Leveraging Multiple Channels in Ad Hoc Networks[*]

Magnús M. Halldórsson[†]   Yuexuan Wang[‡§]   Dongxiao Yu[¶]

Abstract

We examine the utility of multiple channels of communication in wireless networks under the SINR model of interference. The central question is whether the use of multiple channels can result in linear speedup, up to some fundamental limit. We answer this question affirmatively for the data aggregation problem, perhaps the most fundamental problem in sensor networks. To achieve this, we form a hierarchical structure of independent interest, and illustrate its versatility by obtaining a new algorithm with linear speedup for the node coloring problem.

1   Introduction

Diversity in wireless networks – having multiple opportunities for communication – is well known to decrease interference, increase reliability, and improve performance [5, 9]. The question is how much it helps and what the limits are to such improvements. In particular, we seek an answer to the following fundamental question in the context of the SINR model:

> *Can we speed distributed wireless algorithms up linearly with the number of channels, up to a fundamental limit?*

Thus, we are interested in the fundamental limits of the benefits of diversity.

We focus our attention on data dissemination problems, in particular *data aggregation*, sometimes referred to as the "killer-app" for sensor networks: compute a compressible function (e.g., average) of values stored at the nodes [23].

Multiple channels can be available by modulation ranging over frequencies or phases. They can also be simulated by time-division multiplexing (TDMA) by assigning time slots to the different channels. The converse does not hold, however, as multiple channels are a strictly more constrained form of communication. Namely, whereas nodes can listen (and even choose to send) in all slots of a TDMA schedule, they can only listen on one of the channels. Thus, multiple channels can be viewed as a form of *parallelism* in wireless communication and our inquiry involves the parallelizability of fundamental wireless tasks.

Multiple channels have been found to yield linear speedups in graph-based models, such as for broadcast [9], minimum dominating sets [7], leader election [5] and maximal independent sets [4]. In contrast,

[†]ICE-TCS, School of Computer Science, Reykjavik University, Iceland; mmh@ru.is.

[‡]College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, P.R. China.

[§]Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China; amywang@hku.hk.

[¶]Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, P.R. China.; dxyu@hust.edu.cn.

1

essentially the only work on multiple channels in the *signal-to-interference-and-noise ratio* (SINR) model is [37], which attained a sub-linear speedup for local information exchange, but holds only for a restricted number of channels when each message can carry multiple packets. Thus, little has been known about the limits for leveraging multiple channels in an SINR context.

Model. We assume synchronized operation with time measured in *rounds*. Nodes have no power control, no collision detection, but have a carrier sense mechanism in the form of standard signal strength measurements. The SINR model of interference is assumed, but the parameters (α, β, N) are allowed to vary within fixed ranges. We assume for simplicity of exposition that nodes are located in the plane, but the results extend to more general metric spaces known as fading metrics.[1] Nodes are given approximate values of SINR parameters and a polynomial bound on the number of nodes, but have no knowledge of the location of other nodes or their distribution.

Our Results. Let $G = (V, E)$ be the communication graph obtained by connecting pairs of nodes that can potentially communicate with each other directly (please refer to Sec. 2 for detailed definition). Let D be the diameter of G, Δ be its maximum degree, $\mathcal{F}$ be the number of channels, and n the number of nodes (see Sec. 2 for definitions). We say that an event happens *with high probability* (with respect to n), if it happens with probability $1 - 1/n^c$ for some constant $c > 0$.

We give a randomized algorithm that achieves data aggregation in $O(D + \Delta/\mathcal{F} + \log n \log \log n)$ time with high probability. Since Δ is a lower bound for aggregation in single-channel networks, even ones with few hops, we achieve linear speedup up to the additive $\log n \log \log n$ term. This is essentially best possible for a setting where high probability guarantees are required.

Our data aggregation algorithm is based on a data aggregation structure that can be constructed in $O(\log^2 n)$ time. If a $\log^{O(1)} n$-approximation of Δ is known, the time for constructing the aggregation structure is $O(\Delta/\mathcal{F} + \log n \cdot \log \log n)$. Hence, in this case, the total time for accomplishing data aggregation (taking into account the time for structure construction) is $O(D + \Delta/\mathcal{F} + \log n \log \log n)$ with high probability.

The aggregation structure is of independent interest, as it can be used to solve other core problems. To illustrate its applicability, we give an algorithm for the node coloring problem that runs in $O(\Delta/\mathcal{F} + \log n \log \log n)$ time with high probability.

Lower Bounds. We indicate here briefly why our bounds are close to best possible. Any global task involving communication requires at least D steps, which yields a lower bound on every instance. Similarly, $\lceil \log n \rceil$ is a lower bound for data aggregation, since at most half the items can be coalesced in a single round. Thus, independent of the parallelization in the form of multiple channels, $\Omega(D + \log n)$ steps are needed.

In a single channel, the term Δ is necessary for any communication task that involves all nodes when using fixed power assignment such as uniform power. In particular, consider the "exponential chain", where point i is located at position 2^i on the real line, $i = 1, 2, \ldots, n$. Then, when using uniform power, at most one successful transmission can occur in a time slot (assuming $\beta \geq 2^{1/\alpha}$) [25]. In particular, aggregation and coloring require Δ steps in single-channel networks, and clearly $\mathcal{F}$ channels can reduce the time requirement at most to $\Delta/\mathcal{F}$. While no proof is known, it is unlikely that power control reduces this bound in the distributed SINR setting; known distributed algorithms all feature time complexity of either the distance diversity (which can be as large as n) [2, 14, 27] or terms linear in Δ [13, 15, 22, 33].

Related Work.

---

[1] A metric space is said to be *fading* if the path loss exponent α is strictly greater than the doubling dimension of the metric. This is a generalization of the standard requirement of $\alpha > 2$ in the two-dimensional Euclidean space, as the two-dimensional Euclidean space has a doubling dimension of 2. For more details on fading metric, see [12].

Data Aggregation. In single-channel networks, there is a long line of research on data aggregation under different settings in the protocol model [30, 31, 32] and the SINR model [1, 2, 10, 14, 16, 17, 23, 24]. Regarding distributed solutions in the SINR model, a distributed aggregation algorithm with uniform power assignment was proposed in [24], which achieves a latency upper bound of $O(D + \Delta)$. Assuming a model where every node in the network knows its position, the network diameter and the number of neighbors, Li et al. [23] presented a distributed algorithm with a latency bound of $O(K)$, where K is the logarithm of the ratio between the length of the longest link and that of the shortest link. This result additionally needs that nodes can adjust the transmission power arbitrarily. In [17], Hobbs et al. gave a deterministic algorithm which can accomplish data aggregation in $O(D + \Delta \log n)$ rounds. An entirely different approach is to use (significant) precomputation to build a fast aggregation structure. In particular, aggregation can be achieved in optimal $O(D + \log n)$ time [2, 14], but this uses $O(K \log^2 n)$ time for precomputation and also relies heavily on arbitrary power control.

In multi-channel networks, the multiple-message broadcast algorithm given in [4] can be adapted to solve the data aggregation problem in a graph-based interference model in $O(D + \Delta + \frac{\log^2 n}{\mathcal{F}} + \log n \log \log n)$ rounds with high probability, but it restricts the number of channels to at most $\log n$. An algorithm for the related broadcast problem was given in [9] for a similar setting but also allowing disruptions on channels. The work closest to ours is a recent treatment of the local information exchange problem in multi-channel SINR networks [37], where Yu et al. gave a distributed algorithm attaining a sub-linear speedup. In the algorithm, the number of channels that can be used effectively is limited to $O(\sqrt{\Delta / \log n})$, resulting in an $\Omega(\log n \cdot \sqrt{\Delta \log n})$ lower bound on the performance of the algorithm.

Coloring. The distributed node coloring problem has been extensively studied since the 1980s as a classical symmetry breaking paradigm [3]. Most work has been in message passing models that ignore interference and collisions. Assuming a graph-based model that defines only direct interference from neighbors, Moscibroda and Wattenhofer [26] gave an $O(\Delta \log n)$ time randomized algorithm using $O(\Delta)$ colors for bounded-independence graphs, which was lated improved to a $\Delta + 1$-coloring in $O(\Delta + \log \Delta \log n)$ time by Schneider and Wattenhofer [29]. Derbel and Talbi [8] showed that the algorithm of [26] can also work in the SINR model with the same time and color bounds. Yu et al. [36] gave a randomized algorithm with running time $O(\Delta \log n + \log^2 n)$ that achieves a $\Delta + 1$-coloring in the SINR model. All of the above results are for wireless networks with a single channel, and it appears no work has previously addressed the coloring problem in multiple channel networks, let alone in the SINR model.

Backbone Network Construction. Another line of related work is finding dominating sets and/or a broadcast/ aggregation network in a multi-hop scenario. The work we directly use is that of [28] with an algorithm that finds a dominating set in the SINR model in $O(\log n)$ time. An algorithm was given in [26] that finds a maximal independent set running in $O(\log^2 n)$ time in the quasi unit disk model, later converted to the SINR model in [36]. Broadcast or aggregation networks among dominators are formed in some works such as [2, 14, 18, 20, 21, 34, 35]. These works either highly rely on strong assumptions on the connectivity of the network [34, 35], use precise location information [18, 20, 21], or arbitrary power adjustment [2, 14]. All these works are only for single-channel networks.

Roadmap. The formal model, problem definitions and preliminaries are given in Sec. 2. Sec. 3 contains a technical overview. In Sec. 4, an algorithm to find ruling sets is introduced, which is invoked frequently in the structure construction. The algorithm for constructing the aggregation structure is given in Sec. 5 and the data aggregation algorithm in Sec. 6. Sec. 7 contains the coloring algorithm making use of the aggregation structure.

2 Model, Problem Formulations and Preliminaries

The network consists of a set V of n nodes with unique IDs that are positioned arbitrarily on a plane. We focus on the setting of a *uniform* power assignment, where all nodes use the same transmission power P. For two nodes u and v, denote by $d(u,v)$ the Euclidean distance between u and v.

Multiple Communication Channels and Synchronization. Nodes communicate through a shared medium divided into $\mathcal{F}$ non-overlapping channels. Time is divided into synchronized rounds, where each round may contain a constant number of synchronized slots. All nodes start the algorithm at the same time. In each slot of every round, each node can select one of the $\mathcal{F}$ channels and either transmit or listen on that channel. A node that operates on a channel in a given slot learns nothing about events on other channels.

Interference and SINR model. Simultaneous transmissions on the same channel interfere with each other. The SINR model captures the interference by stipulating that a message sent by node u to node v can be correctly received at v iff (i) u and v operate on the same channel and v does not transmit, and (ii) the following signal-to-interference-and-noise-ratio (SINR) is above a hardware-defined threshold $\beta \geq 1$:

$$SINR(u,v) := \frac{P/d(u,v)^\alpha}{N + \sum_{w \in S \setminus \{u\}} \frac{P}{d(w,v)^\alpha}} \geq \beta \, , \tag{1}$$

where $\alpha > 2$ is the "path-loss" constant, N is the ambient noise, and S is the set of nodes transmitting simultaneously with u.

The *transmission range* R_T is the maximum distance at which a transmission can be successfully decoded (in the absence of other transmissions); by the SINR condition (1), $R_T = (\frac{P}{\beta \cdot N})^{1/\alpha}$.

We assume that listening nodes can measure the SINR (only in the case of a successful reception), and the total received power. Nodes can also use this feature to infer (approximate) distances from the sender of a received message. This power reception feature is comparable to the RSSI function of actual wireless motes [28]. In our algorithm, indeed, it is enough to determine whether the SINR (of a successful reception) or the total received power crosses a single fixed threshold.

It is always of theoretical interest to determine the tradeoffs between different model assumptions, and to identify the least set of primitives that suffice for efficient execution. We posit, however, that the default model for wireless algorithms in physical models should feature receiver-side carrier sense ability. Given that such a feature is so standard in even the cheapest hardware and so easily implementable, it would be counterproductive to exclude it. Note that we assume no transmitter-side detection ability.

Communication Graph and Notations. For parameter c, $0 < c < 1$, denote $R_c := (1-c)R_T$. The *communication graph* $G(V,E)$ of a given network consists of all network nodes and edges (v,u) such that $d(v,u) \leq R_\epsilon$, where $0 < \epsilon < 1$ is a fixed model parameter. Since nodes of distance very close to R_T can only communicate in the absence of other activity in the network arbitrarily far away, we adopt the standard assumption that a slightly smaller range, R_ϵ, is sufficient to communicate [2, 6, 19].

We use standard graph terminology: $N(u)$ is the set of neighbors of node u; $d_u = |N(u)|$ is the degree of u; and Δ is the maximum degree of a node. The *diameter* D of a graph G is the maximum, over all pairs of nodes u, v, of the shortest hop-distance between u and v.

An *r-ball* is a disk in the plane of radius r. Denote by E_v^r the r-ball centered at node v, and overload the notation to refer also to the set of nodes in the ball. A node u is an *r-neighbor* of (not necessarily distinct) node v if $d(u,v) \leq r$. An *r-dominating set* is a subset S of nodes (called *dominators*) such that each node in V has an r-neighbor in S. The *density* of an r-dominating set is the maximum number of dominators in an r-ball (over all balls in the plane). A set S of nodes is *r-independent* if no two nodes in S are r-neighbors. An r-independent set S is *maximal* if it is also r-dominating.

Knowledge of Nodes. Nodes know a polynomial approximation to n (i.e., the value of $\log n$, up to constant factors). For simplicity of description, we also use n to denote this estimate. We assume that nodes do not know the precise value of the SINR parameters α, β and N but instead know only upper and lower bounds for the parameters (i.e., α_{min} and α_{max}, β_{min} and β_{max}, N_{min} and N_{max}). For simplicity, we perform calculations assuming that exact values of these parameters are known; to deal with uncertainty regarding those parameters, it suffices to choose their maximal/minimal values depending on whether upper or lower estimates are needed. Nodes have no other information, such as the network topology, their neighbors or their location coordinates.

Data Aggregation. Initially, each node has a data value. The *data aggregation* problem is to compute an aggregate function (e.g., maximum or average) on the input data from all nodes in the network, and inform all nodes of this value as quickly as possible.

Preliminaries. The following Chernoff bounds will be used in the analyses of algorithms. The proofs of these bounds can be found in most textbooks on probability theory or randomized algorithms.

Lemma 1 (Chernoff bounds). *Let $X_1, X_2, \ldots, X_n$ be independent Bernoulli random variables. Let $X := \sum_{i=1}^{n} X_i$ and $\mu := \mathbb{E}[X]$. Then, for any $\delta > 0$, it holds that*

$$Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu .$$

More precisely,
$$Pr[X \geq 2\mu] \leq (e/4)^\mu \leq e^{-\mu/3}. \tag{2}$$

On the other hand,
$$Pr[X \leq \frac{1}{2}\mu] \leq \left(\frac{e^{-1/2}}{(1/2)^{1/2}} \right)^\mu = (e/2)^{-\mu/2} \leq e^{-\mu/8} . \tag{3}$$

We use a frequently-used argument that shows that well-separated communication can proceed independently. The proof of this lemma uses the standard technique of bounding interference within concentric circles.

Lemma 2. *Let r_1, r_2 be distance parameters such that $r_2 \leq \min\{\left(\frac{\alpha-2}{48\beta(\alpha-1)} \right)^{\frac{1}{\alpha}} \cdot r_1, R_T/2\}$. Suppose the set S_F of nodes transmitting on a channel F is r_1-independent. Then, the transmission of each node $v \in S_F$ is received by all r_2-neighbors of v that are listening on F.*

Proof. By assumption, the set S_F satisfies $d(u,v) > r_1$, for any pair of nodes $u, v \in S_F$. For a node $w \in S_F$, we compute the interference experienced by a node $x \in Q_F \cap E_w^{r_2}$, where Q_F is the set of nodes selecting to operate on a channel F. Let C_t be the annulus with distance from w in the range $[tr_1, (t+1)r_1)$ for $t \geq 1$. Without confusion, C_t is also used to denote the set of nodes in C_t that operate on F. Because any two transmitting nodes are separated by r_1, an area argument implies that $|C_t| \leq 8(2t+1)$. Then we

5

bound the interference at a node $x \in E_w^{r2}$ caused by other transmitters in S_F as follows.

$$
\begin{aligned}
I_x = \sum_{y \in S_F \setminus \{w\}} \frac{P}{d_{yx}^\alpha} &\leq \sum_{t=1}^\infty \frac{N\beta R_T^\alpha}{(tr_1)^\alpha} \cdot 8(2t+1) \\
&\leq 24 r_1^{-\alpha} N\beta R_T^\alpha \sum_{t=1}^\infty t^{-\alpha+1} \\
&\leq 24 r_1^{-\alpha} N\beta R_T^\alpha \cdot \frac{\alpha-1}{\alpha-2} \\
&\leq (\frac{R_T^\alpha}{r_2^\alpha} - 1)N.
\end{aligned}
$$

Then by the SINR condition, x can receive the message sent by u. $\qquad\square$

Given that each node u transmits with a probability p_u, let $P_r(v) = \sum_{u \in E_v^r \cap Q_F} p_u$ be the sum of transmission probabilities of nodes in E_v^r that operate on channel F. Using a similar argument as in proving Lemma 2 and further considering the transmission probabilities of nodes, we can get the following result.

Lemma 3. *Let $R \in \Omega(R_T)$ be a distance, F be a channel and Q_F the set of nodes operating on the channel. Suppose that each node u transmits on F with probability p_u, satisfying $P_R(v) := \sum_{u \in E_v^R \cap Q_F} p_u \leq \psi$. Then, whenever a node v transmits on F, with constant probability $\kappa := e^{-O((R_T/R)^2 \cdot \psi)} \in \Omega(1)$, it is heard by all its R-neighbors $E_v^R \cap Q_F$ on the channel.*

Lemma 3 has been implicitly proved in previous work, such as [11] (Lemma 4.1 and Lemma 4.2). The basic idea of proving Lemma 3 is bounding the interference at the neighbors of a transmitter v from other nearby transmitters (within a specified distance that is a constant times R) and faraway transmitters respectively. For the interference from nearby transmitters, based on the facts that these transmitters can be covered by a constant number of R-balls and the sum of transmission probabilities of transmitters in each R-ball is upper bounded by a constant (as given in the condition), it is easy to show that there are no nearby transmitters with a certain constant probability. For the interference from faraway nodes, it suffices to compute the expected interference at the neighbors of a transmitter v based on the transmission probabilities of nodes, instead of computing the real interference in Eq. (6). Because the sum of transmission probabilities of each node's neighbors is bounded by a constant (as given in the condition), which means that there are a constant number of transmitters in expectation within the neighborhood of each node, the expected interference at the neighbors of the transmitter v can be bounded by a small constant using the same concentric argument as in Eqt. (6). Then by Markov Inequality, it can shown that with constant probability (determined by R_T/R and ψ), the interference at every neighbor of the transmitter v is still upper bounded by a small constant, which is enough to ensure successful receptions. Combining the results for interference bounding from nearby nodes and faraway nodes, Lemma 3 can be proved. For more details, please refer to [11].

In this work, we use Lemma 3 for only two different distances. So there is a constant lower bound for the probability of successful transmissions. In the subsequence, we still use κ to denote this lower bound.

3 Technical Overview

Our approach is to build a multi-purpose dissemination structure that we then use in each of our problems. The structure has global and local parts, which are linked through the *dominators*, the local leaders that

collaborate to carry out the global task.

After finding a low-density set of dominators, the other nodes are partitioned into local clusters, each headed by a dominator. These clusters are then colored to disperse the clusters of same color, effectively eliminating interference from other clusters. The clusters are arranged into a communication tree to carry out the global task. These constructions are by now all fairly well known, so we build on previous work, in particular using the $O(\log n)$-round clustering process from [28].

Our main contributions are in the treatment of the intra-cluster aspects. We first estimate the size of each cluster, in order to adjust the contention. We distribute the cluster nodes randomly into channels, and run leader election processes to elect a *reporter* in each channel in $O(\log n)$ rounds. We then form a binary tree of $O(\log \mathcal{F})$ levels on the reporters, which is used to aggregate the data to the dominator. The total time needed for reporter election and reporter tree construction is $O(\log^2 n)$, while the aggregation cost in the clusters is $O(\Delta/\mathcal{F} + \log n \log \log n)$. If a $\log^{O(1)} n$-approximation of Δ is known, the reporter election and the reporter tree construction can be done in $O(\Delta/\mathcal{F} + \log n \log \log n)$ time as well.

4 Ruling Set Algorithm

We present an algorithm that will be invoked frequently in subsequent sections. A (r, s)-*ruling set* is a subset S of nodes that is both r-independent and s-dominating. The algorithm presented finds a $(r, 2r)$-ruling set, where r satisfies $r \leq R_T/2 = \frac{1}{2}(P/(N\beta))^{1/\alpha}$.

The algorithm has two phases. In the first phase, a constant density r-dominating set X is found in $O(\log n)$ rounds using the algorithm of Scheideler et al. [28]. Let μ denote an upper bound on the density guaranteed by their algorithm. In the remainder of this section, we focus on the second phase, computing a maximal r-independent set S among the dominators. Namely, S is r-independent and each node in X is within distance r from a node in S. Then, by the triangular inequality, S forms a $2r$-dominating set of the full set V of nodes.

The strength of signals and interference can yield precious indications about the origin of the signal, and even of interferers.

Definition 4. *A* clear reception *occurs at a node, for a parameter r, if: a) the message originates from an r-neighbor of the node, and b) the interference sensed is at most $T_s = N \cdot \min\{\frac{2^\alpha - 1}{2^\alpha}, (\frac{1}{2})^\alpha \cdot \beta\}$. The latter condition ensures that no other $4r$-neighbor transmitted.*

Based on our model assumptions, a node can detect clear receptions.

The second phase of the algorithm uses three kinds of messages: HELLO, ACK, and IN. Let $\gamma = 3/(\kappa/2\mu)^2 = 12\mu^2/\kappa^2$, where κ is the constant of Lemma 3. The phase consists of $\gamma \ln n$ rounds, each consisting of three slots:

- **Slot 1.** Each node transmits HELLO independently with probability $1/(2\mu)$.

- **Slot 2.** If a node gets a clear reception of HELLO, it sends ACK independently with probability $1/(2\mu)$.

- **Slot 3.** If a node sent HELLO and received ACK from an r-neighbor, it then joins the set S, transmits IN and halts. Otherwise, the node listens; if it receives IN from an r-neighbor, it halts.

If a node is still active after all $\gamma \ln n$ rounds, it then enters the set S. This completes the specification of the second phase, and thus the algorithm.

We first argue the correctness of the last step, when dominated nodes bow out.

Lemma 5. *If a node joins S in a round, then all of its (still active) r-neighbors halt after that round.*

Proof. Let Y be the set of nodes that joined S during the given round, and let u be a node in Y. We claim that all nodes in E_u^r receive IN message from u. Let w be a node in E_u^r and observe that the strength of the signal from u received on w is at least P/r^α. Thus, it suffices to show the total interference $I_{Y_u}(w)$ from $Y_u = Y \setminus u$ received by w is at most $\frac{1}{\beta}P/r^\alpha - N$.

Let v be the r-neighbor of u that sent it ACK and y be a node in Y_u. Since v had a clear reception, $d(v, y) \geq 4r$, while $d(v, w) \leq 2r$, since they are both r-neighbors of u. Thus, $d(y, w) \geq d(y, v) - d(v, w) \geq \frac{1}{2}d(y, v)$. Also, the interference $I_{Y_u}(v)$ received by v is then at most $T_s \leq \frac{2^\alpha - 1}{2^\alpha}N$. Hence,

$$I_{Y_u}(w) = \sum_{y \in Y_u} \frac{P}{d(y, w)^\alpha} \leq 2^\alpha I_{Y_u}(v) \leq 2^\alpha T_s \leq (2^\alpha - 1)N \leq \frac{1}{\beta}P/r^\alpha - N \ ,$$

as desired. $\qquad\square$

The main correctness issue is to ensure independence. While the above lemma handles nodes that enter the ruling set during the main rounds, we use a probabilistic argument to argue that neighbors are unlikely to survive all the rounds to be able to enter the set S at the end of the execution.

Lemma 6. *The algorithm correctly computes a $(r, 2r)$-ruling set S in $O(\log n)$ rounds, with high probability.*

Proof. By definition of the algorithm, the nodes halt by either joining S or after receiving IN from a neighbor. Thus, the solution is an r-dominating set of X, and hence a $2r$-dominating set of V. It remains to show that S is r-independent.

Let u, v be nodes in S, and suppose without loss of generality that u was added no later than v. If both joined S during the same round then they must be of distance at least $3r$ apart (since an r-neighbor of u experienced a clear reception). If v joined S later, it must be more than r away from u, since u notified all its r-neighbors with an IN message, by Lemma 5. Finally, we show that, with high probability, no r-neighbors remain active after all the $\gamma \ln n$ rounds.

Let u and v be r-neighbors. Observe that the sum of transmission probabilities of nodes in any r-ball E_w^r is at most $1/2$ (as the density is at most μ and each node transmits with probability $1/(2\mu)$). This allows us to apply Lemma 3 to determine successful transmissions. If u transmits a HELLO in a given round, then its neighbors receive it clearly with probability is at least $\frac{1}{2\mu} \cdot \kappa$, and if a clear reception occurs, then u receives ACK, also with probability at least $\frac{1}{2\mu} \cdot \kappa$. Hence, if both u and v are active at the beginning of a round, they stay active after that round with probability at most $1 - (\kappa/2\mu)^2$. Thus, the probability that they stay active for all $\gamma \ln n$ rounds is at most $(1 - (\kappa/2\mu)^2)^{\gamma \ln n} \leq e^{-3\ln n} = n^{-3}$. By the union bound, the probability that some r-adjacent pairs remains active is at most n^{-1}. $\qquad\square$

5 Aggregation Structure Construction

We give in this section an algorithm to form a hierarchical aggregation structure. The algorithm has three parts: forming a dominating set, coloring the dominators to separate them spatially, and finally forming a tree of reporters to speed up aggregation using the multiple channels.

5.1 Communication Backbone

To reduce computation and communication, we construct an overlay in the form of a connected dominating set. The dominators function as local leaders of their respective *clusters*, managing the local computation, as well as participating in disseminating the information globally. The dominators are colored to ensure good spatial separation between clusters of same color, which in turn allows the local computation to ignore interference from other clusters.

5.1.1 Computing a Dominating Set

We first form a *clustering*, which is a function assigning each node a dominator within a specified distance r.

Let $t = \left(\frac{\alpha-2}{48\beta(\alpha-1)}\right)^{1/\alpha}$ and $r_c = \min\{\frac{t}{2t+2} \cdot R_{\epsilon/2}, \frac{\epsilon R_T}{4}\}$. Recall that $R_{\epsilon/2} = (1 - \frac{\epsilon}{2})R_T$. We adapt the algorithm of Scheideler et al. [28] to compute an r_c-dominating set of constant density. In that algorithm, a node that receives a message from a dominator becomes a dominatee; here, we simply additionally require that the node receive a message from a dominator within distance r_c. Using the same argument as in [28], we have the following result.

Lemma 7. *There is a distributed algorithm running in time $O(\log n)$ that produces, with high probability, an r_c-dominating set of constant density μ, along with the corresponding clustering function.*

5.1.2 Cluster Coloring and a TDMA Scheme of Clusters

To separate the clusters spatially, we color the dominators so that those within distance $R_{\epsilon/2}$ are assigned different colors, as done by the following algorithm.

The algorithm operates in ϕ phases, where ϕ is an upper bound on the number of dominators in any disk of radius $R_{\epsilon/2}$. A standard area argument gives an upper bound of $\phi := 4\mu(R_{\epsilon/2} + r_c/2)^2/r_c^2 \in O(1)$. In each phase i, dominators that are still not colored compute a $(R_{\epsilon/2}, R_\epsilon)$-ruling set, using the algorithm of Sec. 4, and assign the nodes of the ruling set the color i.

The following result follows easily from Lemma 6.

Lemma 8. *Given an upper bound ϕ on the dominator density, there is an algorithm for coloring the dominators (assigning $R_{\epsilon/2}$-neighbors different colors) using ϕ colors in $O(\log n)$ rounds.*

The cluster coloring yields the following TDMA scheme of ϕ rounds: only nodes in clusters of color i transmit in the i-th round, for $i = 1, 2, \ldots, \phi$. A clustering with a proper coloring as described above is called *well-separated*. Lemma 2 and the setting of r_c imply the following result.

Lemma 9. *If at most one node transmits in each cluster (on a given channel), and only in clusters of a particular color, then each such transmission is received by all nodes within the same cluster.*

Thus, when using the TDMA scheme, communication within clusters can proceed deterministically without concern for outside interference (as long as only one node transmits in a cluster). For simplicity, in the subsequent sections, we implicitly assume that clusters of the same color execute the algorithm together in the rounds assigned by the TDMA scheme and only consider the algorithm execution of the clusters with a particular color. This assumption incurs an overhead of only a constant factor ϕ on the running time.

5.2 Reporter Tree Construction in Clusters

The tree construction proceeds in three steps. We first estimate the number of nodes in the cluster, which determines the number of channels to which to assign the nodes randomly. Within each channel, a leader known as a reporter is then elected. Finally, the reporters automatically organize themselves into a complete binary tree, using the channel number as a heap number in the tree.

Denote by C_v the cluster consisting of dominator v and its dominatees. Denote by $f_v = \min\{\lceil |C_v|/(c_1 \log n)\rceil, \mathcal{F}\}$ the number of channels used in cluster C_v, where $c_1 = 24$. The setting of f_v ensures (by Chernoff bound) that, with high probability, each channel is assigned at least one node.

The following theorem summarizing the results of this subsection follows from Lemmas 14, 15, and 16 given later.

Theorem 10. *Suppose clusters are well-separated. There is an algorithm that for each cluster elects a reporter on each of its channels and organizes them into a complete binary tree, using $O(\log^2 n)$ rounds with high probability. If a $\log^{O(1)} n$-approximation of Δ is given, then $O(\Delta/\mathcal{F} + \log n \cdot \log\log n)$ rounds suffice.*

Since the number of channels used in a cluster depends on its size, we first need to approximate that quantity and make it known to all dominatees.

5.2.1 Cluster Size Approximation

Suppose an upper bound $\hat{\Delta}$ on the size of any cluster is known. Consider the following **Cluster-Size-Approximation (CSA)** problem: Given a set of nodes partitioned into well-separated clusters, each of size at most $\hat{\Delta}$, compute a constant approximation of the cluster size and disseminate it to all nodes in the cluster. In the most general case, $\hat{\Delta}$ can be taken to be n.

5.2.1.1 Cluster Size Approximation with Large $\hat{\Delta}$

The CSA algorithm uses only the first channel, i.e., all nodes operate on a single channel. The stage is divided into $\lceil \log \hat{\Delta} \rceil$ phases, each of which contains $\gamma_1 \ln n + 1$ rounds, where γ_1 is a constant to be determined.

In all but the last round of a phase, each dominatee u transmits with a specified probability, while the dominators listen. In rounds of phase j, the common transmission probability p_j is $\frac{\lambda}{\hat{\Delta}} \cdot 2^{j-1}$, where $\lambda = 1/2$. Namely, the initial probability is $\lambda/\hat{\Delta}$, and the probability is doubled after each phase. In the last round of each phase, the dominator sends out a notification if it received enough messages from the nodes in its cluster, in which case all the nodes terminate the algorithm. If a dominator receives at least $\omega_1 \ln n$ messages in phase j from nodes in its cluster, where $\omega_1 = 36$, then it settles for the estimate of $|\hat{C}_v| := \lceil \hat{\Delta} \cdot 2^{-j+1} \rceil$ for the number of nodes in its cluster. Note that, if the contention $P_c(v)$ is constant when the algorithm terminates, then $|\hat{C}_v| = \Theta(|C_v|)$, a constant approximation of the true cluster size.

We start with preliminary results before deriving the main result on CSA.

Lemma 11. *Let v be a dominator and consider a phase of the CSA algorithm. The following holds with probability $1 - n^{-3}$: If $P_c(v) < \omega_1/(4\gamma_1)$, then v receives fewer than $\omega_1 \ln n$ messages in the phase, while if $P_c(v) \in (\lambda/2, \lambda]$ and $P_c(w) \leq \lambda = 1/2$ for every dominator w, then v receives at least $\omega_1 \ln n$ messages.*

Proof. Suppose first that $P_c(v) < \omega_1/(4\gamma_1)$. The dominator v receives a message in a round with probability at most $P_c(v)$, and therefore receives at most $\gamma_1 \ln n \cdot \omega_1/(4\gamma_1) = \frac{\omega_1}{4} \ln n$ messages during the phase, in expectation. By Chernoff bound (2) using $\omega_1 = 36$, it holds with probability $1 - n^{-3}$ that v receives at most $\frac{\omega_1}{2} \ln n$ messages.

Suppose now that $P_c(v) \in (\lambda/2, \lambda]$ and that $P_c(w) \leq \lambda$ for every dominator w. By Lemma 3, if a dominatee transmits in a round, its dominator receives the message with constant probability κ. The probability that v receives some message in a given round of the phase is then at least $\sum_{w \in C_v} p_w \cdot \kappa = \kappa \cdot P_c(v) \geq \kappa \cdot \frac{\lambda}{2}$. Then during the first $\gamma_1 \ln n$ rounds of phase j, v receives at least expected $\kappa \cdot \frac{\lambda}{2} \cdot \gamma_1 \ln n$ messages from its dominatees. Setting $\gamma_1 \geq 2\omega_1 \cdot \frac{2}{\kappa\lambda}$, it follows from Chernoff bound (3) that v receives at least $\omega_1 \ln n$ messages during the first subphase, with probability $1 - n^{-3}$, in which case it notifies its dominatees to terminate the algorithm. $\square$

Lemma 12. *With a known upper bound $\hat{\Delta}$ on the maximum cluster size, the CSA algorithm approximates the size of each cluster within a constant factor in $O(\log \hat{\Delta} \cdot \log n)$ rounds, with high probability. Using the naive bound of $\hat{\Delta} \leq n$, the running time is $O(\log^2 n)$.*

Proof. By the first part of Lemma 11, using union bounds, it holds with probability at least $1 - 1/n$ that whenever a dominator v explicitly terminates the algorithm, then $\omega_1/(4\gamma_1) \leq P_c(v) \leq \lambda$. Assume that v terminates the algorithm in phase j. The transmission probability during phase j is $p_j = \frac{1}{2\hat{\Delta}} \cdot 2^{j-1}$. Then, $|C_v| = \frac{P_c(v)}{\frac{1}{2\hat{\Delta}} \cdot 2^{j-1}} \in [2\hat{\Delta} \cdot 2^{-j+1} \cdot \omega_1/4\gamma_1, 2\hat{\Delta} \cdot 2^{-j+1} \cdot \lambda]$. Hence, $|\hat{C}_v| = \lceil \hat{\Delta} \cdot 2^{-j+1} \rceil \in \Theta(|C_v|)$. In other words, the estimate $|\hat{C}_v|$ obtained is always a constant approximation of the true cluster size $|C_v|$. The algorithm is run for at most $\log \hat{\Delta} = O(\log n)$ phases, for a $O(\log^2 n)$ bound on the time complexity. It remains to argue that the algorithm is explicitly terminated.

By applying the union bounds on the second part of Lemma 11, it holds with probability at least $1 - 1/n$ that $P_c(v) \leq \lambda$ is satisfied for every vertex in each phase. Initially, $p_1 = \lambda/\hat{\Delta}$, for each dominatee u, in which case $P_c(v) \leq \lambda$ is satisfied. If the algorithm operates for all the $\lceil \log \hat{\Delta} \rceil$ phases, then $p_j \geq \lambda/2$ in the last phase j, in which case $P_c(v) \geq \lambda/2$. Thus, for each dominator v, there is a phase in which the conditions of the second part of Lemma 11 are satisfied, in which case the dominator terminates the algorithm, with high probability. $\square$

5.2.1.2 Cluster Size Approximation with Small $\hat{\Delta}$

For the case that $\hat{\Delta} \leq \mathcal{F} \log^c n$, for constant $c \geq 1$, the Cluster-Size-Approximation problem can be solved more efficiently. The basic process is as follows: First, each dominatee selects a channel uniformly at random. Then, on each channel, the nodes selecting that channel elect a leader and execute the CSA algorithm to obtain constant approximation of the number of dominatees in the channel. Finally, the dominator obtains a constant approximation of the cluster size by polling the estimates from the leaders on each channel, and sends the estimate to its dominatees on the first channel in the last round. The detailed algorithm and analysis of the following result are given in the Appendix.

Lemma 13. *Given knowledge of $\hat{\Delta}$ satisfying $\hat{\Delta} \leq \mathcal{F} \log^c n$ for some constant $c \geq 1$, we can get a constant approximation of the size of each cluster in $O(\log n \cdot \log \log n)$ rounds, with high probability.*

We can combine the two cluster size estimation procedures if $\hat{\Delta}$ is a $\log^{\hat{c}} n$-approximation of Δ for some constant $\hat{c} \geq 0$: When $\hat{\Delta}/\mathcal{F} \leq \log^{\hat{c}+2} n$, Lemma 13 gives a bound of $O(\log n \cdot \log \log n)$ rounds, while otherwise the bound of Lemma 12 is $O(\log^2 n) = O(\Delta/\mathcal{F})$ rounds. Hence, based on Lemmas 12 and 13, we have the following result.

Lemma 14. *There is a constant-approximation algorithm for **Cluster-Size-Approximation** that runs in $O(\log^2 n)$ rounds, with high probability. When given a $\log^{O(1)} n$-approximation of Δ, there is a constant-approximation algorithm that runs in $O(\Delta/\mathcal{F} + \log n \cdot \log \log n)$ rounds, with high probability.*

For simplification, we shall simply use $|C_v|$ to denote the size estimate $|\hat{C}_v|$ derived for cluster C_v. Since it is a constant approximation, it will not affect the asymptotic running times.

5.2.2 Reporter Election and Aggregation Tree Formation

In this stage, reporters are elected in each cluster simultaneously by running the ruling set algorithm of Sec. 4. For a cluster C_v, a reporter is elected on each of the channels $F_1, F_2, \ldots, F_{f_v}$. To argue correctness, it suffices in light of Lemma 6 to show that every channel gets assigned some node. The expected number of nodes in C_v choosing a channel is $|C_v|/f_v$. Chernoff bound (3) and the union bound then imply the desired result with high probability.

Lemma 15. *In each cluster C_v, with high probability, exactly one reporter is elected on each of the f_v channels in $O(\log n)$ rounds.*

We refer to dominatees that are not reporters as *followers*. Thus, C_v is partitioned into one dominator, f_v reporters, and $|C_v| - f_v - 1$ followers. In subsequence, we use $X_v = \{u_1, \ldots, u_{f_v}\}$ and Y_v to denote the sets of reporters and followers in C_v, respectively, where u_i is the reporter elected on channel F_i. Let $u_0 = v$ refer to the dominator. We define a complete binary tree rooted at the dominator, with the reporters ordered in level-order, like a binary heap. Thus, $u_{\lfloor k/2 \rfloor}$ is the parent of u_k in the tree, for $k = 1, \ldots, f_v$.

Once the reporters are elected, the aggregation tree is then ready to use.

Lemma 16. *A complete binary tree of $\lfloor \log(f_v + 1) \rfloor$ levels is constructed on the reporters for each cluster C_v. Operating on well-separated clusters, it can perform a convergecast operation deterministically in time $2\lfloor \log(f_v + 1) \rfloor$.*

6 Data Aggregation

The data aggregation algorithm consists of three procedures executed in parallel: The intra-cluster aggregation involves two processes: a) collecting the data from followers to the reporters, b) aggregating the data of dominatees using the reporter tree to the dominator, and finally aggregating the data among the dominators. The first two procedures can together be referred to as *intra-cluster aggregation*, while the last one is *inter-cluster* aggregation. In each round there are five slots for these three procedures: a pair of send/acknowledge slots for each of the first two, and a single slot for the last one.

Aggregation from Followers to Reporters. The execution of this process is divided into phases, each with $\Gamma + 1$ rounds, where $\Gamma := \gamma_2 \ln n$ and γ_2 is to be determined. For a cluster C_v, the first f_v channels are used for transmissions. The first channel is special in that the dominator listens on it to estimate the contention. In each phase, the operations of nodes are as follows:

(i) A follower $u \in Y_v$, in each of the first Γ rounds, selects one of the first f_v channels uniformly at random, transmits on the selected channel with a specified probability p_u in the first slot, and listens in the second slot for an acknowledgement (*ack*) from its reporter. Initially, p_u is set as $p_u = \lambda f_v/|C_v|$ with $\lambda = 1/2$. If u receives an ack, it halts.

In the last round, u listens on the first channel. After each phase, if u receives a backoff message from its dominator in the last round, it keeps p_u unchanged, and doubles p_u otherwise.

(ii) A reporter $w \in X_v$ operates on the channel where it is elected. In each of the first Γ rounds, w listens in the first slot. If it receives a message from a follower in its cluster, it returns an acknowledgement in the second slot. In the last round, w does nothing.

(iii) The dominator v listens on the first channel during the first Γ rounds. In the last round it transmits a *backoff message* if and only if it heard at least $\Omega := \omega_2 \ln n$ messages from followers during the preceding rounds.

In the above algorithm, we set the constant parameters as follows: $\omega_2 = 96/\kappa_1$ and $\gamma_2 = 8\omega_2/\kappa_1$, where $\kappa_1 \leq 1$ is a constant that will be given in Lemma 18 of the analysis.

Aggregation on Reporter Tree. The execution of this process is divided into phases, where each phase contains $\lfloor \log(\mathcal{F} + 1) \rfloor - 1$ rounds.

For a cluster C_v, the first f_v channels are used for transmissions. As before, use $X_v = \{u_1, \ldots, u_{f_v}\}$ to denote the set of reporters and T_v to denote the reporter tree. We enumerate the levels of T_v from bottom, i.e., with the leaves at level 1.

In the s-th round of a phase, nodes at level s and $s + 1$ of T_v execute the algorithm to aggregate from level s to level $s + 1$, while other nodes keep silent. Each reporter u_k at level s operates on the same channel as its parent, i.e., on the channel $\lfloor k/2 \rfloor$. If k is odd (even), then u_k transmits its data to its parent $u_{\lfloor k/2 \rfloor}$ in the third (fourth) slot of round s, respectively.

Inter-cluster Aggregation. In this procedure, we use a known approach for disseminating data on a constant-density backbone network (e.g., see Section 5.2 in [2]). The basic idea of the algorithm is to use flooding (with continuous constant-probability transmissions) to produce an aggregation/broadcast tree, with which data can be aggregated and then broadcast to all nodes in $O(D + \log n)$ rounds with high probability.

6.1 Analysis

The main effort of the analysis is on the first procedure, aggregating from the followers to reporters. We address the other two in the final theorem.

To bound the time spent on aggregating from the followers we show that we maintain linear throughput while the contention is high enough. Namely, while the contention is above a fixed constant threshold, each reporter makes progress with constant probability, where progress means aggregating a message from one more follower. To this end, we show that contention always remains bounded from above, and whenever it becomes low, the transmission probabilities double. When the contention dips below the threshold, we need only doubly logarithmic number of phases to increase the transmission probabilities to constant and aggregate the remaining followers.

The sum of transmission probabilities of followers in a cluster is referred to as the *contention* in the cluster, and denoted by $P_c(v) = \sum_{u \in Y_v} p_u$.

Definition 17 (Bounded Contention). Bounded Contention *is achieved in a given round if the contention in each cluster is at most half the number of channels alotted, i.e., $P_c(v) \leq \lambda f_v = \frac{1}{2} f_v$, for each cluster C_v.*

Even if the contention in each cluster is bounded, we cannot directly use the result in Lemma 3, as the contention on a particular channel may not be constant bounded. But because followers select the operating channel uniformly at random, it can be seen that the expected contention on each channel can be bounded by λ. This is enough to use the interference bounding technique used for proving Lemma 3, and we can get the following Lemma 18.

We say a follower *succeeds* (to transmit) if its message is properly received by a reporter on a channel. The proof detail is omitted because it is very similar to the standard argument given in [11] (Lemma 4.1 and Lemma 4.2).

Lemma 18. *Assuming Bounded Contention holds, whenever a follower transmits, it succeeds with probability at least κ_1, for a universal constant $\kappa_1 > 0$.*

The TDMA scheme ensures that when a follower succeeds, it receives the ack message in the subsequent slot, as argued in Lemma 9.

Using Lemma 18, we can argue the Bounded Contention property.

Lemma 19. *Bounded Contention holds in every round, with probability $1 - n^{-1}$.*

Proof. We prove the Lemma by contradiction. Assume that cluster C_v is the first one to violate the Bounded Contention property, and that the violating phase is j. The initial transmission probability implies that $j > 1$. We focus on phase $j - 1$. In this phase, by assumption, we have $P_c(y) \leq \lambda f_y$ for each cluster C_y, and since the transmission probability of followers is at most doubled between phases, $P_c(v) \in (\lambda f_v/2, \lambda f_v]$. The expected number of transmissions by followers in C_v during phase $j - 1$ is then at least $\lambda f_v/2 \cdot \Gamma$. Since Bounded Contention holds in phase $j - 1$, each transmission is successful with some constant probability κ_1. Hence, there are $\lambda/2 \cdot \Gamma \kappa_1 = \frac{8}{4}\omega_2 \ln n = 2\Omega$ successful transmissions on each channel, in expectation. Using Chernoff bound (3), the dominator v receives at least Ω transmissions with probability $1 - n^{-3}$ (as $\omega_2 \geq 12$). Then, by Lemma 9, v sends a backoff message to all the followers, who keep their transmission probability unchanged after this phase $j - 1$. As a result, the λf_v bound will not be broken in phase j, which contradicts with our assumption. So C_v cannot be the first violating cluster with probability $1 - n^{-3}$. The Lemma is then proved by the union bound. ∎

A phase is *increasing* if the transmission probability of the reporters in C_v is doubled after the phase, i.e., the dominator v receives less than $\omega_2 \ln n$ messages, and otherwise it is *unchanging*. Let N_v^j denote the total number of transmissions by followers in C_v during phase j. A transmission by a follower $u \in Y_v$ is *successful* if u succeeds in transmitting the data to a reporter.

Lemma 20. *Consider a cluster C_v. If a phase j is unchanging, then, with probability at least $1 - n^{-3}$, there are at least $\Omega/4 = \frac{\omega_2}{4} f_v \ln n$ transmissions in the phase, of which at least $12 f_v \ln n$ are successful.*

Proof. Suppose there are fewer than $\Omega/4$ transmissions in phase j. Then, since channels are chosen with equal probability, the expected number of transmissions in the first channel is at most $\frac{\omega_2}{4} \ln n$. Then, by Chernoff bound (2) (since $\omega_2 \geq 36$), at most $\frac{\omega_2}{2} \ln n$ transmissions are made in the channel, with probability $1 - n^{-3}$, which implies that the phase is increasing. Thus, the first part of the lemma holds: if a phase if unchanging, then at least $\Omega/4$ transmissions occur. By Lemma 18, the expected number of successful transmissions is then at least $\Omega/4 \cdot \kappa_1 = (\omega_2 f_v \ln n/4) \cdot \kappa_1 = 24 f_v \ln n$. Using Chernoff bound (3), the number of successful transmissions is at least $12 f_v \ln n$, with probability $1 - O(n^{-3})$. ∎

Based on above analysis, we can now get the result for the first procedure.

Lemma 21. *In each cluster, the data of all followers can be aggregated to the reporters in $O(\frac{\Delta}{\mathcal{F}} + \log n \log \log n)$ rounds, with probability $1 - O(n^{-1})$.*

Proof. Consider a cluster C_v. There are at most $O(|C_v|/(12 f_v \ln n)) = O(1 + \Delta/(\mathcal{F} \log n))$ unchanging phases, by Lemma 20, with probability $1 - n^{-3}$. Also, when the transmission probability of a follower is

increased to a constant $\lambda/2$ in a phase, it can successfully send its data to a reporter with probability $\lambda \kappa_1/2$ in each round of the phase by Lemma 18, and the $\gamma_2 \ln n \geq \frac{3 \ln n}{\lambda \kappa_1/2}$ rounds in the phase ensure successful transmission with high probability. Hence, there are at most $O(\log(|C_v|/f_v)) = O(\log(\Delta/\mathcal{F}) + \log \log n)$ increasing phases for each cluster, given the initial transmission probability of followers. Combined, the number of phases is $O(\Delta/(\mathcal{F} \log n) + \log(\Delta/\mathcal{F}) + \log \log n) = O(\Delta/(\mathcal{F} \log n) + \log \log n)$, and thus the number of rounds is $O(\Delta/\mathcal{F} + \log n \log \log n)$, with probability $1 - n^{-3}$. The lemma then follows from the union bound over the clusters. $\qquad \square$

Theorem 22. *Data aggregation can be accomplished in $O(D + \frac{\Delta}{\mathcal{F}} + \log n \log \log n)$ rounds, with high probability.*

Proof. We can combine the high probability bounds on each of the three procedures. By Lemma 21, the aggregation from followers to reporters is achieved in $O(\frac{\Delta}{\mathcal{F}} + \log n \log \log n)$ rounds (with probability $1 - O(n^{-1})$). In each cluster C_v, the data aggregation from the reporters to the dominator can be accomplished in $O(\log \mathcal{F})$ rounds. Namely, the construction of the aggregation tree ensures that when a reporter transmits, it is the only one from the same cluster in the same channel, and thus, by Lemma 9, each transmission is successful. The number of rounds to aggregate from reporters to dominator then equals the height of the tree, or $\lfloor \log(\mathcal{F} + 1) \rfloor$. Finally, Theorem 3 in [2] achieves (the inter-cluster) aggregation on the dominators in $O(D + \log n)$ rounds. $\qquad \square$

7 Coloring

Using the aggregation structure, the data of dominatees can be efficiently aggregated to a dominator, as shown in Sec. 6. This aggregation structure can be used to solve fundamental problems other than data aggregation, which we illustrate on the node coloring problem.

Algorithm. In the constructed aggregation structure, the dominators are colored with cluster colors $1, 2, \ldots, \phi$ for some constant ϕ such that dominators within distance $R_{\epsilon/2}$ receive different cluster colors (refer to Sec. 5.1.2). We then allocate to each dominator of cluster color i the sequence of colors $k\phi + i : k = 0, 1, 2, \ldots$ to assign to its cluster nodes.

Operating on each cluster C_v, the algorithm consists of four procedures:

1. The followers execute the data aggregation algorithm of Sec. 6 to send their IDs to the reporters, by which each reporter will acquire the knowledge of all of its followers. An aggregation tree on all nodes in C_v is then constructed based on the reporter tree by adding links that connect each reporter and the followers following it.

2. Each reporter forwards the number of nodes in its subtree (including the reporters and the followers) to its parent in the reporter tree.

3. The color range (the range of k, which determines the set of available colors) of each reporter and its followers is then disseminated to each reporter via the reporter tree. In particular, on the reporter tree, each node u (recall that the root is the dominator) determines the color ranges of its two children based on the color range assigned to u and the number of nodes in the subtree of its children. The distribution of the color range uses an inverse process to the aggregation on the reporter tree given in Sec. 6.

4. For a reporter u, let B_u denote the set of colors assigned to it (which can be derived using the color range assigned to u). Each reporter u then assigns a different color in B_u to each of its followers and announces the color assignment one by one to its followers.

Because the first procedure uses a randomized algorithm, and the other three procedures are done by letting nodes execute the deterministic TDMA scheme given in Section 5.1, to avoid the interference between the executions of different procedures among clusters, we run procedures in separate slots of each round. Specifically, in each round, there are four slots for the execution of each of the four procedures.

Analysis.

Lemma 23. *For each cluster C_v, after $O(\frac{\Delta}{\mathcal{F}} + \log n \log \log n)$ rounds, each node in C_v will get a different color with high probability. And the total number of colors used is $O(\Delta)$.*

Proof. As for the time complexity, the first procedure takes $O(\Delta/\mathcal{F} + \log n \log \log n)$ rounds, with high probability, by Lemma 21; the second and third procedures take $O(\log \mathcal{F})$ rounds, or proportional to the height of the tree (using Lemma 16); and finally, the fourth procedure takes as many rounds as a reporter has followers, or $O(|C_v|/f_v + \log n)$, since these messages are successfully received because the reporters transmit on different channels. Because nodes have a constant approximation of $|C_v|$, and knowledge of n (a polynomial estimate) and the number of channels $\mathcal{F}$, they can each determine the completion time of each procedure.

The time bound can be obtained by the execution time of each procedure. We next show that each node in C_v gets a different color and the total number of colors used is $O(\Delta)$.

By Lemma 21, each follower can send its ID to a reporter with high probability. We claim that each follower transmits its ID to only one reporter. This follows from Lemma 9. By this Lemma, once a follower transmits a message to a reporter, it will receive an ack message in the same round. Hence, the sets of followers of reporters are disjoint. With this claim, we can see that the aggregation tree on all nodes in C_v is correctly constructed in the first procedure, i.e., every node is in the tree and has exactly one parent. Then in the second procedure, each reporter will get the exact number of nodes in its subtree by the analysis in Theorem 22. Based on this knowledge and because the aggregation tree is correctly constructed, after the third procedure, reporters will get disjoint color ranges and the number of colors used is $|C_v| \in O(\Delta)$. Hence, after the fourth procedure, each node will get a different color. $\qquad\square$

Theorem 24. *A proper coloring with $O(\Delta)$ colors can be computed in $O(\frac{\Delta}{\mathcal{F}} + \log n \log \log n)$ rounds with high probability.*

Proof. The total time used for the coloring is given in Lemma 23. By the algorithm, it is easy to see that the total number of colors used is $\phi \cdot O(\Delta) \in O(\Delta)$.

We next show the correctness of the coloring algorithm. For any two neighboring nodes u, v that are in different clusters, their dominators have distance at most $\epsilon R_T/4 + R_\epsilon + \epsilon R_T/4 = R_{\epsilon/2}$. By the algorithm, the color sets given to the clusters in which u, v stay are disjoint. Hence, u, v will not get the same color. For any pair of neighboring nodes in the same cluster, they will also be assigned different colors by Lemma 23. $\qquad\square$

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] M. H. Bodlaender, M. M. Halldórsson, and P. Mitra. Connectivity and Aggregation in Multihop Wireless Networks. In *PODC*, pages 355–364. ACM, 2013.

[3] R. Cole and U. Vishkin. Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking. *Inf. Control*, 70(1):32–53, July 1986.

[4] S. Daum, M. Ghaffari, S. Gilbert, F. Kuhn, and C. Newport. Maximal independent sets in multichannel radio networks. In *PODC*, pages 335–344, 2013.

[5] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Leader election in shared spectrum radio networks. In *PODC*, pages 215–224, 2012.

[6] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Broadcast in the Ad Hoc SINR Model. In *DISC*, pages 358–372, 2013.

[7] S. Daum, F. Kuhn, and C. Newport. Efficient symmetry breaking in multi-channel radio networks. In *DISC*, pages 238–252, 2012.

[8] B. Derbel and E. Talbi. Distributed Node Coloring in the SINR Model. In *ICDCS*, pages 708–717, 2010.

[9] S. Dolev, S. Gilbert, M. Khabbazian, and C. Newport. Leveraging channel diversity to gain efficiency and robustness for wireless broadcast. In *DISC*, pages 252–267, 2011.

[10] H. Du, Z. Zhang, W. Wu, L. Wu, and K. Xing. Constant-approximation for optimal data aggregation with physical interference. *Journal of Global Optimization*, 56(4):1653–1666, 2013.

[11] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC '08*, pages 35–44, 2008.

[12] M. M. Halldórsson. Wireless scheduling with power control. *ACM Trans. Algorithms*, 9(1):7:1–7:20, Dec. 2012.

[13] M. M. Halldórsson and P. Mitra. Nearly Optimal Bounds for Distributed Wireless Scheduling in the SINR Model. In *ICALP*, pages 625–636, 2011.

[14] M. M. Halldórsson and P. Mitra. Distributed Connectivity of Wireless Networks. In *PODC*, 2012.

[15] M. M. Halldórsson and P. Mitra. Towards tight bounds for local broadcasting. In *FOMC*, 2012.

[16] M. M. Halldórsson and P. Mitra. Wireless connectivity and capacity. In *SODA*, pages 516–526. SIAM, 2012.

[17] N. Hobbs, Y. Wang, Q.-S. Hua, D. Yu, and F. Lau. Deterministic distributed data aggregation under the SINR model. In *TAMC'12*, pages 385–399, 2012.

[18] T. Jurdzinski, D. Kowalski, M. Rozanski, and G. Stachowiak. Distributed randomized broadcasting in wireless networks under the SINR model. In *DISC*, pages 373–387. 2013.

[19] T. Jurdzinski, D. Kowalski, M. Rozanski, and G. Stachowiak. On the impact of geometry on ad hoc communication in wireless networks. In *PODC '14*, 2014.

[20] T. Jurdzinski and D. R. Kowalski. Distributed backbone structure for algorithms in the SINR model of wireless networks. In *DISC*, pages 106–120, 2012.

[21] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak. Distributed Deterministic Broadcasting in Uniform-power Ad Hoc Wireless Networks. In *FCT'13*, pages 195–209, 2013.

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

[23] H. Li, Q.-S. Hua, C. Wu, and F. Lau. Minimum-latency Aggregation Scheduling in Wireless Sensor Networks Under Physical Interference Model. In *MSWiM '10*, pages 360–367, 2010.

[24] X.-Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi. Efficient data aggregation in multi-hop wireless sensor networks under physical interference model. In *MASS '09*, pages 353–362, 2009.

[25] T. Moscibroda and R. Wattenhofer. The complexity of connectivity in wireless networks. In *INFO-COM*, pages 1–13, April 2006.

[26] T. Moscibroda and R. Wattenhofer. Coloring unstructured radio networks. *Distributed Computing*, 21(4):271–284, 2008.

[27] G. Pei and A. K. S. Vullikanti. Distributed approximation algorithms for maximum link scheduling and local broadcasting in the physical interference model. In *INFOCOM*, 2013.

[28] C. Scheideler, A. Richa, and P. Santi. An $O(\log N)$ Dominating Set Protocol for Wireless Ad-hoc Networks Under the Physical Interference Model. In *MobiHoc '08*, pages 91–100, 2008.

[29] J. Schneider and R. Wattenhofer. Coloring Unstructured Wireless Multi-hop Networks. In *PODC*, pages 210–219, 2009.

[30] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia. Minimum-latency aggregation scheduling in multihop wireless networks. In *MobiHoc '09*, pages 185–194, 2009.

[31] X. Xu, M. Li, X. Mao, S. Tang, and S. Wang. A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(1):163–175, Jan 2011.

[32] B. Yu, J. Li, and Y. Li. Distributed data aggregation scheduling in wireless sensor networks. In *INFOCOM'09*, pages 2159–2167, 2009.

[33] D. Yu, Q.-S. Hua, Y. Wang, and F. Lau. An $O(\log n)$ Distributed Approximation Algorithm for Local Broadcasting in Unstructured Wireless Networks. In *DCOSS '12*, pages 132–139, 2012.

[34] D. Yu, Q.-S. Hua, Y. Wang, H. Tan, and F. Lau. Distributed multiple-message broadcast in wireless ad-hoc networks under the SINR model. In *SIROCCO'12*, pages 111–122, 2012.

[35] D. Yu, Q.-S. Hua, Y. Wang, J. Yu, and F. Lau. Efficient distributed multiple-message broadcasting in unstructured wireless networks. In *INFOCOM'13*, pages 2427–2435, 2013.

[36] D. Yu, Y. Wang, Q.-S. Hua, and F. Lau. Distributed (Δ+1)-coloring in the physical model. In *ALGO-SENSORS'11*, pages 145–160, 2011.

[37] D. Yu, Y. Wang, Y. Yan, J. Yu, and F. Lau. Speedup of information exchange using multiple channels in wireless ad hoc networks. In *INFOCOM'15*, 2015.

A   Cluster Size Approximation with Small $\hat{\Delta}$

When the contention is known to be small relative to the number of channels, we can reduce the time complexity for computing the cluster size. Here we consider the case that $\hat{\Delta} \leq \mathcal{F} \log^c n$ for some constant $c \geq 1$.

Algorithm. For each cluster C_v, the algorithm consists of four procedures:

1. Initially, each dominatee in C_v selects a channel from $\mathcal{F}$ uniformly at random. On each channel, the nodes selecting the channel elect a leader by executing the ruling-set algorithm given in Sec. 4. This procedure consists of $\gamma_3 \ln n$ rounds, where γ_3 is set to be a sufficiently large constant such that there are enough rounds for the execution of the algorithm in Sec. 4.

2. On each channel, nodes execute the CSA Algorithm with $\hat{\Delta} = \gamma_3 \ln^c n$, where the leader functions as the dominator on the channel.

3. The leaders aggregate the number of nodes that selected the channels they dominate. This procedure consists of $O(\log \mathcal{F})$ rounds. In particular, denote by $U_v = \{x_1, \ldots, x_{\mathcal{F}}\}$ the set of leaders in cluster C_v. Note that there may be some channels without nodes selecting it and thus without leaders elected on them. Hence, there may be some nodes x_i missing. For each channel that does not have nodes, we add an auxiliary node, and it will be introduced how to deal with these auxiliary nodes in the aggregation process.

We first construct a binary tree on these $\mathcal{F}$ nodes rooted at the dominator using the same manner as the reporter tree construction in Sec. 5.2. Then we use the data aggregation algorithm on the reporter tree given in Sec. 6 to aggregate the number of nodes to the dominator. Specifically, we need to handle here the auxiliary nodes. The solution is to divide each slot in each round into two sub-slots (recall that there are two slots in each round for the data aggregation on reporter trees), and make a parent send the ack message when it receives a message from its children. For each node x_j transmits, if it does not receive the ack message from its parent, which means that its parent is an auxiliary node, x_j will function as its parent in the subsequent aggregation process.

4. Finally, in a single round, v broadcasts the estimate of the cluster size to its dominatees on the first channel.

Analysis.

Proof of Lemma 13. Consider a cluster C_v. We analyze the four procedures one by one. We first bound the number of nodes operating on each channel in the first procedure.

Claim 25. *For a cluster C_v, in the first procedure, there are at most $2 \ln^c n$ nodes on each channel with probability $1 - n^{-2}$.*

Proof. Because dominatees select channels uniformly at random, the expected number of dominatees selecting each channel is at most $\ln^c n$. Consider a channel F. Using Chernoff bound (3), we get that the number of dominatees selecting F is at most twice the expectation, with probability $1 - n^{-3}$. By the union bound on all channels, the result follows. □

A channel F is *nonempty* with respect to a cluster C_v if there are dominatees in C_v selecting it in the first procedure. Using a similar argument for proving Lemma 15, we have the following result for the first procedure.

Claim 26. *For each cluster and each nonempty channel F, exactly one leader is elected on F in $O(\log n)$ rounds, with probability $1 - n^{-2}$.*

Using a similar argument for proving Lemma 12, we have the following result for the second procedure.

Claim 27. *Each leader in each cluster can get an absolute constant approximation of the number of dominatees selecting its channel in $O(\log n \log \log n)$ rounds, with probability $1 - n^{-2}$.*

By Lemma 9, a node will receive an ack message after it sends a message to its parent if its parent is not an auxiliary node. Hence, the auxiliary nodes will not affect the aggregation process in the third procedure. Hence, we have the following result.

Claim 28. *For a cluster C_v, the estimates of leaders will be aggregated to the dominator v in $O(\log \mathcal{F})$ rounds.*

After the estimates of leaders are aggregated to the dominator, the dominator v will get a constant approximation of the cluster size by Claim 27. Then in the fourth procedure, v can send the estimate of the cluster size to all dominatees by Lemma 9. Adding the time used in each procedure, each node in cluster C_v will get a constant approximation of the cluster size in $O(\log n \log \log n)$ rounds with probability $1 - O(n^{-2})$. The result is then proved by the union bound. $\square$