



Distributed bare-bones communication in wireless networks

Bogdan S. Chlebus¹  · Dariusz R. Kowalski¹ · Shailesh Vaya²

Received: 30 July 2018 / Accepted: 4 November 2021 / Published online: 25 November 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

We consider wireless networks operating under the SINR model of interference. Nodes have limited individual knowledge and capabilities: they do not know their positions in a coordinate system in the plane, further they do not know their neighborhoods, nor do they know the size of the network n , and finally they cannot sense collisions resulting from simultaneous transmissions by at least two neighbors. Each node is equipped with a unique integer name, where N as an upper bound on the a range of names. We refer as a backbone to a subnetwork induced by a diameter-preserving dominating set of nodes. Let Δ denote a maximum number of nodes that can successfully receive a message transmitted by a node when no other nodes transmit concurrently. We study distributed algorithms for communication problems in three settings. In the single-node-start case, when one node starts an execution and other nodes are awoken by receiving messages from already awoken nodes, we present a randomized broadcast algorithm that wakes up all nodes in $\mathcal{O}(n \log^2 N)$ rounds with high probability. For the synchronized-start case, when all nodes start an execution simultaneously, we give a randomized algorithm computing a backbone in $\mathcal{O}(\Delta \log^7 N)$ rounds with high probability. In the partly-coordinated-start case, when a number of nodes start an execution together and other nodes are awoken by receiving messages from the already awoken nodes, we develop an algorithm that creates a backbone in time $\mathcal{O}(n \log^2 N + \Delta \log^7 N)$ with high probability

Keywords Wireless network · Signal-to-interference-plus-noise ratio · Broadcast · Backbone

1 Introduction

We consider wireless networks in which the effects of interference are determined by the signal-to-interference-plus-noise ratio (SINR) model. The extent to which such networks can support distributed communication depends on nodes' capabilities, like the ability to detect signals' collisions (caused by two or more neighbors transmitting simultaneously), and on the information that can be used in codes of algorithms, such as coordinates of nodes in a Cartesian coordinate system in a plane. We demonstrate that efficient distributed communication can be carried out by a wireless network whose nodes have severely limited power.

We assume that the nodes of a wireless network are positioned in a two-dimensional Euclidean space. This is abstracted into an associated graph structure called the communication graph of the network. The nodes of the network

serve as vertices of the communication graph. A pair of such vertices (u, w) makes an edge in the communication graph if w can successfully receive a message transmitted by u when no other node transmits simultaneously, and vice versa. We use Δ to denote a maximum node-degree in a communication graph, and D to denote a diameter of this graph.

We want to distinguish certain vertices of a communication graph such that they together make a *backbone* of the graph. We call these vertices *backbone vertices*. Backbone vertices induce a subgraph of a constant degree in the communication graph that has its diameter asymptotically equal to that of the whole communication graph and such that every node outside the backbone is connected to some backbone vertex.

Nodes of a wireless network communicate directly subject to restrictions on the signal-to-interference-plus-noise ratio (SINR). A transmission is successfully received by a node depending on a ratio of the signal strength to the strength of other signals plus ambient noise, when evaluated at a receiver. Let \mathcal{T} be a set of nodes that transmit together at a round, and let two nodes v and u be such that $v \in \mathcal{T}$ and $u \notin \mathcal{T}$. The signal strength of v 's transmission as reach-

✉ Bogdan S. Chlebus
bchlebus@augusta.edu

¹ School of Computer and Cyber Sciences, Augusta University, Augusta, Georgia, USA

² Xerox Research Centre India, Bangalore, India

ing u is determined as $\mathcal{P}(v, u) = P_v \cdot \text{dist}(v, u)^{-\alpha}$, where $\text{dist}(v, u)$ is the distance between u and v , quantity P_v is the transmission strength, and $\alpha > 2$ is a path loss. The interference at u means $\mathcal{I}(v, u, \mathcal{T}) = \sum_{w \in \mathcal{T} \setminus \{v\}} \mathcal{P}(w, u)$. The quantity $\text{SINR}(v, u, \mathcal{T})$ is defined as $\text{SINR}(v, u, \mathcal{T}) = \mathcal{P}(v, u) / (\mathcal{N} + \mathcal{I}(v, u, \mathcal{T}))$, where $\mathcal{N} > 0$ is the ambient noise. For a node u to successfully receive a transmission by node v , it is necessary for the inequality $\text{SINR}(v, u, \mathcal{T}) \geq \beta$ to hold, where $\beta > 0$ is some threshold. We work with a SINR model that combines weak-connectivity assumptions, as formulated by Daum et al. [9], with weak-sensitivity (weak devices) assumptions, as formulated by Jurdziński et al. [26], see also Jurdziński and Kowalski [21]. In a weak-sensitivity setting, a node can never successfully receive a transmission from the borderline of its transmission range, while weak connectivity means that all links that could successfully transfer a message in a suitably favorable scenario are included as links of a network, see Sect. 2 for details. We also assume that nodes cannot detect collisions produced by interfering transmissions from at least two neighbors.

Algorithms are restricted in what information can be used in their codes. A network is said to be *bare-bones* when it is subject to a specific set of such restrictions, which are used in this paper and can be summarized as follows. Each node among the n nodes in a network has a unique name in the range $\{1, \dots, N\}$, for some positive integer $N \geq n$. Each node knows its name and the number N . Nodes do not know their neighbors in the communication graph but they know the parameter Δ .

We consider distributed algorithms to synchronize and organize a wireless network. The way a communication task is initiated impacts how a communication algorithm may be designed. We consider the following natural modes of initialization of communication tasks. If all the nodes begin an execution together in the same round, then such an execution is said to be performed from a *synchronized start*. Executing algorithms from an *single-node start* means that just a single node is awoken at the beginning of an execution, while all the other nodes do not send messages until they hear a message. A case of *multiple-nodes synchronized start* means that a group of nodes start an execution together, while the remaining nodes wait to be awoken by hearing messages. Synchronized start is a global configuration of a network that facilitates executing a communication algorithm with all nodes starting at the same time.

We consider implementations of three communication primitives useful in developing distributed algorithms. One primitive is to prepare a synchronized-start configuration from a single-node start. This means to make all nodes reach a “start” global state from which to begin in unison an execution of some algorithm. Another primitive is to synchronize start from a multiple-nodes synchronized start, which has a similar goal but subject to the stated different initial condi-

tions. These two tasks can be accomplished by a broadcast with a wake-up functionality, like coordinating round numbers at nodes. The third primitive is to build a backbone of the communication graph of a network, provided that a synchronized-start configuration has already been reached.

Next, we briefly summarize the results presented in this paper, and put them in the context of previous and related work.

A summary of the results.

We develop randomized distributed algorithms implementing certain communication primitives in wireless networks operating under the SINR regime of interferences among concurrent transmissions. Each of them has the property that a node uses only $\mathcal{O}(\log^3 N)$ random bits in an execution. The model is of weak-sensitivity and weak-connectivity.

We begin by developing a randomized distributed algorithm to perform a broadcast from a single-node start in $\mathcal{O}(n \log^2 N)$ expected time, where it is the source that starts as an activated node. Once an execution is completed, every node will have been activated and all the nodes have their round numbers synchronized. This may be used as a preparation to begin an execution of a distributed communication algorithm from a synchronized start.

We present two randomized algorithms to build a backbone. One of them completes the task from a synchronized start in $\mathcal{O}(\Delta \log^7 N)$ rounds with high probability. The other one creates a backbone from multiple-nodes coordinated-start in $\mathcal{O}(n \log^2 N + \Delta \log^7 N)$ rounds with high probability.

There are known lower bounds $\Omega(n \log N)$ and $\Omega(D\Delta)$ on time to broadcast from a single-node start, given by Jurdziński et al. [26]. These lower bounds hold for randomized algorithms and when nodes know their coordinates in a system of coordinates, so these lower bounds apply to our less demanding settings as well. Therefore the time performance of our broadcast algorithm misses a lower bound $\Omega(n \log N)$ by a factor of $\mathcal{O}(\log n)$.

Building a backbone from a single-node start could proceed by way of first coordinating all the nodes such that they can start simultaneously an execution of a distributed communication algorithm to build a backbone. For $D = \Delta = \Theta(\sqrt{n})$, both constructing a backbone and performing broadcast could be performed in $\mathcal{O}(\Delta \text{polylog } N) = \mathcal{O}(\sqrt{n} \text{ polylog } N)$ rounds, while they may need as much as $\Omega(D\Delta) = \Omega(n)$ rounds for a single-node start. The performance of our algorithm for building a backbone from a synchronized start implies that having all the nodes of a network synchronized, so that they can start an execution simultaneously, makes it possible to perform some distributed tasks faster than otherwise, in that building a virtual backbone network is among such tasks.

Our work extends the results of Jurdziński and Kowalski [20] and Jurdziński et al. [26], whose solutions rely on the knowledge of coordinates of nodes in a coordinate system in a plane, to a setting where such knowledge is not available. These are the first algorithms with performance bounds close to optimal in the model of weak sensitivity and weak connectivity of wireless networks, in view of the lower bounds in the settings considered in Jurdziński et al [26] that apply to the setting of this paper as well.

We introduce a novel approach to collision avoidance, based on strongly selective families with specifically chosen parameters, in order to compensate for lack of node coordinates in a system of coordinates in the plane. We combine them with a number of graph-related and geometry-related algorithmic techniques to synchronize and build a backbone sub-network.

A preliminary version of this paper appeared as [6].

Previous work.

Communication algorithms in SINR wireless networks known in the literature have been designed under various assumptions regarding the underlying models. Such specifications are sometimes mutually exclusive so that the respective algorithms cannot be directly compared. To put our work in a proper context, we first clarify the relevant aspects of wireless networks and then concentrate on the broadcasting primitive.

We categorize communication models of SINR wireless networks following a methodology popularized by Jurdziński and Kowalski [21]. This methodology is based to the following two independent criteria. One pertains to weak versus strong “sensitivity,” which is also known as weak versus strong “devices,” according to the terminology used by Jurdziński et al. [26]. The other relates to weak versus strong “links” in reachability graphs, according to the terminology introduced by Daum et al. [9]. These assumptions together determine four different settings, which we explain in detail in Sect. 2. Intuitively, weak-sensitivity makes it impossible for a node to ever receive a transmission from the borderline of its transmission range, while strong sensitivity determines a success of a transmission entirely by a suitable magnitude of the SINR ratio. Weak connectivity means that all links that could possibly transfer a message under favorable circumstances are considered as valid links, unlike strong connectivity, in which borderline neighbors are not connected by links. Along with these stipulations, additional assumptions determine which parameters of a network are known to the nodes, so that they can be used in codes of algorithms.

The *weak-sensitivity and weak-connectivity* model, which is used in this work, was considered by Jurdziński et al. [26]. They proved the lower bounds $\Omega(n \log N)$ and $\Omega(D\Delta)$ on time to broadcast from a single-node start. These lower

bounds hold for randomized algorithms even when the nodes know their coordinates in the plane. They also developed a deterministic algorithm that accomplishes broadcast in time $\mathcal{O}(\min\{D\Delta \log^2 N, n \log N\})$. For this model, Jurdziński and Kowalski [20] gave a deterministic distributed algorithms building a backbone from a synchronized start in $\mathcal{O}(\Delta \text{polylog } n)$ rounds. The algorithms presented in Jurdziński and Kowalski [20] and Jurdziński et al. [26] rely on nodes knowing their position in a coordinate system.

For the model of *weak sensitivity and strong connectivity*, Yu et al. [35] gave broadcast algorithms operating in times $\mathcal{O}(D + \log^2 n)$ and $\mathcal{O}(D \log^2 n)$ with high probability, where the bound depends on how broadcast is initiated. These algorithms additionally resort to a power-control mechanism. This approach applies to broadcasting multiple messages and was generalized by Yu et al. [36] to a scenario when nodes are activated in arbitrary rounds. The model of wireless networks used in the papers Yu et al. [35] and Yu et al. [36] incorporates additional parameters, with the suitable assumptions on these parameters on which efficiency of the algorithms depends; one could expect that in such environments even more involved communication tasks might have solutions with running time proportional to D , with other parameters contributing sub-linear factors.

For the model of *strong sensitivity and weak connectivity*, Daum et al. [9] gave a randomized broadcast algorithm operating in time $\mathcal{O}(n \log^2 n)$. Compared to the algorithm in this paper, the algorithm in [9] uses nearly exponentially more, namely $\Omega(n \log^3 N)$, random bits per node. They showed a lower bound $\Omega(n)$, which holds in networks of diameter 2.

Finally, for the model of *strong sensitivity and strong connectivity*, Jurdziński et al. [22] gave a broadcast algorithm working in time $\mathcal{O}(D \log n + \log^2 n)$ with high probability, which relies on nodes knowing their coordinates. Jurdziński et al. [23] gave another algorithm that works in time $\mathcal{O}(D \log^2 n)$ with high probability. The latter algorithm does not rely on nodes knowing their coordinates, improving the performance of algorithms for this model given by Daum et al. [9] for a suitable range of model parameters. A solution in Jurdziński et al. [23] was generalized to the wake-up problem with non-synchronized start-ups by Jurdziński et al. [24]. For this model of strong sensitivity and strong connectivity, Jurdziński et al. [25] studied deterministic solutions for the single-broadcast problem when nodes know their own coordinates in the plane and those of their neighbors. Their deterministic algorithm for a single-node start operates in time $\mathcal{O}(D \log^2 n)$, and another deterministic algorithm for a synchronized start operates in time $\mathcal{O}(D + \log^2 n)$. No deterministic algorithms for this model are known that do not rely on the knowledge of coordinates of nodes in a coordinate system in the plane and that are of comparable time performance.

Table 1 Time performance bounds on randomized broadcasting for the four sensitivity and connectivity settings

| | Bounds | Strong connectivity: $\varepsilon_c > \varepsilon_s$ | Weak connectivity: $\varepsilon_c = \varepsilon_s$ |
|--|--------|---|---|
| Strong sensitivity: $\varepsilon_s = 0$ | Lower | $\Omega(D)^\dagger$ | $\Omega(n)$ [9] |
| | Upper | $\mathcal{O}(D \log n \log^{\alpha+1} R_s)$ [9]; $\mathcal{O}(D \log^2 n)$ [23] | $\mathcal{O}(n \log^2 n)^\ddagger$ [9] |
| Weak sensitivity: $\varepsilon_s > 0$ | Lower | $\Omega(D)^\dagger$ | $\Omega(\min\{D\Delta, n\})$ [26] |
| | Upper | $\mathcal{O}(D \log^2 n)^*$ [35] | $\mathcal{O}(n \log^2 n)$ this paper |

The following parameters occur in the bounds: Δ denotes the maximum node degree in a communication graph, D is the diameter of this graph, α is the path loss, and R_s is the maximum ratio between distances of neighbors in communication graph. It is assumed that nodes' names are in a range $[1, N]$ such that $N = \mathcal{O}(n)$ and N is known to all nodes. The lower bounds marked with the dagger \dagger follow by a distance argument. The algorithm giving an upper bound marked with the asterisk $*$ requires a power control mechanism and assumes $\varepsilon_c = 2/3$. The algorithm giving an upper bound marked with \ddagger requires $\Omega(n \log^3 N)$ random bits per node

Table 2 Time performance bounds to build a backbone in a randomized manner for four sensitivity and connectivity settings, with spontaneous wake-up

| | bounds | Strong connectivity $\varepsilon_c > \varepsilon_s$ | Weak connectivity $\varepsilon_c = \varepsilon_s$ |
|--|--------|--|--|
| Strong sensitivity: $\varepsilon_s = 0$ | Lower | $\Omega(\log n)^\dagger$ [29] | $\Omega(\Delta)^\ddagger$ [20] |
| | Upper | $\tilde{\mathcal{O}}(\log^2 n)$ [18] | $\mathcal{O}(N\Delta)^\diamond$ |
| Weak sensitivity: $\varepsilon_s > 0$ | Lower | $\Omega(\log n)^\dagger$ [29] | $\Omega(\Delta)^\ddagger$ [20] |
| | Upper | $\mathcal{O}(\log n)^*$ [38,40] | $\mathcal{O}(\Delta \log^7 n)$ this paper |

The lower bounds marked with the dagger \dagger hold in the model of radio networks, which represents interference through neighborhoods of nodes in graphs of arbitrary topology. The upper bound marked with the asterisk $*$ requires knowing the coordinates in the plane and the ability to exchange them in messages and assumes that the parameter ε_c is greater than some absolute positive constant. The lower bounds marked with \ddagger holds even with known node coordinates. The algorithm giving the upper bound marked with the diamond \diamond consists of executing single-node transmissions in a round-robin manner $\mathcal{O}(\Delta)$ times

We summarize known lower and upper bounds on time performance of randomized broadcasting in Table 1 and building backbone in Table 2.

Broadcasting can be considered with a limited goal to inform only the neighbors in the communication graph. Such *local broadcasting* in wireless networks was studied by Barenboim and Peleg [2], Goussevskaia et al. [14], Kesselheim and Vöcking [27], Yu et al. [37], Halldórsson et al. [16], Halldórsson and Mitra [17] and Fuchs and Wagner [12]. Halldórsson et al. [19] studied broadcasting in dynamic wireless networks. Centralized algorithms for the SINR model were surveyed by Goussevskaia et al. [15].

Broadcasting of multiple messages was considered by Reddy et al. [31,32] and Yu et al. [35,36]. Chlebus et al. [5] considered multi-communication primitives for radio networks; that paper also used breadth-then-depth trees, similarly as the broadcast algorithm in Sect. 4. Derbel and Talbi [11] showed how to estimate node degrees in radio networks with nodes initially not knowing their neighbors.

Scheideler et al. [33] gave an algorithm to find a dominating set in time $\mathcal{O}(\log n)$. That algorithm relies on manipulating thresholds in physical carrier sensing. Halldórsson and

Tonoyan [18] demonstrated how nodes in wireless networks can leverage indirect information received during collisions to infer neighborhoods and construct backbones; building on this, they showed applications of backbones to various communication problems in wireless networks. Moses and Vaya [30] developed deterministic algorithms for multi-broadcast and building backbones in wireless networks. Yu et al. [38] considered distributed algorithms to construct dominating sets in dynamic wireless networks. Zou et al. [40] studied building backbones in wireless networks subject to adversarial jamming. Constructions and applications of virtual backbones and dominating sets in wireless networks were surveyed by Yu et al. [39].

There has been a great variety of approaches to signal strength and geometric decay in representations of wireless networks. Some authors have sought to specialize SINR-like settings to the Euclidean space, in order to leverage its specific topological properties, as for example, Avin et al. [1] in their work on the SINR diagrams. Others have sought to abstract from the coordinates in a plane but maintain a metric space of bounded growth that defines distances, like in the papers of Daum et al. [9] and Jurdziński et al. [23]. Still

others, like Bodlaender and Halldórsson [3] and Halldórsson et al. [19], attempted to abstract the geometry even further and consider models that relax the properties of distance as represented by metric spaces.

2 Technical preliminaries

Executions of algorithms are synchronous, in that they are structured as sequences of rounds of equal duration. The size of messages transmitted by nodes is scaled to the length of a round, so that one message can be transmitted and received in a round. We do not assume the existence of a global clock giving the same consecutive round numbers to each node.

There are n nodes in a network. Nodes have unique names assigned to them. Each node's name is an integer in the range $[1, N]$, for some positive N . We treat the numbers n and N as independent parameters in formulas, subject only to the assumption that $N \geq n$.

A broadcast algorithm disseminates some contents throughout the network; this contents is referred to as a *rumor*. Rumors and messages are also mutually scaled to each other such that a message consists of at most one rumor. A message may contain a sequence of control bits, whenever needed, to help nodes coordinate their concurrent actions. This set of restrictions on communication is often referred to in the literature as of *separate* or *bounded* messages, which comes from the interpretation that multiple rumors require multiple messages, a separate message per each rumor. The number of control bits per message in our algorithms is always $\mathcal{O}(\log N)$.

Models of wireless communication.

Networks are embedded in a two-dimensional Euclidean space. Each node can be identified by its coordinates in a Cartesian coordinate system. These coordinates determine the Euclidean distance $\text{dist}(x, y)$ between any pair of points x and y .

The SINR interference model involves the following parameters: *path loss* $\alpha > 2$, *ambient noise* $\mathcal{N} > 0$, *transmission success threshold* $\beta > 0$, *sensitivity* ε_s and *connectivity* ε_c such that $0 \leq \varepsilon_s \leq \varepsilon_c < 1$. The *transmission strength* of a node v is a positive real number denoted by P_v .

If node v transmits and u is a different node then the *signal strength* of v 's transmission as reaching u is denoted by $\mathcal{P}(v, u)$ and is defined as follows:

$$\mathcal{P}(v, u) = P_v \cdot \text{dist}(v, u)^{-\alpha}. \quad (1)$$

We will consider only *uniform* networks in which all transmission strengths of nodes are equal, and denoted as P .

Let \mathcal{T} be a set of nodes that transmit together in a round. For any two nodes v and u , where $v \in \mathcal{T}$ and $u \notin \mathcal{T}$, the *interference at u* of this configuration is denoted by $\mathcal{I}(v, u, \mathcal{T})$ and is defined as follows:

$$\mathcal{I}(v, u, \mathcal{T}) = \sum_{w \in \mathcal{T} \setminus \{v\}} \mathcal{P}(w, u). \quad (2)$$

Observe that $\mathcal{I}(v, u, \mathcal{T}) = 0$ if and only if $\mathcal{T} = \{v\}$. The *signal-to-interference-plus-noise ratio* in this configuration is denoted by $\text{SINR}(v, u, \mathcal{T})$ and is defined as follows:

$$\text{SINR}(v, u, \mathcal{T}) = \frac{\mathcal{P}(v, u)}{\mathcal{N} + \mathcal{I}(v, u, \mathcal{T})}. \quad (3)$$

We say that *node u hears node v* in a round when the following holds: (1) v transmits in this round, (2) u does not transmit in this round, and (3) u successfully receives the message transmitted by v . A node u is in the *hearing range of node v* if u can hear a message transmitted by v in a round in which v is the only node in the network that transmits. Nodes execute algorithms driven by the following two kinds of events only: either hearing a whole message from a node or not hearing a whole message from a node. Nodes do not react to any other medium-sensing feedback from the wireless network.

Next, we explain the categorization of models with respect of sensitivity, which determines when nodes can hear transmitted messages. Again, let a set of nodes \mathcal{T} consist of these nodes that transmit in a round, and let nodes v and u be such that $v \in \mathcal{T}$ and $u \notin \mathcal{T}$.

In the model of *strong sensitivity*, a node u hears node v in this round when the inequality $\text{SINR}(v, u, \mathcal{T}) \geq \beta$ holds, where parameter $\beta > 0$ is a transmission success threshold. This condition determines the hearing range of v as a distance from v , which can be determined as follows. A distance d from which a message transmitted by a node v can be heard by u is determined by the formula (3) and satisfies the inequality

$$\frac{P \cdot d^{-\alpha}}{\mathcal{N} + \mathcal{I}(v, u, \mathcal{T})} \geq \beta. \quad (4)$$

The maximum magnitude of such a distance d can be found as follows. The ambient noise \mathcal{N} is a fixed part of the left-hand side of (4), but the interference part (2) can vary and it is minimized when $\mathcal{I}(v, u, \mathcal{T}) = 0$, which holds when $\mathcal{T} = \{v\}$. It follows that the maximum d attainable in (4) is determined by setting $\mathcal{I}(v, u, \mathcal{T})$ to 0 and equals $(P/(\mathcal{N}\beta))^{1/\alpha}$; we call this quantity the *network radius* and denote by r . For the strong-sensitivity model, the meaning of the network radius r is such that a node u is in the hearing range of v if and only if $d(v, u) \leq r$. In uniform networks, the network radius is a number determined by a network that does not depend on

a transmitting node, while acting as a single transmitter, nor on listening nodes.

We may remark that if $\text{SINR}(v, u, \mathcal{T}) \geq \beta$ then also

$$\mathcal{P}(v, u) > \beta \cdot \mathcal{I}(v, u, \mathcal{T}) = \beta \cdot \sum_{w \in \mathcal{T} \setminus \{v\}} \mathcal{P}(w, u),$$

because $\mathcal{N} > 0$. It follows that, in the model of strong sensitivity, if $\beta \geq 1$ and a node $u \notin \mathcal{T}$ hears a node v then node u cannot hear any other node in \mathcal{T} . In this paper we assume only that $\beta > 0$.

In the model of *weak sensitivity*, a node u hears another node v in a round when v transmits and $\text{SINR}(v, u, \mathcal{T}) \geq \beta$ and $\text{dist}(v, u) \leq (1 - \varepsilon_s) \cdot r$, where $r = (P/(\mathcal{N}\beta))^{1/\alpha}$ is the network radius again. We use the notation $R = (1 - \varepsilon_s) \cdot r$ and call R the *hearing radius*.

The categorization of models into two classes of weak and strong sensitivities (devices) was introduced by Jurdziński et al. [26]. Weak sensitivity may be justified by the fact that it is often too costly for wireless devices to perform signal acquisition continuously, due to the constraints of the technology of wireless communication, see Goldsmith and Wicker [13]. An alternative is to wait for an energy spike, as represented by the condition $\text{SINR}(v, u, \mathcal{T}) \geq \beta$. Once nodes experience it, they may start sampling and correlating to synchronize and acquire a potential packet preamble, see Schmid and Wattenhofer [34]. After that they can detect signals as determined by the formulas of weak sensitivity. Observe that if a node wants to transmit to a distance of the network’s radius r , then this node needs to be the only transmitting one in a network, but when a range of desired transmission is restricted to $(1 - \varepsilon_s) \cdot r$, then several nodes may succeed in transmitting concurrently. In this paper, we use the model of weak sensitivity.

There is a natural algorithmic interpretation of weak sensitivity for wireless networks operating under the SINR regime. Namely, such a network could be understood as a unit disc graph with transmission range R , where R is the hearing radius, which is further restricted by the property that only a “dominating” signal from a station within the transmission range can be heard. One may observe that algorithms for weak-sensitivity wireless networks could employ techniques similar to the ones used in unit disc graphs to develop transmission schemes relying on sets of transmitters positioned relatively “sparsely” throughout the network. Guided by this observation, we use strongly selective families with suitably chosen parameters to implement message exchanges among the nodes.

Graphs.

Let $G = (V, E)$ be a simple graph. A set of vertices $M \subseteq V$ is *independent* when no two vertices in M are connected by an edge in E . An independent set of vertices of G is

maximal independent (or is a *MIS*) when it is maximal in the sense of inclusion among all independent sets. A set of vertices $B \subseteq V$ is *dominating* in G when every vertex not in B is adjacent to at least one vertex of B . A dominating set in *minimal dominating* when it is minimal in the sense of inclusion among all dominating sets. A maximal independent set in G is also a minimal dominating set of G .

We consider the length of a path to be the number of hops, so the length of each edge is 1. A shortest path connecting two members of a maximal independent set that does not include at least one other member of the maximal independent set has length at most three, and the number three is smallest as such a bound in general. In other words, if there are at least two members in an independent set, then a vertex in this independent set is connected to some other member by a path of length at most three.

Communication graphs.

A wireless network differs from wireline ones by lacking physical links. Wires used as links allow to interpret the network as a graph with nodes acting as vertices and links serving as edges. Associating a graph with a network helps to interpret communication algorithms as working in a clean abstract model of graphs, where information flows through edges. A similar benefit could be obtained for wireless networks by associating graphs with them. Such graphs are called “communication graphs” in this paper. The nodes serve the purpose to be vertices of communication graphs, but what determines edges is less apparent.

The *communication graph* of a wireless network is defined as follows: all the network’s nodes are its vertices, and for any two nodes u and v , they are connected by an edge in the communication graph when the inequality $\text{dist}(u, v) \leq (1 - \varepsilon_c) \cdot r$ holds, where r is the network’s radius and a connectivity coefficient ε_c satisfies $0 \leq \varepsilon_s \leq \varepsilon_c < 1$. We say that nodes u and v are k -hops away from each other, or are k -hop neighbors, when k is the length of a shortest path connecting u to v in the communication graph. A simultaneous transmission of two or more neighbors of a node v is called a *collision* at this node v . A collision at v does not produce any special medium-sensing feedback at v , but results in no message successfully received, for $\beta \geq 1/2 + 1/\mathcal{N}$.

If $\varepsilon_c = \varepsilon_s$ then the model is of *weak connectivity*, and if $\varepsilon_c > \varepsilon_s$ then the model is of *strong connectivity*. In the model of weak connectivity, any two nodes u and v are connected by an edge in the communication graph if the inequality $\text{dist}(u, v) \leq R$ holds, where R is the hearing radius.

The categorization of wireless networks with respect to weak and strong connectivity was introduced by Daum et al. [9]. The problem of broadcasting allows to differentiate between the two models. Strong connectivity allows to develop a broadcast algorithm of running time that depends linearly on the diameter D while other parameters contribute

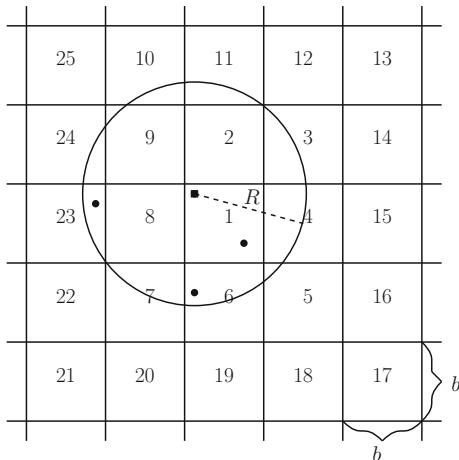


Fig. 1 A depiction of a pivotal grid. The circle of radius R centered at the square dot point is the hearing range of this point. The round dot point in box 1 belongs to the same box as the center of the circle. The dot point in box 6 belongs to the 3×3 square centered at the box containing the center of the circle. The dot point in box 23 belongs to the 5×5 square centered at the box containing the center of the circle

sublinear factors, see Table 1. In contrast to that, weak connectivity demand time that is either $\Omega(n)$ or $\Omega(\min\{D\Delta, n\})$, depending on sensitivity.

We assume the models of weak sensitivity and weak connectivity and use hearing and communication graphs as determined by these models. To simplify notations, we denote the parameter $\varepsilon_c = \varepsilon_s$ by ε .

Grids and boxes.

Communication algorithms for wireless networks with nodes interpreted as points in a Euclidean space may be designed to leverage the geometric properties of this space. For a constant $b > 0$, we consider a grid of lines parallel to the coordinate axes that partition the space into $b \times b$ disjoint boxes, and such that $(0, 0)$ is an intersection of a horizontal line and a vertical one; see Fig. 1 for an illustration. Specifically, a *box* determined by its internal point includes the points on its left side without the top endpoint and the points on its bottom side without the rightmost endpoint, and the other points on the boundary are excluded. Two boxes are *adjacent* when their interiors are separated by a line of the grid. Two boxes *share a corner* when they share exactly one point on their boundaries. The *grid distance* between two boxes is understood in the Manhattan-metric sense, in that it is a natural number equal to the minimum number of hops between two adjacent boxes needed to move from one of them to the other. In particular, adjacent boxes are of distance 1, and there are $4j$ boxes of grid distance j from any box.

If $b = R/\sqrt{2}$, where $R = (1 - \varepsilon) \cdot r$ is the hearing radius, then the grid is called *pivotal*. We will use only the pivotal grid in this paper. Any two nodes in a box of the pivotal

grid are within the Euclidean distance R from each other. It follows that there are at most $\Delta + 1$ nodes in a box, since all these nodes induce a clique and Δ is an upper bound on the degree of a vertex in a communication graph.

All the neighbors of a vertex v belong to 25 boxes in the “ 5×5 square” of boxes centered at the box of v . This is depicted in Fig. 1, where the square dot represents such a vertex v . To verify this, observe that one hop in horizontal direction from a vertical edge of a box covers distance at most R , which is less than the width of two columns of boxes, because $R < \sqrt{2}R = 2 \cdot \frac{R}{\sqrt{2}}$, while $5 = 2 + 1 + 2$.

All the vertices 2-hop away from a vertex v belong to 49 boxes in the “ 7×7 square” of boxes centered at the box containing v . To verify this, observe that two hops in horizontal direction from a vertical edge of a box covers distance at most $2R$, which is less than the width of three columns of boxes, because $2R < 3 \cdot \frac{R}{\sqrt{2}}$, and we have $7 = 3 + 1 + 3$.

All the vertices 3-hop away from a vertex v belong to 121 boxes in the “ 11×11 square” of boxes centered at the box containing v . To see this, observe that three hops in horizontal direction from a vertical edge of a box cover distance at most $3R$, which is less than five columns of boxes, because $3R < 5 \cdot \frac{R}{\sqrt{2}}$, and $11 = 5 + 1 + 5$.

Backbones.

A backbone of a network is a subnetwork that facilitates global communication tasks, similarly as a spanning tree does. Given a communication graph G , its subgraph H that is a backbone of G is required to be connected, similarly to a tree, but rather than spanning G it is a dominating set of G . A backbone is required to have asymptotically the same diameter as that of G , so that implementing broadcast of a rumor by flooding inside a backbone does not incur an extra distance to cover. At the same time, degrees of nodes in a backbone are required to be small, which facilitates collision resolution in wireless communication. The notion is more involved though, because each node is additionally equipped with specialized algorithms that facilitate using a backbone efficiently. A precise specification is as follows.

Consider a network with a communication graph G of diameter D . A *subnetwork H of G* is an induced subgraph of G . Backbones are subnetworks that have the suitable topological properties along with local algorithms associated with nodes. We say that a subnetwork H is a *backbone of G* when it has the following topological properties:

1. The nodes of H form a connected dominating set of G .
2. Each node's degree in H is $\mathcal{O}(1)$.
3. The diameter of H is $\mathcal{O}(D)$.
4. For each node v in $G \setminus H$, there is exactly one neighbor w in H assigned to v ; the node w is called a *representative* of v and v is said to be *associated* with v .

The definition above refers to implicit constants in the asymptotic notation that do not depend on the size of the communication graph G . A method of construction of backbones is expected to assign representatives to neighbors of the obtained backbone in some systematic manner.

There are two *local algorithms* INTRA_H and INTER_H that provide communication functionality of a backbone H for a network G , that implement intra-backbone communication among the nodes in the backbone and inter-backbone communication between nodes not in the backbone and their representatives in the backbone, respectively. These algorithms have the following functionality:

Algorithm INTRA_H : It facilitates exchanging messages between each pair of neighbors in H . In the process of executing INTER_H , each message received by a node $v \in H$ is also delivered to all the nodes in G associated with v , that is, to the nodes for which v is a representative. Algorithm INTER_H : It facilitates delivering a message from a node to its representative.

The notion of a backbone we use is defined similarly as in Jurdziński and Kowalski [20]. Backbones can be used as a generic tool for many communication and computation tasks in the network, see Halldórsson and Tonoyan [18], Jurdziński and Kowalski [20], and Yu et al. [39].

Bare-bones algorithms.

The following restrictions on algorithms for wireless communication make them *bare-bones*: each node knows only its name and the numbers N and Δ , and additionally the size of messages is constrained such that a single message carries $\mathcal{O}(\log N)$ bits. The motivation for studying communication algorithms under bare-bones constraints is that the restrictions imposed on the algorithms make them easily portable and the obtained performance bounds widely applicable.

Algorithms with similar bare-bones restrictions were considered by Jurdziński et al. [23] in the case of strong-sensitivity and strong-connectivity of wireless networks, while we consider bare-bones algorithms for a combination of weak sensitivity and weak connectivity of wireless networks. The algorithms in Jurdziński and Kowalski [20] and Jurdziński et al. [26], being also for weak-connectivity and weak-sensitivity of the wireless communication, were designed with the assumption that nodes know their coordinates in a plane; in this work we develop efficient algorithms that do not use this information.

Rooted spanning trees.

Consider a simple connected graph $G = (V, E)$ with a distinguished *source* vertex. The source generates a token which traverses the graph and in the process builds a subgraph. Here “traversal” means that the token hops from a visited vertex to its neighbor by traversing the connecting edge. The specific manner of token traversal and construct-

ing the subgraph that we use is called *breadth-then-depth* and is described as follows.

The source generates a token, and so becomes a first vertex visited by the token. Vertices are categorized into *discovered* and *hidden*. Initially, the source is discovered and every other vertex is hidden. Suppose that the token visits a vertex v . If this is a first visit of the token then all the neighbors of v that are still hidden become discovered and the edges connecting them to v become added to the constructed subgraph; the newly added vertices become the *children* of v and vertex v becomes their *parent*. The source vertex is considered to be its own parent and is a *root*. The vertex v dispatches the token to one of its children that has not been visited by the token yet, if such vertices exist. If the token comes back again to v and there are still children of v not visited by the token then v dispatches the token to one of them, and otherwise sends the token back to its parent. The traversal terminates when the source is about to send the token to itself. A tree produced during a breadth-then-depth traversal of a graph is called a *breadth-then-depth* spanning tree. Such a tree is rooted at the source.

Chlebus et al. [5] used breadth-then-depth trees in their distributed algorithms in radio networks. Next, we summarize the relevant properties of breadth-then-depth traversal as Fact 1.

Fact 1 *The breadth-then-depth traversal of a simple connected graph creates a spanning tree of the graph. The token visits the vertices by traversing the obtained spanning tree in a depth-first manner.*

Proof The vertices and edges that are added to the constructed graph belong to the original traversed graph, so the constructed graph is a subgraph. We show next that the edges connecting children to parents make a subgraph that is connected, has no cycles, and it includes all the vertices.

Connectivity follows from the fact that edges traversed by a token determine a walk, which creates a path to the source vertex after being pruned of repetitions of edges. A cycle cannot occur because an edge connects two vertices such that when one of them was first visited by the token then it was already discovered while the other became its child because it was still hidden. Since the created subgraph is connected and acyclic, it is a tree.

We show now that all vertices become discovered, and then they get connected as children to their respective parents. If there is only one vertex then it is the source and it belongs to the tree; otherwise let w be a vertex different from the source. Let v_1, \dots, v_k be the children of the source, in the order in which the source sends the token to them. Observe that if a token is sent by the source to its child then it comes back to the source through the same child, as otherwise the token would traverse a cycle. Let v_i be such that i is smallest with the property that there is a path connecting v_i with w

that does not pass through the source: this path traverses only undiscovered vertices before the token visits v_i . When the token visits v_i and starts traversing the graph then it will discover and visit w , because the token will discover and visit all the vertices on the path to w . Since all the vertices are incorporated into the created tree, it is a spanning tree.

The traversal of the spanning tree, as it unfolds for the traveling token, is specified by the depth-first search traversal principle: the token explores all the discoverable vertices before it returns to the parent. \square

Sets and sequences of sets

For a natural number N , the notation $[N]$ denotes the set $\{1, 2, \dots, N\}$. We consider subsets of $[N]$ and sequences of such subsets; the notation \mathcal{S} will denote a sequence of subsets of $[N]$. Such a sequence $\mathcal{S} = (S_0, \dots, S_{t-1})$ of subsets of $[N]$ consisting of t terms is said to be of *length* t . By writing $B \in \mathcal{S}$ we mean that B is a term of such a sequence \mathcal{S} , as if \mathcal{S} were an unordered family of sets.

We identify a sequence of sets $\mathcal{S} = (S_0, \dots, S_{t-1})$ with a broadcast schedule \mathcal{S}' consisting of t consecutive rounds, in which a node v transmits in round i if and only if $v \in S_i$, for $0 \leq i < t$. Such a broadcast schedule is said to be performed by *executing* \mathcal{S} .

Let \mathcal{S} be a sequence of subsets of $[N]$. For a subset $A \subseteq N$ and $a \in A$, we say that a is *selected from* A by \mathcal{S} if there is a set $S \in \mathcal{S}$ such that $A \cap S = \{a\}$.

Let N , x and y be positive integers such that $y \leq x \leq N$. We say that a sequence \mathcal{S} of subsets of $[N]$ is a (N, x, y) -*selector* if for each set $A \subseteq [N]$ of x elements there are at least y elements in A that can be selected from A by \mathcal{S} . For any fixed ζ such that $0 < \zeta < 1$, there is an $(N, x, \zeta x)$ -selector of size $\mathcal{O}(x \log N)$, see Chlebus and Kowalski [4] and DeBonis et al. [10].

For an integer $c > 0$, a sequence \mathcal{S} is (N, c) -*strongly-selective* if, for every non-empty subset Z of $[N]$ such that $|Z| \leq c$, and for each element $z \in Z$, this z can be selected from Z by \mathcal{S} . For each $c > 0$ such that $c \leq N$, there exists an (N, c) -strongly-selective sequence of length $\mathcal{O}(c^2 \log N)$; see Clementi et al. [7].

Whenever we use combinatorial structures such as selectors or strongly-selective sequences in algorithm design, then it is assumed that they are a part of code.

Odds and ends.

Performance bounds of an algorithm hold *with high probability* if, for any constant $d \geq 1$, the performance bounds can be made to hold with probability at least $1 - n^{-d}$ by suitably adjusting constants in a code of the algorithm. We assume that the numbers n , N and Δ are all powers of 2, for the simplicity of exposition; in a general case, these parameters can be rounded up to the nearest power of 2. The notation $\lg x$ means logarithm of x to the base 2.

3 Algorithmic tools

We present building blocks of algorithms and concepts and tools used in deriving their performance bounds.

We propose a construction of an induced subgraph H of a simple graph G such that H is connected dominating, which we call *shortcut connecting*. Start with a maximal independent set M as an initial dominating set; it will grow to eventually make the vertices of a connected subgraph H . We grow H by iterating the following process. Suppose that there exist two vertices v_1 and v_2 in M that are connected in G by a path of length at most three but they are not connected in H by a path of length at most three: add to H either one or two vertices along with all the newly induced edges that provide a missing shortcut such that v_1 and v_2 are now connected in H by a path of length at most three. The vertices added this way are called *connectors* for M . The final graph H consists of the initial maximal independent set M and all the added connectors.

Proposition 1 *For a simple connected graph $G = (V, E)$ of diameter D and a maximal independent set of vertices $M \subseteq V$, if an induced subgraph H is obtained from M by shortcut connecting then H is connected, its vertices make a dominating set of G , and the diameter of H is at most $3D + 2$.*

Proof The set of vertices of the subgraph H is dominating in G because M is already such. To show it is connected, suppose otherwise, to arrive at a contradiction. Each connected component of H includes at least one element of M . This is because otherwise adding any single vertex from this component to M would make it larger and still independent. Consider a shortest path in G connecting two vertices v_1 and v_2 in M that are in different connected components of H . This shortest path does not include vertices from M , by its minimality. So v_1 and v_2 are connected by a path in H of length at most three. This means that H is not the final subgraph produced by shortcut connecting, which is a contradiction.

Next we estimate the diameter of H . Take a simple path $P = (v_1, \dots, v_k)$ in G that has both endpoints in H . For each vertex v_i on P if $v_i \in M$ then denote v_i also by w_i and otherwise if v_i does not belong to M then let w_i be a neighbor of v_i that belongs to M . Consider a sequence of vertices (w_1, \dots, w_k) . For each pair w_i, w_{i+1} , these vertices are connected by a path of length at most three in G , by their selection. Since both w_i and w_{i+1} are in M , they are connected by a path of length at most three in H . It follows that for each path in G connecting two vertices in H of length L there exists a path in H connecting the same pair of vertices of length at most $3L + 2$. \square

The following Proposition 2 gives a useful estimate of signal strength with our assumptions about the model of communication.

Proposition 2 *In the model of weak sensitivity and weak connectivity, if a node v transmits and another node u is such that u and v are neighbors in the communication graph then the signal strength at u of this transmission is at least $(1+\varepsilon)\mathcal{N}\beta$.*

Proof The signal strength $\mathcal{P}(v, u)$ at u of a transmission is given by the formula (1). The distance between u and v is at most R , by the specification of edges of the communication graph. The signal strength can be estimated as follows:

$$\begin{aligned}\mathcal{P}(v, u) &\geq P \cdot R^{-\alpha} \\ &= P \cdot (1-\varepsilon)^{-\alpha} (P/(\mathcal{N}\beta))^{1/\alpha}^{-\alpha} \geq (1+\varepsilon)\mathcal{N}\beta,\end{aligned}$$

where we used $(1-\varepsilon)^{-\alpha} > 1+\varepsilon$ for $0 < \varepsilon < 1$ and $\alpha \geq 1$.

□

The communication graph could be discovered by the nodes of a wireless network by having them transmit one by one in a systematic manner: when a node u hears a message from another node v that is the only transmitter in the whole network then it may come from a neighbor in the communication graph. To determine if the inequality $\text{dist}(u, v) \leq (1-\varepsilon_c) \cdot r$ holds, the node u could compute the distance $\text{dist}(u, v)$ by resorting to the nodes' coordinates, assuming the nodes know their own coordinates and include them in transmitted messages as a “signature” identifying the sender, and also that they know the network radius $r = (P/(\mathcal{N}\beta))^{1/\alpha}$ along with the connectivity coefficient ε_c . After a preprocessing that allows all the nodes to learn and remember their neighbors in the communication graph, this could be used in distributed algorithms to construct subgraphs of the communication graph (like backbone) when such algorithms rely on neighbors in the communication graph exchanging messages. Namely, when a message is heard and it does not come from a neighbor in the communication graph then it could be ignored. A simpler approach suffices in the model of weak sensitivity and weak connectivity, as is summarized in the following Proposition 3. This fact justifies why nodes executing algorithms we develop need to know so little, and in particular, they do not need to know their positions in a system of coordinates.

Proposition 3 *In the model of weak sensitivity and weak connectivity, if a node u hears a message transmitted by a node v in a round then u and v are neighbors in the communication graph.*

Proof Let \mathcal{T} be the set of nodes transmitting in a round, where $v \in \mathcal{T}$. The node u hears the message transmitted by v if both inequalities $\text{SINR}(v, u, \mathcal{T}) \geq \beta$ and $\text{dist}(v, u) \leq R$ hold, by weak sensitivity. The nodes u and v are neighbors in the communication graph when the inequality $\text{dist}(v, u) \leq R$ holds, by weak connectivity. □

Shortcut connecting in communication graphs.

We obtain a connected dominating set by adding certain vertices to an independent set of a wireless network. The resulting subgraph of the communication graph has properties useful for implementing communication algorithms in wireless settings.

Proposition 4 *For a connected communication graph $G = (V, E)$ of a wireless network and a maximal independent set of vertices $M \subseteq V$ in G , if a subgraph H of G is obtained from M by shortcut connecting then there is a constant upper bound on node degrees of H and the number of nodes in H is $\mathcal{O}(m)$, where m is the size of the smallest connected subgraph of G dominating in G .*

Proof Each box of the pivotal grid contains at most one member of M . A node has neighbors in at most 25 boxes, including its own. A member of M is connected via connectors to at most 24 other members of M . A node that is a connector plays this role for at most 24×25 pairs of members of M . It follows that there is a constant upper bound on the number of nodes in H in one box of the pivotal grid. This gives a constant upper bound on degrees of every node of H .

Let K be an arbitrary set of nodes of G such that K is dominating, the subgraph induced by K in G is connected, and the size of K is smallest with these two properties. We observe that each node v of K is connected to a constant number of nodes in H . This is because the neighbors of v in H have to belong to at most 25 boxes of the pivotal grid, determined by the box of v , and there is a constant upper bound on the number of nodes in H in one box of the pivotal grid. It follows that the size of H is at most a constant multiple of the size of K . □

Next we discuss the efficiency of using strongly selective sequences to facilitate communication among neighbors in a communication graph. The following Proposition 5 states a critical technical insight that we will use to reason about properties of wireless communication in Euclidean space with the weak sensitivity and weak connectivity. A similar approach was applied in Jurdziński et al. [26]. We assume that some nodes are *active* in that only they can transmit.

Proposition 5 *In the model of weak sensitivity and weak connectivity, if there are at most z active nodes per box, then there is a number c such that $c = \Theta(z^3)$ and with the property that if all the active nodes execute an (N, c) -strongly-selective sequence, then, for each active node v , there is a round in which the node v is heard by all its neighbors in the communication graph.*

Proof Let us define $\eta = \delta z / (\varepsilon \mathcal{N})$, where z is smallest such that there are at most z active nodes in each box and $\delta > 0$ is a constant parameter to be determined later. The quantity η is a

linear function of z , since ε and \mathcal{N} are both fixed parameters in wireless networks.

Consider an active node v in a network under consideration. Let the other active nodes in the boxes within the grid distance of at most η of the box of node v make a set A . The number of such active nodes is $|A| = \mathcal{O}(z\eta^2)$. Let δ be large enough so that all the neighbors of v are in A .

We want a strongly selective sequence such that when it is executed then there is a round in which v transmits but none of the nodes in A does. To this end, it is sufficient to take an (N, c) -strongly-selective sequence, where $c = Kz\eta^2$ with a sufficiently large constant $K > 0$. Let t denote this round, counting from the first round of executing the strongly-selective sequence.

We estimate the total interference at a neighbor of v coming from the active nodes that are not in A . There are $4j$ boxes of distance $j > 0$ from the box of v , each with at most z nodes transmitting among the active nodes. The signal strength from each of these nodes, at any node that is a neighbor of v , is $\Theta(j^{-\alpha})$. The total interference at a neighbor of v from the transmitting nodes not in A , as specified in (2), is bounded from above by a quantity proportional to

$$z \sum_{j>\eta} j \cdot j^{-\alpha} = \mathcal{O}(z\eta^{1-\alpha} \cdot \sum_{j \geq 1} j^{1-\alpha}) = \mathcal{O}(z\eta^{1-\alpha}), \quad (5)$$

by the estimate $\sum_{j \geq 1} j^{1-\alpha} = \mathcal{O}(1)$, for $\alpha > 2$. The total interference at a neighbor of v , as expressed by (5), could be made smaller than $\varepsilon\mathcal{N}$ by choosing a sufficiently large δ in the specification of η , since $1/\eta = \varepsilon\mathcal{N}/(\delta z)$. It follows that the total interference plus noise at such a node is at most $\mathcal{N} + \varepsilon\mathcal{N} = (1 + \varepsilon)\mathcal{N}$.

The signal strength of a neighbor of v in the communication graph is at least $(1 + \varepsilon)\mathcal{N}\beta$, by Proposition 2. The value of SINR at such a node, according to the defining Eq. (3), is at least β . This means that the conditions of hearing are satisfied, so that node v is heard by all its neighbors in the communication graph in round t . \square

In applying Proposition 5, we will use (N, c) -strongly-selective sequence for sufficiently large c , with c being either $c = \mathcal{O}(1)$ or $c = \Theta(\log^3 N)$, depending on whether $z = \mathcal{O}(1)$ or $z = \mathcal{O}(\log N)$, respectively, and of the respective lengths $\mathcal{O}(\log N)$ and $\mathcal{O}(\log^7 N)$.

4 Broadcasting to coordinate start

We present a randomized algorithm for a single-source broadcast. Nodes other than the source join an execution only after they get activated by receiving messages. A “rumor” that the source wants to disseminate among all the nodes fits into

a message that can be transmitted in one round and consists of $\mathcal{O}(\log N)$ bits of information.

The broadcast algorithm we develop may be used to synchronize a network. This can be accomplished by forwarding a counter of rounds along with each message generated by the broadcast algorithm. Such a counter is inherited by each awoken node and is incremented with each round of broadcasting. A predetermined threshold for the counter values may be established, determined by the running time of broadcast, such that when the counter reaches this threshold then this indicates reaching a synchronized start. Once such a synchronized-start round is reached, an algorithm designed for a synchronized start can be invoked simultaneously by all the nodes, like the backbone algorithm given in Sect. 5.

The broadcast algorithm involves a token traversing the network by hopping along the edges of its communication graph. The token is initiated by the source node and performs a breadth-then-depth traversal, as summarized in Fact 1 in Sect. 2. Nodes do not know their neighbors in the communication graph when the traversal starts and they need to learn them to be able to control the moves of the token. Discovering edges of the communication graph and sending a token across them is accomplished by sending messages, so the token traverses the edges of the communication graph of the wireless network, by Proposition 3.

We will use two auxiliary randomized distributed routines to coordinate movements of the token. They will be used to implement a breadth-then-depth traversal and we use the relevant terminology as explained in Sect. 3. One is to estimate the number of hidden neighbors of a node in the communication graph. The other is to discover the hidden neighbors, based on knowing an estimate of their number. The two auxiliary algorithms are discussed in the next two Subsections.

4.1 Estimating the number of hidden neighbors

We discuss a randomized procedure to estimate the number of hidden neighbors. The procedure is initiated and coordinated by a node, denoted s , when it holds the token in the course of its traversal of the network. Only the nodes of distances at most R from s participate, where R is the hearing radius, since this distance determines neighborhoods in the communication graph in the weak-connectivity case. The procedure is called ESTIMATE- HIDDEN. Its pseudocode is in Fig. 2.

The node s estimates the number of hidden neighbors by counting messages it hears and comparing the outcome to some threshold value. The hidden neighbors of the node s execute $1 + \lg N$ stages. In stage i , a hidden neighbor iterates the inner loop $d \lg N$ times. A node transmits in one iteration of the inner loop in stage i with probability 2^{-i} , independently from other nodes. If k is the latest stage for which the number of messages heard by s is at least $d \lg N \cdot 2^{-4}$ then

Fig. 2 Pseudocode for a node s and its neighbor v .

Transmissions of a dummy message are performed by v and heard and counted by s . Constant d is a parameter to be determined in analysis. The number 2^k returned by s is interpreted as an approximation of the number of hidden neighbors of s

```

procedure ESTIMATE-HIDDEN
1.  $s$  initializes  $k \leftarrow 0$ 
2.  $s$  transmits a message inviting hidden neighbors to participate
3. for  $i \leftarrow 0$  to  $\lg N$  do /* stage  $i$  */
   if  $v$  is a hidden neighbor of  $s$  then
      for  $j \leftarrow 1$  to  $d \lg N$  do
          $v$  carries out a random trial with probability  $2^{-i}$  of success
         if a success occurs then  $v$  transmits
      if  $s$  hears a message at least  $2^{-5} \cdot d \lg N$  times in this stage then  $k \leftarrow i$ 
4. return  $2^k$  at  $s$ 

```

node s considers the number 2^k to be an upper bound on the number of hidden neighbors.

Lemma 1 *For each $a > 0$, if $\rho > 1$ is the number of hidden neighbors of s then the number 2^k returned by algorithm ESTIMATE-HIDDEN satisfies $\rho \leq 2^k \leq 2^5 \cdot \rho$ with probability at least $1 - n^{-a}$, for a sufficiently large parameter $d > 0$ and all sufficiently large n .*

Proof The procedure operates by nodes exchanging messages, which always arrive at a node from neighbors in the communication graph, by Proposition 3.

Let the interval of integers $[2, 2^{1+\lg N} - 1]$ be partitioned into disjoint segments as follows:

$$[2, 3], [4, 7], \dots, [2^i, 2^{i+1} - 1], \dots, [2^{\lg N}, 2^{1+\lg N} - 1].$$

The number ρ is in precisely one of these ranges. Let ℓ be such that $\rho \in [2^\ell, 2^{\ell+1} - 1]$, where $\ell \geq 1$.

The probability that a specific hidden neighbor of s transmits during stage ℓ , while the other hidden neighbors do not transmit, can be estimated as follows:

$$\begin{aligned} \frac{1}{2^\ell} \left(1 - \frac{1}{2^\ell}\right)^{\rho-1} &\geq \frac{1}{2^\ell} \cdot \left(1 - \frac{1}{2^\ell}\right)^{2^\ell \frac{\rho}{2^\ell}} \geq \\ \frac{1}{2^\ell} \cdot \left(1 - \frac{1}{2^\ell}\right)^{2^{\ell+1}} &\geq \frac{1}{2^\ell} \cdot \frac{1}{16}, \end{aligned}$$

because $\rho < 2^{\ell+1}$. The probability that exactly one hidden neighbor of s transmits is at least

$$\rho \cdot \frac{1}{2^\ell} \cdot \frac{1}{16} \geq \frac{1}{2^4}, \quad (6)$$

because $\rho \geq 2^\ell$. Let us define an indicator random variable X_i^j such that $X_i^j = 1$ when a single transmission occurs in trial j of stage i , otherwise $X_i^j = 0$. The estimate of (6) can be interpreted to mean that $\Pr(X_\ell^j = 1) \geq \frac{1}{2^4}$. Define a random variable X_i as follows:

$$X_i = \sum_{j \in [d \lg N]} X_i^j.$$

By the linearity of expectation, the following inequality holds:

$$\mathbb{E}[X_\ell] \geq 2^{-4} \cdot d \lg N.$$

By the Chernoff bound, the number of successful transmissions received by node s in stage ℓ of algorithm ESTIMATE-HIDDEN is smaller than $2^{-5} d \lg N$ with probability that is at most n^{-2a} , for sufficiently large d . It follows that the inequality $2^k \geq \rho$ does not hold with a probability that is at most n^{-2a} .

Consider a stage i of the algorithm such that $\ell + 7 \leq i \leq 1 + \lg N$. In a single round of this stage, the probability that there is at least one transmitting node is at most

$$\rho \cdot 2^{-i} \leq \rho \cdot 2^{-\ell-7} < 2^{-6},$$

because $\rho < 2^{\ell+1}$. Thus the expected number of rounds with at least one transmission in stage i is smaller than $d \lg N \cdot 2^{-6}$. By the Chernoff bound, the number of times node s hears a message in stage i is at least $d \lg N \cdot 2^{-5}$ with a probability at most n^{-3a} , for a sufficiently large d . The number of such stages is at most $\lg N$. By the union bound, some of these stages result in producing at least $d \lg N \cdot 2^{-5}$ messages that are heard with probability at most $\lg N \cdot n^{-3a}$, which is at most n^{-2a} for sufficiently large n . Therefore, the inequality $2^k \leq 2^5 \rho$ does not hold with probability at most n^{-2a} .

We conclude, by the union bound, that both the inequalities $\rho \leq 2^k \leq 2^5 \cdot \rho$ hold with probability at least $1 - n^{-a}$, for a sufficiently large parameter d and all sufficiently large n . \square

4.2 Discovering hidden neighbors

Let us assume that we have an upper bound x on the size ρ of the hidden neighborhood of a node s , where $\rho \leq x \leq 2^5 \rho$. The node s could obtain such an estimate x by executing procedure ESTIMATE-HIDDEN presented in Sect. 4.1. Given such a bound x for a node s , a routine DISCOVER(x) allows the node to learn its neighborhood in $\mathcal{O}(x \log N)$ rounds, as we show next in Lemma 2.

Fig. 3 Pseudocode for a node s . The number x is an upper bound on the number of hidden neighbors, it is a power of 2 and is passed as an argument. The used selectors are of length $\mathcal{O}(2^{-j}x \log N)$ and are part of code. The discovered neighbors become children of s in a breadth-then-depth tree

```

procedure DISCOVER( $x$ )
1. initialize list of discovered neighbors  $R \leftarrow \emptyset$  at  $s$ 
2. for  $j = 0, 1, \dots, \lg x - 1$  do /* stage  $j$  */
   notify neighbors  $v$  of  $s$ :
      if  $v$  is hidden then  $v$  starts executing  $(N, 2^{-j}x, 2^{-j-1}x)$ -selector
      during executing a selector by neighbors:
         if a name  $v$  is heard then  $v$  gets discovered: node  $s$  adds  $v$  to  $R$ 
         notify all neighbors discovered in this stage  $j$ :
          $s$  transmits their names one by one
3. return  $R$  as the list of discovered neighbors at  $s$ 

```

Pseudocode of procedure DISCOVER is given in Fig. 3. The procedure works by repeating *stages* $\lg x$ times, each stage an iteration of the for-loop, with the goal to decrease by half an estimate on the number of hidden neighbors. This in turn is accomplished by having the hidden neighbors execute selectors of lengths determined by stage numbers.

Lemma 2 *If a node has at most x hidden neighbors that an execution of DISCOVER(x) makes this node learn all its hidden neighbors in time $\mathcal{O}(x \log N)$.*

Proof The procedure DISCOVER operates by nodes exchanging messages. A transmitted message always hops from a node to its neighbors in the communication graph, by Proposition 3.

During executing procedure DISCOVER(x) by a node s , at most $x/2$ nodes remain hidden after the first stage, by the definition of selectors. This pattern continues, such that after a j th stage the number of hidden neighbors is at most $x/2^{j-1}$. Indeed, the proof is by induction on j . We just argued that it holds at the end of the execution of loop “for” applied for $j = 0$, thus assume that it holds for some $j \geq 0$. By the definition of $(N, 2^{-j}x, 2^{-j-1}x)$ -selector applied to the set of hidden neighbors, of size at most $x/2^j$ by induction, at most $x/2^{j-1}$ of them remain unselected by the selector, hence hidden. The lengths of these selectors decreases geometrically, so the lengths of the stages sum up to $\mathcal{O}(x \log N)$. \square

4.3 Algorithm for broadcasting

We present an algorithm to have a token traverse the network. The algorithm is called TRAVERSE- TO- BROADCAST. It is summarized as a pseudocode in Fig. 4. The token can carry any contents piggybacked on it with the goal to broadcast it. A traversal is initialized by the node which is the source of a broadcast message. There is no prior coordination among the nodes of the network to participate in an execution. The token’s traversal involves building a breadth-then-depth spanning tree in the communication graph, rooted at the source node, and the token traverses it in a depth-first manner, according to Fact 1.

When the algorithm is invoked then nodes do not know their neighbors yet. Each node uses the procedures ESTIMATE- HIDDEN and DISCOVER to discover the hidden neighbors in the communication graph. If a node hears a message from a neighbor that is executing ESTIMATE- HIDDEN with an invitation to its hidden neighbors to join in disclosing themselves and becoming children in a breadth-then-depth tree, then this is a first signal the node obtains that a broadcast has been initiated. Upon a token’s visit to a discovered neighbor, the visited node creates a list of its hidden neighbors just to be discovered. The token will be dispatched to visit these nodes one by one after each return. When the list of the discovered neighbors gets exhausted, the token is returned to the parent node from which it arrived. If the token returns to the initiating source node and all its neighbors have been already visited by the token then the traversal terminates.

Theorem 1 *Algorithm TRAVERSE- TO- BROADCAST accomplishes a broadcast from a single-node start in $\mathcal{O}(n \log^2 N)$ rounds with high probability. Each node uses $\mathcal{O}(\log^3 N)$ random bits.*

Proof Each node obtains a correct upper bound on the number of hidden neighbors with high probability when executing ESTIMATE- HIDDEN, by Lemma 1. If the bound holds true then procedure DISCOVER identifies the hidden neighbors correctly, by Lemma 2. The token’s traversal is implemented by nodes exchanging messages, and this always occurs only between neighbors in the communication graph, by Proposition 3. The token traverses the obtained breadth-then-depth spanning tree, by Fact 1, and accomplishes broadcasting in the process.

Now we estimate the running time. Each node executes procedure ESTIMATE- HIDDEN and DISCOVER once, by the pseudocode in Fig. 4. These procedures are executed sequentially, started by receiving the token for the first time. Executing ESTIMATE- HIDDEN takes $\mathcal{O}(\log^2 N)$ rounds, since it consists of two nested loops, by the pseudocode in Fig. 2, each taking $\mathcal{O}(\log N)$ iterations. Therefore the total time spent on executing ESTIMATE- HIDDEN is $\mathcal{O}(n \log^2 N)$ with high probability. A node participates only

Fig. 4 Pseudocode executed by a node s upon receiving a token while holding it. The source initiates a token and is the first node holding the token

```

algorithm TRAVERSE-TO-BROADCAST
  1. upon receiving and holding the token for the first time:
    if  $s$  is different from the source then
      record the name of the node from which the token arrived as parent
      execute procedure ESTIMATE-HIDDEN
        /* returns an upper bound  $x$  on the number of hidden neighbors */
      execute procedure DISCOVER( $x$ )           /* returns a list  $R$  of discovered neighbors */
    2. upon receiving and holding the token:
      if the token arrived from a node  $v \in R$  then remove  $v$  from  $R$ 
      if  $R$  nonempty then pass the token to a node in  $R$  else
        if  $s$  is different from the source
          then return the token to the parent and exit
        else exit

```

```

algorithm BACKBONE-SYNCHRONIZED-START
  stage 1: call FIND-MIS
  stage 2: call CONNECT-To-MIS
  stage 3: build an implementation of  $\text{INTRAH}_H$ 
  stage 4: build an implementation of  $\text{INTER}_H$ 

```

Fig. 5 Pseudocode for all nodes to start simultaneously. Procedure CONNECT- TO- MIS starts from the maximal independent set found by FIND- MIS

once as a hidden neighbor of a node executing DISCOVER. The sum of upper bounds on the number of hidden neighbors is at most $2^5 n$ with high probability, by Lemma 1. It follows that the time spent on all the executions of DISCOVER is $\mathcal{O}(n \log N)$ with high probability.

Randomness is used only in procedure ESTIMATE- HIDDEN. Lemma 1 gives the needed estimates on probability. A node performs $\mathcal{O}(\log^2 N)$ experiments, each requiring $\mathcal{O}(\log N)$ random bits. \square

5 Backbone from synchronized start

We develop a randomized algorithm to build a backbone from a synchronized start, which means that all the nodes begin an execution together. The algorithm runs in $\mathcal{O}(\Delta \log N + \log^c N)$ time, where c is a positive constant. The running times of algorithms INTER_H and INTRAH_H , associated with a backbone, are $\mathcal{O}(\log N)$ and $\mathcal{O}(\Delta \log N)$, respectively.

The algorithm is called BACKBONE- SYNCHRONIZED- START. Its pseudocode is in Fig. 5. An execution begins by calling two procedures. One of them elects a maximal independent set of nodes in the communication graph; we present it in Sect. 5.1. The other one inter-connects the nodes in the obtained maximal independent set into a connected dominating and also connects the remaining nodes to it; this procedure is discussed in Sect. 5.2. The execution concludes with finding transmission schedules for algorithms INTER_H and INTRAH_H associated with the backbone; the details are given in Sect. 5.3.

5.1 Finding a maximal independent set

We use a maximal independent set as a minimal dominating set. The procedure to find a maximal independent set is called FIND- MIS. Its pseudocode is given in Fig. 6. The invoked $(N, \Theta(\log^3 N))$ -strongly-selective sequence of length $\mathcal{O}(\log^7 N)$ exists, by the fact that for each $c > 0$ such that $c \leq N$, there exists an (N, c) -strongly-selective sequence of length $\mathcal{O}(c^2 \log N)$.

The procedure FIND- MIS works in phases numbered by integers i , where $1 \leq i \leq 1 + \lg \Delta$. A phase i consists of $\gamma \lg N$ sub-phases, for a constant $\gamma > 0$. A sub-phase j of phase i consists of two stages: the first one serves the purpose to elect new members of a maximal independent set and the other one determines for some nodes that they will not belong to a maximal independent set under construction, such nodes will be categorized as workers.

Here is a summary of categorizations of nodes we will use. In the beginning of an execution, all the nodes have the status of being *neutral*. This status may change so that a node becomes either a *candidate*, a *member* or a *worker*. We want the status of either neutral or a candidate to be temporary and eventually a node to become either a member or a worker, such that this status, of a member or a worker, stays unchanged.

A neutral node can become a candidate in the beginning of the first stage of a phase, as determined by an outcome of a random experiment. By the end of the first stage, it is also determined which among the candidates graduate to members and which become neutral again. If a candidate in a sub-phase hears in the first stage that one of its neighbors is a candidate too then it reverts to the neutral status. A candidate in a sub-phase that does not hear that its neighbor is a candidate becomes a member. It follows that there are no candidates at the end of a phase. Once a node becomes a member, at the end of the first stage, then this status is permanent. Neutral nodes that hear in the second stage of a sub-phase that some of their neighbors are members immediately become workers. Once a node becomes a worker, at the end of the second stage, then this status is permanent. The

Fig. 6 Pseudocode for a node s . The invoked $(N, \Theta(\log^3 N))$ -strongly-selective sequence is a part of code and is of $\mathcal{O}(\log^7 N)$ length. The independent set obtained as output consists of the nodes that become members. The constant γ is determined by the analysis

```

procedure FIND-MIS
  for  $i = 1$  to  $\lg(\Delta + 1)$  do      /* phase  $i$  */
    for  $j = 1$  to  $\gamma \lg N$  do      /* sub-phase  $j$  */
      (i) Incorporate new members: /* first stage of sub-phase  $j$  */
        if  $s$  is neutral then  $s$  becomes a candidate with probability  $2^i/(\Delta + 1)$ .
        if  $s$  is a candidate then
           $s$  executes an  $(N, \gamma \log^3 N)$ -strongly-selective sequence to announce its name
          take a record of the names from the messages heard from candidates
        if  $s$  is a candidate then
          if  $s$  heard a message from some other candidate
            then  $s$  goes back to neutral
          else  $s$  becomes a member
      (ii) Convert neutral nodes to workers: /* second stage of sub-phase  $j$  */
        if  $s$  became a member in the preceding stage then
           $s$  executes an  $(N, \gamma \log^3 N)$ -strongly-selective sequence to announce its name
          take a record of the names heard from new members
        if  $s$  is neutral and  $s$  has just heard some member's name then
           $s$  becomes a worker
         $s$  assigns a member with the smallest name just heard as its representative

```

sets of neutral nodes, as considered in the end of sub-phases, are monotonically decreasing, in the sense of inclusion.

At the end of an execution, each node becomes either a member or a worker or it remains neutral. The set of nodes that is produced as outcome, and which is to be a maximal independent one, consists of the nodes that end up as members.

Next, we argue about the correctness and efficiency of FIND- MIS. This procedure operates by nodes exchanging messages, which successfully arrive to neighbors in the communication graph once transmitted by a node, by Proposition 3. We may assume that when an execution begins then phase 0 is has been just completed. We begin with formulating an invariant for a phase i , for $0 \leq i \leq \lg \Delta$, which is understood to hold at the end of phase.

Invariant for phase i of procedure FIND- MIS:

Each box of the pivotal grid contains at most $(\Delta + 1)/2^i$ neutral nodes.

Observe that initially the invariant holds for $i = 0$, by the properties of pivotal grid, see Sect. 2. Specifically, if a node belongs to a box, then all other nodes in the box are its neighbors in the communication graph, so there can be at most Δ of them. If the invariant holds for a phase i then it holds also for phase $i + 1$ with high probability, which we discuss next, starting from Lemma 3 through Lemma 6.

Lemma 3 *If the invariant holds for a phase of procedure FIND- MIS, then during a sub-phase of this phase for each neutral node v the number of neutral neighbors of node v that become candidates is at most $(a + 1) \lg N$ with probability at least $1 - n^{-a}$, for sufficiently large n .*

Proof In the beginning of the phase, there are at most $(\Delta + 1)/2^i$ neutral nodes in each box, by the invariant. For a sub-phase, each neutral node becomes a candidate with prob-

ability $2^i/\Delta$. So the expected number of neutral nodes in a box that become candidates in this sub-phase is at most 2. Let v be a neutral node. Its neighbors are in at most 25 boxes, so the expected number of neutral neighbors of node v that become candidates is at most 50.

Suppose X is the number of successes in a number of independent Bernoulli trials, with the mean number of successes equal to μ . We use the Chernoff bound which states that $\Pr(X \geq b) \leq 2^{-b}$ for $b \geq 6\mu$. Let $b = c \lg n$, for $a > 0$. Then we have that $\Pr(X \geq c \lg n) \leq n^{-c}$, for sufficiently large n . Let a random variable X be specifically the number of neighbors of v that become candidates in a sub-phase. Choosing $c = a + 1$, we obtain that at least $(a + 1) \lg n$ neighbors of v become candidates with probability at most n^{-a-1} . This applies to each node in the network with probability at most n^{-a} , by the union bound. \square

Lemma 4 *For any $a > 0$ there exists $\gamma > 0$ such that if the invariant holds for a phase of procedure FIND- MIS then each node that executes a strongly-selective sequence to announce its name in a sub-phase of the phase is heard by all of its neighbors with probability at least $1 - n^{-a}$.*

Proof For every node, the number of its neighbors that become candidates in a sub-phase is at most $(a + 1) \lg N$ with probability at least $1 - n^{-a}$, by Lemma 3. These nodes will execute a strongly-selective sequence to announce their names during a sub-phase. By Proposition 5, there exists $c = \Theta(((a + 1) \lg N)^3)$ such that if all the candidates execute an (N, c) -strongly-selective sequence, then, for each such a candidate node v , there is a round in which the node v is heard by all its neighbors. It is sufficient to take γ at least such that $c = \gamma \lg^3 N$. \square

Lemma 5 *For any $a > 0$ there exists $\gamma > 0$ such that if the invariant holds for a phase $i < \lg(\Delta + 1)$ of procedure*

FIND- MIS and there are at least $(\Delta + 1)/2^{i+1}$ neutral nodes in phase $i + 1$ in a box then there are no neutral nodes in the box after phase $i + 1$ with probability at least $1 - n^{-a}$.

Proof Denote $x = \frac{\Delta+1}{2^{i+1}}$. Each node in the box decides with probability $\frac{1}{x}$ to become a candidate in phase $i + 1$. A node in the box has fewer than $25 \cdot \frac{\Delta+1}{2^i} = 50x$ neighbors in the communication graph, by the invariant. If exactly one neutral node in the box becomes a candidate in a sub-phase while none of its neighbors choose the same then we call it a *success* in the box. If a success in a box occurs during a sub-phase then the node that is a candidate becomes a member and its neighbors become workers, because the candidate node does not hear from another candidate and its neighbor nodes learn of its candidate status during the sub-phase with probability at least $1 - n^{-2a}$, by Lemma 4, for a suitable γ . The probability of a success in a sub-phase is at least

$$x \cdot \frac{1}{x} \left(1 - \frac{1}{x}\right)^{50x} > \frac{1}{2e^{50}} = p,$$

where the number p is a constant. There exists a sufficiently large $c > 0$ such that if $c \lg n$ independent trials are performed, each with the probability p of success, then all of them are failures with probability at most n^{-2a} . It suffices to take $\gamma > c$. \square

Now we are ready to give a fact that summarizes the key properties of procedure FIND- MIS.

Lemma 6 For any $a > 0$ there is a $\gamma > 0$ such that procedure FIND- MIS works in $\mathcal{O}(\log^8 N \log \Delta)$ rounds and produces a maximal independent set with probability at least $1 - n^{-a}$.

Proof The number of rounds the procedure is executed is determined by its pseudocode in Fig. 6. There are $\lg(\Delta + 1)$ phases, each consisting of $\gamma \lg N$ sub-phases. A sub-phase takes $\mathcal{O}(\log^7 N)$ rounds, because of the length of the used strongly-selective sequence. This contributes $\mathcal{O}(\log^8 N \log \Delta)$ rounds in total.

Next, we argue about the correctness. We show first that the invariant holds for each phase with probability at least $1 - n^{-a}$, for a suitable $\gamma > 0$. The argument is by induction on the phase number i . Specifically, conditional on the invariant holding for a phase, it holds for the next phase with probability at least $1 - n^{-2a}$, for a suitable constant $\gamma > 0$. The base of induction occurs for the conceptual phase number 0. The invariant holds for this phase since each box contains at most $\Delta + 1$ nodes by the definitions of Δ and the pivotal grid. In the inductive step, we show that if the invariant holds for a phase i then it holds for the next phase $i + 1$ with probability at least $1 - n^{-2a}$. Consider a box. If there are at most $\frac{\Delta+1}{2^{i+1}}$ neutral vertices in this box then certainly the invariant holds for the next phase, and otherwise there are at least $\frac{\Delta+1}{2^{i+1}}$ vertices in the box. If so then, by Lemma 5, there are no neutral

vertices after phase $i + 1$ with probability at least $1 - n^{-2a}$, for a suitable constant $\gamma > 0$. This completes the argument by induction. The invariant holds for each phase with a probability that is a product of all these conditional probabilities for each phase, that is with a probability at least

$$(1 - n^{-2a})^{\lg(\Delta+1)} \geq 1 - \lg(\Delta + 1) \cdot n^{-2a} \geq 1 - n^{-a},$$

by the Bernoulli's inequality.

Next we argue that when an execution is completed then the set of nodes that are members makes a maximal independent set, with a suitably high probability. The assumption is that the invariant is satisfied for each phase. The set of members after a phase is an independent set of nodes. This is because when two neighbors are candidates in a sub-phase then they hear each other's names in a sub-phase with a high probability, by Lemma 4, and so retreat to being neutral. To show that the independent set of members is maximal independent, it is enough to demonstrate that no neutral node remains after an execution is over: this is because then each node is either a worker or a member, so there is no room for more members. It suffices to argue that there are no neutral node in every box. Suppose otherwise, that there remain neutral nodes in a box. Let i be the smallest integer such that when phase i occurs then the number of neutral nodes in the box is at least $\frac{\Delta+1}{2^{i+1}}$ and at most $\frac{\Delta+1}{2^i}$. By Lemma 5, there are no neutral nodes in the box after the phase is over with a probability at least $1 - n^{-3a}$, for a suitable $\gamma > 0$. Since there are at most n boxes, the probability of some neutral nodes surviving all the phases is at most n^{-2a} , by the union bound.

We conclude that all possible unfavorable events occur with a probability that is at most n^{-a} , again by the union bound. \square

5.2 Implementing shortcut connecting

We describe a procedure to build a connected dominating set starting with a maximal independent set by way of implementing shortcut connecting, as it is defined in Sect. 2. This is done by designating connectors which together with the independent set make a connected dominating subgraph. It is assumed that a maximal independent set is given and each node knows whether it belongs to this set or not. A node in such an independent set is called a *member*, to be consistent with the categorization of some nodes as members in the sense of procedure FIND- MIS. The algorithm is called CONNECT- TO- MIS and its pseudocode is given in Fig. 7. It is structured as four consecutive parts, which we describe in detail next.

The goal of the first part is for all the non-member nodes to learn the names of their member neighbors. The members transmit their own names using a (N, c) -strongly-selective

Fig. 7 Pseudocode for a node s . The (N, c) -strongly-selective sequence is of length $\mathcal{O}(\log N)$, where the constant $c > 0$ is determined in analysis. Communication in sub-parts is by way of executing suitable strongly-selective sequences

```

procedure CONNECT-TO-MIS
part 1: Non-members learn their neighbors:
    if  $s$  is a member then it executes a  $(N, c)$ -strongly selective sequence
        else  $s$  records the names of member neighbors
part 2: Members learn about other members of distance at most three:
    sub 1: any two adjacent non-members exchange information about their member neighbors
    sub 2: non-members pass what they have learned in sub-part 1 to member neighbors
part 3: Members choose connectors and notify them:
    sub 1: connectors of distance one from their members get notified
    sub 2: connectors of distance two from their members get notified
part 4: Nodes learn the neighborhoods in the backbone:
    if  $s$  belongs to the backbone then
         $s$  executes a  $(N, c)$ -strongly selective sequence
         $s$  simultaneously records the names of neighbors heard from

```

sequence of length $\mathcal{O}(\log N)$, for a sufficiently large constant $c > 0$, to be determined in analysis. Every node that is not a member records the names of members, as they are heard. The j -th heard name heard by a non-member node is called the j -th member of this non-member node.

During the second part, members learn about other members of distance at most three in the communication graph. This is done by having each member communicate with its neighbors and nodes of distance two, as those were learned in the first part. Let $x = \mathcal{O}(\log^7 N)$ be the length of an $(N, \Theta(\log^3 N))$ -strongly-selective sequence, which is the same sequence as in the implementation of procedure FIND-MIS. This is organized as two sub-parts such that during the first sub-part two adjacent nodes that do not belong to the maximal independent set exchange information about their neighbors that are members, and during the second sub-part each node passes this knowledge along with the information about its own neighbors that are members to the member neighbors, of whom there are at most 25. The details are given next.

The first sub-part proceeds as follows. First, each non-member node v chooses a number t_v that is $\Theta(\Delta)$ uniformly at random from a suitable range, which is determined in analysis. Time is partitioned into consecutive blocks of length x , each devoted to executing some $(N, \Theta(\log^3 N))$ -strongly-selective sequence by suitable nodes. Additionally, we join blocks into consecutive groups of 25 blocks each. Then node v is active during group t_v in every round in $[25xt_v + 1, \dots, 25x(t_v + 1)]$, in the following sense: in the j -th block of group t_v , where $0 \leq j \leq 24$. More precisely, in rounds $[25xt_v + jx + 1, \dots, 25xt_v + (j + 1)x]$, node v transmits its j -th member and its own name using an $(N, \Theta(\log^3 N))$ -strongly-selective sequence. Each node w records all the names heard in this execution. For every name that u heard, node w records the round number f_u when the name of u was heard and the node g_u who sent it.

The second sub-part proceeds similarly, with the following modifications. First, each group consists of 49 blocks. This

is because there is at most one member in a box of the pivotal grid and 49 is an upper bound on the number of boxes containing nodes that are two hops away from a node, as argued in Sect. 2. Second, each block is associated with some known member, either in one-hop neighborhood, as learned in the first part, or in two-hop neighborhood, as learned in the first sub-part of this second part. In the latter case, such member's name, say u , is transmitted along with its associated forwarding node g_u and the successful round number f_u . Third, at the end of this sub-part, each member w additionally records all members' names u heard in this sub-part, together with one or two node(s) associated with this name, and one or two, respectively, round number(s) when successful transmissions between the node(s) and the member took place. In case there are many intermediate nodes or pairs thereof, an arbitrary selection of one such a configuration per a member's name is made: for each member's name u , let $g_{u,1}$ denote a one-hop neighbor and $g_{u,2}$ be two-hop neighbor associated with this name, and let $f_{u,i}$ be associated with a successful transmission round of node $g_{u,i}$, for $i = 1, 2$. In case there is only one connector associated with a member of name u , we denote it by g_u and its successful round number by f_u , to simplify the notation. These nodes are designated as connectors.

In the third part, members inform connectors chosen in the second part. Similarly as in the second part, this is carried out in two sub-parts, with first connectors of distance one and then of distance two becoming informed. In the first sub-part, every member node w transmits according to a (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$, for a sufficiently large constant $c > 0$, precisely 121 times, one after another. The number 121 is an upper bound on the number of member nodes in any 3-hop neighborhood. In the j -th execution of a (N, c) -strongly-selective sequence, node w transmits its j -th heard member u 's name along with its associated connector g_u and round value f_u , or a pair of connectors $g_{u,1}, g_{u,2}$ and round numbers $f_{u,1}$ and $f_{u,2}$. Upon receiving such a message containing only one connector g_u

and a value f_u , connector g_u records that it is a connector from member u to member w , as well as the successful round f_u when forwarding took place. Upon receiving such a message containing two connectors $g_{u,1}$ and $g_{u,2}$ and values $f_{u,1}$ and $f_{u,2}$, connector $g_{u,1}$ records that it is a connector from a member u , via node $g_{u,2}$, to a member w , as well as the successful round $f_{u,1}$ when the forwarding from it to the member w took place. In the second sub-part, only such connectors $g_{u,1}$ are active in relaying messages. Similarly as in the first sub-part, they use a (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$, for a sufficiently large constant $c > 0$, precisely 121 times one after another. In the j -th execution of the (N, c) -strongly-selective sequence, node $g_{u,1}$ transmits its j -th heard associated connector $g_{u,2}$ for the member u , along with its associated pair of members u and w and round value $f_{u,2}$. Upon receiving such a message, connector $g_{u,2}$ records that it is a connector from the member u to the member w , via node $g_{u,1}$, as well as the successful round $f_{u,2}$ when the forwarding from it to the connector $g_{u,1}$ took place. A non-member node is in the backbone when it is a connector to some pair of members.

In the final fourth part, all backbone nodes, both members and connectors, learn their neighborhoods in the backbone. In order to accomplish it, they transmit their own names using an (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$, for a sufficiently large constant $c > 0$. Each backbone node records all the received names as its neighbors in the backbone.

Lemma 7 *Procedure CONNECT- TO- MIS builds a connected dominating subnetwork satisfying all the specifications of a backbone in $\mathcal{O}(\Delta \log^7 N)$ time with high probability, when starting with a maximal independent set of nodes.*

Proof It is sufficient to show that the algorithm CONNECT- TO- MIS implements shortcut connecting in wireless networks, which makes Propositions 1 and 4 applicable. This follows from the fact that an execution of CONNECT- TO- MIS is a systematic enlargement of a maximal independent set of nodes according to the requirements for shortcut connecting. We rely on the property that nodes add edges by sending messages to neighbors in the communication graph, which always arrive successfully, by Proposition 3. A detailed argument follows.

The first part makes each non-member learn the names of all its member neighbors. There are at most 25 such neighbors, since there is at most one member in a box of the pivotal grid. If the constant c is sufficiently large, than executing a (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$ is sufficient to achieve this task, by Proposition 5.

During the second part, each member obtains either direct or relayed information from each of its one- or two-hop neighbors, about their at most 121 member neighbors, each of distance at most three from the member. By letting each non-

member node to choose a random number in range $\Theta(\Delta)$, for a sufficiently large range proportional to Δ , in every box of the pivotal grid, there are $\mathcal{O}(\log N)$ nodes that selected the same round, with high probability. In order to allow each of these nodes to transmit alone, an $(N, \Theta(\log^3 N))$ -strongly-selective sequence of length $x = \mathcal{O}(\log^7 N)$ can be used, by Proposition 5 for $b = \mathcal{O}(\log N)$ active nodes per box with high probability, by an argument similar to that in the proof of Lemma 5. This applies to both sub-parts. Additionally, a factor of 25 guarantees that in the first sub-part such situations will occur at least 25 times, so each non-member node will be able to successfully transmit all the names of its member neighbors. There might be at most 49 members of one or two hops away from a node. The second sub-part accomplishes its goal for similar reasons. This results in each member learning at most 121 other members of at most three hops away, together with at most two connectors.

In the third part, each member needs to send to at most two connectors the information summarizing what it learned about other members of distance at most three away. To accomplish this, the members execute an (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$ in the first sub-part, similarly as in part one, each time for a different name of such a member, repeating for at most 121 names. In the second sub-part, each connector that receives such a message addressed to it, in which it occurs with another connector, relays this message to the other connector using the same procedure as in the first sub-part. Each node could be chosen as a connector to at most 121 pairs of members, hence repeating the (N, c) -strongly-selective sequence 121 times, each time for a different pair of members for which it was chosen as a connector, is sufficient for relaying all such messages to all two-hop connectors. Proposition 5 is used, for a constant number z of active nodes per box, these being only members and connectors, to guarantee a successful message exchange between the neighboring pairs of active nodes.

This is repeated one more time in part four to assure that both the members and the connectors will know about the other nodes that are either members or connectors and that are in their range. The argument for correctness is similar to the one in part one, except that instead of at most one member in each box of the pivotal grid we have a constant number of backbone nodes in a box. \square

5.3 Implementing local algorithms

The nodes in a backbone use local algorithms INTRA_H and INTER_H . Both INTRA_H and INTER_H are deterministic, though the latter is pre-computed by a randomized algorithm.

Algorithm INTRA_H facilitates communication among the nodes in the backbone. It schedules all members and connectors to transmit using some (N, c) -strongly-selective sequence of length $\mathcal{O}(\log N)$, for sufficiently large con-

stant $c > 0$. One run is executed for preprocessing to confirm neighboring backbone nodes. Additionally, if a non-backbone node does not have its associated backbone node yet, which is its representative node, then it selects the smallest among the heard names in the backbone as its representative.

Algorithm INTER_H facilitates communication between the nodes not in the backbone and their representatives in the backbone. It is specified as follows. We refer to the length of an $(N, \Theta(\log^3 N))$ -strongly-selective sequence by $x = \mathcal{O}(\log^7 N)$. Each non-backbone node selects a number t in the range $[1, y]$ uniformly at random, where $y = \Theta(\Delta \log N)$, with a sufficiently large constant in front of Δ , and it transmits its own name by executing an $(N, \Theta(\log^3 N))$ -strongly-selective sequence in rounds $[t \cdot x + 1, t \cdot x + x]$. A backbone node records the names of the associated non-backbone nodes from which it receives such message as a potential representative. After time $y \cdot x$, each backbone node arranges all its associated nodes into a list and processes it as follows. For each associated node on the list, it executes some (N, c) -strongly-selective sequence of length $z = \mathcal{O}(\log N)$, for sufficiently large constant $c > 0$, transmitting the associated node's name whenever scheduled. These executions occur for every node on the list one after another. The part is completed within Δz rounds, as at most Δ associated nodes are on the list. Each associated node v remembers its position on the list of associated nodes of its potential representative, denoted σ_v , upon receiving a message addressed to it from such a backbone node.

Algorithm INTRAH_H facilitates communication among neighbors in a backbone. It is defined as follows: every non-backbone node v transmits according to a (N, c) -strongly-selective sequence of length z in rounds $[\sigma_v \cdot z + 1, \sigma_v \cdot z + z]$. The component σ_v makes it possible to schedule all the workers to avoid collisions at their shared representatives, as already determined, while the (N, c) -strongly-selective sequence component facilitates avoiding clashes between a constant number of backbone neighbors.

Lemma 8 *The local algorithms are successfully constructed in $\mathcal{O}(\Delta \log^7 N)$ rounds with high probability. INTRAH_H operates in $\mathcal{O}(\log N)$ time and INTER_H operates in $\mathcal{O}(\Delta \log N)$ time.*

Proof By Proposition 3, messages are exchanged between neighbors in the communication graph. Proposition 5 guarantees a successful message exchange between neighboring pairs of backbone nodes, as their number per box is constant. Next, we argue about computing algorithms INTER_H and INTRAH_H associated with the backbone.

Random selections allow for all non-backbone nodes to avoid collisions at their potential representatives with high probability, while an $(N, \Theta(\log^3 N))$ -strongly-selective sequence allows to avoid clashes between $\mathcal{O}(\log N)$ non-

backbone nodes of different potential representatives with high probability, since the density of backbone nodes is constant per box. One auxiliary execution of INTER_H allows to assign by each non-backbone node its representative in the backbone, as it can hear all its neighbors in the backbone due to the fact that there is a constant number of them.

A schedule for algorithm INTRAH_H can be found similarly as for part two of CONNECT- TO- MIS. Selecting a random number and a strongly-selective sequence allows each backbone node to hear all the names of neighboring non-backbone nodes with high probability. Backbone nodes send acknowledgements to their non-backbone neighbors using the same (N, c) -strongly-selective sequence as in the schedule of INTER_H , such that all non-backbone nodes receive them.

Algorithm INTRAH_H makes non-backbone nodes transmit, each using an (N, c) -strongly-selective sequence, in the order determined by the list created during the first part of procedure CONNECT- TO- MIS, see Fig. 7. Correctness follows by the property that each execution of an (N, c) -strongly-selective sequence is associated with a set of pairs involving backbone and non-backbone nodes. Hence, due to a constant density per boxes of the pivotal grid, applying the strongly-selective sequence guarantees successful deliveries of each of these pairs, by Proposition 5.

The time this computation takes is determined by the durations of all parts. Performance bounds of algorithms INTRAH_H and INTER_H follow from their specifications. A strongly-selective sequence is of length $\mathcal{O}(\log N)$. Algorithm INTER_H uses it no more than Δ times. \square

5.4 Combining ingredients into a backbone

An algorithm that builds a backbone from a synchronized start is called BACKBONE- SYNCHRONIZED- START. Its pseudocode is in Fig. 5. The algorithm begins by executing FIND- MIS, which produces a maximal independent set. This is followed by CONNECT- TO- MIS, which works with the maximal independent set produced by FIND- MIS. Finally, a backbone is completed by augmenting nodes with local algorithms, as specified in Sect. 5.3.

Theorem 2 *Algorithm BACKBONE- SYNCHRONIZED- START builds a backbone in $\mathcal{O}(\Delta \log^7 N)$ rounds with high probability, such that INTRAH_H operates in $\mathcal{O}(\log N)$ time and INTER_H in $\mathcal{O}(\Delta \log N)$ time.*

Proof The first stages produces a maximal independent set with high probability, by Lemma 6. Given a success of the first stage, the second stage constructs a connected dominating set that satisfies the specifications of a backbone, by Lemma 7. The time to construct local algorithms and their performance bounds follow from Lemma 8. \square

6 Backbone from general start

We build on algorithms presented in Sects. 4 and 5 to obtain a general distributed algorithm to construct a backbone. This algorithm is called BACKBONE- GENERAL- START. Its pseudocode is given in Fig. 8.

We assume that an arbitrary set of nodes is activated to start an execution. An activated node begins by executing algorithm FIND- MIS from Sect. 5.1. During an execution, if a node not originally activated receives a message, it ignores it and does not join yet. The outcome maximal independent set is nonempty, since the initial set S of awoken nodes is non-empty.

Next, each node in the obtained maximal independent set initiates an execution of TRAVERSE- TO- BROADCAST from Sect. 4.3 with itself as source. This may create multiple tokens that traverse the network concurrently. Each token carries its source name, to which we refer as the token's *name*. The tokens compete for survival by comparing their names. When tokens visit nodes, they leave a name behind as a trace of visit. A token that comes to a node formerly visited by other tokens is compared, by the name it brings, to the previous visitors.

A token carries a round number, which is interpreted as a round by which a full synchronization of the network will be accomplished, as determined by its originator. We want nodes to maintain consistency of such synchronization times. This is done by following the proposal of a "highest bidder," understood as a token of the maximum name seen so far. The source nodes initialize this with their own tokens, while the other nodes wait for a token to visit.

When processing messages generated in tokens traversals, all nodes participate. If there is a conflict between two messages needed to be sent at the same round then a message for the token with a greater source name takes precedence. To process received tokens, a node refers to a variable max-name. Once a node obtains a token, it proceeds in one of the following ways:

- (i) if this is a first token ever received then the node sets max-name to the name of the token, adopts the proposed round number propagated by the token and proceeds to facilitate this token's further traversal by sending the required messages;
- (ii) if this is a token with a smaller name than max-name then this token is destroyed by the node not sending messages to facilitate the token's further traversal;
- (iii) if this is a token with a name greater than max-name then the node sets max-name to the name of the visiting token, adopts the information propagated by the token, and proceeds to facilitate this token's further traversal by sending the suitable messages.

If there is a node that keeps a token's name and the associated information it carries then we say that the token is *operative* at the node. It is critical to have a bound on the number of tokens that are operative in nodes' neighborhoods, since each active node may contribute to collisions by generating messages needed for the corresponding token's traversal. This is reflected in the size of strongly-selective sequences used to resolve collisions.

Lemma 9 *There is a constant upper bound on the number of tokens that are operative in a box of the pivotal grid at any round.*

Proof Tokens start their traversal originating at nodes of a maximal independent set. There is at most one node per box in this set, since the nodes in a box induce a clique in the communication graph. If a token arrives at a node then the node establishes its children in the breadth-then-depth tree. This means that each node in the box learns about this token's arrival and if this token arrives to a child in the box while a token with a larger name has already visited the box then the smaller token gets destroyed. A token may arrive through a node's neighbor, and a node has neighbors in at most 25 boxes. If we start with at most one operative token in each box then this leads to an invariant that there are at most 25 operative tokens in a box at all times. \square

A token with the largest name eventually becomes the only survivor. Its traversal activates the whole network and synchronizes the nodes to begin a target algorithm in the same round. This algorithm is BACKBONE- SYNCHRONIZED- START from Sect. 5.4, which constructs a backbone.

Theorem 3 *Algorithm BACKBONE- GENERAL- START creates a backbone network, for any set of initially activated nodes, in $\mathcal{O}(n \log^2 N + \Delta \log^7 N)$ rounds with high probability. The total number of random bits per node is poly-logarithmic in N , and the associated local algorithms backbone algorithms operate in the same times as produced by algorithm BACKBONE- SYNCHRONIZED- START.*

Proof The algorithm resorts to the procedures that were already discussed so it suffices to revisit their performance bounds and the assumptions under which they are valid. Procedure FIND- MIS works in $\mathcal{O}(\log^8 N \log \Delta)$ rounds and produces a maximal independent set with high probability, by Lemma 6. Algorithm TRAVERSE- TO- BROADCAST accomplishes a single-source broadcast from a single-node start in $\mathcal{O}(n \log^2 N)$ rounds with high probability, with each node generating $\mathcal{O}(\log^3 N)$ random bits, by Theorem 1. These bounds remain valid with multiple tokens. This is because of a constant upper bound on the number of tokens operative that are operative in a box of the pivotal grid in any round, by Lemma 9. Algorithm BACKBONE- SYNCHRONIZED- START

```

algorithm BACKBONE-GENERAL-START
stage 1: active nodes execute procedure FIND-MIS
stage 2: each node in the maximal independent set executes TRAVERSE-To-BROADCAST
         using a dedicated token
stage 3: all nodes execute BACKBONE-SYNCHRONIZED-START at the same round

```

Fig. 8 Active nodes start from the beginning, while the other nodes join after contacted by a node visited by a token. The surviving token imposes on all the nodes a round number to be used to start executing BACKBONE-SYNCHRONIZED-START simultaneously

finds a backbone in $\mathcal{O}(\Delta \log^7 N)$ rounds with high probability, by Theorem 2. These bounds added together make $\mathcal{O}(n \log^2 N + \log^8 N \log \Delta + \Delta \log^7 N) = \mathcal{O}(n \log^2 N + \Delta \log^7 N)$ rounds with high probability. \square

7 Conclusion

We developed algorithms for broadcasting and building a backbone in the model of weak-sensitivity and weak-connectivity of wireless networks, where nodes do not know their position in a coordinate system. This extends the related results obtained for the case when nodes can refer to their coordinates in the plane, as obtained by Jurdziński and Kowalski [20] and Jurdziński et al. [26].

Jurdziński et al. [26] showed a lower bound $\Omega(\min\{D\Delta, n\})$ on time performance of broadcast and wake-up in wireless networks with a single-node start. Combining it with the performance of algorithm for building a backbone we developed demonstrates that preprocessing a network, in order to synchronize the nodes such that they can start an execution simultaneously, decreases the expected-time performance requirements for some communication tasks in wireless networks, for the model of weak sensitivity and weak connectivity.

Algorithms for a weak-sensitivity weak-connectivity SINR regime can be compared to those developed for a related model of (geometric) radio networks. Radio networks allow for randomized broadcast algorithms of running time proportional to the diameter D such that other parameters contribute sub-linear factors, see Kowalski and Pelc [28] and Czumaj and Rytter [8]. Solutions for SINR networks have to efficiently combine methods of resolving collisions not only coming from simultaneously transmitting neighbors in the communication graph but also interferences coming from other parts of the network. Broadcast requires time $\Omega(\min\{D\Delta\})$ in a weak-sensitivity weak-connectivity SINR model.

The concept of pivotal grid defined in terms of a coordinate system in a plane but it only serves the purpose to argue about correctness and performance bounds of algorithms but does not impact actions of nodes. It would be interesting to develop algorithms to build a backbone with comparable performance

bounds in general metric spaces, such as bounded-growth ones.

Backbones provide natural infrastructure of wireless networks. It would be interesting to explore leveraging backbones to support dynamic communication tasks, such as periodic broadcast, convergecast, and routing.

Funding The work of the first author was partly supported by the National Science Foundation Grant No. 1016847. The work of the second author was partly supported by the National Science Foundation Grant No. 2131538 and the Polish National Science Center NCN grant UMO-2017/25/B/ST6/02553

References

1. Avin, C., Emek, Y., Kantor, E., Lotker, Z., Peleg, D., Roditty, L.: SINR diagrams: convexity and its applications in wireless networks. *J. ACM* **59**(4), 18 (2012)
2. Barenboim L., Peleg D.: Nearly optimal local broadcasting in the SINR model with feedback. In: Post-Proceedings of the 22nd International Colloquium on Structural Information and Communication Complexity (SIROCCO), Lecture Notes in Computer Science vol. 9439, pages 164–178. Springer, 2015
3. Bodlaender, M.H.L., Halldórsson, M.M.: Beyond geometry: towards fully realistic wireless models. In: Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing (PODC), pages 347–356, 2014
4. Chlebus, B.S., Kowalski, D.R.: Almost optimal explicit selectors. In: Proceedings of the 15th International Symposium on Fundamentals of Computation Theory (FCT), Lecture Notes in Computer Science vol. 3623, pages 270–280. Springer, 2005
5. Chlebus, B.S., Kowalski, D.R., Pelc, A., Rokicki, M.A.: Efficient distributed communication in ad-hoc radio networks. In: Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), Part II, volume 6756 of Lecture Notes in Computer Science, pages 613–624. Springer, 2011
6. Chlebus, B.S., Vaya, S.: Distributed communication in bare-bones wireless networks. In: Proceedings of the 17th International Conference on Distributed Computing and Networking (ICDCN), pages 1:1–1:10. ACM, 2016
7. Clementi, A.E.F., Monti, A., Silvestri, R.: Distributed broadcast in radio networks of unknown topology. *Theor. Comp. Sci.* **302**(1–3), 337–364 (2003)
8. Czumaj, Artur, Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. *J. Algorithms* **60**(2), 115–143 (2006)
9. Daum, S., Gilbert, S., Kuhn, F., Newport, C.C.: Broadcast in the ad hoc SINR model. In: *Proceedings of the 27th International Symposium on Distributed Computing (DISC)*, Lecture Notes in Computer Science vol. 8205, pages 358–372. Springer, 2013

10. De Bonis, A., Gąsieniec, L.: Optimal two-stage algorithms for group testing problems. *SIAM J. Comput.* **34**(5), 1253–1270 (2005)
11. Derbel, B., Talbi, E.G.: Radio network distributed algorithms in the unknown neighborhood model. In: *Proceedings of the 11th International Conference on Distributed Computing and Networking (ICDCN)*, volume 5935 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2010
12. Fuchs, F., Wagner, D.: On local broadcasting schedules and CONGEST algorithms in the SINR model. In: *Proceedings of the 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, volume 8243 of *Lecture Notes in Computer Science*, pages 170–184. Springer, 2013
13. Goldsmith, Andrea J., Wicker, Stephen B.: Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Commun.* **9**(4), 8–27 (2002)
14. Goussevskaia, O., Moscibroda, T., Wattenhofer, R.: Local broadcasting in the physical interference model. In: *Proceedings of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, pages 35–44, 2008
15. Goussevskaia, O., Pignolet, Y.A., Wattenhofer, R.: Efficiency of wireless networks: approximation algorithms for the physical interference model. *Found. Trends Network.* **4**(3), 313–420 (2010)
16. Halldórsson, M.M., Holzer, S., Lynch, N.A.: A local broadcast layer for the SINR network model. In: *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 129–138, 2015
17. Halldórsson, M.M., Mitra, P.: Towards tight bounds for local broadcasting. In: *Proceedings of the 8th ACM International Workshop on Foundations of Mobile Computing (FOMC)*, page 2, 2012
18. Halldórsson M.M., Tonoyan, T.: Sparse backbone and optimal distributed SINR algorithms. *ACM Transactions on Algorithms*, **17**(2):article 17, 2021
19. Halldórsson, M.M., Tonoyan, T., Wang, Y., Yu, D.: Brief announcement: Data dissemination in unified dynamic wireless networks. In: George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 199–201. ACM, 2016
20. Jurdziński, T., Kowalski, D.R.: Distributed backbone structure for algorithms in the SINR model of wireless networks. In: *Proceedings of the 26th International Symposium on Distributed Computing (DISC)*, Lecture Notes in Computer Science vol. 7611, pages 106–120. Springer, 2012
21. Jurdziński, T., Kowalski, D.R.: Distributed randomized broadcasting in wireless networks under the SINR model. In: Kao, M.Y. (ed.) *Encyclopedia of Algorithms*. Springer, US (2014)
22. Jurdziński, T., Kowalski, D.R., Różański, M., Stachowiak, G.: Distributed randomized broadcasting in wireless networks under the SINR model. In: *Proceedings of the 27th International Symposium on Distributed Computing (DISC)*, Lecture Notes in Computer Science vol. 8205, pages 373–387. Springer, 2013
23. Jurdziński, T., Kowalski, D.R., Różański, M., Stachowiak G.: On the impact of geometry on ad hoc communication in wireless networks. In: *Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 357–366, 2014
24. Jurdziński, T., Kowalski, D.R., Różański M., Stachowiak, G.: On setting-up asynchronous ad hoc wireless networks. In: *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2191–2199, 2015
25. Jurdziński, T., Kowalski, D.R., Stachowiak, G.: Distributed deterministic broadcasting in uniform-power ad hoc wireless networks. In: *Proceedings of the 19th International Symposium on Fundamentals of Computation Theory (FCT)*, Lecture Notes in Computer Science vol. 8070, pages 195–209. Springer, 2013
26. Jurdziński, T., Kowalski, D.R., Stachowiak, G.: Distributed deterministic broadcasting in wireless networks of weak devices. In: *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, Lecture Notes in Computer Science vol. 7966, pages 632–644. Springer, 2013
27. Kesselheim, T., Vöcking, B.: Distributed contention resolution in wireless networks. In: *Proceedings of the 24th International Symposium on Distributed Computing (DISC)*, Lecture Notes in Computer Science vol. 6343, pages 163–178. Springer, 2010
28. Kowalski, Dariusz R.: Broadcasting in undirected ad hoc radio networks. *Distrib. Comput.* **18**(1), 43–57 (2005)
29. Kushilevitz, E., Mansour, Y.: An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. Comput.* **27**(3), 702–712 (1998)
30. Moses, W.K., Jr., Vaya, S.: Deterministic protocols in the SINR model without knowledge of coordinates. *J. Comput. Syst. Sci.* **115**, 121–145 (2021)
31. Reddy S.P., Kowalski D.R., Vaya S.: Multi-broadcasting under the SINR model. *CoRR*, abs/1504.01352, 2015
32. Reddy S.P., and Vaya S.: Brief announcement: Multi-broadcasting under the SINR model. In: *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 479–481. ACM, 2016
33. Scheideler C., Richa A., Santi P.: An $O(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In: *Proceedings of the 9th ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 91–100, 2008
34. Schmid S., Wattenhofer R.: Algorithmic models for sensor networks. In: *Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006
35. D. Yu, Q.S. Hua, Y. Wang, H. Tan, and F.C. Lau.: Distributed multiple-message broadcast in wireless ad-hoc networks under the SINR model. In: *Proceedings of the 19th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, Lecture Notes in Computer Science vol. 7355, pages 111–122. Springer, 2012
36. D. Yu, Q. S. Hua, Y. Wang, J. Yu, and F. C. Lau.: Efficient distributed multiple-message broadcasting in unstructured wireless networks. In: *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM)*, pages 2427–2435, 2013
37. D. Yu, Y. Wang, Q.S. Hua, and F. C. Lau.: Distributed local broadcasting algorithms in the physical interference model. In: *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 1–8, 2011
38. D. Yu, Y. Zou, Y. Zhang, F. Li, J. Yu, Y. Wu, X. Cheng, and F.C. Lau.: Distributed dominating set and connected dominating set construction under the dynamic SINR model. In: *Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 835–844. IEEE, 2019
39. Yu, J., Wang, N., Wang, G., Yu, D.: Connected dominating sets in wireless ad hoc and sensor networks—A comprehensive survey. *Comput. Commun.* **36**(2), 121–134 (2013)
40. Y. Zou, D. Yu, L. Wu, J. Yu, Y. Wu, Q.S. Hua, and F.C. Lau.: Fast distributed backbone construction despite strong adversarial jamming. In: *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*, pages 1027–1035. IEEE, 2019

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.