Connectivity and Aggregation in Multihop Wireless Networks

Marijke H.L. Bodlaender
ICE-TCS
School of Computer Science
Reykjavik University, Iceland.
mbodlaender@gmail.com

Magnús M. Halldórsson
ICE-TCS
School of Computer Science
Reykjavik University, Iceland.
magnusmh@gmail.com

Pradipta Mitra
ICE-TCS
School of Computer Science
Reykjavik University, Iceland.
ppmitra@gmail.com

ABSTRACT

We present randomized distributed algorithms for connectivity and aggregation in multi-hop wireless networks under the SINR model. The connectivity problem asks for a set of links that strongly connect a given set of wireless nodes, along with an efficient schedule. Aggregation asks for a spanning in-arborescence (converge-cast tree), along with a schedule that additionally obeys the partial order defined by the tree. Here we treat the multi-hop case, where nodes have limited power that restricts the links they can potentially form. We show that connectivity is possible for any set of n nodes in $O(\log n)$ slots, which matches the best centralized bound known, and that aggregation is possible in $O(D + \log n)$ time (D being the maximum hop-distance), which is optimal.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation—*Probabilistic computation*

General Terms

Algorithms, Design, Theory

Keywords

Wireless Networks, SINR, Multihop, Connectivity, Aggregation, Distributed Algorithms

1. INTRODUCTION

We deal in this paper with two fundamental and related problems in wireless algorithmics: connectivity and aggregation. Given a set of n wireless nodes in the Euclidean plane, the *connectivity* problem asks for a set of links (i.e. directed edges) that strongly connect the nodes along with an efficient schedule for those links. The *aggregation* problem is similar: find both a set of links forming a tree directed toward a root and a schedule that obeys the aggregation order, with links entering a node scheduled before the link leaving it. The limiting factor in both cases is interference: all of the communication goes through a single wireless channel (divided into time slots), thus links cannot be arbitrarily scheduled together. The algorithms and the results necessarily depend crucially on the adopted model of interference.

While wireless interference is notoriously difficult to model, the physical or *SINR model* has garnered reputation as a relatively close fit to reality [19, 23], and has recently received increased attention in the algorithms community. Connectivity (and aggregation) actually received the first worst-case algorithm analysis in the SINR model by Moscibroda and Wattenhofer [22]. Their result — that any set of n points can be connected in $O(\log^4 n)$ slots — has been improved over the years to $O(\log n)$ [7], and more recently that bound has been obtained with a distributed algorithm [6]. All of these results assume, however, either that power is unlimited or that the environmental noise is negligible, so that any pair of nodes could communicate, no matter how far apart.

We consider here the full multi-hop picture, where noise affects reachability and power limits restrict the way power control can overcome interference. We find that excellent connectivity and aggregation is possible in the power limited setting. Assuming connectivity is possible at all (with a little bit of slack), $O(\log n)$ slots suffice to connect any set of points, matching the bound for unbounded power. For aggregation we show that $O(D+\log n)$ slots suffice, where D is the appropriately defined diameter. This is asymptotically optimal for any input and implies therefore constant-factor approximation. Moreover, we show that these results are achievable using a distributed algorithm, where nodes begin with minimal information about the network. These are the first such results that do not depend on local density or similar structural properties of the input.

For distributed algorithms, it is important to distinguish between the initial running time of the algorithm on one hand, and the quality of the final schedule (which can then be repeatedly reused) on the other hand. The latter has already been mentioned. The initial running time of our distributed algorithm is $O(D+\log \Delta \log^2 n)$ or $O(D+g^2 \log^2 n)$ depending on one of two variations, where both Δ and g are length diversity measures discussed later.

We also take one of the first steps to relax the inherent determinism of the basic SINR model. Whereas the basic model states that reception is successful iff the measured

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
PODC'13, July 22–24, 2013, Montréal, Québec, Canada.
Copyright 2013 ACM 978-1-4503-2065-8/13/07 ...$15.00.

SINR is above a certain threshold, experimental results generally suggest an intermediate gray area, where transmissions succeed largely randomly. Motivated by these observations, we adopt a graded version of the SINR model, and show that our results hold equally well in this extended model.

In the process, we also obtain several more subtle technical contributions. We show that those results can be obtained without making strong assumptions about the environment. In particular, we do without location or neighborhood information, which would greatly simplify the task of distributed algorithms, and we make do with absolutely minimal assumptions about connectivity. We also obtain some improvements for the single-hop case, both regarding the time complexity and a new method for power assignment.

2. MODEL

Given are n wireless nodes located at points on the plane. The nodes have synchronized clocks, and start running the distributed algorithm simultaneously using slotted time. Each node has a globally unique ID.

A *link* is a directed edge between two nodes, indicating a transmission from the first node (the sender) to the second (the receiver). A link between u and v is denoted by (u, v) or by the generic ℓ. The Euclidean *distance* between two nodes u and v is denoted by $d(u, v)$ (which is the *length* of the link (u, v)). If clear from the context ℓ is also used to denote the length. We say that a link $\ell = (u, v)$ transmits to mean that the sender u transmits with v as the intended receiver. We will use, for two links $\ell = (u, v)$ and $\ell' = (u', v')$, the asymmetric distance $d_{\ell\ell'} = d(u, v')$, for the distance between the sender of ℓ and the receiver of ℓ'.

In the SINR (*signal-to-interference-and-noise ratio*) model of interference, a link ℓ is successful if,

$$SINR(\ell, L) \equiv \frac{p_\ell/\ell^\alpha}{N + \sum_{\ell' \in L \setminus \{\ell\}} p_{\ell'}/d_{\ell'\ell}^\alpha} \geq \beta \ , \quad (1)$$

where N is the ambient *noise*, β is the required SINR level, $\alpha > 2$ is the so-called path loss constant, p_ℓ is the *power* used by the sender of link ℓ, and L is the set of links transmitting simultaneously. A set of links is said to be *feasible* if it is possible to assign power to the senders to satisfy (1) for each receiver simultaneously.

We enhance this model, by allowing an intermediate region: if $SINR \in [\beta_1, \beta)$ (for some $\beta_1 < \beta$), then the transmission succeeds with probability $0 < \kappa < 1$ (for lower levels of SINR, the transmission fails with probability 1). This version of the SINR model is similar to some previous graded versions of the SINR model (e.g., [25]). In adopting this, we are motivated by experimental results that demonstrate such transitional regions where transmission is spotty, and a smaller region (corresponding to SINR $\geq \beta$) where transmissions are reliable [3, 28].

Both of our problems (aggregation and connectivity) ask for a set of links along with a schedule meeting the required conditions. If a number of links transmit in the same slot, the successes of individual links are defined by the SINR constraints, measured at the receivers. The set L in this case is the set of senders from all concurrent links. For a set of links, we say that we have a schedule of *cost* C if there is an algorithm that succeeds with high probability to successfully schedule the links in C slots. This "algorithm" could in principle be anything, but in practice will either be an assignment of each link to a fixed slot in $\{1, 2 \ldots C\}$, or a simple probabilistic algorithm. We will say that a link set is scheduled "in the aggregation order" to mean that links into a node (i.e., links for which it is the receiver) are scheduled before the link leaving it (the link for which it is the sender).

The minimum distance between nodes is 1 (which is a matter of choosing a unit). It is assumed that nodes are restricted in their use of power by a maximum power P. The SINR constraint implies that the maximum distance a node can possibly transmit is $\Delta \equiv \left(\frac{P}{N\beta_1}\right)^{1/\alpha} > 1$. For any fixed constant $\epsilon < 1$, let $G = G_\epsilon$ be the undirected graph on the nodes with edges between nodes of distance at most $(1-\epsilon)\Delta$. We assume that G is connected, for some given value ϵ; we then say that the set of nodes is ϵ-*connected*. Let D be the diameter of G. Set the constant $\epsilon' = \min(\epsilon, \left(\frac{\beta}{2\beta_1}\right)^{1/\alpha}, \frac{1}{2}^\alpha)$

A *length class* is a set of (potential) links with lengths within a factor 2. We define g as the number of non-empty length classes, or:

$$g = |\{i \leq \log \Delta + 1 : \exists\ x, y \text{ such that } d(x, y) \in [2^i, 2^{i-1}\}|$$

While $\log \Delta + 1$ is an obvious upper bound on the number of length classes into which the links can be partitioned, g provides a tighter bound by counting only non-empty classes.

We shall use "time" to refer to the the initial number of slots required for algorithm to form the required aggregation or connectivity network, and "cost" to refer to the number of slots in the final schedule. When clear from the context, we may drop the qualification "with high probability".

We shall use the following particular Chernoff bound (see Chapter 4 in [20]): Let $\{X_i\}$ be independent Poisson trials such that $X = \sum_i X_i$ and $\mathbb{E}(X) = C \ln n$, for some $C > 0$. Then,

$$\mathbb{P}(X \leq (1 - \delta)\mathbb{E}(X)) \leq n^{-C\delta^2/2} \ . \quad (2)$$

Assumptions

Recall that we assumed that it suffices to use links of length at most $(1 - \epsilon)$-fraction of maximum possible. This assumption is unavoidable, since arbitrarily weak links can require arbitrarily bad schedules. We assume the nodes know (a lower bound on) ϵ.

We assume that nodes know the signal transmission parameters $N, \alpha, \beta, \beta_1$. These values can be arbitrary, except necessarily $\alpha > 2$ (as otherwise limited interference integrated over the whole plane goes to infinity). Also, that they know a polynomial approximation to n (i.e., the value of $\log n$, up to constant factors), without which there is a $\Omega(n/\log n)$-lower bound [11]. In the Δ-based algorithms, the nodes need a polynomial approximation of Δ, while the g-based algorithms do not assume prior knowledge of g.

We assume that nodes can measure the SINR (in case of a successful reception), or total received power (in other cases). This power reception feature is comparable to the RSSI function of real wireless motes. Nodes can use this feature to measure distances from the sender of a received message. In some parts of our algorithm, only measuring whether or not the SINR crosses a threshold is needed. We clarify this usage in the respective sections.

We do not assume nodes have knowledge of their locations in space, the diameter D, nor anything about their neighborhood.

Implications of the graded model

Note that the graph G is defined in terms of β_1, not β. Thus the algorithm is forced, at least on occasion, to use links that succeed only probabilistically. Note that by repeating the same transmission $\approx \log n$ times, links with $SINR \in [\beta_1, \beta)$ can be converted to a link succeeding with high probability. We are interested in bounds better than those that incur this obvious multiplicative $O(\log n)$ factor. Indeed, the bounds we derive are optimal even if links succeeded with probability 1 in the $[\beta_1, \beta)$ range.

3. RESULT AND RELATED WORK

THEOREM 1. *There is a distributed algorithm that runs in time $O(D + \log \Delta \log^2 n)$ (alternatively, $O(D + g^2 \log^2 n)$) that produces an aggregation network of cost $O(D + \log n)$ and connectivity network of cost $O(\log n)$ on any given ϵ-connected set of n nodes, for any fixed ϵ.*

The SINR model was first proposed in an influential paper of Gupta and Kumar [5], who showed that $O(\log n)$-slots suffice to connect a set of n uniformly distributed nodes. Moscibroda and Wattenhofer were the first to formalize the connectivity problem from a worst case perspective in the SINR model in [22]. They proposed a centralized algorithm that connects an arbitrary set of n nodes in $O(\log^4 n)$ slots. This was improved to $O(\log^3 n)$ [24], $O(\log^2 n)$ [21], and $O(\log n)$ [7].

Distributed algorithms for aggregation in the SINR model include [18], [9] and [1]. These works have network costs that are polynomials in $O(\log \Delta)$ or $O(d)$ (where d is the max-degree, i.e., the number of nodes within a radius of $\approx \Delta$). Dependence on degrees or $\log \Delta$ make these works closer to disk graph models, and the power of the SINR model in handling density (as seen by the works cited in the previous paragraph) is not fully utilized.

In [6] a distributed algorithm for finding a tree that connects an arbitrary set of n nodes in $O(\log n)$ slots was given, matching the best centralized result known [7]. Our work builds upon this work, and we extensively use, modify or improve techniques from this paper. It was also shown there that such results require the use of arbitrary power control; namely, algorithms using power assignments that are functions of link length alone are forced to use $\Omega(n)$ slots in worst case.

Another set of related results involves finding dominating sets and/or a broadcast network in a multi-hop scenario. The work we directly use is that of [26], where an algorithm is provided to find dominating sets in the SINR model in $O(\log n)$ time. Also relevant is [17], where a dominating set is constructed using a distributed algorithm in a quasi unit disk model, which can be converted to the SINR model (see [27]). These works do not attempt to construct a network among dominators. Broadcast or aggregation networks among dominators are formed in some works [27, 10] (as well as the works in the previous paragraph), but none give quite what we need. They either need a large ϵ (at least 2/3) to work [27, 9], use precise location information [10], and/or do not form an aggregation network [10, 27].

4. ALGORITHM OUTLINE

Our algorithm has two major parts. In the first part, we select a small set of dominators such that all (other) nodes are within a small distance of a dominator. We then construct a low-cost aggregation or connectivity network between the dominators. This network can be thought of as the backbone of the combined network. To achieve this, we use existing work to find a dominating set, and then show to how form the network among them. For this part, the initial running time (as well as the cost of the final schedule) is $O(D + \log n)$ for aggregation and $O(\log n)$ for connectivity. This is detailed in Section 5.

The second part deals with clusters (i.e. a dominator and nodes dominated by it). Each cluster is a single-hop environment, so we can apply ideas from [6], with technical changes needed to take care of power limits and the fact we are computing networks for different clusters simultaneously. We also improve the running time. After these conversions, we compute an aggregation (or connectivity) network. The initial running time is either $O(\log \Delta \log^2 n)$ or $O(g^2 \log^2 n)$, depending on which algorithm is used, and the final schedule cost is $O(\log n)$. These bounds apply both for connectivity and aggregation. This is detailed in Section 6.

Each part is divided into multiple sub-parts, which are described in the appropriate subsections.

5. DOMINATING NETWORK

5.1 Finding a dominating set

An *R-dominating set* is a subset of nodes such that each input node is within a distance R from a dominator (possibly itself). A *clustering* is a function f assigning each node a valid dominator (i.e., within distance R).

An *R-ball* is a disk in the plane of radius R. The *density* of an R-dominating set is the maximum number of dominators in an R-ball (over all balls in the plane). Let

$$\eta = \epsilon' \Delta / 4 \ .$$

By running the dominating-set algorithm of Scheideler et al. [26] with adjusted power settings, their Theorem 2.1 can be rephrased as:

THEOREM 2. *There is a distributed algorithm running in time $O(\log n)$ that produces, with high probability, a η-dominating set of constant density, along with the corresponding clustering function.*

Note that this result does not hold immediately in the graded SINR region of $[\beta_1, \beta)$, but since $\epsilon' \leq \left(\frac{1}{\beta_1}\beta\right)^{1/\alpha}$, by definition, no communication uses the graded SINR region.

5.2 Network Formation

We next seek an efficient network on top of the η-dominating set. Note that dominators can be far enough apart that the use of the transitional region is unavoidable. This is the only section where we use such links.

THEOREM 3. *There is a distributed algorithm that runs in time $O(D + \log n)$ and forms an aggregation network with cost $O(D + \log n)$ on any given η-dominating set of constant density.*

In proving this, we focus on forming an aggregation network, with a connectivity network achieved along the way.

Note that we only deal with dominators in this section. Thus when we talk about properties of the node set (like the minimum ID, or number of neighbors) we mean the set of dominators, not the original set.

Since we have an η-dominating set, the graph G' formed by connecting nodes at distance at most $\Delta(1-\epsilon'+2\frac{\epsilon'}{4}) = \Delta(1-\frac{1}{2}\epsilon')$ is connected. In this section, graph terminology such as "neighbors" refers to G'.

We shall use the following two simple primitives:

com A COM consist of a slot in which each nodes transmits with a (low enough) fixed probability q.

ncomm An NCOMM (neighborhood communication) consists of $O(\log n)$ COM slots (with a sufficiently large implicit constant).

We shall say that a node u *informs* a neighbor v if it successfully transmitted to v in a given step. The following lemma encapsulates our requirements of the interference model. The results of this section will apply to any model for which this lemma holds.

LEMMA 4. *During* COM, *any given node u informs any given neighbor v with probability at least $\zeta \doteq \frac{1}{2}\kappa q(1-q)$.*

The proof of this lemma (given in Appendix A) uses the now standard technique of bounding interference within concentric circles.

A straightforward application of the Chernoff bound (2) gives that each node successfully transmits to each of its neighbors $\Omega(\log n)$ times (details provided in Appendix A).

LEMMA 5. *During* NCOMM, *each node informs each of its neighbors ($\Omega(\log n)$ times), w.h.p.*

For simplicity, in what follows, we will consider communication with neighbors during an NCOMM as a deterministic event.

Note that COM and NCOMM apply in the transitional region – and of course in the safe region of $SINR \geq \beta$. These lemmas abstract away this particular issue, which then needs no further consideration.

Using these primitives as building blocks, the algorithm can be outlined as follows: The eventual root of the network will be the node with the minimum ID. All nodes start a flooding process to find out the minimum node (assuming initially that the node itself is the minimum). Eventually the global minimum wins and the edges involved in the flooding initiated by the root form the final tree.

An initial NCOMM informs all nodes of the IDs of their neighbors. Node u maintains the following state:

$r(u)$ the smallest ID currently known by u. Initially $r(u) = u$.

$p(u)$ the "parent" of u, or the node that *first* informed u about $r(u)$. Initially $p(u) = u$.

We use the term "child" (and children) of a node u to mean a node v such that $p(v) = u$.

In each slot, a node transmits a message using COM. There are two types of messages — M and A. Intuitively, an M-message is a broadcast message intended to inform neighbors about a node's current state (especially $r(u)$), while an A-message acts as an acknowledgment intended for the node's parent (and ultimately the root). They both contain the same information (along with the message type) — the sender u, and the current values of $p(u)$ and $r(u)$. The conditions for sending these messages and the behavior upon receiving them for a node u are as follows:

1. For each neighbor v, let $m(v)$ be the last message received by u from v. Condition SameRoot holds if the $r(v)$ value contained in $m(v)$ is the same as the current $r(u)$. Condition ChildAck holds true for $m(v)$ if the $p(v)$ value in the message is equal to u. Node u will decide on an A-message in slot $t+1$ if u has received at least one message from each of its neighbors v by the end of slot t, and for each message: Condition SameRoot holds, and either ChildAck is false, or ChildAck holds and the message is an A-message. If the conditions hold but $p(u) = u$, the node decides to become the root instead of sending an A-message.

2. If the conditions of the previous paragraph do not apply, u decides on an M-message with the current values of $r(u)$ and $p(u)$.

The node transmits the message it has decided on with probability q, and listens for messages otherwise.

At the end of every step, node u will update its state as follows: If u has not received a message in the slot, nothing happens. Otherwise, suppose u received a message from v. If $r(u) > r(v)$ ($r(v)$ as included in the message), then u sets $r(u) \leftarrow r(v)$ and $p(u) \leftarrow v$.

The following lemma claims a time bound on how quickly nodes learn about the global minimum.

LEMMA 6. *Let $u_{\min}$ be the node with the globally minimum ID. Then, for each node u, $r(u) = u_{\min}$ by $O(D+\log n)$ slots, w.h.p.*

PROOF. It is clear that once $r(v) = u_{\min}$ for a node v, it remains that way. Let us call a node for which this is true *communicated*. The message type does not matter since $r(u)$ is updated regardless (if it has not already been communicated).

Let R_t be the set of nodes communicated after t slots with the convention $R_0 = \{u_{\min}\}$. Now fix a node u and define d_t be the distance from u to the nearest node in R_t. Note that $d_0 \leq D$. Define $\delta_t = d_{t-1} - d_t$ as the progress made at time t. Now, δ_t is a Bernoulli random variable with $\mathbb{E}[\delta_t] \geq \zeta$ for all t, as long as $d_t > 0$; $\delta_t = 0$ after that. Let $\tilde{\delta_t}$ be Bernoulli random variable where $\tilde{\delta_t}$ has the distribution of δ_t as long as $d_t > 0$ (equivalently $\sum_{i=1}^{t}\delta_i < d_0$), and an i.i.d. Bernoulli random variable with expectation ζ after that. Let $\tilde{\Delta}_t = \sum_{i=1}^{t}\tilde{\delta}_t$. It is clear that for any t, $\mathbb{P}\left(\sum_{i=1}^{t}\delta_i < d_0\right) = \mathbb{P}\left(\tilde{\Delta}_t < d_0\right)$, thus we can focus on $\tilde{\delta_i}$ exclusively.

Define $Z_t = \tilde{\Delta}_t - \zeta t$. Now, $\mathbb{E}(Z_t|\delta_1,\delta_2\ldots\delta_{t-1}) = \tilde{\Delta}_{t-1} - \zeta(t-1) + \mathbb{E}(\tilde{\delta_t}) - \zeta \geq Z_{t-1}$. Thus, Z_t is a sub-martingale. If $t \geq 5c_1\frac{1}{\zeta}(D+\log n)$ for a large enough constant c_1, then the event that $\tilde{\Delta}_t < d_0$ implies that $Z_t < d_0 - \zeta t \leq D - 5c_1(D+\log n) \leq -4c_1(D+\log n)$. We can now upper bound $\mathbb{P}(Z_t \leq -4c_1(D+\log n))$ using the Azuma-Hoeffding inequality (see, e.g., [20, Thm. 12.4]). The Azuma-Hoeffding inequality for sub-martingale Z_t with $|Z_t - Z_{t-1}| \leq c_t$ looks as follows

$$P(|Z_t - Z_0| \leq -x) \leq \exp\left(\frac{-x^2}{2\cdot\sum_{k=1}^{x} {c_t}^2}\right) .$$

Filling in the value for t, $c_t \leq 1$, and $x = 4c_1 \frac{1}{\zeta}(D + \log n)$ yields

$$P(Z_t \leq -4c_1(D + \log n)) \leq \exp\left(\frac{-(4c_1(D+\log n))^2}{2 \cdot \sum_{k=1}^{(5c_1(D+\log n)} 1}\right) \leq \exp(-2c_1(D + \log n)/5) \ .$$

A union-bound over all nodes u then yields the lemma. □

Lemma 6 describes dissemination of root information to other nodes, but we also need to ensure that $u_{\min}$ realizes this fact quickly.

LEMMA 7. *Let $u_{\min}$ be the node with the globally minimum ID. Then by $O(D + \log n)$ slots, $u_{\min}$ decides to become the root.*

PROOF. Assume that the process described in Lemma 6 has communicated everyone. The passing of messages continues, and we are now interested in tracking the progress of the A-messages. A node starts to transmit A-messages once it has received the same from all of its children. Our goal will be to claim that $u_{\min}$ will fulfill the requirements of sending an A-message in $O(D + \log n)$ steps, and, according to the algorithm description, decide to become the root.

Consider a modified process, where within the M-messages that a node transmits, it forwards the A-messages from its descendants. Thus, an A-message will not be stopped by waiting for a child, but will travel up the tree independent of what happens in subtrees outside that path. This modified process completes once A-messages from all nodes in the tree have been forwarded to the root. It should be clear that this occurs at exactly the same time as when the root receives the last A-message from its children in the original process.

In the modified process, we can track the progress of each path $\langle u_0, u_1, \ldots, u_{k-1}, u_{\min}\rangle$ from a leaf u_0 to the root. The process will remove nodes from this path in the order $u_0, u_1 \ldots$. The current node u_i is removed with a probability $\mathbb{P}(u_i)$, and this probability is set to the probability of a message being successfully transmitted from u_i to u_{i+1}, which can be lower bounded by ζ (as described in Lemma 6). The process can be modeled as a sub-martingale essentially identically the process in Lemma 6 and given a similar concentration bound. A union bound over all leaf-to-root paths then yields the lemma. □

We also need to make sure of the following.

LEMMA 8. *No node other than the global minimum decides to become the root.*

PROOF. By contradiction, assume $v > u_{\min}$ did decide to become the root. As before, consider the tree between nodes with $r(u) = v$ with directed links from nodes to their parents. All nodes with $p(u) = v$ must clearly be part of v. Also, to become the root, v needs to receive an A-message from its children in the tree, and this recursively is true for all nodes in the tree. Thus the links in the tree represent A-messages as well.

Consider any simple path from a node in the tree to $u_{\min}$. This path will not be empty since $u_{\min}$ will never be part of the tree. Let x be the first node on this path and y be a neighbor of x in the tree. If $r(x) = v$ then x has to be part of the tree, a contradiction. On the other hand, if $r(x) \neq v$ then y will never fulfill the requirement needed for an A-message and will not be part of the tree. □

5.3 Stopping criterion and Schedule construction

The algorithm as described does not have a stopping criterion. This issue is easily dealt with. Once deciding to become a root, $u_{\min}$ can flood a high priority termination message that will force out other messages and inform all nodes in $O(D + \log n)$ rounds that a root has been found.

Since links used in this section can have $SINR \in [\beta_1, \beta)$, we cannot compute a fixed schedule that is guaranteed to work (since, links can fail even without any interference). However, the following scheme will, with high probability, schedule all links in $O(D + \log n)$ time in the aggregation order. The links are simply the tree of A-messages ending up in $u_{\min}$. Nodes can identify these links by one use of NCOMM. When scheduling these links in future, a node transmits its outgoing link in each slot with probability q once it has received messages from its children. This is essentially a repetition of the process described in Lemma 7 and will have the same running time.

5.4 Carrier sense assumption

For the algorithms of this section to work, we do not need the full SINR or received power primitive assumed in Section 2. It is enough to know whether or not the received power has exceeded a certain threshold. This applies both to the result from [26] to form the dominating set, and the network formation part. In the later part, nodes need to make sure that they accept messages only from neighbors. For this, it is enough to know if the distance from a sender has crossed a certain threshold, equivalent to the crossing a received power threshold, as mentioned.

6. CONNECTIVITY IN A CLUSTER

Our goal in this section is to form a network among nodes in a cluster, for all clusters simultaneously. In this part, links will only form between nodes in a cluster, ending up with an aggregation tree with the dominator as the root. As the previous section provides aggregation among the dominators, these two structures, combined, provide the total aggregation tree. Connectivity is achieved similarly. Since links will only form between nodes in a cluster, which have small radii, we need not worry about the transitional SINR phase for this section: all established links in this section will have $SINR \geq \beta$ and thus will succeed with probability 1.

We use the following definition:

DEFINITION 9. *A set of clusters is* well-separated *if nodes in different clusters are of distance at least*

$$\hat{R} \doteq \Upsilon \cdot \epsilon' \cdot \Delta \ ,$$

for a sufficiently large constant Υ.

Claiming the following:

CLAIM 6.1. *The clusters are well-separated.*

To achieve this, we run a a simple coloring scheme using $O(1)$ colors, running in time $O(\log^2 n)$ (which is subsumed by the overall runtime of this section). Details are provided in Lemma 30 of Appendix B. The final aggregation schedule contains the schedule for each color in order, followed by the schedule for the dominators.

Since nodes can measure distances from message senders, they can easily filter out stray communication from nodes of different clusters via Claim 6.1.

We provide two methods to form connectivity or aggregation in a cluster. Both algorithms are based on the single-hop algorithm from [6]. The first is identical to it except for the power assignment part (where we improve the running time), and the second is a modification to achieve a potentially faster g-based run-time. We will address aggregation in the following; identical results apply for connectivity in all cases.

6.1 $O(\log \Delta)$-based runtime

THEOREM 10. *There is a distributed algorithm running in time $O(\log \Delta \log^2 n)$ that computes, simultaneously for all clusters, an aggregation network with the dominator as the root, and a schedule of length $O(\log n)$.*

We prove this by following closely arguments from [6], with repeated use of an algorithm finding a large feasible subset according to the following lemma:

LEMMA 11. *There is a distributed algorithm running in time $O(\log \Delta \log n)$ that, given a set of m nodes divided into b disjoint clusters, finds a feasible set L of intra-cluster links with $\mathbb{E}(|L|) = \delta \cdot (m - b)$, for some fixed constant δ.*

We call a node v a *top* node with respect to a link set L if v is not a sender of a link in L. We use the following algorithmic framework (defined for a single cluster):

Algorithm 1 ClusterTree

```
Set i = 0 and M_i = M (the original input set)
for i = 0, 1, 2 ... until |M_i| = 1 do
    Construct feasible set L on M_i using alg. from Lem. 11
    Let M_{i+1} be the set of top nodes w.r.t. L
end for
Construct a link ℓ between {M_i} and the cluster dominator.
```

We output the schedule where the nodes in set M_i send in slot i. We claim that this process takes $O(\log n)$ steps to complete.

LEMMA 12. *Algorithm 1 ends after $O(\log n)$ iterations, w.h.p., producing a spanning aggregation tree.*

PROOF. Note that by Lemma 11, $\mathbb{E}(|M_{i+1}|) \leq |M_i| - \mathbb{E}(|L|) \leq (1 - \frac{1}{2}\delta)|M_i|$. We use this to argue termination.

CLAIM 6.2. *$\mathbb{P}(|M_t| > 1) \leq n^{-4}$, for $t = 6\frac{1}{\delta}\ln n$.*

PROOF. Since M_i is non-increasing in i, for contradiction, condition on all $M_i \geq 2$ for $i \leq t$. Then by Lemma 11

$$\mathbb{E}(|M_t|) \leq \left(1 - \frac{1}{2}\delta\right)^{6\frac{1}{\delta}\ln n} \frac{1}{n} \leq e^{-3\ln n}\frac{1}{n} = n^{-4} ,$$

from which the claim follows by Markov's inequality. ☐

For any i, each node in $M_{i+1} \setminus M_i$ is connected by a link to a node in M_i. Thus, every node is connected to a root, and thus the structure is an aggregation tree. Note that by the way the algorithm proceeds, the scheduling order of the links follows the direction of the links in the tree. Also, note that since each iteration uses a single slot, the bound on iterations implies the bound on the number of slots in the schedule, thus proving the theorem.

In the remainder of Sec 6.1, we will prove Lemma 11.

6.1.1 Choosing a feasible set

First, a few definitions. Given ν, define a *ν-class-partition* to be a partition of a link set L into ν length classes $L_1 \dots L_\nu$ sorted in descending order (with links in L_i longer than those in L_{i+1}), along with the assumption that all links (i.e. nodes involved in the links) *know* the index i of the length class they belong to. Given a class-partition and a link ℓ in class i, define $S_\ell^- = \cup_{j \leq i} L_j \setminus \{\ell\}$ (i.e., the set of links in the same or longer length classes), and similarly, $S_\ell^+ = \cup_{j \geq i} L_j \setminus \{\ell\}$.

The following was essentially shown in [6]:

THEOREM 13. *Consider a set of n nodes partitioned into well-separated clusters. Then, there is an algorithm running in time $O(\log \Delta \log n)$ that finds a set L of links of expected $\Omega(n)$ size, along with a $\log \Delta$-class-partition, with the property that for each link ℓ in L,*

$$\sum_{\ell' \in S_\ell^+} \left(\frac{\ell'^\alpha}{d_{\ell\ell'}^\alpha} + \frac{\ell'^\alpha}{d_{\ell'\ell}^\alpha} \right) \leq \tau, \text{ with } \tau = \frac{1}{2\beta(1 + 4 \cdot 3^\alpha)} . \quad (3)$$

In [6] (Sections 6, 7 and 8), this result was proven for one cluster. But given well-separated clusters (Claim 6.1), the arguments go through easily (details omitted). From this, the following lemma follows easily (the proof is provided in Appendix B):

LEMMA 14. *The link set L found by the algorithm of Thm. 13 satisfies*

$$\sum_{\ell' \in S_\ell^-} \frac{\ell^\alpha}{d_{\ell'\ell}^\alpha} = O(1) , \quad (4)$$

for each link $\ell \in L$.

To find a feasible power assignment we need to tighten property in Lemma 14. The proof of the following lemma is given in Appendix B.

LEMMA 15. *Let L' be a set of n links, along with a ν-class-partition, satisfying Eqn. 4, such that for each link ℓ in L',*

$$\sum_{\ell' \in S_\ell^-} \frac{\ell^\alpha}{d_{\ell'\ell}^\alpha} \leq C ,$$

for some constant C. Then, there is a distributed algorithm to find in $O(\nu)$ time a subset L of $\Omega(n)$ links, each satisfying

$$\sum_{\ell' \in L} \frac{\ell^\alpha}{d_{\ell'\ell}^\alpha} \leq \frac{1}{4\beta} . \quad (5)$$

6.1.2 Finding a feasible power assignment.

With these results in hand we give a distributed algorithm computing a feasible power assignment respecting the power limit. Assume we are given a set L of links satisfying Eqn. 5 along with a class-partition $L_1, \dots L_\nu$.

The algorithm runs in ν rounds, with links in L_i computing their powers simultaneously in round i in two steps.

First, all links ℓ' in $\cup_{j<i} L_j$ (which have already received their power) transmit with their assigned power $p_{\ell'}$. Each link ℓ in L_i measures the interference I_ℓ^- at its receiver:

$$I_\ell^- = N + \sum_{\ell' \in \cup_{j<i} L_i} p_{\ell'}/d_{\ell'\ell}^\alpha \ .$$

Then, ℓ computes its power as

$$p_\ell = 2\beta \ell^\alpha I_\ell^- \ . \tag{6}$$

To show feasibility with limited power we need to show that: a) power limits are not exceeded, and b) the SINR constraints are fulfilled.

LEMMA 16. *All powers assigned by the algorithm above are at most the maximum power P.*

PROOF. The proof is by induction, closely resembling the proof of Thm. 2.22 from [14]. Note that since we only have intra-cluster links, we have that $\ell \leq \frac{1}{2}\epsilon'\Delta \leq \frac{1}{2}\left(\frac{P}{2N\beta}\right)^{1/\alpha}$, by definition of ℓ, ϵ' and Δ, which implies that $P \geq 4\beta N \ell^\alpha$.

Links $\ell \in L_1$ compute $p_\ell = 2\beta\ell^\alpha N$, which is clearly less than P. For $i > 1$, all links ℓ' in length classes $L_j, j < i$, have $p_{\ell'} \leq P$, by the inductive hypothesis. Thus, the power p_ℓ of link ℓ in L_i satisfies

$$\begin{aligned} p_\ell &= 2\beta N \ell^\alpha + 2\beta \sum_{\ell' \in \cup_{j<i} L_j} \frac{p_{\ell'}}{d_{\ell'\ell}} \ell^\alpha \\ &\leq 2\beta N \ell^\alpha + 2\beta P \sum_{\ell' \in S_\ell^-} \frac{\ell^\alpha}{d_{\ell'\ell}} \\ &\leq 2\beta N \ell^\alpha + 2\beta P \frac{1}{4\beta} \\ &\leq P \ , \end{aligned}$$

using the inductive hypothesis, Eqn. 5, and the bound $4\beta N \ell^\alpha \leq P$, respectively. □

THEOREM 17. *The power assignment computed on set L is feasible.*

PROOF. As mentioned, the algorithm is a parallel version of an algorithm presented in [14, 15]. We show that this is not a problem, and proof ideas from those papers carry through.

We can bound the final interference received by a link ℓ as $I_\ell^+ + I_\ell^-$, where $I_\ell^+ = \sum_{\ell' \in S_\ell^+} p_{\ell'}/d_{\ell'\ell}^\alpha$ and $I_\ell^- = N + \sum_{\ell' \in S \setminus (S_\ell^+ \cup \{\ell\})} p_{\ell'}/d_{\ell'\ell}^\alpha$. Note that $I_\ell^- = \frac{1}{2\beta} p_\ell/\ell^\alpha$.

We first expand I_ℓ^+ using the powers assigned:

$$I_\ell^+ = \sum_{\ell' \in S_\ell^+} p_{\ell'} \frac{1}{d_{\ell'\ell}^\alpha} \tag{7}$$

$$= \sum_{\ell' \in S_\ell^+} \left(2\beta N \ell'^\alpha \frac{1}{d_{\ell'\ell}^\alpha} + 2\beta \sum_{\ell'' \in S_{\ell'}^-} \frac{1}{d_{\ell'\ell}^\alpha} \left(p_{\ell''} \frac{\ell'^\alpha}{d_{\ell''\ell'}^\alpha} \right) \right) \ . \tag{8}$$

The first term is bounded by $2\beta N \sum_{\ell' \in S_\ell^+} \ell'^\alpha / d_{\ell'\ell}^\alpha \leq 2\beta N \tau$, by Eqn. 3. By rearranging indices, the second term is bounded by

$$\begin{aligned} &\sum_{\ell' \in S_\ell^+} 2\beta \sum_{\ell'' \in S_{\ell'}^-} \frac{1}{d_{\ell'\ell}^\alpha} \left(p_{\ell''} \frac{\ell'^\alpha}{d_{\ell''\ell'}^\alpha} \right) \\ &\leq 2\beta \left(\sum_{\ell'' \in S_\ell^-} \sum_{\ell' \in S_\ell^+} \frac{p_{\ell''} \cdot \ell'^\alpha}{d_{\ell'\ell}^\alpha \cdot d_{\ell''\ell'}^\alpha} + \sum_{\ell'' \in S_\ell^+} \sum_{\ell' \in S_\ell^+} \frac{p_{\ell''} \cdot \ell'^\alpha}{d_{\ell'\ell}^\alpha d_{\ell''\ell'}^\alpha} \right) \ . \end{aligned}$$

The internal sum $\sum_{\ell' \in S_\ell^+} \frac{p_{\ell''} \cdot \ell'^\alpha}{d_{\ell'\ell}^\alpha d_{\ell''\ell'}^\alpha}$ can be bounded in both cases by $2 \cdot 3^\alpha \cdot \tau \cdot \frac{p_{\ell''}}{d_{\ell''\ell}^\alpha}$, following the precise arguments in Observation 4 of [13]. The first sum becomes

$$2\beta \sum_{\ell'' \in S_\ell^-} 2 \cdot 3^\alpha \cdot \tau \cdot \frac{p_{\ell''}}{d_{\ell''\ell}^\alpha} \leq 2 \cdot 3^\alpha \cdot \tau \cdot p_\ell/\ell^\alpha \ ,$$

by the definition of p_ℓ. Using (7), the second sum is smaller than $4\beta \cdot 3^\alpha \cdot \tau \cdot I_\ell^+$. Thus,

$$I_\ell^+ \leq 2\beta N\tau + 2 \cdot 3^\alpha \cdot \tau \cdot p_\ell/\ell^\alpha + 4\beta \cdot 3^\alpha \cdot \tau \cdot I_\ell^+ \ ,$$

which we can solve for I_ℓ^+, obtaining:

$$I_\ell^+ \leq \frac{2\beta N + 2 \cdot 3^\alpha \cdot p_\ell/\ell^\alpha}{1/\tau - \beta \cdot 3^\alpha} \ . \tag{9}$$

Using the bound $2\beta N \leq p\ell/\ell^\alpha$ in (9) and plugging in the value for τ gives $I_\ell^+ \leq \frac{1}{2\beta} p_\ell/\ell^\alpha$ and thus $I_\ell^+ + I_\ell^- \leq \frac{1}{\beta} p_\ell/\ell^\alpha$, implying the required SINR. This together with the proof for Lemma 16 concludes the proof. □

6.2 *g*-based run-time

To remove dependence on $\log \Delta$, we need to allow links to form without careful round-by-round control. Indeed, this is not too difficult, a $O(g \log n)$ algorithm that connects a set of n nodes is easy to demonstrate (and has been done recently in the unpublished [2]). However, for us, merely forming a network is not enough. It has to have the structure allowing extraction of a large feasible subset with Eqn. 5 holding. To do this, we must exert control over link length without the stringent conditions of the $O(\log \Delta)$-based algorithm.

We prove the following theorem:

THEOREM 18. *There is an algorithm running in time $O(g^2 \log^2 n)$ that forms an aggregation network for all clusters with cost $O(\log n)$.*

We need to prove the following counterpart of Lemma 11:

LEMMA 19. *There is a distributed algorithm running in time $O(g^2 \log n)$ that, given a set of m nodes divided into b disjoint clusters, finds a feasible set L of intra-cluster links with $\mathbb{E}(|L|) = \delta \cdot (m - b)$, for some fixed constant δ.*

Once this lemma is proven, the overall performance of the algorithm from Thm. 18 follows from the same argumentation for the $\log \Delta$ based algorithm.

We first provide an algorithm to form a set in $O(g \log n)$ time that connects the nodes (without a cost guarantee at this point):

LEMMA 20. *There is an algorithm that connects the nodes (and finds an aggregation and a complementary broadcasting network) in each cluster in time $O(g \log n)$.*

Once we prove this, we will relate it to feasibility and power assignments in Section 6.2.1 to achieve Lemma 19.

The algorithm proceeds in rounds running in $O(\log n)$ time and continues until there is only one active node in each cluster. When this happens the dominators of the clusters will terminate the algorithm. We will prove in the analysis that the latter will happen within g rounds, thus explaining the running time.

Each round contains three phases. The first two phases run in $O(\log n)$ time. In the first phase active nodes decide whether to participate and in the second phase links are formed. The last phase only takes a constant number of slots for the dominator to decide whether to proceed to the next round or to terminate the algorithm. All nodes start out active.

Participation Decision (PD): During this phase, an active node u transmits with probability q for $\frac{48}{q(1-q)} \ln n$ slots. When not transmitting, it listens for messages. If, at the end of this phase, u has heard from *some* node more than $12 \ln n$ times, it sets $\delta_u = d(u, z)$, where z is the nearest node from which it has heard in this phase. Otherwise it sets $\delta_u = 0$.

Link Formation (LF): This phase contains $\frac{48}{q(1-q)} \ln n$ slot pairs (a slot pair is simply two consecutive slots). In the first slot still active nodes transmit with probability q. If a non-transmitting (but active and participating) node u receives such a message from v such that $d(u, v) \leq 4 \min\{\delta_u, \delta_v\}$ (where δ_v is encoded in the message from v), then u acknowledges the message with probability q. If v receives the acknowledgment, the link $\ell = (v, u)$ is created and v becomes inactive.

Termination Decision: In the first slot all active nodes transmit and the dominators listen. If a dominator received no message, the algorithm proceeds to the next round. If the dominator did receive a message from a node x, then the dominator broadcasts in the following slot telling all active nodes except x to retransmit in the next slot. If in this next slot the dominator receives a message, the algorithm proceeds to the next round. If the dominator doesn't receive a message, it measures the received power. If the received power is $> 2P/(\epsilon'\Delta)^\alpha$ the algorithm proceeds to the next round (note that this is the lower bound on the received power from a node in the same cluster). Otherwise the dominator decides that x is the only active node in the cluster and the cluster stops trying to form new nodes. In the last slot of the phase the dominators broadcast their decisions.

We claim that this algorithm connects the set (with an aggregation and a complementary broadcasting network) in time $O(g \log n)$.

We prove the following:

THEOREM 21. *Within the first g rounds, the above algorithm finds a set of links that connects each cluster.*

To prove this we first show the following:

LEMMA 22. *Consider the execution of any round of the algorithm, and assume that the minimum distance among active nodes is d. Let $\phi(u)$ be the distance from active node u to the nearest active node. Then:*

1. *If $\phi(u) \in [d, 2d)$, then $\delta_u = \phi(u)$.*
2. *If $\delta_u > 0$, then $\delta_u = \phi(u)$.*

PROOF. The first part says that nodes that have the closest possible neighbors (up to a factor of 2), hear from them $\Omega(\log n)$ times, thus setting $\phi(u)$ as claimed in the Lemma. We prove this using a technique very similar to Lemma 4 (also known from [6] – Lemmas 5 and 6) and the use of a Chernoff bound.

CLAIM 6.3. *The probability of node u hearing from a given node v with $d(u, v) \leq 2d$ in a given time slot of the PD or LF-phases is at least $\frac{1}{2}q(1-q)$, when q is sufficiently small.*

PROOF. The proof of this claim is almost identical to that of Lemma 4. We set $\kappa = 1$, since the graded SINR region does not play a role. Furthermore, instead of a given density we use a minimal distance of d between active nodes but this does not influence the proof. □

CLAIM 6.4. *If $\phi(u) \in [d, 2d)$, then $\delta_u = \phi(u)$.*

PROOF. Let v be the closest neighbor of u, in which case $d(u, v) \leq 2d$. By Claim 6.3, the probability that u receives a message from v in any given time slot is at least $\frac{1}{2}q(1-q)$. Letting X be the number of successful (u, v) transmission in the PD-phase, we get that $\mathbb{E}(X) \geq 24 \ln n$. Using the Chernoff bound (2) with $\delta = 1/2$, it follows that $P(X \leq 12 \ln n) \leq n^{-3}$. □

The second claim from the lemma is a bit different. Note that this claim places no restriction on $\phi(u)$. What the claim says is that if u does hear from *some* node $\geq 12 \ln n$ times, it will be able to compute $\phi(u)$ correctly.

In the following definition we capture the case where a node y would receive a message from a node x if the latter was transmitting and the former was not. The definition specifies no behavior for the two nodes, beyond the fact that such a transmission would succeed were it to occur.

DEFINITION 23. *Consider two nodes x and y forming the (potential) link $\ell = (x, y)$. The event $ps(x, y)$ is said to occur in a slot if $SINR(\ell, T) \geq \beta$, where T is the set of nodes transmitting in the slot.*

CLAIM 6.5. *Consider any three points u, x and y such that $d(u, x) \leq d(u, y)$. Then, $\mathbb{P}(ps(x, u)) \geq \mathbb{P}(ps(y, u))$.*

PROOF. Since interference is computed at the receiver, the interference from any point z received at u is identical for (x, u) and (y, u). The signal is at least as strong for (x, u) since x is the closer node. Thus, for any configuration for which (y, u) succeeds, so will (x, u). The claim follows. □

CLAIM 6.6. *Let u be a node and v be a node from which u received at least $12 \ln n$ messages during the PD-phase. Then, during each slot of the PD-phase, $\mathbb{P}(ps(v, u)) \geq \frac{1}{12}$.*

PROOF. Let $\mathbb{P}(ps(v, u)) = \rho$. The probability that v successfully transmits to u in any given slot is $q(1-q) \cdot \rho$, so the expected number over the whole phase is $\mathbb{E}(X) = q(1-q) \cdot \rho \cdot \frac{48}{q(1-q)} \ln n = 48\rho \ln n$. Suppose for contradiction that $\rho < \frac{1}{12}$. Let X be the number of successful (v, u) transmissions in the phase. Noting that success is i.i.d. across slots, we employ the following Chernoff-type bound [20]: $\mathbb{P}(X \geq (1+\delta)\mathbb{E}(X)) < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}(X)}$. Using $\delta = 2$ it gives that $\mathbb{P}(X \geq 12 \ln n) \leq \left(\frac{e^2}{27}\right)^{4 \ln n} \leq n^{-4}$, implying that with high probability the message (u, v) will not succeed the required $12 \ln n$ times. □

Now we can prove the second claim in Lemma 22. If $\delta_u > 0$, there was a v from which u received at least $12 \ln n$ messages in the PD-phase From Claim 6.6, $ps(v, u) \geq \frac{1}{12}$. Let z be the node closest to u (thus $d(z, u) = \phi(u)$). Then by Claim 6.5, $ps(z, u) \geq \frac{1}{12}$. Now the probability that u has never heard from z is at most $\left(1 - \frac{1}{12} q(1-q)\right)^{\frac{48}{q(1-q)}} \ln n \leq e^{-4 \ln n} = n^{-4}$.

Which proves Lemma 22 and leads to the following:

Lemma 24. *After each round, the minimum distance among active nodes increases by a factor of 2.*

Proof. Let d be the minimum distance at the beginning of the round. Consider any two active nodes u, v with $d(u, v) \leq 2d$. Now by the first part of Lemma 22, $\delta_u > 0$ and $\delta_v > 0$ (i.e. both of them actively participate in the link formation phase). By Claim 6.3 there is a constant probability of the link (u, v) forming in each slot pair of the link formation phase. An application of the Chernoff bound over the $\Omega(\log n)$ slot pairs proves the Lemma. □

The proof of Thm. 21 now follows automatically, since each of the g non-empty length classes are exhausted in $O(\log n)$ slots.

To prove the actual runtime and correctness of the algorithm we now need to show that dominators terminate at the correct moment.

Theorem 25. *Every dominator correctly determines within g rounds that there is at most one active node in its cluster and succeeds to subsequently terminate the algorithm for his cluster.*

Proof. We need to prove:

Claim 6.7. *A dominator does not mistakenly decide that there is only one active node in the cluster.*

Proof. If a message was not received in the slot, the dominator decides correctly. If a message was received from x, let y be another active node in the cluster. Since it transmits in the second slot, from the cluster radius it can be verified that the received power at the dominator will be larger than the threshold and thus the dominator will decide correctly to enter the next round. □

Moreover,

Claim 6.8. *Within g rounds, a dominator decides that there is only one active node in the cluster.*

Proof. After g rounds there will be exactly one active node in each cluster. By Lemma 32, in the first slot, each dominatee will hear from this node . Thus, we enter the case of second slot, where no one transmits. The noise level is not enough to break the received power threshold and thus the dominators will correctly decide the existence of only one active node. □

If each dominator broadcasts a message, all nodes in the respective clusters will receive the relevant message (details provided in Lemma 32 in Appendix B). And thus every dominator succeeds to inform their cluster of their decision. A union bound over all nodes completes the proof of the Theorem.

6.2.1 Link selection and power control

We now show how to select a feasible set from the link set constructed above. We need a crucial definition from [6]:

Definition 26. *A set L of links is ψ-**sparse** if, for every closed ball B in the plane,*

$$B \cap L(8 \cdot rad(B)) \leq \psi \ ,$$

where $rad(B)$ is the radius of B, $L(d)$ is the set of links in L of length at least d, and $B \cap Q$ denotes the links in a set Q with at least one endpoint in ball B.

Lemma 27. *For the link set constructed in the previous section, consider the nodes with degree bounded by some (large constant) C. The link set induced by these nodes is $O(C)$-sparse and has expected size $\Omega(n)$.*

Proof. Every time a node forms a link, it has probability $\frac{1}{2}$ of becoming inactive (and thus forming no more links). Thus, the probability of a node having more than C incident links falls exponentially. The statement about the expected size being $\Omega(n)$ follows from this (also see Thm. 8, [6]).

For sparsity, consider any disc B of radius ρ in the plane. Let L be the set of links induced by the node considered in the statement of the Lemma. We claim that at most one node in B has a node incident to a link in $L(8 \cdot \rho) \cap B$, from which C-sparsity follows since the node has degree at most C.

For contradiction, assume that there are two such nodes u and v. Assume without loss of generality that v is active in the slot pair when u forms the first link in $L(8 \cdot \rho) \cap B$ (call this link ℓ). Then $\phi(u) \leq d(u, v) \leq 2\rho < \ell$. But then, by Lemma 22 and the description of the link formation phase, the link ℓ cannot be formed. This is a contradiction. □

It is known that $O(1)$-sparsity implies Eqn. 4 [7]. It would appear that the link selection and power control algorithm of Section 6.1 applies at this point. Specifically, one could invoke Lemma 13 and the algorithm for power assignment from Section 6.1.2 setting $\nu = g$ (instead of $\nu = \log \Delta$), achieving a $O(g)$ running time. However, in this case, the assumption that link ℓ **knows** the L_i it belongs to is missing. In the g-based algorithm, links can form out of order, thus the round i in which it was formed is not indicative of the L_i it belongs to.

To this end, links of length in the range $[\frac{\hat{R}}{n^{10/\alpha}}, \Delta]$ and links of smaller lengths are considered separately. The larger of the two feasible sets extracted from these two sets enter the solution. For the longer set, link $\ell \in [\frac{\Delta}{2^{i-1}}, \frac{\Delta}{2^{i-2}})$ gets the index i, for a total of $O(\log n)$ indices. This is precisely the same type of partition employed in the $O(\log \Delta)$ based algorithm, thus a $O(\log^2 n)$ time algorithm suffices to choose and assign powers to a constant factor feasible set, in the same way described for the $O(\log \Delta)$ based algorithm. This runtime is subsumed by the runtime of other parts of the algorithm.

Let us assume now that all links are shorter than $\frac{\hat{R}}{n^{10/\alpha}}$. For these links the index i can be learned with some extra cost:

Lemma 28. *Assume all links have length shorter than $\frac{\hat{R}}{n^{10/\alpha}}$. All links can learn the L_i ($i \leq g$) to which they belong in $O(g^2 \log n)$ time.*

PROOF. Let us limit ourselves to a single cluster first. Let M be the maximum length of a link in L. We simply aggregate this value up to the root. Aggregating from nodes up to a dominator takes time $O(g \log n)$ (since the connected set formed in time $O(g \log n)$ includes an aggregation tree). This value can be transmitted back from the root to each node in a single broadcast, and links with length in $[M, M/2)$ set $i = 1$. Now the maximum link length smaller than $M/2$ is aggregated in a similar way, and recursively until all links are exhausted. This process takes g rounds of aggregation for a total cost of $O(g^2 \log n)$.

As mentioned, the above ignores the possibility that links from different clusters of widely varying lengths may be assigned the same i. We argue that since inter-cluster separation is much larger than the lengths of the links, this does not matter. We shall demonstrate this using the case of Eqn. 3, the case of others are similar. The equation is as follows:

$$\sum_{\ell' \in S_\ell^+} \frac{\ell'^\alpha}{d_{\ell\ell'}^\alpha} + \frac{\ell'^\alpha}{d_{\ell'\ell}^\alpha} \leq \tau \ .$$

Note that by Claim 6.1, if ℓ and ℓ' are in different clusters, $d_{\ell'\ell} \geq \hat{R}$. On the other hand, ℓ and ℓ' are both upper bounded by $\frac{\hat{R}}{n^{10/\alpha}}$. Thus, $\frac{\ell'^\alpha}{d_{\ell\ell'}^\alpha} + \frac{\ell'^\alpha}{d_{\ell'\ell}^\alpha}$ is bounded by $\frac{2}{n^{10}}$, which even summed over at most n possible links is a minuscule value which does not asymptotically affect the bound. □

With this lemma in hand, the arguments of Lemma 13 and Section 6.1.2 now apply directly, with $\nu = g$, thus achieving the claimed results from Lemma 19 and consequently Thm. 18.

7. REFERENCES

[1] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen. Minimum latency data aggregation in the physical interference model. *Computer Communications*, 35(18):2175–2186, 2012.

[2] E. I. Ásgeirsson, J. T. Foley, H. Gudmundsdottir, A. Gudmundsson, M. M. Halldórsson, S. F. Kristjansson, P. Mitra, H. Ulfarsson, and Y. Vigfusson. A clean-slate design of a low-latency wireless aggregation network. Submitted, Dec 2012.

[3] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *TOSN*, 8(4):34:1–34:33, 2012.

[4] O. Goussevskaia, M. M. Halldórsson, R. Wattenhofer, and E. Welzl. Capacity of Arbitrary Wireless Networks. In *INFOCOM*, pages 1872–1880, 2009.

[5] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Information Theory*, 46(2):388–404, 2000.

[6] M. M. Halldórsson and P. Mitra. Distributed connectivity of wireless networks. In *PODC*, pages 205–214, 2012.

[7] M. M. Halldórsson and P. Mitra. Wireless Connectivity and Capacity. In *SODA*, pages 516–526, 2012.

[8] M. M. Halldórsson and R. Wattenhofer. Wireless Communication is in APX. In *ICALP*, pages 525–536, 2009.

[9] N. Hobbs, Y. Wang, Q.-S. Hua, D. Yu, and F. C. M. Lau. Deterministic distributed data aggregation under the SINR model. In *TAMC*, pages 385–399, 2012.

[10] T. Jurdzinski, D. R. Kowalski, T. Maciejewski, and G. Stachowiak. Distributed broadcasting in wireless networks under the SINR model. *CoRR*, abs/1207.6732, 2012.

[11] T. Jurdzinski and G. Stachowiak. Probabilistic algorithms for the wakeup problem in single-hop radio networks. In *ISAAC*, pages 535–549, 2002.

[12] R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 62:665–671, 1939.

[13] T. Kesselheim. A Constant-Factor Approximation for Wireless Capacity Maximization with Power Control in the SINR Model. In *SODA*, pages 1549–1559, 2011.

[14] T. Kesselheim. *Approximation Algorithms for Spectrum Allocation and Power Control in Wireless Networks.* PhD thesis, RWTH Aachen, 2012.

[15] T. Kesselheim. Approximation algorithms for wireless link scheduling with flexible data rates. In *ESA*, pages 659–670, 2012.

[16] T. Kesselheim and B. Vöcking. Distributed contention resolution in wireless networks. In *DISC*, pages 163–178, 2010.

[17] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and Sensor Networks. In *MOBICOM*, pages 260–274, 2004.

[18] H. Li, Q. S. Hua, C. Wu, and F. C. M. Lau. Minimum-latency aggregation scheduling in wireless sensor networks under physical interference model. In *MSWIM*, pages 360–367, 2010.

[19] R. Maheshwari, S. Jain, and S. R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *SenSys*, pages 141–154, 2008.

[20] M. Mitzenmacher and E. Upfal. *Probability and computing - Randomized algorithms and probabilistic analysis.* Cambridge University Press, 2005.

[21] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *IPSN*, pages 1–10, 2007.

[22] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *INFOCOM*, 2006.

[23] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-based Models. In *HotNets*, 2006.

[24] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology control meets SINR: The scheduling complexity of arbitrary topologies. In *MOBIHOC*, pages 310–321, 2006.

[25] P. Santi, R. Maheshwari, G. Resta, S. Das, and D. Blough. Wireless link scheduling under a graded SINR interference model. In *FOWANC '09*, pages 3–12, 2009.

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

[27] 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 *SIROCCO*, pages 111–122, 2012.

[28] M. Z. Zamalloa and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *TOSN*, 3(2), 2007.

APPENDIX

A. MISSING PROOFS FROM SECTION 5

We restate Lemma 4 as follows:

LEMMA 29. *Let L be an η-dominating set with constant density ϕ. Then if each node in L transmits with (small enough) probability q, the probability of a transmission from node u to node v is successful, given that u and v are neighbors, is $\kappa\frac{1}{2}q(1-q)$.*

PROOF. Let link $\ell = (u, v)$. Note that $d(u, v) \leq \Delta(1 - 1/2\epsilon')$ since u and v are neighbors. Let B_r be the set of nodes transmitting in a given time slot. Note that

$$\mathbb{P}(u \in B_r \text{ and } v \notin B_r) = q(1-q) \ .$$

For $t = 0, 1, \ldots$ define C_t to be the ball around v of radius $\Delta(t+1) + \eta$, and define the annulus A_t as $A_0 = C_0, A_t = C_t \setminus C_{t-1}$ for $t \geq 1$.

By definition of constant density, L contains at most ϕ nodes in every ball with radius η. We cover the annulus A_t with balls of radius η.

The limit of the ratio of the area of A_t to the area of smaller balls is $\frac{3\sqrt{3}}{2\pi}$ [12]. All balls intersecting A_t are completely within the annulus containing the ball around v with radius $\Delta(t+1) + 2\eta$, but not containing the ball around v with radius $\Delta t - 2\eta$. And thus we can write for A_t

$$\frac{\pi(\Delta(t+1) + 2\eta)^2 - \pi(\Delta t - 2\eta)^2}{\chi_t \pi \eta^2} \geq \frac{3\sqrt{3}}{2\pi} \ ,$$

where χ_t is the number of smaller balls needed to cover A_t. This equation has the solution

$$\chi_t \leq \frac{32(1+\epsilon')(1+2t)}{3\sqrt{3}\epsilon'^2\pi} \ .$$

Note that A_t contains at most $\phi \cdot \chi_t$ nodes in L. This means there is a constant $\eta_{\epsilon'}$ depending on ϵ' such that $|A_t \cup L| \leq \eta_{\epsilon'} t$.

Solving a similar equation for A_0 gives $\chi_0 \leq \frac{\Delta^4(4\epsilon' + \epsilon'^2)^2}{384\sqrt{3}\pi}$. We will denote the upper bound on $|A_0 \cup L| \leq \chi_0 \cdot \phi$ with η_0.

We use the notion of *affectance*, introduced in [4, 8] and refined in [16]. The affectance $a_w(\ell)$ on link ℓ from a sender w is the interference of w on v relative to the power received, or

$$a_w(\ell) = \min\left\{1, c_\ell \frac{P_w}{P_u} \cdot \left(\frac{d(u,v)}{d(w,v)}\right)^\alpha\right\} \ ,$$

where $c_\ell = \beta_1/(1 - \beta_1 N d(u,v)^\alpha / P_u)$ depends only on the parameters of the link ℓ. The equation $SINR \geq \beta_1$ can be rewritten as $a_L(\ell) \leq 1$ and thus is ℓ successful with probability κ iff $a_L(\ell) \leq 1$.

We bound the affectance on ℓ by the other nodes in B_r. For $x \in B_r \cap A_0, a_x(\ell) \leq 1$, by the definition of affectance. For $x \in B_r \cap A_t$ for $t \geq 1$, $d(x,v) \geq \Delta \cdot t$ and thus $a_x(\ell) \leq c_\ell \frac{(\Delta(1-1/2\epsilon'))^\alpha}{(\Delta \cdot t)^\alpha)}$. Using $\Delta = \left(\frac{P}{\beta_1 N}\right)^{1/\alpha}$ we get $c_\ell = \frac{\beta_1}{1-(1-1/2\epsilon')^{1/\alpha}}$. Now we rewrite $a_x(\ell) \leq c_\ell \frac{(\Delta(1-1/2\epsilon'))^\alpha}{(\Delta \cdot t)^\alpha)} = \eta_{\epsilon',\Delta}\left(\frac{1}{t}\right)^\alpha$ where $\eta_{\epsilon',\Delta}$ is a constant depending on ϵ' and Δ.

$$\begin{aligned}\mathbb{E}(a_{B_r}(\ell)) &= \mathbb{E}(a_{B_r \cap A_0}(\ell)) + \sum_{t \geq 1} \mathbb{E}(a_{B_r \cap A_t}(\ell)) \\ &\leq q\eta_0 + q\beta_1 \eta_{\epsilon'} \eta_{\epsilon',\Delta} \sum_{t \geq 1} \left(\frac{1}{t}\right)^\alpha \cdot t \\ &\leq q\eta_0 + q\beta_1 \eta_{\epsilon'} \eta_{\epsilon',\Delta} \frac{1}{\alpha - 2} \ ,\end{aligned}$$

using the bound $\zeta(x) = \sum_{n \geq 1} \frac{1}{n^x} \leq \frac{1}{x-1}$ on the Riemann zeta function. Thus, for any $q \leq (4\eta_0 q + q\eta_{\epsilon'}\eta_{\epsilon',\Delta}\frac{1}{\alpha-2})^{-1}$, we get that $\mathbb{E}(a_{B_r}(\ell)) \leq 1/2$. By Markov's inequality, $a_{B_r}(\ell) \leq 1$ with probability at least $1/2$. Thus,

$$\mathbb{P}(a_{B_r}(\ell) \leq 1 \text{ and } u \in B_r \text{ and } v \notin B_r) \geq \frac{1}{2}q(1-q) \ ,$$

and thus the probability of a successful transmission is $\kappa\frac{1}{2}q(1-q)$, which concludes the proof. □

Similarly, we restate Lemma 5.

Lemma 5: *Let every node transmit with sufficiently low probability q during $O(\log n)$ slots. Then each node succeeds in transmitting to each of its neighbors $\Omega(\log n)$ times, with high probability.*

PROOF. By Lemma 4 the probability that a node u succeeds in transmitting to a fixed neighbor v is $\frac{1}{2}\kappa \cdot q(1-q)$.

Now let X be the number of successful transmissions in $(1/2)\kappa \cdot q(1-q))^{-1} C \ln n$ slots, where C is a constant. Clearly $\mathbb{E}(X) = C \ln n$. Set $\delta = 1 - 1/C$, then $(1-\delta)\mathbb{E}(X) = \ln n$.

The Chernoff bound states here

$$\mathbb{P}(X \leq \ln n) \leq n^{-C(1-1/C)^2/2} \ ,$$

which means that the number of successful transmissions is $O(\log n)$ with high probability. □

B. MISSING PROOFS FROM SECTION 6

Coloring

LEMMA 30. *Given an η-dominating set with constant density ϕ, the dominators can color themselves using a constant number of colors in $O(\log^2 n)$ slots so that the corresponding clusters having dominators with the same color satisfy Claim 6.1.*

PROOF. This proof uses the definition of neighbor and NCOMM from Section 5. Recall that for a dominator, a neighbor is a dominator within distance $\Delta(1 - \epsilon'/2)$. The primitive NCOMM allows dominators to contact each neighboring dominator within $O(\log n)$ slots.

Since the dominators form an η-dominating set we need the minimum distance between two dominators to be at least $\Upsilon \cdot \epsilon' \cdot \Delta + 2 \cdot \eta$ to satisfy the claim.

Since the density is constant, the number of other dominators within a distance of $(\Upsilon + 1/2)\epsilon' \cdot \Delta$ of a dominator is a (larger) constant γ. We assume ϵ' is small enough such that $(\Upsilon + 1/2)\epsilon' < 1 - \epsilon'$ (otherwise, nodes can choose ϵ' small enough to ensure this, as Υ is independent of ϵ'). Nodes within this distance of each other are clearly neighbors.

Nodes start out with the same color. Nodes can have one of two states – finished or violator. All nodes start out as violator. The algorithm contains $O(\log n)$ rounds, each with $O(\log n)$ slots. In the first slot of the round, violators choose a color from $\{1 \dots 10\gamma\}$. Then an NCOMM is used for all nodes to be informed about their neighbors colors, which takes $O(\log n)$ slots. Nodes that are not finished and have the same color as a neighbor remain violators. The probability of a violator to become finished is at least $(1 - \frac{1}{10\gamma})^{10\gamma-1}$, which is lower bounded by a constant. Thus in $O(\log n)$ rounds, all nodes will become finished w.h.p. It is clear that a valid coloring of the finished nodes exists. □

Proof of Lemma 14

In Thm. 3 in [13] it is proved that a set with property described in Eqn. 3 is feasible. The remainder of the proof is analogous to Thm. 1 of [13].

We use the signal strengthening technique from [8] to decompose S_ℓ^- into $\lceil 2 \cdot 3^\alpha/\beta \rceil^2$ sets, which makes the set feasible with $\beta' = 3^\alpha$ for every transmission. Since there is a constant number of sets we prove the lemma for one set.

Consider links ℓ' and ℓ'' with $d_{\ell'\ell''} \le d_{\ell''\ell'}$ in such a set $S_\ell^{-'}$. We know the SINR condition is fulfilled with $\beta' = 3^\alpha$ and thus for for some arbitrary power assignment we have

$$\frac{p(\ell')}{\ell'^\alpha} \ge 3^\alpha \frac{p(\ell'')}{d_{\ell''\ell'}} \text{ and } \frac{p(\ell'')}{\ell''^\alpha} \ge 3^\alpha \frac{p(\ell')}{d_{\ell'\ell'}} ,$$

and by multiplying

$$d_{\ell''\ell'} \cdot d_{\ell'\ell''} \ge 9 \cdot \ell' \cdot \ell'' . \tag{10}$$

By the triangle inequality we get

$$d_{\ell'\ell''} \cdot d_{\ell''\ell'} \le dne\ell'\ell'' \cdot (\ell'' + d_{\ell'\ell''} + \ell') . \tag{11}$$

Note that $d_{\ell'\ell''} \ge \min\{2\ell', 2\ell''\}$. By contradiction, if $d_{\ell'\ell''} < 2\ell'$ and $d_{\ell'\ell''} < 2\ell''$ then by Eqn. 11, we get $d_{\ell'\ell''} \cdot d_{\ell''\ell'} < 2\min(\ell', \ell'') \cdot 4\max(\ell', \ell'') = 8 \cdot \ell' \cdot \ell''$ which contradicts Eqn. 10. Remember that $\ell', \ell'' \ge \frac{1}{2}\ell$ and thus $d_{\ell''\ell'} \ge d_{\ell'\ell''} \ge \ell$ and thus the distance between any of the nodes in $\{s, s', r, r'\}$ is at least $\frac{1}{2}\ell$. To simplify notation we use $d = \frac{1}{2}\ell$.

For the remainder of the proof we use the same technique as in the proof for Lemma 4. For $t = 0, 1, \dots$ define C_t to be the ball around node v (the sender of ℓ) of radius $d(t+1)$, and define the annulus A_t as $A_0 = C_0, A_t = C_t \setminus C_{t-1}$ for $t \ge 1$. We compute

$$\text{Area}(A_t) = \pi d^2(2t+1) . \tag{12}$$

Since the distance between any two points in $S_\ell^{-'}$ is at least d, balls of radius $\frac{d}{4}$ around any pair of points in $S_\ell^{-'}$ do not intersect. Combining this with Eqn. 12, we see that for $t \ge 1$, A_t contains at most $16(2t+1) \le 33t$ nodes in $S_\ell^{-'}$ (note that $|A_0| \le 16$).

Since L is feasible we know for any link $\ell' \in L$ that $\frac{\ell^\alpha}{d_{\ell'\ell}} = O(1)$ and thus that there is a constant c such that $\frac{\ell^\alpha}{d_{\ell'\ell}} \le c$ for any ℓ' in $|A_0|$.

$$\begin{aligned} \sum_{\ell' \in S_\ell^-} \min\left\{1, \frac{\ell^\alpha}{d_{\ell'\ell}^\alpha}\right\} &\le c \cdot |A_0| + \sum_{t \ge 1} |A_t| \cdot \left(\frac{(2d)}{d(t+1)}\right)^\alpha \\ &\le c \cdot |A_0| + \sum_{t \ge 2} 33t \left(\frac{2}{t}\right)^\alpha \\ &\le c \cdot 16 + 2^\alpha \frac{1}{\alpha - 2} 33 \\ &= O(1) . \end{aligned}$$

□

Proof of Lemma 15

The algorithm proceeds in rounds. The solution set S starts out empty, with links in L_i adding themselves to the solution simultaneously in round i, explaining the runtime.

In round i, nodes currently in S transmit with power P. Any node ℓ in L_i transmits with power P with small constant probability $\frac{1}{16\beta C}$, and adds itself to the solution set if it succeeds with $SINR \ge 4\beta$. Thus,

$$\frac{P/\ell^\alpha}{\sum_{\ell' \in S_{\ell^-}} (P/d_{\ell'\ell}^\alpha) + N} \ge 4\beta ,$$

which implies

$$\sum_{\ell' \in S_\ell^-} \frac{\ell^\alpha}{d_{\ell\ell'}^\alpha} \le \frac{1}{4\beta} - \frac{N}{P\ell^\alpha} ,$$

from which bound claimed in the Lemma follows.

Let L_i' be the links sending in phase i and let L_i^* be the links that succeeded. We claim that using a sufficient power assignment at least a constant fraction of L_i gets selected. For every link ℓ in L_i' the expected value for the sum is

$$\mathbb{E}\left(\sum_{\ell' \in S \cup L_i'} \frac{\ell^\alpha}{d_{\ell\ell'}^\alpha}\right) \le \frac{1}{16\beta} .$$

Using Markov's inequality we argue that for every i at least half the nodes in L_i will have

$$\mathbb{E}\left(\sum_{\ell' \in S \cup L_i'} \frac{\ell^\alpha}{d_{\ell\ell'}^\alpha}\right) \le \frac{1}{8\beta} .$$

Assuming $P \ge \frac{8\beta}{N\ell^\alpha}$ these links will have $SINR \ge 4\beta$ and consequently will add themselves to the set. From which the lemma follows. □

Cluster broadcast

LEMMA 31. *Assume a set of well-separated clusters. Fix a cluster and a node x in it, and assume that exactly one node is transmitting from each of the other clusters. Then the received power at x from the transmitting nodes in other clusters is $\le \frac{P}{(2\beta\epsilon'\Delta)^\alpha}$.*

PROOF. For $t = 1, 2 \dots$ define $A_t = C_{t+1} \setminus C_t$, where C_t is the $\hat{R}t/2$-ball around x. It is clear that all transmitting nodes in other clusters must belong to an A_t. Furthermore, it is easily computed that the area of A_t is

$$\text{Area}(A_t) = \pi\hat{R}^2/4(2t+1) . \tag{13}$$

Since the clusters are well-separated, $\hat{R}/2$-balls around nodes in different clusters do not intersect. Combining this with Eqn. 13 and the assumption that exactly one node is transmitting in each cluster, we see that A_t contains at most $2t + 1 \geq 3t$ transmitting nodes, for all $t \geq 1$.

The power received at x from these nodes is then at most

$$\sum_{t \geq 1} \frac{3tP}{(\hat{R}t/2)^\alpha} = \frac{3P}{(\hat{R}/2)^\alpha} \sum_{t \geq 1} \frac{1}{t^{\alpha-1}} = \frac{3P}{(\hat{R}/2)^\alpha} \frac{1}{\alpha - 2} \leq \frac{P}{\beta(\epsilon'\Delta)^\alpha}$$

Here the first equality is by rearrangement, the second is by bounding the sum by using the bound $\zeta(x) = \sum_{n \geq 1} \frac{1}{n^x} \leq \frac{1}{x-1}$ on the Riemann zeta function, and the last inequality is by assuming $\Upsilon \geq 2 \left(\frac{3\beta}{\alpha-2}\right)^{1/\alpha}$. ☐

Lemma 31 implies the following:

LEMMA 32. *Assume at most one node transmits in each cluster. Then all nodes in a cluster that has a transmitting node will receive the message.*

PROOF. Consider any node x in a cluster where a node y in transmitting. The signal received at x from y is at least $\frac{2P}{(\epsilon'\Delta)^\alpha}$. By Lemma 31 the interference at x from other transmitting nodes is at most $\frac{P}{\beta(\epsilon'\Delta)^\alpha}$. Plugging in the value of noise, $N = \frac{P}{\beta_1 \Delta^\alpha}$, gives

$$SINR = \frac{\frac{2P}{(\epsilon'\Delta)^\alpha}}{\frac{P}{\beta_1 \Delta^\alpha} + \frac{P}{\beta(\epsilon'\Delta)^\alpha}} = \frac{2}{\left(\frac{1}{\beta_1} + \frac{1}{\beta\epsilon'^\alpha}\right)\epsilon'^\alpha} \geq \frac{2}{2\frac{1}{\beta\epsilon'^\alpha}\epsilon'^\alpha} = \beta\,,$$

Here the equalities are by rearrangement and the inequality is by using $\epsilon' \leq \left(\frac{\beta_1}{\beta}\right)^{1/\alpha}$. ☐