood sizes and do not have *feedback* on the success of a transmission (i.e., they cannot tell whether a transmission has successfully received by all their neighbors). For this scenario, an algorithm with time[1] $O(\bar{\Delta} \log^3 n)$ was devised in [7]. Later it was improved in a series of works due to Yu et al. [13,14], where currently the best known algorithm has time $O(\bar{\Delta} \log n + \log^2 n)$ [13]. On the other hand, several researchers have observed that by considering slightly less harsh settings, one can improve the performance of the algorithms significantly. Moreover, these slightly stronger settings are still feasible for practical use. Although they may require more advanced devices, such as a carrier-sense mechanism that measures signal strength, they still can be implemented in hardware at a reasonable cost [9]. Already in the work of [7] it was observed that if vertices have knowledge about their neighborhood size, then the running time of the $O(\bar{\Delta} \log^3 n)$-time algorithm can be improved to $O(\bar{\Delta} \log n)$. Another result of this nature was obtained by Halldórsson and Mitra [9] who showed that in networks with free feedback (but with other properties that are similar to the harshest setting) the running time becomes $O(\bar{\Delta} + \log^2 n)$. In the current work we continue this line of

[1] All running times mentioned in our paper refer to randomized algorithms and hold *with high probability*, unless stated otherwise. *High probability* is $1 - 1/n^c$, for an arbitrarily large constant c. Note that if we are given $O(n)$ independent events, each of which occurs with high probability, then the event that all of them occur holds with high probability as well.

research and devise significantly improved algorithms in settings that are slightly less harsh than the harshest setting.

Our Results. We devise a local broadcasting algorithm for networks with feedback requiring $O(\bar{\Delta}+\log n \log\log n)$ time. This improves the best previously-known result for networks with feedback that has running time $O(\bar{\Delta}+\log^2 n)$ [9]. Moreover, the running time of our algorithm is tight up to a $\log\log n$ factor, in view of the lower bounds $\Omega(\Delta)$ and $\Omega(\log n)$ [13]. In addition, it shows that the conjecture of [9] that the $\log^2 n$ term is necessary does not hold in some settings. (On the other hand, the conjecture may still be true in weaker settings, such as settings without feedback. This is an intriguing open problem.) We consider a slotted setting with simultaneous wake up. This is somewhat stronger than the settings of [7,9]. However, there are standard methods that allow to weaken these requirements [7]. Also, similarly to other works, we assume that vertices know (upper bounds on) n and $\bar{\Delta}$.

We also consider a closely related problem, called *distance-k coloring.* In this problem the goal is to color the vertices with $O(\bar{\Delta})$ colors, such that each pair of neighbors at *geometrical distance* at most $k \cdot \bar{R}$ from one another are assigned distinct colors. If k is a constant, then a distance-k coloring with $O(\bar{\Delta})$ colors always exists. If k is a sufficiently large constant, this coloring constitutes a feasible SINR schedule. Note, however, that this problem is more challenging than k-hop coloring in which the goal is to obtain a coloring such that any pair of vertices at *graph distance* at most k have distinct colors. Indeed, any distance-k coloring is a k-hop coloring, but not vice versa. Moreover, the vertices are not always able to compute a distance-k coloring. For example, if two vertices are at distance greater than R from one another, they may not be able to communicate. On the other hand, a distance-k coloring requires them to select distinct colors, which cannot be achieved without communication when the required probability is sufficiently large. To address this problem we propose to employ *helper vertices*, also known as Steiner points. We employ an optimal number of Steiner points (up to constant factors), and obtain a distance-k coloring with $O(\bar{\Delta})$ colors in $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ time. This coloring gives rise to an optimal SINR schedule of length $O(\bar{\Delta})$ after a preprocessing stage of $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ time.

An interesting question deals with the influence of the path-loss exponent α on the performance of algorithms. Intuitively, a lower path-loss exponent means less obstacles and better signal strength. Moreover, from the point of view of each vertex, its signal should be as strong as possible in order to allow a successful transmission. Therefore, it seems reasonable that a lower path-loss exponent implies a better SINR schedule. In other words, transmitting in vacuum where $\alpha = 2$ should be the option for best performance. Surprisingly, we prove that the opposite is true! Specifically, any feasible SINR schedule for an environment with $\alpha = 2$ has length $\Omega(\bar{\Delta} \log n)$. This is an unconditional lower bound, no matter how strong the setting is. We present a network in which any shorter schedule will certainly fail. Hence we illustrate a gap between settings with $\alpha = 2$, and settings with $\alpha > 2$, where an $O(\bar{\Delta} + \log n \log\log n)$-schedule can be achieved. This interesting phenomenon can be explained by noting that obstacles do not only weaken signals - they also weaken noise. Our findings demonstrate that modifying α may affect noise more significantly than transmission signals. Therefore, in some occasions it might be better to add obstacles in order to block noise, instead of removing them in order to strengthen the signal.

Our Techniques. The main idea of our local broadcasting algorithm is gradually reducing the sizes of neighborhoods. In other words, as the algorithm proceeds, more and more vertices succeed and terminate. Consequently, the remaining vertices have less competition, and they are able to perform transmission trials more intensively. Specifically, in each phase consisting of $O(\bar{\Delta})$ rounds a constant fraction of vertices in each neighborhood terminates, with high probability. (A vertex terminates once it has successfully transmitted to all its neighbors.) This reduces the bound $\bar{\Delta}$ on the maximum neighborhood size, which allows to execute each phase more efficiently than the previous one. These improvements, however, are only possible as long as $\bar{\Delta} > \log n$. Once $\bar{\Delta}$ reaches $\log n$ we cannot proceed in the same way, since the probability that all sizes of neighborhoods are reduced is no longer large enough. Hence we switch to another method that increases the number of trials after each phase. Although the number of trials becomes greater than $\bar{\Delta}$ it is still bounded by $O(\log n)$ per phase. The number of these phases is $O(\log \log n)$, contributing a factor $O(\log n \log \log n)$ to the running time. This is in addition to the $O(\bar{\Delta})$ term for the first part of the algorithm.

We employ our local broadcasting algorithm in order to compute distance-k colorings using Steiner points. Once appropriate Steiner points are deployed we can make sure that any pair of vertices in the network can communicate (not necessarily directly). We observe that the resulting communication graph G is a unit disk graph. Moreover, the graph G^k obtained by adding an edge between any pair of vertices at *geometrical distance* $O(\bar{R} \cdot k)$ from one another has bounded growth. We then employ an algorithm due to Schneider and Wattenhofer for $O(\bar{\Delta})$-coloring graphs with bounded growth in $O(\log^* n)$ time [11]. Invoking it on G^k results in the desired distance coloring. This algorithm, however, is designed for networks with no interference. Nevertheless, each round of the algorithm can be simulated using local broadcasting. More precisely, $O(k)$ executions of local broadcasting are required in order to propagate a message to distance k. The propagation is possible thanks to the Steiner points. Consequently, a single round of the algorithm of [11] is simulated within $O(k \cdot (\bar{\Delta} + \log n \log \log n)) = O(\bar{\Delta} + \log n \log \log n)$ rounds, since k is a constant. Thus we obtain an overall running time $O(\log^* n \cdot (\bar{\Delta} + \log n \log \log n))$. For a sufficiently large constant k, we show that all vertices of the same color can transmit in parallel in the SINR setting without interference. Thus we obtain an SINR schedule of length $O(\bar{\Delta})$.

For our lower bound in the scenario when $\alpha = 2$ we consider a grid of vertices of size roughly $\sqrt{n} \times \sqrt{n}$. Our goal is to show that in any partition of vertices into $o(\bar{\Delta} \log n)$ subsets, there must be a subset that causes too much noise that results in a failure of some transmission. By calculating the overall noise of all n vertices we conclude that whenever the length of a schedule is too short, there must be a subset generating noise that is too strong. We then show that this noise necessarily disturbs a certain transmission. Hence, in any $o(\bar{\Delta} \log n)$-schedule there must be a round in which the noise is too strong at a certain vertex that tries to receive a message. Consequently, at least one vertex will fail during the transmission.

Related Work. In their pioneering work Goussevskaia, Moscibroda and Wattenhofer [7] devised a local broadcasting algorithm with time $O(\bar{\Delta} \log n)$ when neighborhood sizes are known, and $O(\bar{\Delta} \log^3 n)$ time when the sizes are unknown. For the latter scenario, Yu et al. obtained improved local broadcasting algorithms that require $O(\bar{\Delta} \log^2 n)$ time [14] and $O(\bar{\Delta} \log n + \log^2 n)$ time [13]. By using carrier-sense (a mechanism that allows receiving feedback) Yu et al. [14] obtained an algorithm with time $O(\bar{\Delta} \log n)$. An improved algorithm for the latter scenario of networks with feedback was devised by Halldórsson and Mitra [9]. The running time of the algorithm of [9] is $O(\bar{\Delta} + \log^2 n)$. The feedback mechanism of [9] is similar to the one used in the current paper.

Several works obtained the optimal (up to constant factors) $O(\bar{\Delta})$-schedule at the expense of performing a preprocessing stage, and employing some additional mechanisms that are not available in the weaker settings mentioned above. Specifically, Derbel and Talbi [3] perform preprocessing of $O(\bar{\Delta} \log n)$ time and employ power-level adjustments. Jurdzinski and Kowalski [10] perform preprocessing of $O(\bar{\Delta} \log^3 n)$ time, do not require power-level adjustments, but require location information. In the latter setting, a better result was obtained recently by Fuchs and Wagner [5] whose algorithm has $O(\bar{\Delta} \log n)$ preprocessing time. Note that the result of [10] is deterministic, while the other results are randomized. It is natural to compare these results with our new randomized algorithm that obtains $O(\bar{\Delta})$-schedule with $O(\log^* n \cdot (\bar{\Delta} + \log n \log \log n))$ preprocessing time. Instead of employing power-level adjustments or location-information mechanisms, our algorithm employs Steiner points in networks with feedback. This allows us to break the $O(\bar{\Delta} \log n)$ barrier in the preprocessing time, and outperform the running time of the above-mentioned algorithms.

The problem of $O(\bar{\Delta})$-coloring is closely related to local broadcasting, and has been intensively studied in the SINR model as well. However, it is weaker than local broadcasting in the following sense. Given a feasible local-broadcasting schedule, no two vertices of the same neighborhood transmit in the same time. Therefore, all vertices that transmit in the same time form a proper color class. On the other hand, given a proper coloring, all vertices of the same color will not necessarily be able to transmit in parallel. In order to allow this, a geometrical distance-k coloring is required. Still, $O(\bar{\Delta})$-coloring has attracted much attention. Derbel and Talbi [3] devised an $O(\bar{\Delta})$-coloring algorithm with $O(\bar{\Delta} \log n)$ time, and a distance-coloring algorithm with the same time that requires power-level adjustments. Yu et al. [15] devised a $(\bar{\Delta}+1)$-coloring algorithm that requires power-level adjustments and runs in $O(\bar{\Delta} \log n + \log^2 n)$ time. They also devised an algorithm that does not require power-level adjustments and has running time $O(\bar{\Delta} \log^2 n)$. Fuchs and Prutkin [4] obtained a $(\bar{\Delta}+1)$-coloring in $O(\bar{\Delta} \log n)$ time. Coloring problems have been very intensively studied in additional settings, such as wireless radio networks and networks without interference. The best currently-known $(\Delta + 1)$-coloring algorithm for radio networks has time $O(\Delta + \log^2 n)$ [12]. The best currently-known $(\Delta + 1)$-coloring algorithm for networks without interference has running time $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ [2]. For an extensive overview of distributed coloring algorithms we refer the reader to [1].

2 Local Broadcasting in Networks with Feedback

In this section we devise a local broadcasting algorithm for networks with feedback that requires $O(\bar{\Delta} + \log n \log \log n)$ time. We start with the following claim. Suppose that all vertices $v \in V$ perform trials in which each vertex transmits with probability $1/(c \cdot \bar{\Delta})$, and listens with probability $1 - 1/(c \cdot \bar{\Delta})$, for a sufficiently large constant c. Then a transmitting node successfully performs its local broadcasting, with probability at least $1/2$. This is similar to a phenomenon observed in [7]. (We omit its prof from the current paper due to lack of space.) We refer to the set of vertices at distance at most $\bar{R}$ from the vertex $v \in V$ (excluding v) as the *neighborhood* of V, and denote it by $\Gamma_{\bar{R}}(v)$.

Lemma 1. *For a sufficiently large constant c, suppose that all vertices perform transmissions with probability $1/(c \cdot \bar{\Delta})$. Then a transmission of a sender $v \in V$ is successfully received in v's neighborhood $\Gamma_{\bar{R}}(v)$, namely, within radius $\bar{R}$ from v, with probability at least $1/2$.*

Next, we devise a procedure called *Feedback-Broadcasting* for performing local broadcasting in networks with feedback, namely, networks in which any vertex $v \in V$ can decide whether a transmission was successfully received by all vertices in its neighborhood $\Gamma_{\bar{R}}(v)$. The procedure consists of two phases. In the first phase, vertices repeatedly perform the following trials: each vertex transmits with probability $1/(c \cdot \bar{\Delta})$ for $\hat{c} \cdot \bar{\Delta}$ times, where $\hat{c} > c$ is a sufficiently large constant. If a vertex v discovers (using the feedback mechanism) that it has succeeded to transmit to its entire neighborhood $\Gamma_{\bar{R}}(v)$, then v terminates. If v has failed in all these $\hat{c} \cdot \bar{\Delta}$ trials, then it updates the bound on $\bar{\Delta}$ by setting $\bar{\Delta} := \frac{1}{2} \cdot \bar{\Delta}$, and performs another stage of $\hat{c} \cdot \bar{\Delta}$ trials. This continues as long as $\bar{\Delta} > \log n$, and then the first phase of the procedure terminates.

In the second phase, it holds that $\bar{\Delta} \leq \log n$, with high probability. This phase consists of $O(\log \log n)$ stages, each of which consists of $O(\log n)$ trials in which each vertex transmits with probability $1/(c \cdot \bar{\Delta})$. In the end of each stage, all unsuccessful vertices update $\bar{\Delta}$ by setting $\bar{\Delta} = \frac{1}{2} \cdot \bar{\Delta}$. We later prove that once the second stage has been completed, all vertices have succeeded with high probability. Next, we provide the pseudocode of the procedure. (Note that $\Gamma_{\bar{R}}(v)$ denotes all neighbors of v including those that have terminated. In other words, the feedback in lines 8 and 24 of the algorithm has to be received for all neighbors, namely, the active and the terminated ones.)

Algorithm 1. Procedure Feedback-Broadcasting($V, \bar{\Delta}$) (code for vertex $v \in V$)

Let $c, \hat{c}$ be sufficiently large constants, and $\hat{c} > c$.

```
success := F
(* Phase 1 *)
while Δ̄ > log n do
   (* Stage k *)
   for i = 1, 2, ..., ĉ · Δ̄ do
      (* trial i of stage k *)
      transmit with probability 1/(c · Δ̄)
      if all neighbors of v in Γ_R̄(v) receive the transmission successfully then
         success := T
      end if
   end for
   if success = T then
      terminate
   else
      Δ̄ := ⌊(1/2) · Δ̄⌋
   end if
end while
(* Phase 2 *)
for k = 1, 2, ..., ⌊log log n⌋ do
   (* Stage k *)
   for i = 1, 2, ..., ⌊ĉ · log n⌋ do
      (* trial i of stage k *)
      transmit with probability 1/(c · Δ̄)
      if all neighbors of v in Γ_R̄(v) receive the transmission successfully then
         success := T
      end if
   end for
   if success = T then
      terminate
   else
      Δ̄ := max{⌊(1/2) · Δ̄⌋, 1}
   end if
end for
```

We say that a vertex is *active* if it has not terminated yet. The invariant that the algorithm attempts to preserve is $\texttt{Bound}(\bar{\Delta}) \equiv$ "the parameter $\bar{\Delta}$ is an upper bound on the maximum neighborhood size (counting only active vertices)". The correctness of the algorithm follows from the observation that this invariant holds at all stages of the algorithm, with high probability. This observation, in turn, follows from the fact that in each stage the number of active neighbors of each vertex is reduced by a factor of 1/2. We prove this in the next lemma.

Lemma 2. *Suppose that Procedure Feedback-Broadcasting is invoked by all vertices with a parameter* $\bar{\Delta}$ *that satisfies* $\texttt{Bound}(\bar{\Delta})$. *Then the invariant* $\texttt{Bound}(\bar{\Delta})$ *holds throughout the entire execution, with high probability.*

Proof. The assertion holds trivially in the beginning of the execution of the procedure. We have to prove that each time the value of $\bar{\Delta}$ is updated, its new value indeed satisfies $\texttt{Bound}(\bar{\Delta})$. Note that $\bar{\Delta}$ is updated only in the end of a stage. (Lines 4 - 16 constitute a stage of Phase 1; lines 20 - 32 are a stage of Phase 2.) We start by analyzing the first phase (lines 2 - 17). Assuming that $\texttt{Bound}(\bar{\Delta})$ holds in the beginning of stage k of Phase 1, we show that in the end of stage k, $\texttt{Bound}(\bar{\Delta})$ still holds, namely, for each vertex $v \in V$, the number of neighbors of v that are still active is at most $\frac{1}{2} \cdot \bar{\Delta}$, with high probability.

Suppose that in the beginning of stage k the number d of active neighbors of v is at least $\frac{1}{2} \cdot \bar{\Delta}$. (Otherwise, the assertion holds already in the beginning of the stage, and will hold in the end of the stage since the number of active neighbors can only decrease.) Let X_i, $i = 1, 2, ..., \hat{c} \cdot \bar{\Delta}$, be a random indicator variable that equals 1 if a neighbor of v succeeds in trial i of stage k, and 0 otherwise, and let $X = \sum_{i=1}^{\hat{c} \cdot \bar{\Delta}} X_i$. The probability that exactly one neighbor of v tries in trial i is $d \cdot (c \cdot \bar{\Delta} - 1)^{d-1} / (c \cdot \bar{\Delta})^d > d/(4 \cdot c \cdot \bar{\Delta})$. Thus, by lemma 1, the probability that it succeeds is at least $d/(8 \cdot c \cdot \bar{\Delta})$. Therefore, $\mathbb{E}(X) \geq (\hat{c}/c) \cdot d/8$. By Chernoff bound, as the X_i's are independent,

$$Pr(X < \mathbb{E}(X)/2) \leq e^{-\mathbb{E}(X)/8} \leq e^{-(\hat{c}/c) \cdot d/64}.$$

In other words, for a sufficiently large constant $\hat{c}$, we can obtain at least $2d$ successful trials in a stage, with high probability. (Recall that $d > \frac{1}{2}\bar{\Delta} > \frac{1}{2} \log n$.) However, the trials were performed with repetitions, and thus, the number of successful neighbors may be smaller than d. Next, we analyze the probability that it is smaller than $d/2$. Since in each iteration the vertices have equal chances of performing a trial, this problem is equivalent to balls-into-bins, where vertices are bins and successful trials are balls. The value $2d$ denotes the number of balls, d denotes the number of bins, and we would like to analyze the probability that more than $d/2$ bins contain balls. We calculate the probability of the complementary event, i.e., that at most $d/2$ bins contain balls. This probability is at most $\binom{d}{d/2} \cdot (1/2^{2d}) < (1/2^d)$. Note that by increasing the constant $\hat{c}$ we can have an arbitrarily large constant multiplicative factor, instead of the factor 2 in the term $2d$. Since $d = \Omega(\log n)$, at least $d/2$ neighbors succeed, with probability $1 - 1/poly(n)$. By the union bound, for all vertices, all neighborhoods are (at least) halved, with high probability. Thus the size of the maximum neighborhood is reduced by a factor of at least 2 in each stage of the first phase, with high probability.

Consequently, within $O(\log \bar{\Delta})$ stages of Phase 1, the maximum neighborhood size becomes at most $\log n$, with high probability. Therefore, once Phase 2 (lines 18 - 33) starts, it holds that $\bar{\Delta} \leq \log n$ is an upper bound on the maximum neighborhood size, as required. Denote again by d the number of active neighbors of a vertex $v \in V$ that is still active. In each trial of a stage of Phase 2 (lines 22 - 26), the probability that exactly one active neighbor of v succeeds is at least $d/(4 \cdot c \cdot \bar{\Delta}) = \Omega(1)$, if $d \geq \bar{\Delta}/2$. Consequently, the expected number of successful trials is $\mathbb{E}(X) = \Omega(\log n)$, where the constant hidden in the Ω-notation can be made as large as desired by choosing a sufficiently large constant

$\hat{c}$. Thus, by Chernoff bound, the number of successful trials is $\Omega(\log n)$, with high probability. Next, we analyze the probability that at least $d/2$ different neighbors have succeeded. Again we reduce the problem to balls-to-bins, where here we have $\Omega(\log n)$ balls and d bins. Therefore, this probability is $1 - \binom{d}{d/2} \cdot (1/2^{\Omega(\log n)}) \geq 1 - \binom{\log n}{(\log n)/2} \cdot (1/2^{\Omega(\log n)})$, i.e., high probability, for a sufficiently large constant $\hat{c}$. Using the union bound we obtain this result for all vertices, and thus the maximum neighborhood size is at least halved in each stage, with high probability. Hence throughout the entire execution $\bar{\Delta}$ is an upper bound on the maximum neighborhood size, with high probability. □

By Lemma 2, within $\log \log n - 1$ stages of Phase 2 the neighborhood size of all vertices becomes $O(1)$, with high probability. In stage $\lfloor \log \log n \rfloor$ of the second phase each active vertex succeeds with a constant probability since $\bar{\Delta} = O(1)$. (See Lemma 1.) Therefore, within $O(\log n)$ iterations of this stage all remaining active vertices succeed, with high probability. Thus we obtain the following result.

Theorem 1. *Procedure Feedback-Broadcasting performs a successful local broadcasting of all vertices, with high probability.*

Next we analyze the running time of the procedure. Each stage of the first phase requires $O(\bar{\Delta})$ time. However, $\bar{\Delta}$ is halved in each stage, and thus the overall running time of the first phase is $O(\bar{\Delta} + \bar{\Delta}/2 + \bar{\Delta}/4 + ...) = O(\bar{\Delta})$. The second phase requires $O(\log n \log \log n)$ time. Hence the overall running time is $O(\bar{\Delta} + \log n \log \log n)$.

Theorem 2. *Local broadcasting in networks with feedback can be performed in* $O(\bar{\Delta} + \log n \log \log n)$ *time.*

3 Distant Coloring

Our local-broadcasting algorithm produces an $O(\bar{\Delta} + \log n \log \log n)$ time schedule. In other words, a distributed algorithm for networks without interference can be simulated in SINR networks, where each round of the original algorithm is simulated by $O(\bar{\Delta} + \log n \log \log n)$ rounds of the local-broadcasting procedure. This is, however, not optimal, since a schedule of length $O(\bar{\Delta})$ always exists. It is easy to verify that the latter bound is the best possible (up to constant factors). Indeed, given a vertex v, in order to receive the messages of all the $\bar{\Delta}$ vertices at distance at most $\bar{R}$ from v, each of them must transmit in a distinct round.

A schedule of length $O(\bar{\Delta})$ can be obtained by computing a *distance-k coloring*, for a sufficiently large constant k. In this coloring each pair of vertices at (geometrical) distance less than $k \cdot \bar{R}$ from one another are colored by distinct colors. Since the number of vertices in each disk of radius $k \cdot \bar{R}$ is $O(\bar{\Delta})$, a distance-k coloring can always employ $O(\bar{\Delta})$ colors, for any constant k. Unfortunately, it is impossible to obtain such a coloring in the SINR setting (as will be explained shortly), even though it is possible to achieve a *k-hop-coloring*, namely, a coloring in which any pair of vertices within at most k hops from one another are colored by distinct colors. In models without interference a k-hop $O(\bar{\Delta})$-coloring can be computed in $O(\log^* n)$ time for any constant k. This is done by computing an $O(\bar{\Delta})$-coloring of growth-bounded graphs on G^k. Since G is a unit disk graph, G^k is of bounded growth. The running time of the algorithm is $O(\log^* n)$ [11]. Consequently, in SINR networks a k-hop-coloring can be computed within $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ rounds by performing single-round simulations. (See Theorem 2.) This, however, may increase message size by a factor of poly$(\bar{\Delta})$ as a consequence of simulating G^k.

Corollary 1. *A k-hop coloring can be computed in $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ rounds (with high probability) in the SINR setting with uniform power.*

However, a k-hop-coloring does not necessarily produce a feasible SINR schedule. Consider, for instance, three vertices a,b,c, such that $\text{dist}(a, b) = \bar{R}$, and $\text{dist}(a, c) = \text{dist}(b, c) = \bar{R} + \epsilon$, for some $\epsilon > 0$. Then $\varphi(a) = 1$, $\varphi(b) = 2$, $\varphi(c) = 1$ is a proper k-hop coloring for any k, since c cannot receive messages from a and b. On the other hand, by the SINR formula, if a and c transmit simultaneously, they cause interference that prevents b from receiving the message of a. Thus a distance-k coloring is desirable. But it cannot be computed since a and b cannot communicate with c, and cannot make sure they all select distinct color. To solve this problem we propose to use *Steiner points*. In other words, we add some helper vertices that allow to compute a distance-k coloring of the original vertex set. For each original vertex, we add $O(k)$ Steiner vertices in the way illustrated in Figure 1(a). These Steiner vertices have exactly the same status as that of the original vertices of V, i.e., a Steiner vertex is a processor with a transmitter and a receiver. Note that as a result $\bar{\Delta}$ increases only by a multiplicative constant factor of at most 5. Let V' denote the new set of vertices, including vertices of V.

We compute a $2k$-hop-coloring of V' by invoking the algorithm of Corollary 1. We next prove that it results in a *distance-k* coloring of V that employs $O(\bar{\Delta})$ colors.

Lemma 3. *A $2k$-hop coloring of V' is a distance-k coloring of V that employs $O(\bar{\Delta})$ colors.*

Proof. The number of vertices of V' in any disk of radius $\bar{R}$ is at most five times the number of vertices of V in this disk. Consequently, the number of employed colors is $O(5\bar{\Delta}) = O(\bar{\Delta})$. Let $u, v \in V$ be two vertices at distance at most $\bar{R} \cdot k$ from one another. Then there exists a path of at most $2k$ vertices connecting u and v, such that each pair of neighboring vertices on the path are at distance at most $\bar{R}$ from one another. (See Figure 1(b).) Consequently, u and v are colored by distinct colors by the $2k$-hop-coloring algorithm. □

By Lemma 3 and Corollary 1 we obtain the following result.

Theorem 3. *A distance-k coloring of V can be obtained within $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ rounds, with high probability, using at most $4k \cdot n$ Steiner points.*

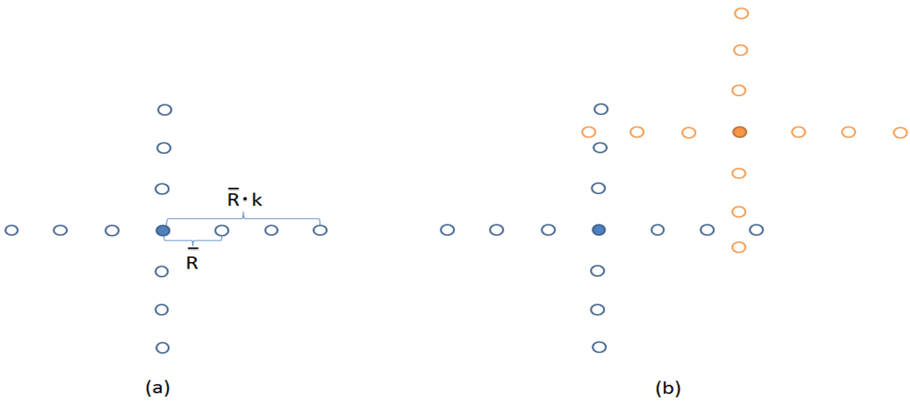

Fig. 1. (a) The vertex in the center is $v \in V$, and it is surrounded by $4k$ Steiner points. (b) If u and v are not too far from one another, there is a path connecting u and v.

Consider all vertices colored by the same color z of a distance-k coloring, for a sufficiently large constant k. Let v be such a vertex. Then the number of vertices in L_i (the ith ring of width $\bar{R}$ around v) whose color is z is $O(i)$. The constant in the O-notation can be made as small as one wishes, by increasing k. As a result, by similar arguments to those in the analysis of Lemma 1 (see also [7]), the interference in the disk of radius $\bar{R}$ of v is sufficiently small to allow a successful broadcast of v. Specifically, for a sufficiently large constant k, no vertices except for v transmit in L_1 and L_2. Hence, the interference I_1 experienced by neighbors of v, i.e., by vertices in L_1 is at most

$$I_1 = \sum_{i=3}^{\infty} P \cdot O(i) \cdot 1/(\bar{R}(i-2))^{\alpha} \;\leq\; \sum_{i=3}^{\infty} (1/\bar{R})^{\alpha} \cdot P \cdot O((i-2)^{\alpha-1})$$
$$= (1/\bar{R}^{\alpha}) \cdot P \cdot \sum_{i=1}^{\infty} O(1/i^{\alpha-1}) \;=\; (1/\bar{R}^{\alpha}) \cdot P \cdot O((\alpha-1)/(\alpha-2)),$$

where the constant hidden in the O-notation can be made as small as one wishes. In other words, $I_1 \leq \epsilon \cdot P/\bar{R}^{\alpha}$, for an arbitrarily small constant $\epsilon > 0$. This interference is sufficiently small to allow all vertices at distance at most $\bar{R}$ from v to receive the message of v. Consequently, if all vertices of the same color in the k-hop coloring (and only them) transmit simultaneously, they all succeed. Thus Theorem 3 implies the following result.

Theorem 4. *A schedule of length $O(\bar{\Delta})$ can be obtained within $O(\log^* n \cdot (\bar{\Delta} + \log n \log\log n))$ rounds, with high probability, using at most $4k \cdot n$ Steiner points.*

As noted earlier, the schedule length is optimal up to constant factors. Next we show that the number of Steiner points is optimal as well. Consider a vertex set V whose vertices are placed on a line, such that the distance between any pair of neighboring vertices is $k \cdot \bar{R}$. In order to compute a k-hop coloring, the graph induced by V' must be connected. The minimum number of vertices that must be added to V in order to satisfy this requirement is $(n-1)(k-1) = \Omega(k \cdot n)$, since for any pair of neighboring vertices on the line, at least $k-1$ vertices must be placed between them. This is summarized in the next Theorem.

Theorem 5. *The number of Steiner points required for k-hop coloring is $\Omega(kn)$.*

4 A Lower Bound for $\alpha = 2$

In this section we prove that if the path loss exponent α equals 2, then any feasible schedule in the SINR model has length $\Omega(\bar{\Delta} \log n) = \Omega(\Delta \log n)$. To this end consider a grid of size $k \cdot k = n$ of vertices, such that the distance between any vertex and its closest neighbors on the X-axis and Y-axis is exactly one unit. The dimensions of the square containing this grid is $(k-1)\times(k-1)$. Let v be a vertex in a corner of the grid. Suppose that all other vertices $u \in V \setminus \{v\}$ transmit, and let $\bar{R} \geq 1$ be a parameter defining the transmission range, as in Section 2. (Note that $\bar{\Delta} = \Theta(\bar{R}^2)$). Denote by t_u the interference experienced by v as a result of the transmission of u. Then the overall interference experienced by v is at least the interference I_{far} caused by the subset $\bar{V}$ of vertices at distance greater than $\bar{R}$ from v. This interference satisfies

$$\begin{aligned} I_{far} = \sum_{u \in \bar{V}} t_u &= \sum_{u \in \bar{V}} P/d_{uv}^{\alpha} = \sum_{u \in \bar{V}} P/d_{uv}^2 \\ &\geq \sum_{i=\bar{R}+1}^{k-1} P \cdot (2i+1)/(2i^2) = \sum_{i=1}^{k-1} P \cdot (2i+1)/(2i^2) - \sum_{i=1}^{\bar{R}} P \cdot (2i+1)/(2i^2). \end{aligned}$$

The last inequality follows from the observation that the number of vertices on the boundary of a square of size $(i+1) \times (i+1)$ that do not belong to the inner square of size $i \times i$ is $i + i + 1 = 2i + 1$. On the other hand, each such vertex is at distance at most $\sqrt{2}i$ from v.

Consequently, the interference experienced by v as a result of the transmissions of all other vertices is at least $P \cdot \sum_{i=\bar{R}+1}^{k-1} 1/i$. Whenever $\bar{R} \leq k^{1-\epsilon}$, for an arbitrarily small constant $\epsilon > 0$, we have $P \cdot \sum_{i=\bar{R}+1}^{k-1} 1/i = \Omega(P \cdot \log k - P \cdot \log \bar{R}) = \Omega(P \cdot \log n)$. This is summarized below.

Lemma 4. *Let v be a corner vertex and $\bar{V}$ be the set of vertices at distance greater than $\bar{R}$ from v. If all vertices in $\bar{V}$ transmit, then the interference experienced by v is $\Omega(P \cdot \log n)$.*

Next, Assume for contradiction that there exists a feasible SINR schedule of length $\ell = o(\bar{\Delta} \cdot \log n)$. Then, let $V_1, V_2, ..., V_\ell$ be a partition of $V \setminus \{v\}$, such that the vertices in each V_i, $i \in [\ell]$, can transmit successfully in parallel. Let $j \in [\ell]$ be the index of the set V_j, such that vertices of $V_j \cap \bar{V}$ cause the maximum interference at the corner vertex v. Then, by the Pigeonhole principle,

$$\sum_{u \in V_j \cap \bar{V}} t_u \geq \omega(P/\bar{\Delta}). \tag{1}$$

Note that $\bar{\Delta}$ depends on P linearly. Indeed, increasing P by a multiplicative factor of q results in an increase of the transmission range by $\sqrt{q}$, and thus the number of vertices at distance at most $\sqrt{q} \cdot \bar{R}$ becomes $\Theta(q \cdot \bar{\Delta})$. If we normalize P to be equal to 1 in the case of a transmission of an only-transmitting vertex to a distance of one unit, then $\omega(P/\bar{\Delta}) = \omega(1)$, for any P.

Let $w \in V_j \cap \bar{V}$ be the closest vertex to v. Let $y \in V \setminus V_j$ be a vertex at distance at least $\bar{R} - 1$ and at most $\bar{R}$ from w. See Figure 2. Let $M_w = (V_j \cap \bar{V}) \setminus \{w\}$.

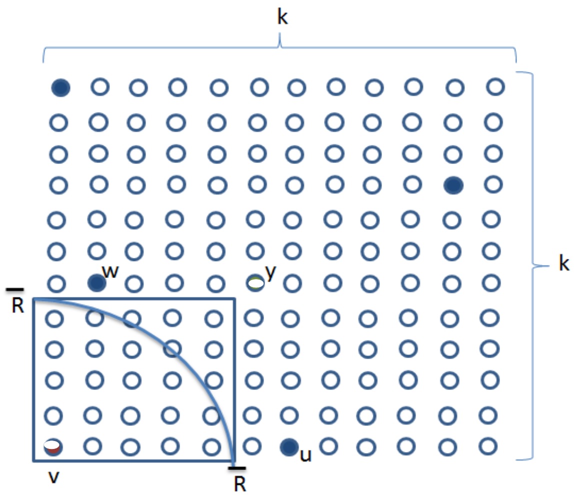

Fig. 2. The set of transmitting vertices $V_j \cap \bar{V}$ is depicted by filled circles.

Note that for any $u \in V_j \cap \bar{V}$ the distance between u and v is at least $1/\sqrt{2}$ the distance between u and w. Thus the interference experienced by w when vertices of $V_j \cap \bar{V}$ transmit is at least $\frac{1}{2} \sum_{u \in M_w} t_u$. The interference experienced by y is at least $\frac{1}{8} \sum_{u \in M_w} t_u$, since the distance between w and each vertex $x \in M_w$ is at least $1/2$ the distance between y and x. Thus, when all vertices of $V_j \cap \bar{V}$ transmit, the SINR formula that determines whether y receives the message of w successfully satisfies (for $\alpha = 2$)

$$\frac{P/d_{wy}^{\alpha}}{N + \sum_{u \in M_w} P/d_{uy}^{\alpha}} \leq \frac{P/(\bar{R}-1)^2}{N + \frac{1}{8} \sum_{u \in M_w} t_u} = \frac{\Theta(P/\bar{\Delta})}{\omega(P/\bar{\Delta})} = o(1) < 1,$$

as $t_w = \Theta(P/\bar{\Delta})$ and $\sum_{u \in M_w} t_u = \omega(P/\bar{\Delta}) - \Theta(P/\bar{\Delta}) = \omega(P/\bar{\Delta})$ by (1).

Hence y fails to receive the message of w, and thus $V_1, V_2, ..., V_\ell$ is not a feasible schedule; contradiction. In summary, we get the following theorem.

Theorem 6. *In settings with loss-path exponent $\alpha = 2$, any feasible SINR schedule with uniform power has length $\Omega(\Delta \log n)$.*

References

1. Barenboim, L., Elkin, M.: Distributed Graph Coloring: Fundamentals and Recent Developments. Morgan & Claypool Synthesis Lectures on Distributed Computing Theory (2013)
2. Barenboim, L., Elkin, M., Pettie, S., Schneider, J.: The locality of distributed symmetry breaking. In: Proc. 53rd Symp. on Foundations of Computer Science (FOCS 2012), pp. 321–330 (2012)
3. Derbel, B., Talbi, E.: Distributed Node Coloring in the SINR Model. In: Proc. 30th IEEE Int. Conf. on Distributed Computing Systems (ICDCS 2010), pp. 708–717 (2010)
4. Fuchs, F., Prutkin, R.: Simple Distributed $(\Delta + 1)$-coloring in the SINR model (2015), http://arxiv.org/abs/1502.02426
5. Fuchs, F., Wagner, D.: On Local Broadcasting Schedules and CONGEST Algorithms in the SINR Model. In: Proc. 9th Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2013), pp. 170–184 (2013)
6. Fuchs, F., Wagner, D.: Local broadcasting with arbitrary transmission power in the SINR model. In: Halldórsson, M.M. (ed.) SIROCCO 2014. LNCS, vol. 8576, pp. 180–193. Springer, Heidelberg (2014)
7. Goussevskaia, O., Moscibroda, T., Wattenhofer, R.: Local Broadcasting in the Physical Interference Model. In: Proc. 5th ACM Int. Workshop on Foundations of Mobile Computing (DialM-POMC 2008), pp. 35–44 (2008)
8. Goussevskaia, O., Pignolet, Y., Wattenhofer, R.: Efficiency of wireless networks: Approximation algorithms for the physical interference model. Foundations and Trends in Networking 4(3), 313–420 (2010)
9. Halldórsson, M., Mitra, P.: Towards Tight Bounds for Local Broadcasting. In: Proc. 8th ACM Int. Workshop on Foundations of Mobile Computing (FOMC 2012), Article No 2 (2012)
10. Jurdzinski, T., Kowalski, D.: Distributed Backbone Structure for Algorithms in the SINR Model of Wireless Networks. In: Proc. 26th Int. Symp. on Distributed Computing (DISC 2012), pp. 106–120 (2012)
11. Schneider, J., Wattenhofer, R.: A Log-Star Distributed Maximal Independent Set Algorithm For Growth Bounded Graphs. In: Proc. 27th ACM Symp. on Principles of Distributed Computing (PODC 2008), pp. 35–44 (2008)
12. Schneider, J., Wattenhofer, R.: Coloring unstructured wireless multi-hop networks. In: Proc. 28th ACM Symp. on Principles of Distributed Computing (PODC 2009), pp. 210–219 (2009)
13. Yu, D., Hua, Q., Wang, Y., Lau, F.: An $O(\log n)$ Distributed Approximation Algorithm for Local Broadcasting in Unstructured Wireless Networks. In: Proc. 8th Int. Conf. on Distributed Computing in Sensor Systems (DCOSS 2012), pp. 132–139 (2012)
14. Yu, D., Wang, Y., Hua, Q., Lau, F.: Distributed Local Broadcasting Algorithms in the Physical Interference Model. In: Proc. 2011 Int. Conf. on Distributed Computing in Sensor Systems (DCOSS 2011), pp. 1–8 (2011)
15. Yu, D., Wang, Y., Hua, Q.-S., Lau, F.C.M.: Distributed $(\Delta + 1)$-Coloring in the Physical Model. In: Erlebach, T., Nikoletseas, S., Orponen, P. (eds.) ALGOSENSORS 2011. LNCS, vol. 7111, pp. 145–160. Springer, Heidelberg (2012)