Information Gathering in Ad-Hoc Radio Networks with Tree Topology

Marek Chrobak* Kevin Costello† Leszek Gasieniec‡ Dariusz R. Kowalski‡

Abstract

We study the problem of information gathering in ad-hoc radio networks without collision detection, focussing on the case when the network forms a tree, with edges directed towards the root. Initially, each node has a piece of information that we refer to as a rumor. Our goal is to design protocols that deliver all rumors to the root of the tree as quickly as possible. The protocol must complete this task within its allotted time even though the actual tree topology is unknown when the computation starts. In the deterministic case, assuming that the nodes are labeled with small integers, we give an $O(n)$-time protocol for the model with unbounded messages, and an $O(n \log n)$-time protocol for the model with bounded messages, where any message can include only one rumor. We also consider fire-and-forward protocols, in which a node can only transmit its own rumor or the rumor received in the previous step. We give a deterministic fire-and-forward protocol with running time $O(n^{1.5})$, and we show that it is asymptotically optimal. We then study randomized algorithms where the nodes are not labelled. In this model, we give an $O(n \log n)$-time protocol and we prove that this bound is asymptotically optimal.

1 Introduction

We consider the problem of information gathering in ad-hoc radio networks, where initially each node has a piece of information called a *rumor*, and all these rumors need to be delivered to a designated target node as quickly as possible. A radio network is defined as a directed graph G with n vertices. At each time step any node v of G may attempt to transmit a message. This message is sent immediately to all out-neighbors of v. However, an out-neighbor u of v will receive this message only if no other in-neighbor of u attempted to transmit at the same step. The event when two or more in-neighbors of u transmit at the same time is called a *collision*. We do not assume any collision detection mechanism; in other words, not only this u will not receive any message, but it will not even know that a collision occurred.

One other crucial feature of our model is that the topology of G is not known at the beginning of computation. We are interested in distributed protocols, where the execution of a protocol at a node v depends only on the identifier (label) of v and the information gathered from the received

*Department of Computer Science, University of California at Riverside, USA. Research supported by NSF grants CCF-1217314 and OISE-1157129.

†Department of Mathematics, University of California at Riverside, USA. Research supported by NSA grant H98230-13-1-0228

‡Department of Computer Science, University of Liverpool, UK. Research partially supported by Network Sciences and Technologies (NeST) initiative.

messages. Randomized protocols typically do not use the node labels, and thus they work even if the nodes are indistinguishable from each other. The protocol needs to complete its task within the allotted time, independent of the topology of G.

Several primitives for information dissemination in ad-hoc radio networks have been considered in the literature. Among these, the two most extensively studied are *broadcasting* and *gossiping*.

The *broadcasting problem* is the one-to-all dissemination problem, where initially only one node has a rumor that needs to be delivered to all nodes in the network. Assuming that the nodes of G are labelled with consecutive integers $0, 1, ..., n-1$, the fastest known deterministic algorithms for broadcasting run in time $O(n \log n \log\log n)$ [23] or $O(n \log^2 D)$ [12], where D is the diameter of G. The best lower bound on the running time in this model is $\Omega(n \log D)$ [11]. (See also [9, 20, 5, 6] for earlier work.) Allowing randomization, broadcasting can be accomplished in time $O(D \log(n/D) + \log^2 n)$ with high probability [12], even if the nodes are not labelled. This matches the lower bounds in [2, 21].

The *gossiping problem* is the all-to-all dissemination problem. Here, each node starts with its own rumor and the goal is to deliver all rumors to each node. There is no restriction on the size of messages; in particular, different rumors can be transmitted together in a single message. (Models with restrictions on message size have also been studied, see [17, 8].) With randomization, gossiping can be solved in expected time $O(n \log^2 n)$ [12] (see [22, 10] for earlier work), even if the nodes are not labelled. In contrast, for deterministic algorithms, with nodes labelled $0, 1, ..., n-1$, the fastest known gossiping algorithm runs in time $O(n^{4/3} \log^4 n)$ [18], following earlier progress in [9, 30]. (See also the survey in [16] for more information.) For graphs with arbitrary diameter, the best known lower bound is $\Omega(n \log n)$, the same as for broadcasting. Reducing the gap between lower and upper bounds for deterministic gossiping to a poly-logarithmic factor remains a central open problem in the study of radio networks with unknown topology.

Our work has been inspired by this open problem. It is easy to see that for arbitrary directed graphs gossiping is equivalent to information gathering, in the following sense. On one hand, trivially, any protocol for gossiping also solves the problem of gathering. On the other hand, we can apply a gathering protocol and follow it with a protocol that broadcasts all information from the target node r; these two protocols combined solve the problem of gossiping. So if we can solve information gathering in time $O(n \operatorname{polylog}(n))$, then we can also solve gossiping in time $O(n \operatorname{polylog}(n))$.

Our results. To gain better insight into the problem of gathering information in radio networks, we focus on networks with tree topology. Thus we assume that our graph is a tree $\mathcal{T}$ with root r and with all edges directed towards r. In this model, a gathering protocol knows that the network is a tree, but it does not know its topology.

We consider several variants of this problem, for deterministic or randomized algorithms, and with or without restrictions on the message size or processor memory. We provide the following results:

- In the first part of the paper we study deterministic algorithms, under the assumption that the nodes of $\mathcal{T}$ are labelled $0, 1, ..., n-1$. First, in Section 4, we examine the model without any bound on the message size. In particular, such protocols are allowed to aggregate any number of rumors into a single message. (This is a standard model in existing gossiping protocols in unknown radio networks; see, for example, the survey in [16]). We give an optimal, $O(n)$-time protocol using unbounded messages.

- Next, in Section 5, we consider the model with bounded messages, where a message may contain only one rumor. For this model we provide an algorithm with running time $O(n \log n)$.
- In Section 6 we consider an even more restrictive model of protocols with bounded messages, that we call fire-and-forward protocols. In those protocols, at each step, a node can only transmit either its own rumor or the rumor received in the previous step (if any). Fire-and-forward protocols are very simple to implement and require very little memory, since nodes do not need to store any received messages. (We remark that our model of fire-and-forward protocols resembles hot-potato packet routing in networks that has been well studied; see, for example [4, 15].) For deterministic fire-and-forward protocols we provide a protocol with running time $O(n^{1.5})$ and we show a matching lower bound of $\Omega(n^{1.5})$.
- We then turn our attention to randomized algorithms (Sections 7 and 8). For randomized algorithms we assume that the nodes are not labelled. In this model, we give an $O(n \log n)$-time gathering protocol and we prove a matching lower bound of $\Omega(n \log n)$. The upper bound is achieved by a simple fire-and-forward protocol that, in essence, reduces the problem to the coupon-collector problem. Our main contribution here is the lower bound proof. For the special case of trees of depth two, we show that our lower bound is in fact optimal even with respect to the leading constant.

All our algorithms for deterministic protocols easily extend to the model where the labels are drawn from a set $0, 1, ..., L$ where $L \geq n$. In this model we assume that all nodes know L, but they do not need to know n. If $L = O(n)$ then the running times of our algorithms are not affected. If L is arbitrary, the $O(n)$-time algorithm for the unbounded-message model can be implemented in time $O(n^2 \log L)$, and the $O(n \log n)$-time algorithm for the bounded-message model can be implemented in time $O(n^2 \log n \log L)$.

We remark that some communication protocols for radio networks use forms of information gathering on trees as a sub-routine; see for example [3, 7, 19]. However, these solutions typically focus on undirected graphs, which allow feedback. They also solve relaxed variants of information gathering where the goal is to gather only a fraction of rumors in the root, which was sufficient for the applications studied in these papers (since, with feedback, such a procedure can be repeated until all rumors are collected). In contrast, in our work, we study directed trees without any feedback mechanism, and we require all rumors to be collected at the root.

Our work leaves several open problems, with the most intriguing one being whether there is a deterministic algorithm for trees that does not use aggregation and works in time $o(n \log n)$. We conjecture that such an algorithm exists. It would also be interesting to refine our results by expressing the running time in terms of the tree depth D and the maximum degree Δ. In terms of more general research directions, the next goal should be to establish some bounds for gathering in arbitrary acyclic graphs.

2 Preliminaries

Radio networks. In this section we give formal definitions of radio networks and gathering protocols. We define a radio network as a directed graph $G = (V, E)$ with n nodes, with each node assigned a different label from the set $[n] = \{0, 1, ..., n-1\}$. (As mentioned earlier, all the results in the paper hold if the label range is $[L]$, where $L \geq n$ and $L = O(n)$.) Denote by $\mathsf{label}(v)$ the

label assigned to a node $v \in V$. One node r is distinguished as the *target* node, and we assume that r is reachable from all other nodes. Initially, at time 0, each node v has some piece of information that we will refer to as *rumor* and we will denote it by ρ_v. The objective is to deliver all rumors ρ_v to r as quickly as possible, according to the rules described below.

The time is discrete, namely it consists of time steps numbered with non-negative integers $0, 1, 2, ...$ At any step, a node v may be either in the *transmit state* or the *receive state*. A gathering protocol $\mathcal{A}$ determines, for each node v and each time step t, whether v is in the transmit or receive state at time t. If v is in the transmitting state, then $\mathcal{A}$ also determines what message is transmitted by v, if any. This specification of $\mathcal{A}$ may depend only on the label of v, time t, and on the content of all messages received by v until time t. We stress that, with these restrictions, $\mathcal{A}$ does not depend on the topology of G and on the node labeling.

All nodes start executing the protocol simultaneously at time 0. If a node v transmits at a time t, the transmitted message is sent immediately to all out-neighbors of v, that is to all u such that (v, u) is an edge. If (v, u) and (v', u) are edges and both v, v' transmit at time t then a *collision* at u occurs and u does not receive a message. More specifically, u will receive this message if and only if (i) u is in the receive state at time t, and (ii) no collision at u occurs at time t. We do not assume any feedback from the transmission channel or any collision detection features, so, in case of a collision, neither the sender nor any node within its range knows that a collision occurred.

Throughout the paper, we will focus on the case when the graph is a rooted tree, with all edges directed towards the root. We will typically use notation $\mathcal{T}$ for this tree and r for its root.

The running time of a deterministic gathering protocol $\mathcal{A}$ is defined as the minimum time $T(n)$ such that, for any tree $\mathcal{T}$ with root r and n nodes, any assignment of labels from $[n]$ to the nodes of $\mathcal{T}$, and any node v, the rumor ρ_v of v is delivered to r no later than at step $T(n)$. In case of randomized protocols, we evaluate them either using the expectation of their running time $T(n)$, which is now a random variable, or by showing that $T(n)$ does not exceed a desired time bound with high probability.

We consider three types of gathering protocols:

Unbounded messages: In this model a node can transmit arbitrary information in a single step. In particular, multiple rumors can be aggregated into a single message.

Bounded messages: In this model no aggregation of rumors is allowed. Each message consists of at most one rumor and $O(\log n)$ bits of additional information.

Fire-and-forward: In a fire-and-forward protocol, each message consists of just one rumor, without any other information. If a node v received a message in the previous step, then v must transmit (forward) this message in the current step. Otherwise, v can either be idle or transmit its own rumor (that is, "fire"). Thus each message consists of just one rumor ρ_x, which, after being fired from its origin node x, travels towards the root one hop at a time, until either it vanishes (due to a collision or the intended receiver being in the transmit mode), or it successfully reaches the root.

For illustration, consider a protocol called RoundRobin, where all nodes transmit in a cyclic order, one at a time. Specifically, at any step t, RoundRobin transmits from the node v with $\mathsf{label}(v) = t \bmod n$, with the transmitted message containing all rumors received by v until time t. The running time is $O(n^2)$, because initially each rumor ρ_u is at distance at most n from the root and in any consecutive n steps it will decrease its distance to the root by at least 1. (RoundRobin

has been used as a subroutine in many protocols in the literature, and it achieves running time $O(n^2)$ even for gossiping in arbitrary networks, not only for gathering in trees.)

RoundRobin can be adapted to use only bounded messages for information gathering in trees. At any round t and any node v, if v has the rumor ρ_u of node u such that $\mathsf{label}(u) = t \bmod n$, and v has not transmitted ρ_u before, then v transmits ρ_u at time t. Since $\mathcal{T}$ is a tree, each rumor follows the unique path towards the root. Therefore any two sibling nodes will never transmit at the same time, that is, collisions will never occur. This implies that after at most n^2 steps r will receive all rumors.

3 Some Structure Properties of Trees

The running times of our algorithms in Sections 4 and 5 depend on the distribution of high-degree nodes within a tree. To capture the structure of this distribution we use the concept of γ-depth which, roughly, measures how "bushy" the tree is. We define this concept in this section and establish its properties needed for the analysis of our algorithms.

γ-Depth of trees. Let $\mathcal{T}$ be the given tree network with root r and n nodes. Fix an integer γ in the range $2 \leq \gamma \leq n-1$. We define the *γ-height* of each node v of $\mathcal{T}$, denoted $height_\gamma(v)$, as follows. If v is a leaf then $height_\gamma(v) = 0$. If v is an internal node then let g be the maximum γ-height of a child of v. If at least γ children of v have γ-height equal g then $height_\gamma(v) = g + 1$; otherwise $height_\gamma(v) = g$. (For $\gamma = 2$, our definition of 2-height is equivalent to that of Strahler numbers. See, for example, [24, 26].) We then define the *γ-depth of $\mathcal{T}$* as $D_\gamma(\mathcal{T}) = height_\gamma(r)$.

In our proofs, we may also consider trees other than the input tree $\mathcal{T}$. If $\mathcal{H}$ is any tree and v is a node of $\mathcal{H}$ then, to avoid ambiguity, we will write $height_\gamma(v, \mathcal{H})$ for the γ-height of v with respect to $\mathcal{H}$. Note that if $\mathcal{H}$ is a subtree of $\mathcal{T}$ and $v \in \mathcal{H}$ then, trivially, $height_\gamma(v, \mathcal{H}) \leq height_\gamma(v)$.

By definition, the 1-height of a node is the same as its height, namely the longest distance from this node to a leaf in its subtree. For a tree, its 1-depth is equal to its depth. Figure 1 shows an example of a tree whose depth equals 4, 2-depth equals 3, and 3-depth equals 1.

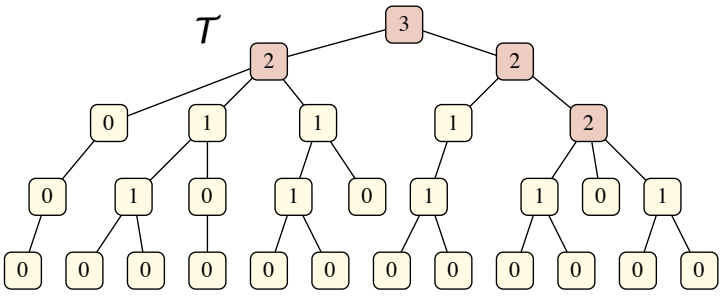

Figure 1: An example illustrating the concept of γ-depth of trees, for $\gamma = 1, 2, 3$. The depth of this tree $\mathcal{T}$ is 4. The number in each node is its 2-height; thus the 2-depth of this tree is 3. All light-shaded nodes have 3-height equal 0 and the four dark-shaded nodes have 3-height equal 1, so the 3-depth of this tree is 1.

The lemmas below spell out some simple properties of γ-heights of nodes that will be useful for the analysis of our algorithms. (In particular, Lemma 1 generalizes the bound from [26] for $\gamma = 2$.) If v is a node of $\mathcal{T}$ then $\mathcal{T}_v$ will denote the subtree of $\mathcal{T}$ rooted at v and containing all descendants of v.

Lemma 1. $D_\gamma(\mathcal{T}) \leq \log_\gamma n$.

Proof. It is sufficient to show that $|\mathcal{T}_v| \geq \gamma^{height_\gamma(v)}$ holds for each node v. The proof is by simple induction with respect to the height of v.

If v is a leaf then the inequality is trivial. So suppose now that v is an internal node with $height_\gamma(v) = g$. If v has a child u with $height_\gamma(u) = g$ then, by induction, $|\mathcal{T}_v| \geq |\mathcal{T}_u| \geq \gamma^g$. If all children of v have γ-height smaller than g then v must have at least γ children with γ-height equal $g - 1$. So, by induction, we get $|\mathcal{T}_v| \geq \gamma \cdot \gamma^{g-1} = \gamma^g$. □

We will be particularly interested in subtrees of $\mathcal{T}$ consisting of the nodes whose γ-height is above a given threshold. Specifically, for $h = 0, 1, ..., D_\gamma(\mathcal{T})$, let $\mathcal{T}^{\gamma,h}$ be the subtree of $\mathcal{T}$ induced by the nodes whose γ-height is at least h (see Figure 2). Note that, since γ-heights are monotonically non-decreasing on the paths from leaves to r, $\mathcal{T}^{\gamma,h}$ is indeed a subtree of $\mathcal{T}$ rooted at r. In particular, for $h = 0$ we have $\mathcal{T}^{\gamma,0} = \mathcal{T}$.

For any h, $\mathcal{T} - \mathcal{T}^{\gamma,h}$ is a collection of subtrees of type $\mathcal{T}_v$, where v is a node of γ-height less than h whose parent is in $\mathcal{T}^{\gamma,h}$. When $h = 1$, all such subtrees contain only nodes of γ-height equal 0, which implies that they all have degree less than γ. In particular, for $\gamma = 2$, each such subtree $\mathcal{T}_v$ is a path from a leaf of $\mathcal{T}$ to v.

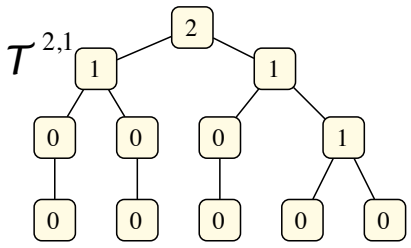

Figure 2: The subtree $\mathcal{T}^{2,1}$ obtained from tree $\mathcal{T}$ in Figure 1. The numbers in the nodes are their 2-heights with respect to $\mathcal{T}^{2,1}$.

Lemma 2. *For any node $v \in \mathcal{T}^{\gamma,h}$ we have $height_\gamma(v, \mathcal{T}^{\gamma,h}) = height_\gamma(v) - h$. Thus, in particular, we also have $D_\gamma(\mathcal{T}^{\gamma,h}) = D_\gamma(\mathcal{T}) - h$.*

Proof. Let $\mathcal{T}' = \mathcal{T}^{\gamma,h}$. The proof is by induction on the height of v in $\mathcal{T}'$. If v is a leaf of $\mathcal{T}'$ then $height_\gamma(v) \geq h$ and $height_\gamma(v, \mathcal{T}') = 0$, by definition. All children of v in $\mathcal{T}$ are outside $\mathcal{T}'$ so their γ-heights are at most $h - 1$. Therefore $height_\gamma(v) = h$ and thus the lemma holds for v.

Suppose now that v is not a leaf of $\mathcal{T}'$ and that the lemma holds for all children of v. This means that each child u of v in $\mathcal{T}$ is of one of two types: either $u \notin \mathcal{T}'$ (that is, $height_\gamma(u) \leq h - 1$), or $u \in \mathcal{T}'$ (that is, $height_\gamma(u) \geq h$) and $height_\gamma(u, \mathcal{T}') = height_\gamma(u) - h$.

Let $height_\gamma(v) = f \geq h$. If v has a child with γ-height equal f then there are fewer than γ such children. By induction, these children will have γ-height in $\mathcal{T}'$ equal $f - h$, and each other child that remains in $\mathcal{T}'$ has γ-height in $\mathcal{T}'$ smaller than $f - h$. So $height_\gamma(v, \mathcal{T}') = f - h$.

If all children of v have γ-height smaller than f then $f \geq h + 1$ (for otherwise v would have to be a leaf of $\mathcal{T}'$) and v must have at least γ children with γ-height $f - 1$. These children will be in $\mathcal{T}'$ and will have γ-height in $\mathcal{T}'$ equal $f - 1 - h$, by induction. So $height_\gamma(v, \mathcal{T}') = f - h$ in this case as well, completing the proof. □

From Lemma 2, we obtain that the operation of taking subtrees $\mathcal{T}^{\gamma,h}$ is, in a sense, transitive.

Corollary 1. *For any $g, h \geq 0$ such that $h + g \leq D_\gamma(\mathcal{T})$, we have $(\mathcal{T}^{\gamma,h})^{\gamma,g} = \mathcal{T}^{\gamma,h+g}$.*

4 Deterministic Algorithms with Aggregation

As explained in Section 2, it is easy to achieve information gathering in time $O(n^2)$. In this section we prove that using unbounded-size messages this running time can be improved to $O(n)$. This is optimal for arbitrary n-node trees, since no better time can be achieved even for paths or star graphs. We start with a simpler protocol with running time $O(n \log n)$, that we use to introduce our terminology and techniques.

We can make some assumptions about the protocols in this section that will simplify their presentation. Since we use unbounded messages, we can assume that each transmitted message contains all information received by the transmitting node, including all received rumors. We also assume that all rumors are different, so that each node can keep track of the number of rumors collected from its subtree. To ensure this we can, for example, have each node v append its label to its rumor ρ_v.

We will also assume that each node knows the labels of its children. To acquire this knowledge, we can precede any protocol by a preprocessing phase where nodes with labels $0, 1, ..., n-1$ transmit, one at a time, in this order. Thus after n steps each node will receive the messages from its children. This does not affect the asymptotic running times of our protocols.

4.1 Warmup: a Simple $O(n \log n)$-Time Algorithm

We now present an algorithm for information gathering on trees that runs in time $O(n \log n)$. In essence, any node waits until it receives the messages from its children, then for $2n$ steps it alternates RoundRobin steps with steps when it always attempts to transmit. A more detailed specification of the algorithm follows.

Algorithm UnbDTree1**.** We divide the time steps into *rounds*, where round s consists of two consecutive steps $2s$ and $2s + 1$, which we call, respectively, the RR-step and the All-step of round s.

For each node v we define its *activation round*, denoted α_v, as follows. If v is a leaf then $\alpha_v = 0$. For any other node v, α_v is the first round such that v has received messages from all its children when this round is about to start.

For each round $s = \alpha_v, \alpha_v + 1, ..., \alpha_v + n - 1$, v transmits in the All-step of round s, and if $\mathsf{label}(v) = s \bmod n$ then it also transmits in the RR-step of round s. In all other

steps, v stays in the receiving state.

Analysis. For any node v we say that v is *dormant* in rounds $0, 1, ..., \alpha_v - 1$, v is *active* in rounds $\alpha_v, \alpha_v + 1, ..., \alpha_v + n - 1$, and that v is *retired* in every round thereafter. Since v will make one RR-transmission when it is active, v will successfully transmit its message (and thus all rumors from its subtree $\mathcal{T}_v$) to its parent before retiring, and before this parent is activated. Therefore, by a simple inductive argument, Algorithm UnbDTree1 is correct, namely that eventually r will receive all rumors from $\mathcal{T}$.

This inductive argument shows in fact that, at any round, Algorithm UnbDTree1 satisfies the following two invariants: (i) any path in $\mathcal{T}$ from a leaf to r consists of a segment of retired nodes, followed by a segment of active nodes, which is then followed by a segment of dormant nodes (each of these segments possibly empty); and (ii) any dormant node has at least one active descendant.

Lemma 3. *Let $d = D_2(\mathcal{T})$. For any $h = 0, 1, ..., d$ and any node v with $height_2(v) = h$, v gets activated no later than in round $2nh$, that is $\alpha_v \leq 2nh$.*

Proof. The proof is by induction on h. By the algorithm, the lemma trivially holds for $h = 0$. Suppose that the lemma holds for $h - 1$ and consider a node v with $height_2(v) = h$. To reduce clutter, denote $\mathcal{Z} = \mathcal{T}^{2,h}$. From Lemma 2, we have that $height_2(v, \mathcal{Z}) = 0$, which implies that $\mathcal{Z}_v$ is a path from a leaf of $\mathcal{Z}$ to v. Let $\mathcal{Z}_v = v_1, v_2, ..., v_q$ be this path, where v_1 is a leaf of $\mathcal{Z}$ and $v_q = v$.

We now consider round $s = 2n(h-1) + n$. The nodes in $\mathcal{T} - \mathcal{Z}$ have 2-height at most $h - 1$, so, by the inductive assumption, they are activated no later than in round $2n(h-1)$, and therefore in round s they are already retired. If $\alpha_v \leq s$ then $\alpha_v \leq 2nh$, and we are done. Otherwise, v is dormant in round s. Then, by invariant (ii) above, at least one node in $\mathcal{Z}_v$ must be active. Choose the largest p for which v_p is active in round s. In round s and later, all children of the nodes $v_p, v_{p+1}, ..., v_q$ that are not on $\mathcal{Z}_v$ do not transmit, since they are already retired. This implies that for each $\ell = 0, ..., q - p - 1$, node $v_{p+\ell+1}$ will get activated in round $s + \ell + 1$ as a result of the All-transmission from node $v_{p+\ell}$. In particular, we obtain that $\alpha_v \leq s + q - p \leq 2nh$, completing the proof of the lemma. □

We have $height_2(r) = d$ and $d = O(\log n)$, by Lemma 1. Applying Lemma 3, this implies that $\alpha_r \leq 2nd = O(n \log n)$, which gives us that the overall running time is $O(n \log n)$. Summarizing, we obtain the following theorem.

Theorem 1. *For any tree with n nodes and any assignment of labels, Algorithm* UnbDTree1 *completes information gathering in time $O(n \log n)$.*

4.2 An $O(n)$-Time Deterministic Algorithm

In this section we show how to improve the running time of information gathering in trees to linear time, assuming unbounded size messages. The basic idea is to use strong k-selective families to speed up the computation.

Recall that a *strong k-selective family*, where $1 \leq k \leq n$, is a collection $F_0, F_1, ..., F_{m-1} \subseteq [n]$ of sets such that for any set $X \subseteq [n]$ with $|X| \leq k$ and any $x \in X$, there is j for which $F_j \cap X = \{x\}$. It is well known that for any $k = 1, 2, ..., n$, there is a strong k-selective family with $m = O(k^2 \log n)$ sets [13, 11]. Note that in the special case $k = 1$ the family consisting of just one set $F_0 = [n]$ is

1-selective. This corresponds to All-transmissions in the previous section. For $k = \omega(\sqrt{n/\log n})$ we can also improve the $O(k^2 \log n)$ bound to $O(n)$ by using the set family corresponding to the RoundRobin protocol, namely the n singleton sets $\{0\}, \{1\}, ..., \{n-1\}$.

In essence, a strong k-selective family can be used to speed up information dissemination through low-degree nodes. Consider the protocol k-Select that works as follows: for any step t and any node v, if $\mathsf{label}(v) \in F_{t \bmod n}$ then transmit from v, otherwise stay in the receive state. Suppose that w is a node with fewer than k children, and that these children collected the messages from their subtrees by time t. Steps $t, t+1, ..., t+m-1$ of this protocol use all sets $F_0, F_1, ..., F_{m-1}$ (although possibly in a different order), so each child of w will make a successful transmission by time $t+m$, which, for $k = o(\sqrt{n/\log n})$, is faster than time $O(n)$ required by RoundRobin.

To achieve linear time for arbitrary trees, we will interleave the steps of protocol k-Select with RoundRobin (to deal with high-degree nodes) and steps where all active nodes transmit (to deal with long paths).

Below, we fix parameters $\kappa = \lceil n^{1/3} \rceil$ and $m = O(\kappa^2 \log n)$, the size of a strong κ-selective family $F_0, F_1, ..., F_{m-1}$. (The choice of κ is somewhat arbitrary; in fact, any $\kappa = \Theta(n^c)$, for $0 < c < \frac{1}{2}$, would work.) Without loss of generality we can assume that $m \leq n$.

Algorithm UnbDTree2**.** We divide the steps into rounds, where each round s consists of three consecutive steps $3s$, $3s+1$, and $3s+2$, that we will call the RR-step, All-step, and Sel-step of round s, respectively.

For each node v we define its *activation round*, denoted α_v, as follows. If v is a leaf then $\alpha_v = 0$. For any other node v, α_v is the first round such that before this round starts v has received all messages from its children.

In each round $s = \alpha_v, \alpha_v + 1, ..., \alpha_v + m - 1$, v transmits in the All-step of round s, and if $\mathsf{label}(v) \in F_{s \bmod m}$ then v also transmits in the Sel-step of round s. In each round $s = \alpha_v, \alpha_v + 1, ..., \alpha_v + n - 1$, if $\mathsf{label}(v) = s \bmod n$ then v transmits in the RR-step of round s. If v does not transmit according to the above rules then v stays in the receiving state.

Analysis. Similar to Algorithm UnbDTree1, in Algorithm UnbDTree2 each node v goes through three stages. We call v *dormant* in rounds $0, 1, ..., \alpha_v - 1$, *active* in rounds $\alpha_v, \alpha_v + 1, ..., \alpha_v + n - 1$, and retired thereafter. We will also refer to v as being *semi-retired* in rounds $\alpha_v + m, \alpha_v + m + 1, ..., \alpha_v + n - 1$ (when it is still active, but only uses RR-transmissions). Assuming that v gets activated in some round, since v makes at least one RR-transmission when it is active, it will successfully transmit its message to its parent before retiring, and before its parent gets activated. By straightforward induction on the depth of $\mathcal{T}$, this implies that each node will eventually get activated, proving that Algorithm UnbDTree2 is correct.

By a similar argument, Algorithm UnbDTree2 satisfies the following two invariants in each round: (i) Any path from a leaf to r consists of a segment of retired nodes, followed by a segment of active nodes (among the active nodes, the semi-retired nodes precede those that are not semi-retired), which is then followed by a segment of dormant nodes. Each segment could be empty. (ii) Any dormant node has at least one active descendant.

It remains to show that the running time of Algorithm UnbDTree2 is $O(n)$. The idea of the analysis is to show that Sel- and All-steps disseminate information very fast, in linear time, through

subtrees where all node degrees are less than κ. (In fact, this applies also to nodes with higher degrees, as long as they have fewer than κ active children left.) The process can stall, however, if all active nodes have parents of degree larger than κ. In this case, a complete cycle of RoundRobin will transmit the messages from these nodes to their parents. We show, using Lemma 1, that, since $\kappa = \lceil n^{1/3} \rceil$, such stalling can occur at most the total of 3 times. So the overall running time will be still $O(n)$. We formalize this argument in the remainder of this sub-section.

Let $\bar{d} = D_\kappa(\mathcal{T})$. From Lemma 1, we have $\bar{d} \leq 3$. We fix some integer $g \in \{0, 1, 2, 3\}$, a node w with $height_\kappa(w) = g$, and we let $\mathcal{Y} = \mathcal{T}_w^{\kappa,g}$; that is, $\mathcal{Y}$ is the subtree of $\mathcal{T}^{\kappa,g}$ rooted at w. This subtree $\mathcal{Y}$ consists of the descendants of w (including w itself) whose κ-height in $\mathcal{T}$ is exactly g, or, equivalently (by Lemma 2), the descendants of w in $\mathcal{T}^{\kappa,g}$ whose κ-height in $\mathcal{T}^{\kappa,g}$ is equal 0. (See Figure 3 for illustration.) Therefore all nodes in $\mathcal{Y}$ have degrees, with respect to $\mathcal{Y}$, strictly smaller than κ.

We also fix $\bar{s}$ to be the first round when all nodes in $\mathcal{T} - \mathcal{T}^{\kappa,g}$ are active or already retired. In particular, for $g = 0$ we have $\bar{s} = 0$. Our goal now is to show that w will get activated in $O(n)$ rounds after round $\bar{s}$.

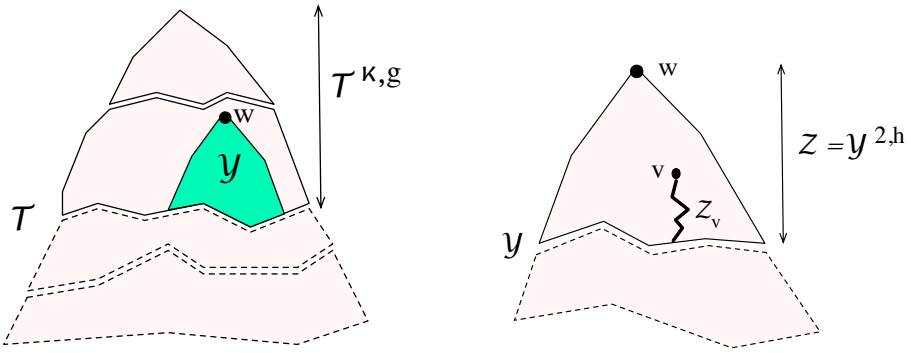

Figure 3: On the left, tree $\mathcal{T}$, partitioned into layers consisting of nodes with the same κ-height. (This figure should not be interpreted literally; for example, nodes of κ-height 3 may have children of κ-height 0 or 1.) The figure on the right shows subtree $\mathcal{Y}$ of $\mathcal{T}$, partitioned into $\mathcal{Z}$ and $\mathcal{Y} - \mathcal{Z}$. Within $\mathcal{Z}$, subtree $\mathcal{Z}_v$ is a path from v to a leaf.

Lemma 4. $\alpha_w \leq \bar{s} + O(n)$.

Proof. Let $d = D_2(\mathcal{Y})$. By Lemma 1, $d = O(\log |\mathcal{Y}|) = O(\log n)$. For $h = 0, ..., d$, let l_h be the number of nodes $u \in \mathcal{Y}$ with $height_2(u, \mathcal{Y}) = h$. The overall idea of the proof is similar to the analysis of Algorithm UnbDTree1. The difference is that now, since all degrees in $\mathcal{Y}$ are less than κ, the number of rounds required to advance through the h-th layer of $\mathcal{Y}$, consisting of nodes of 2-height equal h, can be bounded by $O(m + l_h)$, while before this bound was $O(n)$. Adding up the bounds for all layers, all terms $O(l_h)$ will now amortize to $O(n)$, and the terms $O(m)$ will add up to $O(md) = O(n^{2/3} \log^2 n) = O(n)$ as well. We now fill in the details.

Claim A: Let v be a node in $\mathcal{Y}$ with $height_2(v, \mathcal{Y}) = h$. Then the activation round of v satisfies $\alpha_v \leq s_h$, where $s_h = \bar{s} + 2n + \sum_{i \leq h} l_i + hm$.

First, we observe that Claim A implies the lemma. This is because for $v = w$ we get the bound $\alpha_w \leq \bar{s} + 2n + \sum_{i \leq d} l_i + dm \leq \bar{s} + 2n + n + O(\log n) \cdot O(n^{2/3} \log n) = \bar{s} + O(n)$, as needed.

Thus, to complete the proof, it remains to justify Claim A. We proceed by induction on h.

Consider first the base case, when $h = 0$. We focus on the computation in the subtree $\mathcal{Y}_v$, which (for $h = 0$) is simply a path $v_1, v_2, ..., v_q = v$, from a leaf v_1 of $\mathcal{Y}$ to v. In round $\bar{s} + n$ all the nodes in $\mathcal{T} - \mathcal{Y}$ must be already retired. If v is active or retired in round $\bar{s} + n$, we are done, because $\bar{s} + n \leq s_0$. If v is dormant, at least one node in $\mathcal{Y}_v$ must be active, because of the invariant (ii) and the fact that all nodes in $\mathcal{T} - \mathcal{Y}$ are already retired. So choose p to be the maximum index for which v_p is active. Since we have no interference from outside $\mathcal{Y}_v$, using a simple inductive argument, v will be activated in $q - p$ rounds using All-transmissions. Also, $q - p \leq l_0$, and therefore $\alpha_v \leq \bar{s} + n + q - p \leq \bar{s} + 2n + l_0 = s_0$, which is the bound from Claim A for $h = 0$.

In the inductive step, fix some $h > 0$, and assume that Claim A holds for $h - 1$. Denoting $\mathcal{Z} = \mathcal{Y}^{2,h}$, we consider the computation in $\mathcal{Z}_v$, the subtree of $\mathcal{Z}$ rooted at v. (See Figure 3.) $\mathcal{Z}_v$ is a path $v_1, v_2, ..., v_q = v$ from a leaf v_1 of $\mathcal{Z}$ to v.

The argument is similar to the base case. There are two twists, however. One, we need to show that v_1 will get activated no later than at time $s_{h-1} + m$; that is, after delay of only m, not $O(n)$. Two, the children of the nodes on $\mathcal{Z}_v$ that are not on $\mathcal{Z}_v$ are not guaranteed to be retired anymore. However, they are semi-retired, which is good enough for our purpose.

Consider v_1. We want to show first that v_1 will get activated no later than at time $s_{h-1} + m$. All children of v_1 can be grouped into three types. The first type consists of the children of v_1 in $\mathcal{T} - \mathcal{Y}$. These are activated no later than in round $\bar{s}$, so they are retired no later than in round $\bar{s} + n$. All other children of v_1 are in $\mathcal{Y} - \mathcal{Z}$. Among those, the type-2 children are those that were activated before round $\bar{s} + n$, and the type-3 children are those that were activated at or after round $\bar{s} + n$. Clearly, v_1 will receive the messages from its children of type 1 and 2, using RR-transmissions, before round $\bar{s} + 2n \leq s_{h-1} + m$.

The children of v_1 of type 3 activate no earlier than in round $\bar{s} + n$. Also, since they are in $\mathcal{Y} - \mathcal{Z}$, their 2-height in $\mathcal{Y}$ is strictly less than h, so they activate no later than in round s_{h-1}, by induction. (Note that $s_{h-1} \geq \bar{s} + 2n$.) Thus each child u of v_1 in $\mathcal{Y}$ of type 3 will complete all its Sel-transmissions, that include the complete κ-selector, between rounds $\bar{s} + n$ and $s_{h-1} + m - 1$ (inclusive). In these rounds all children of v_1 that are not in $\mathcal{Y}$ are retired, so fewer than κ children of v_1 are active in these rounds. This implies that the message of u will be received by v_1 using a Sel-transmissions, if no other. Putting it all together, v_1 will receive messages from all its children before round $s_{h-1} + m$, and thus it will be activated no later than in round $s_{h-1} + m$.

From the paragraph above, we obtain that in round $s_{h-1} + m$ either there is an active node in $\mathcal{Z}_v$ or all nodes in $\mathcal{Z}_v$ are already retired. The remainder of the argument is similar to the base case. If v itself is active or retired in round $s_{h-1} + m$ then we are done, because $s_{h-1} + m \leq s_h$. So suppose that v is still dormant in round $s_{h-1} + m$. Choose p to be the largest index for which v_p is active in this round. At round $s_{h-1} + m$ and later, all children of the nodes on $\mathcal{Z}_v$ that are not on $\mathcal{Z}_v$ are either retired or semi-retired. Therefore, since there is no interference, v will get activated in $q - p$ additional rounds using All-transmissions. So $\alpha_v \leq s_{h-1} + m + q - p \leq s_{h-1} + m + l_h = s_h$, completing the inductive step, the proof of Claim A, and the lemma. □

From Lemma 4, all nodes in $\mathcal{T}$ with κ-height equal 0 will get activated in at most $O(n)$ rounds. For $g = 1, 2, 3$, all nodes with κ-height equal g will activate no later than $O(n)$ rounds after the last node with κ-height less than g is activated. This implies that all nodes in $\mathcal{T}$ will be activated within $O(n)$ rounds. Summarizing, we obtain the main result of this section.

Theorem 2. *For any tree with n nodes and any assignment of labels, Algorithm* UnbDTree2 *completes information gathering in time $O(n)$.*

Algorithm UnbDTree2 can be modified to work if we allow labels to be from the range $[L]$, for some $L \geq n$, assuming that all nodes know the value of L. There are two ways to do that. If L is not too large, say $L = O(n)$, then in the algorithm we simply replace n by L. The running time will be still $O(n)$. If L is large, then we can replace the RR-step by an n-selector for the range $[L]$, of size $O(n^2 \log L)$. This will give us an algorithm with running time $O(n^2 \log L)$. Combining these two approaches gives running time $O(\min(L, n^2 \log L))$.

5 Deterministic Algorithms without Aggregation

In this section we consider deterministic information gathering without aggregation, where each message can contain at most one rumor, plus additional $O(\log n)$ bits of information. In this model, we give an algorithm with running time $O(n \log n)$.

To simplify the description of the algorithm, we will temporarily assume that we are allowed to receive and transmit at the same time. Later, we will show how to remove this assumption.

Algorithm BndDTree**.** First, we do some preprocessing, using a modification of Algorithm UnbDTree2 to compute the 2-height of each node v. In this modified algorithm, the message from each node contains its 2-height. When v receives such messages from its children, it can compute its own 2-height, which it can then transmit to its parent.

Let $\ell = \lceil \log n \rceil$. We divide the computation into $\ell+1$ phases. Phase h, for $h = 0, 1, ..., \ell$, consists of steps $3nh, 3nh+1, ..., 3n(h+1)-1$. In phase h, only the nodes of 2-height equal h participate in the computation. Specifically, consider a node v with $height_2(v) = h$. We have two stages:

Stage All: In each step $t = 3nh, 3nh+1, ..., 3nh+2n-1$, if v contains any rumor ρ_u that it still has not transmitted, v transmits ρ_u.

Stage RR: In each step $t = 3nh + 2n + u$, for $u = 0, 1, ..., n-1$, if v has rumor ρ_u, then v transmits ρ_u.

In any other step, v is in the receiving state.

Analysis. By Lemma 1, each node has 2-height at most ℓ, so all nodes will participate in the computation and the whole algorithm will complete computation in $O(n \log n)$ steps. It remains to show that r will receive all rumors.

We claim that, at the beginning of phase h, every node v has rumors from all its descendants in $\mathcal{T} - \mathcal{T}^{2,h}$, namely the descendants whose 2-height is strictly smaller than h. (In particular, if $height_2(v) < h$ then v has all rumors from T_v.) This is trivially true at the beginning, when $h = 0$.

Assume that the claim holds for some $h < \ell$, and consider phase h. We want to show that when phase h ends then each node v has rumors from all descendants in $\mathcal{T} - \mathcal{T}^{2,h+1}$. By the inductive assumption, v has all rumors from its descendants in $\mathcal{T} - \mathcal{T}^{2,h}$. So if v does not have any descendants of height h (in particular, if $height_2(v) \leq h-1$) then we are done.

Let v be a node with $height_2(v) \geq h$. It remains to prove that if v has a child u with $height_2(u) = h$ then right after phase h all rumors from $\mathcal{T}_u$ will also be in v.

The subtree $\mathcal{T}_u^{2,h}$, namely the subtree consisting of the descendants of u with 2-height equal h, is a path $u_1, u_2, ..., u_q = u$, where u_1 is a leaf of $\mathcal{T}^{2,h}$. When phase h starts all rumors from $\mathcal{T}_u$ are in $\mathcal{T}^{2,h}$. We show that, thanks to pipelining, all rumors that are in $\mathcal{T}^{2,h}$ when phase h starts will reach u during Stage All.

In phase h, for each node in $\mathcal{T}^{2,h}$, all children of this node, except possibly the unique child that is also in $\mathcal{T}^{2,h}$, do not transmit. For any step $3nh + s$, $s = 0, 1, ..., 2n - 1$, and for $i = 1, 2, ..., q - 1$, we define $\phi_{s,i}$ to be the number of rumors in u_i that are still not in u_{i+1}, and we let $\Phi_s = \sum_{i=a_s}^{q-1} \max(\phi_{s,i}, 1)$, where a_s is the smallest index for which $\phi_{s,a_s} \neq 0$. We claim that as long as $\Phi_s > 0$, its value will decrease in step $3n + s$. Indeed, for $i < q$, each node v_i with $\phi_{s,i} > 0$ will transmit a new rumor to v_{i+1}. Since $\phi_{s,i} = 0$ for $i < a_s$, node u_{a_s} will not receive any new rumors. We have $\phi_{s,a_s} > 0$, by the choice of a_s. If $\phi_{s,a_s} > 1$ then $\max(\phi_{s,a_s}, 1)$ will decrease by 1. If $\phi_{s,a_s} = 1$ then the index a_s itself will increase. In either case, u_{a_s}'s contribution to Φ_s will decrease by 1. For $i > a_s$, even if u_i receives a new rumor from u_{i-1}, the term $\max(\phi_{s,i}, 1)$ cannot increase, because if $\phi_{s,i} > 0$ then u_i transmits a new rumor to u_{i+1}, and if $\phi_{s,i} = 0$ then this term is 1 anyway. Therefore, overall, Φ_s will decrease by at least 1.

We have $\Phi_0 \leq q + n$, because each rumor contributes to at most one term in Φ_0. Since Φ_s strictly decreases in each step, Φ_s will become 0 in at most $2n$ steps. In other words, in $2n$ steps u will receive all rumors that were in $\mathcal{T}^{2,h}$ when phase h started, and thus all rumors from $\mathcal{T}_u$.

In Stage RR, u will transmit all collected rumors to v, without collisions. As a result, at the beginning of the next phase v will contain all rumors from $\mathcal{T}_u$, completing the proof of the inductive step.

We still need to explain how to modify Algorithm BndDTree to eliminate the assumption that we can transmit and receive at the same time. This can be accomplished by adding $3n$ more steps to each phase, as follows:

- We first add a new stage at the beginning of each phase, consisting of n steps. Note that each node v with $height_2(v) = h$ knows h and also it knows whether it has a child of 2-height h. If v does not have such a child, then v is the initial node of a path consisting of nodes of 2-height equal h. At the very beginning of phase h, v sends a message along this path, so that any node on this path can determine whether it is an even or odd node along this path.

- Then we double the number of steps Stage All, increasing its length from $2n$ to $4n$. In this stage, among the nodes with 2-height h, "even" nodes will transmit in even steps, and "odd" nodes will transmit in odd steps. In this way, each node will never receive and transmit at the same time.

We thus obtain the main result of this section:

Theorem 3. *For any tree with n nodes and any assignment of labels, Algorithm* BndDTree *completes information gathering in time $O(n \log n)$.*

Similar to Algorithm UnbDTree2, we can adapt Algorithm BndDTree to the model where labels are from $[L]$, for some $L \geq n$. The running time will be $O(\min(L \log L, n^2 \log n \log L))$. In particular, for $L = O(n)$, the running time will remain $O(n \log n)$.

6 Deterministic Fire-and-Forward Protocols

We now consider a simple type of protocols that we call *fire-and-forward* protocols. As defined in Section 2, in such protocols each message consists of just one rumor and no other information. They also satisfy the following condition: for any node v, at any time t, if v received a message at step $t-1$ then v must transmit the same message at time t. Otherwise, v can either transmit its own rumor or be in the receiving state. The idea behind fire-and-forward protocols is that all messages travel towards the root without any delay. They may vanish along the way, either due to a collision or being refused by the a node that is in the transmitting state.

Later in Section 7 we will show that there exists a randomized fire-and-forward protocol that accomplishes information gathering in time $O(n \log n)$. This raises the question whether this running time can be achieved by a deterministic fire-and-forward protocol. (As before, in the deterministic case we assume that the nodes are labelled $0, 1, ..., n-1$.) There is a trivial deterministic fire-and-forward protocol with running time $O(n^2)$: release all rumors one at a time, spaced at intervals of length n. In this section we show that this can be improved to $O(n^{1.5})$ and that this bound is optimal.

It is convenient to formulate our arguments in terms of a yet simpler and mathematically more elegant model of *basic fire-and-forward* protocols, that we define shortly. We will refer to the above definition as the *standard fire-and-forward* model. After proving our upper and lower bounds for the basic model, we show later how to adapt our proofs to the standard model.

6.1 Basic Fire-and-Forward (BFF) Protocols

In the *basic fire-and-forward (BFF)* model, we allow nodes to receive and transmit at the same time. Each message consists of one rumor and no other information. If a node v receives a message at time $t-1$, v must *forward* (that is, transmit) the same message at time t. Any any time (including the times when v forwards a message) v can also *fire*, that is transmit its own rumor. Note that this definition allows v to forward and fire at the same time, in which case a collision occurs and the parent of v does not receive any message. One can think of each node as having two independent modules: the forwarding module that unconditionally forwards each received message, and the firing module that transmits v's rumor at certain times.

If rumors fired from two nodes collide at all, they will collide at their lowest common ancestor. This happens only when the difference in times between these two firings is equal to the difference of their depths in the tree. More precisely, let $\mathcal{T}$ be the tree on input, denote by $depth(v)$ the depth of a node v in $\mathcal{T}$, and suppose that some node v fires its rumor at time t. Then this rumor will reach the root if no other node u fires at time $t + depth(v) - depth(u)$.

The BFF protocol we develop below is *oblivious*, in the sense that the decision whether to fire or not depends only on the label of the node and the current time. Such protocols are defined by specifying, for each node v, the set F_v of times when v fires.

In fact, it is easy to show that any protocol can be turned into an oblivious one without affecting its asymptotic running time. The idea is that leaves of the tree receive no information at all during the computation. For any BFF protocol $\mathcal{A}$ that runs in time $f(n)$, and for any tree $\mathcal{T}$, imagine that we run this protocol on the tree $\mathcal{T}'$ obtained by adding a leaf to any node v and giving it the label of v. Label the original nodes with the remaining labels. This at most doubles the number of nodes, so $\mathcal{A}$ will complete in time $O(f(n))$ on $\mathcal{T}'$. (We tacitly assume here that $f(cn) = \Theta(n)$ for any constant c, which is true for the time bounds we consider.) In the execution of $\mathcal{A}$ on $\mathcal{T}'$

the leaves receive no information and all rumors from the leaves will reach the root. This implies that if we apply $\mathcal{A}$ on $\mathcal{T}$ and ignore all information received during the computation, the rumors will also reach the root. In other words, after this modification, we obtain an oblivious protocol $\mathcal{A}'$ with running time $O(f(n))$.

6.2 An $O(n^{1.5})$ Upper Bound in the BFF Model

We now present our $O(n^{1.5})$-time fire-and-forward protocol for the basic model. As explained earlier, this protocol should specify a set of firing times for each label, so that for any mapping $[n] \to [n]$, that maps each label to the depth of the node with this label, each node will have at least one firing time for which there will not be a collision along the path to the root. We want each of these firing times to be at most $O(n^{1.5})$. To this end, we will partition all labels into batches, each of size roughly $\sqrt{n}$, and show that for any batch we can define such collision-avoiding firing times from an interval of length $O(n)$. Since we have about $\sqrt{n}$ batches, this will give us running time $O(n^{1.5})$.

Our construction of firing times is based on a concept of dispersers, defined below, which are reminiscent of various *rulers* studied in number theory, including Sidon sequences [27], Golomb rulers [28], or sparse rulers [29]. The particular construction we give in the paper is, in a sense, a multiple set extension of a Sidon-set construction by Erdös and Turán [14].

We now give the details. For $z \in \mathbb{Z}$ and $X \subseteq \mathbb{Z}$, let $X + z = \{x + z : x \in X\}$. Let also s be a positive integer. A set family $D_1, ..., D_m \subseteq [s]$ is called an (n, m, s)-*disperser* if for each function $\delta : \{1, ..., m\} \to [n]$ and each $j \in \{1, ..., m\}$ we have

$$D_j + \delta(j) \not\subseteq \bigcup_{i \in \{1,...,m\} - \{j\}} (D_i + \delta(i)).$$

The intuition is that D_j represents the set of firing times of node j and $\delta(j)$ represents j's depth in the tree. Then the disperser condition says that, for each depth function δ, some firing in D_j will not collide with firings of other nodes.

Lemma 5. *There exists an (n, m, s)-disperser with $m = \Omega(\sqrt{n})$ and $s = O(n)$.*

Proof. Let p be the smallest prime such that $p^2 \geq n$. For each $a = 1, 2, ..., p - 1$ and $x \in [p]$ define

$$d_a(x) = (ax \bmod p) + 2p \cdot (ax^2 \bmod p).$$

We claim that for any $a \neq b$ and any $t \in \mathbb{Z}$ the equation $d_a(x) - d_b(y) = t$ has at most two solutions $(x, y) \in [p]^2$. For the proof, fix a, b, t and one solution $(x, y) \in [p]^2$. Suppose that $(u, v) \in [p]^2$ is a different solution. Thus we have $d_a(x) - d_b(y) = d_a(u) - d_b(v)$. After substituting and rearranging, this can be written as

$$\begin{aligned}(ax \bmod p) - (by \bmod p) - (au \bmod p) + (bv \bmod p)& \\ = 2p[\,-(ax^2 \bmod p) + (by^2 \bmod p) + (au^2 \bmod p) - (bv^2 \bmod p)\,].&\end{aligned}$$

The expression on the left-hand side is strictly between $-2p$ and $2p$, so both sides must be equal 0. This implies that

$$ax - au \equiv by - bv \pmod p \quad \text{and} \tag{1}$$

$$ax^2 - au^2 \equiv by^2 - bv^2 \pmod p. \tag{2}$$

From equation (1), the assumption that $(x, y) \neq (u, v)$ implies that $x \neq u$ and $y \neq v$. We can then divide the two equations, getting

$$x + u \equiv y + v \pmod{p}. \tag{3}$$

With addition and multiplication modulo p, $\mathbb{Z}_p$ is a field. Therefore for any x and y, and any $a \neq b$, equations (1) and (3) uniquely determine u and v, completing the proof of the claim.

Now, let $m = (p-1)/2$ and $s = 2p^2 + p$. By Bertrand's postulate we have $\sqrt{n} \leq p < 2\sqrt{n}$, which implies that $m = \Omega(\sqrt{n})$ and $s = O(n)$. For each $i = 1, 2, ..., m$, define $D_i = \{d_i(x) : x \in [p]\}$. It is sufficient to show that the sets $D_1, D_2, ..., D_m$ satisfy the condition of the (n, m, s)-disperser.

The definition of the sets D_i implies that $D_i \subseteq [s]$ for each i. Fix some some δ and j from the definition of dispersers. It remains to verify that $D_j + \delta(j) \not\subseteq \bigcup_{i \neq j}(D_i + \delta(i))$. For $x \in [p]$ and $i \in \{1, 2, ..., m\}$, we say that i *kills* x if $d_j(x) + \delta(j) \in D_i + \delta(i)$. Our earlier claim implies that any $i \neq j$ kills at most two values in $[p]$. Thus all indices $i \neq j$ kill at most $2(m-1) = p - 3$ integers in $[p]$, which implies that there is some $x \in [p]$ that is not killed by any i. For this x, we will have $d_j(x) + \delta(j) \notin \bigcup_{i \neq j}(D_i + \delta(i))$, completing the proof that $D_1, ..., D_m$ is indeed an (n, m, s)-disperser. □

We now describe our algorithm.

Algorithm BFFDTree**.** Let $D_1, D_2, ..., D_m$ be the (n, m, s)-disperser from Lemma 5. We partition all labels (and thus also the corresponding nodes) arbitrarily into batches $B_1, B_2, ..., B_l$, for $l = \lceil n/m \rceil$, with each batch B_i having m nodes (except the last batch, that could be smaller). Order the nodes in each batch arbitrarily, for example according to increasing labels.

The algorithm has l phases. Each phase $q = 1, 2, ..., l$ consists of $s' = s + n$ steps in the time interval $[s'(q-1), s'q - 1]$. In phase q, the algorithm transmits rumors from batch B_q, by having the j-th node in B_q fire at each time $s'(q-1) + \tau$, for $\tau \in D_j$. Note that in the last n steps of each phase none of the nodes fires.

Analysis. We now show that Algorithm BFFDTree correctly performs gathering in any n-node tree in time $O(n^{1.5})$. Since $m = \Omega(\sqrt{n})$, we have $l = O(\sqrt{n})$. Also, $s' = O(n)$, so the total run time of the protocol is $O(n^{1.5})$.

It remains to show that during each phase q each node in B_q will have at least one firing that will send its rumor to the root r without collisions. Fix some tree $\mathcal{T}$ and let $\delta(j) \in [n]$ be the depth of the jth node in batch B_q. For any batch B_q and any $v \in B_q$, if v is the jth node in B_q then v will fire at times $s'(q-1) + \tau$, for $\tau \in D_j$. From the definition of dispersers, there is $\tau \in D_j$ such that $\tau + \delta(j) - \delta(i) \notin D_i$ for each $i \neq j$. This means that the firing of v at time $s'(q-1) + \tau$ will not collide with any firing of other nodes in batch B_q. Since the batches are separated by empty intervals of length n, this firing will not collide with any firing in other batches. So v's rumor will reach r.

Summarizing, we obtain our main result of this section.

Theorem 4. *There is a BFF protocol for information gathering in trees with running time $O(n^{1.5})$.*

6.3 An $\Omega(n^{1.5})$ Lower Bound in the BFF Model

In this section we show that any basic fire-and-forward protocol needs time $\Omega(n^{1.5})$ to deliver all rumors to the root for an arbitrary tree. Fix some fire-and-forward protocol $\mathcal{A}$. Without loss of generality, as explained earlier in this section, we can assume that $\mathcal{A}$ is oblivious, namely that the set F_v of firing times of each node v is uniquely determined by its label.

Let T be the running time of $\mathcal{A}$. We first give the proof under the assumption that the total number of firings of $\mathcal{A}$, among all nodes in $[n]$, is at most T, that is $\sum_{v\in[n]} |F_v| \leq T$. Later we will show how to extend our argument to protocols with an arbitrary number of firings.

We will show that if $T = o(n^{1.5})$ then $\mathcal{A}$ will fail even on a "caterpillar" tree, consisting of a path $\mathcal{P}$ of length n with n leaves attached to the nodes of this path. (For convenience we use $2n$ nodes instead of n, but this does not affect the asymptotic lower bound.) This path $\mathcal{P}$ is fixed, and the label assignment to these nodes is not important for the proof, but, for the ease of reference, we will label them $n, n+1, ..., 2n-1$, in order in which they appear on the path, with node labeled $2n-1$ being the root. The leaves have labels from the set $[n] = \{0, 1, ..., n-1\}$. To simplify the argument we will identify the labels with nodes, and we will refer to the node with label l simply as "node l". (See Figure 4.) The objective is to show that there is a way to attach the nodes from $[n]$ to $\mathcal{P}$ to make $\mathcal{A}$ fail, which means that there is at least one node w whose all firings will collide with firings from other nodes.

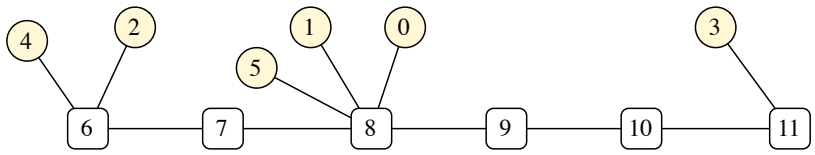

Figure 4: A caterpillar graph from the proof, for $n = 6$. The nodes on the path $\mathcal{P}$ are represented by rectangles, and the leaves are represented by circles. In this example, the root is 11 and $w = 4$.

Without loss of generality, assume that T is a multiple of n, and we let $k = T/n$. We divide the time range $0, 1, ..., T-1$ into k bins of size n, where the ith bin is the interval $[in, (i+1)n-1]$, for $i = 0, 1, ..., k-1$. If a node $v \in [n]$ fires at time t, we say that a node $u \in [n] - \{v\}$ *covers* this firing if u has a firing at time t' such that $t' > t$ and t, t' are in the same bin. For a set $L \subseteq F_v$ of firings of v, denote by $C(L)$ the set of nodes that cover the firings in L.

Lemma 6. *For each node $v \in [n]$ there is a set of firings $L_v \subseteq F_v$ such that $|C(L_v)| < |L_v|$.*

Proof. The proof of the lemma is by contradiction. Suppose that there exists a node $w \in [n]$ with the property that for each $L \subseteq F_w$ we have $|C(L)| \geq |L|$. Then Hall's theorem would imply that there is a perfect matching between the firing times in F_w and the nodes in $[n] - \{w\}$ that cover these firings. Let the firing times of w be $F_w = \{t_1, t_2, \ldots, t_j\}$, and for each $i = 1, 2, ..., j$, let u_i be the node matched to t_i in this matching. By the definitions of bins and covering, each u_i fires at some time $t_i + s_i$, where $0 \leq s_i \leq n-1$. We can then construct a caterpillar tree by attaching w to node n and attaching each u_i to node $n + s_i$ on $\mathcal{P}$. In this caterpillar tree, the firing of w at each time t_i will collide with the firing of u_i at time $t_i + s_i$. So the rumor from w will not reach the root, contradicting the correctness of $\mathcal{A}$. This completes the proof of the lemma. □

For each $v \in [n]$, we now fix a set L_v from Lemma 6. Let A be the set of ordered pairs (u, v) of different nodes $u, v \in [n]$ for which there is a bin which contains a firing from L_u and a firing from L_v, in this order in time.

We will bound $|A|$ in two different ways. On the one hand, each $u \in [n]$ appears as the first element in at most $|C(L_u)|$ pairs in A. Adding up over all u, using Lemma 6, and the assumption that the total number of firings is at most T, we get $|A| \leq \sum_u |C(L_u)| < \sum_u |L_u| \leq T$.

On the other hand, we can also establish a lower bound on $|A|$, as follows. Choose a specific representative firing t_v from each L_v. For each bin i, let n_i be the number of representatives in the ith bin. Any two representatives in bin i contribute one pair to A. So $|A| \geq q$, for $q = \frac{1}{2}\sum_{i=1}^{k} n_i(n_i-1)$. We can assume that n is sufficiently large and $k = o(n)$. Then, since $\sum_{i=1}^{k} n_i = n$, if we let all n_i take real values then the value of q will be minimized when all n_i are equal n/k. This implies that $q \geq cn^2/k$, for some $c > 0$. Thus we get that $|A| \geq cn^2/k = cn^3/T$.

Combining the bounds from the last two paragraphs, we obtain that $T \geq |A| \geq cn^3/T$, which implies that $T = \Omega(n^{1.5})$. This completes the proof of the lower bound, with the assumption that the total number of firings does not exceed T.

We now consider the general case, without any assumption on the total number of firings. Suppose that $\mathcal{A}$ is some protocol with running time $T = o(n^{1.5})$. Using a probabilistic argument and a reduction to the above special case, we show that then we can construct a caterpillar tree and a node w for which $\mathcal{A}$ will fail.

Choose some $\tilde{n}$ such that $\tilde{n} = o(n)$ and $T = o(\tilde{n}^{1.5})$. (For example, we can take $\tilde{n} = n^{1/2}T^{1/3}$.) Let $\tilde{V}$ be a random set of nodes, where each node is included in $\tilde{V}$ independently with probability $\tilde{n}/n$. Let also $\tilde{Z}$ be the set of times at which only nodes from $\tilde{V}$ fire in $\mathcal{A}$. We claim that with probability $1 - o(1)$, the following two properties will hold:

(i) $T \leq |\tilde{V}|^{1.5}$, and

(ii) The total number of firings in $\tilde{Z}$ is at most T.

To justify this claim, note that the probability that (i) is false is $\mathbf{P}[|\tilde{V}| < T^{2/3}] = o(1)$, because $T^{2/3} = o(\tilde{n})$ and the expectation of $\tilde{V}$ is $\tilde{n}$. Next, we consider (ii). If t is a time where $\mathcal{A}$ has $j \geq 1$ firings, then the probability that $t \in \tilde{Z}$ is $(\tilde{n}/n)^j$, so the contribution of t to the expected number of firings in $\tilde{Z}$ is $j(\tilde{n}/n)^j = o(1)$. Therefore the probability that (ii) is violated is $o(1)$.

We can thus conclude that for some choice of $\tilde{V}$ both properties (i) and (ii) hold. Let $\bar{V}$ be this choice, $\bar{n} = |\bar{V}|$, and let $\bar{Z}$ be the corresponding set of times $\tilde{Z}$.

We convert $\mathcal{A}$ into a protocol $\mathcal{B}$ for labels in $\bar{V}$, as follows. (Note that $\bar{V}$ may not be of the form $[\bar{n}]$, but that does not affect the validity of our argument.) For each time $t = 0, 1, ..., T-1$, if $t \in \bar{Z}$ then the firings in $\mathcal{B}$ are the same as in $\mathcal{A}$; if $t \notin \bar{Z}$ then no node in $\bar{V}$ fires. By our earlier argument, there must be a caterpillar tree $\mathcal{T}$ with nodes from $\bar{V}$ and a node $w \in \bar{V}$ for which $\mathcal{B}$ fails.

Let $\mathcal{T}'$ be a modified tree obtained from $\mathcal{T}$ by adding all nodes that are not in $\bar{V}$ as children of the parent of w. Consider now a firing of w in $\mathcal{T}'$ at time t. If $t \in \bar{Z}$, then this firing collides with another firing from $\bar{V}$. If $t \notin \bar{Z}$, then, by the definition of $\bar{Z}$, there is a node u outside of $\bar{V}$ that fires at the same time (otherwise t would be included in $\bar{Z}$) and is a sibling of w in $\mathcal{T}'$. So again, this firing from w will collide with the firing of u. We can thus conclude that the rumor from w will not reach the root, completing the proof of the lower bound.

We thus obtain our lower bound.

Theorem 5. *If $\mathcal{A}$ is a deterministic BFF protocol for information gathering in trees, then the running time of $\mathcal{A}$ is $\Omega(n^{1.5})$.*

6.4 Extension to the Standard Model

It remains to explain how we can extend this result to the standard model, where nodes are not allowed to receive and transmit at the same time. The lower bound from the previous section applies to the standard model without any changes. This is because, for the caterpillar tree $\mathcal{T}$ constructed in this proof, and for any collection of firing time sets $\{F_v\}_{v\in[n]}$ for the leaves, if two messages collide in the basic model then they also collide in the standard model, no matter how the nodes along the main path behave.

For the upper bound, we argue that our Algorithm BFFDTree from Section 6.2 can be adapted to the standard model. We only give a sketch of the argument. The idea is that if some node w transmits a rumor ρ_v and receives a rumor ρ_u, and if v fired at time t, then v fired at time $t + depth(v) - depth(u) + 1$. We can then extend the definition of collisions to include this situation. Incorporating this into the construction from Lemma 5, any index i may now kill more than two x's, but not more than four. So taking $m = \lfloor (p-1)/4 \rfloor$, we still will always have an x that is not killed by any i. By using such modified concept of disperses, we obtain an $O(n^{1.5})$-time protocol in the standard fire-and-forward model.

Theorem 6. (i) *There is a deterministic fire-and-forward protocol for information gathering in trees with running time $O(n^{1.5})$.* (ii) *Any deterministic fire-and-forward protocol for information gathering in trees has running time $\Omega(n^{1.5})$.*

7 An $O(n \log n)$-Time Randomized Algorithm

We now show a randomized algorithm with expected running time $O(n \log n)$ that does not use any labels. Our algorithm is a fire-and-forward algorithm, as defined in Sections 2 and 6. Each message consists only of one rumor and no additional information, and if a message successfully reaches a node v then v forwards this message in the next step, and it cannot transmit it at any time later.

In the description of the algorithm we assume that the number n of nodes is known. Using a standard doubling trick, the algorithm can be extended to one that does not depend on n. (This new algorithm will complete the task in expected time $O(n \log n)$, but it will keep running forever.)

In our presentation, we first specify our algorithm in the basic fire-and-forward (BFF) model; that is, we allow nodes to receive and transmit at the same time. Thus we only need to specify the set of firing times for each node. We explain later how to eliminate this feature. Recall that r is the root of the input tree $\mathcal{T}$. The algorithm is extremely simple:

Algorithm RTree**.** Any node $v \neq r$, at each step t, fires with probability $1/n$, independently of other nodes.

Analysis. We start with the following lemma.

Lemma 7. *At each step $t \geq n$, for each node $z \neq r$, the probability that r receives rumor ρ_z at step t is at least $\frac{1}{n}(1-\frac{1}{n})^{n-1}$. Further, for different t, the events of r receiving ρ_z are independent.*

Proof. To prove the lemma, it helps to view the computation in a slightly different, but equivalent way. Imagine that we ignore collisions, and we allow each message to consist of a set of rumors. If some children of v transmit at a time t, then v receives all rumors in the transmitted messages, and at time $t+1$ it transmits a message containing all these rumors, possibly adding its own rumor, if v also fires at that step. We will refer to these messages as *virtual messages*. In this interpretation, if r receives a virtual message that is a singleton set $\{\rho_z\}$, at some time t, then in Algorithm RTree this rumor ρ_z will be received by r at time t. (Note that the converse is not true, because it may happen that r will receive a virtual message that has other rumors besides ρ_z, but in Algorithm RTree these other rumors may get stopped earlier due to collisions, so they will not collide with ρ_z.)

Fix a time t and some $z \in \mathcal{T} - \{r\}$. By the above paragraph, it is sufficient to show that the probability that at time t the virtual message reaching r is the singleton $\{z\}$ is equal $\frac{1}{n}(1-\frac{1}{n})^{n-1}$. This event will occur if and only if:

- At time $t - \mathit{depth}(z)$, z fires, and
- For each $u \in \mathcal{T} - \{z, r\}$, u does not fire at time $t - \mathit{depth}(u)$.

By the algorithm, all these events are independent, so the probability of this combined event is exactly $\frac{1}{n}(1-\frac{1}{n})^{n-1}$, as needed. □

By Lemma 7, for any step $t \geq n$, the probability of any given rumor ρ_z reaching r in step t is at least as large as the probability of collecting a given coupon in the coupon collector problem. Thus, with probability $1 - o(1)$, all rumors will reach r in time $O(n \log n)$, and, furthermore, the expected time for the last rumor to reach r is also $O(n \log n)$.

It remains to argue that we can convert Algorithm RTree into the standard model, where receiving and transmitting at the same time is not allowed. The algorithm is essentially the same, with the only difference being that if the algorithm decides to fire then it will go into the transmit state, otherwise it will stay in the receive state. With this change, some rumors may get rejected when they are transmitted (without collision) to a node that happens to be in the transmitting state. Extending the concept of collisions to this situation, calculations analogous to those above show that the asymptotic running time remains the same.

Theorem 7. *Algorithm* RTree *has expected running time* $O(n \log n)$. *In fact, it will complete gathering in time* $O(n \log n)$ *with probability* $1 - o(1)$.

8 An $\Omega(n \log n)$ Lower Bound for Randomized Algorithms

In this section, we will show that Algorithm RTree is within a constant factor of optimal for label-less algorithms, even if the topology of the tree is known in advance. Actually we will show something a bit stronger, namely that there is a constant c such that any label-less algorithm with running time less than $cn \ln n$ will almost surely have some rumors fail to reach the root on certain trees.

The specific tree we will use here is that of the star graph, consisting of the root with n children that are also the leaves in the tree. (We use $n+1$ nodes instead of n, for convenience, but this does not affect our lower bound.) In this tree, these leaves are entirely isolated: No leaf receives any information from the root or from any other leaf. Thus at each time step t, each leaf v transmits

with a probability that can depend only on t and on the set of previous times at which v attempted to transmit. Note that, unlike in Algorithm RTree, the actions of v at different steps may not be independent, which complicates the argument. Allowing some dependence, in fact, can help reduce the running time, although only by a constant factor (see Theorem 9).

For the star graph, we can equivalently think of a label-less algorithm running in time T as a probability distribution over all subsets of $\{0, 1, \ldots, T-1\}$ representing the sets of transmission times of each node. Each node v independently picks a subset F_v according to this distribution, and transmits only at the times in F_v. The label-less requirement is equivalent to the requirement that the F_v are identically distributed. Node v succeeds in transmitting if there is a time t such that $t \in F_v$, but $t \notin F_w$ for any $w \neq v$.

The main result of this section is the following lower bound.

Theorem 8. *If $\mathcal{R}$ is a randomized protocol for information gathering on trees then the expected running time of $\mathcal{R}$ is $\Omega(n \ln n)$. More specifically, if we run $\mathcal{R}$ on the n-node star graph for $T \leq cn \ln n$ steps, where $c < \frac{1}{\ln^2 2}$, then with probability $1 - o(1)$ at least one rumor will fail to reach the root.*

As we show, our lower bound is very tight, in the sense that the value of the constant c in the above theorem is best possible for star graphs.

Theorem 9. *If $T = cn \ln n$, where $c > \frac{1}{\ln^2 2}$, then there is a protocol which succeeds on the star graph in time at most T with probability $1 - o(1)$.*

8.1 Proof of Theorem 8 (Under Two Assumptions)

For now we will assume that our transmission distribution satisfies two additional simplifying assumptions for every node v and $n > 1$:

Assumption 1: $|F_v| \leq \ln^{10} n$ with probability 1.

Assumption 2: $\frac{1}{n^3} \leq \mathbf{P}(t \in F_v) \leq \frac{\ln^4 n}{n}$, for each time t.

Later we will show how to remove these assumptions.

For each time t, let q_t be the (unconditional) probability that a given node transmits at time t (which by Assumption 2 lies between $\frac{1}{n^3}$ and $\frac{\ln^4 n}{n}$). Then the conditional probability that a node successfully transmits its rumor at time t, given that it attempted to transmit, is

$$(1 - q_t)^{n-1} \leq e^{-(n-1)q_t}.$$

What we would like to do is use this to obtain an estimate on the probability of successful transmission for a node v, given its set F_v. If the transmissions were independent, this would be easy: just take the product of the failure probabilities at each time in F_v and subtract it from 1. As it turns out, this (nearly) gives a lower bound on the failure probability.

Lemma 8. *For any distribution that satisfies Assumptions 1 and 2, and any fixed v and F_v, we have*

$$\mathbf{P}\left(v \text{ fails} \mid F_v\right) \geq \prod_{t \in F_v} \left(1 - e^{-(n-1)q_t\left(1 - n^{-5/6 + o(1)}\right)}\right),$$

where "v fails" is the event that the rumor from v does not reach the root.

The inequality in this lemma only goes in one direction. As an example, consider the case where $T = 2n$ and each node transmits at times $2t$ and $2t-1$, for a uniformly chosen t. Then if a node's first transmission collides, it makes it significantly more likely (in fact guaranteed) that the second one does as well. What the lemma states, in essence, is that it cannot make it significantly *less* likely. We will assume that Lemma 8 holds for now and return to its proof later on.

From now on, fix some constant $c < \frac{1}{\ln^2 2}$. To allow asymptotic notation in the calculations, we consider n to be some sufficiently large integer. Let X be the random variable equal to the number of nodes whose rumors fail to reach the root in the first T steps of $\mathcal{R}$, where $T \leq cn \ln n$. We next show that the expectation of X is large, which is equivalent to showing that the probability that any fixed node v fails is significantly larger than $1/n$. By Lemma 8, we have

$$\begin{aligned}
\mathbf{P}(v \text{ fails}) &= \mathbf{E}_{F_v}\left[\mathbf{P}\left(v \text{ fails} \,|F_v\right)\right] \\
&\geq \mathbf{E}_{F_v}\left[\prod_{t\in F_v}\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\right)\right] \\
&= \mathbf{E}_{F_v}\left[\prod_{t=0}^{T-1}\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\chi(t \in F_v)\right)\right].
\end{aligned}$$

Here $\chi(t \in F_v)$ is an indicator function equal to 1 if $t \in F_v$ and 0 otherwise. We now apply a variant of Jensen's inequality which states that $f(\mathbf{E}[Y]) \geq \mathbf{E}[f(Y)]$, for a random variable Y and a concave function $f(Y)$. Using this inequality, we have

$$\begin{aligned}
\ln \mathbf{P}(v \text{ fails}) &\geq \ln\left[\mathbf{E}_{F_v}\left[\prod_{t=0}^{T-1}\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\chi(t \in F_v)\right)\right]\right] \\
&\geq \mathbf{E}_{F_v}\left[\ln\left(\prod_{t=0}^{T-1}\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\chi(t \in F_v)\right)\right)\right] \\
&= \sum_{t=0}^{T-1}\mathbf{E}_{F_v}\left[\ln\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\chi(t \in F_v)\right)\right] \\
&= \sum_{t=0}^{T-1} q_t \ln\left(1 - e^{-(n-1)q_t\left(1-n^{-5/6+o(1)}\right)}\right) \\
&\geq \frac{T}{n-1\left(1-n^{-5/6+o(1)}\right)}\inf_{x>0}\left(x\ln\left(1-e^{-x}\right)\right) - O\left(n^{-5/6+o(1)}\right) \\
&= -(c+o(1))\ln^2 2 \ln n.
\end{aligned}$$

The second inequality holds because $x\ln(1-e^{-x})$, for $x > 0$, is minimized when $x = \ln 2$.

For any $c < \frac{1}{\ln^2 2}$, the above bound on $\ln \mathbf{P}(v \text{ fails})$ implies that

$$\mathbf{P}(v \text{ fails}) \geq e^{-(c+o(1))\ln^2 2 \ln n} = n^{-1+\Omega(1)}.$$

This bound holds for any individual node v, so the expected number of nodes which fail is $\mathbf{E}[X] = n^{\Omega(1)}$.

To complete the proof of the Theorem (i.e. to show that X is almost surely positive), we need to establish concentration around this expectation of X, for which we utilize Talagrand's inequality. Think of X as a function of transmission sets, $X = X(F_0, F_1, \ldots, F_{n-1})$. This function X is Lipschitz, in the sense that changing a single F_w can only change X by at most $\ln^{10} n$. (By Assumption 1, node w transmits at most $\ln^{10} n$ times, and each transmission can only interfere with at most one otherwise successful transmission.) Furthermore, X is locally certifiable in the following sense: If $X \geq x_0$ for some x_0, then there is a subset I of at most $2x_0 \ln^{10} n$ nodes such that X remains larger than x_0 no matter how we change the transmission patterns of the nodes outside I. For example, we can construct I as follows. Start with I_0 being a set of x_0 nodes

that failed to transmit successfully. Then, for each $v \in I_0$ let J_v be a set of at most $\ln^{10} n$ nodes such that whenever v transmits then at least one of the nodes in J_v also transmits. Then the set $I = I_0 \cup \bigcup_{v \in I_0} J_v$ has the desired properties.

Let b be the median value of X (not the mean). It follows from the above two properties of X, together with Talagrand's Inequality [25] (see Section 7.7 of [1] for the specific version used here), that for any $\gamma > 0$ we have

$$\mathbf{P}\left(|X - b| \geq \gamma \ln^{15} n\sqrt{b}\right) \leq 4e^{-\gamma^2/4}. \tag{4}$$

In particular, applying this bound with $\gamma = \sqrt{b}\ln^{-15} n$ gives

$$\mathbf{P}(X = 0) \leq \mathbf{P}(|X - b| \geq b) \leq 4e^{-b\ln^{-30} n/4} \tag{5}$$

To finish, it is therefore enough to show that b is large.

Taking $\gamma = 2\ln n$ in (4), we have

$$\mathbf{P}\left(X \geq b + 2\ln^{16} n\sqrt{b}\right) \leq 4e^{-\ln^2 n}. \tag{6}$$

Since X is always at most n, inequality (6) implies that the contribution to $\mathbf{E}[X]$ from the values of X that are at least $b + 2\ln^{16} n\sqrt{b}$ is at most $4e^{-\ln^2 n} \cdot n = o(1)$. On the other hand, the contribution to $\mathbf{E}[X]$ from the remaining values of X can be at most $b + 2\ln^{16} n\sqrt{b}$. It follows that

$$n^{\Omega(1)} = \mathbf{E}(X) \leq b + 2\ln^{16} n\sqrt{b} + o(1),$$

from which we have $b = n^{\Omega(1)}$. Substituting $b = n^{\Omega(1)}$ into the right-hand side of (5), we obtain that $\mathbf{P}(X = 0) = o(1)$, which is exactly the claim in the second part of Theorem 8.

It remains to prove Lemma 8 and to remove Assumptions 1 and 2, that we originally placed on the distribution of transmission times.

8.2 Proof of Lemma 8

We present first a rough idea behind our argument: Suppose we fixed F_v and all of the q_t, and tried to choose our distribution so as to minimize the probability that v fails. (Here and throughout this section we will say that v "fails" if no transmission from v reaches the root. Otherwise we say that v "succeeds"). As the example discussed following the statement of Lemma 8 shows, we can increase the probability of v's failure by choosing a distribution in which a single other node interferes with multiple transmissions from v. Conversely, one might hope that to minimize the failure probability the distribution should be such that no single node interferes with more than one transmission time in F_v. In the remainder of this section we give a rigorous argument that formalizes this intuition.

Let y denote the restriction of our distribution to F_v, defined by

$$y_B = \mathbf{P}(F_u \cap F_v = B) \quad \text{for all } u \neq v \quad \text{and } B \subseteq F_v.$$

This is well defined, as the distribution of F_u is assumed identical for all $u \neq v$. Since the success or failure of v depends only on what happens in times in F_v, the failure probability remains identical if we replace the distribution of F_u by that of y for each $u \neq v$. The intuition in the previous paragraph is formalized in the following claim.

Claim 1. *There is a distribution $\tilde{y}$ on subsets of F_v with the following properties:*

(i) *$\tilde{y}$ is supported on $\{\{t\} : t \in F_v\} \cup \{\emptyset\}$, namely on subsets of F_v of cardinality at most 1.*

(ii) *The expected number of firings at any time $t \in F_v$ is the same when the remaining vertices fire according to $\tilde{y}$ as when they fire according to y.*

(iii) *The probability that v fails if the remaining vertices fire according to $\tilde{y}$ is no larger than if they fire according to y.*

Let us for now assume the truth of Claim 1. It then suffices to give a lower bound on the probability of failure in the case where all vertices apart from v fire according to $\tilde{y}$.

Let $t_1 < t_2 < \cdots < t_k$ be the times in F_v. By Assumption 1, we have $k = n^{o(1)}$. For each $i = 1, 2, ..., k$, let g_i be the number of nodes other than v that transmit at time t_i, and let E_i be the event that $1 \leq g_i \leq n^{1/6}$. We can bound the conditional probability that v fails given F_v from below by the probability that every event E_i occurs, that is

$$\mathbf{P}(v \text{ fails} \mid F_v) \geq \prod_{i=1}^{k} \mathbf{P}\left(E_i \mid E_1, \ldots, E_{i-1}\right). \tag{7}$$

To estimate the right-hand side of (7), we will bound each term in this product directly by separately bounding the conditional probabilities that $g_i = 0$ and that $g_i > n^{1/6}$.

So fix some i and assume that all events $E_1, \ldots, E_{i-1}$ have occurred. Then there must be at least $n - n^{1/6}k = n - n^{1/6+o(1)}$ nodes which did not transmit at any t_j with $j < i$. Under $\tilde{y}$, each of these nodes transmits at time t_i with probability

$$q'_{t_i} = q_{t_i}\Big(1 - \textstyle\sum_{j=1}^{i-1} q_{t_j}\Big)^{-1} \leq q_{t_i}\Big(1 - \frac{\ln^{14} n}{n-1}\Big),$$

where the second inequality holds because $i \leq \ln^{10} n$ (by Assumption 2) and $q_{t_j} \leq \frac{\ln^4 n}{n-1}$ for each j (by Assumption 1). Thus

$$q_{t_i} \leq q'_{t_i} = q_{t_i}(1 + o(1)).$$

We first use this bound to estimate the probability that $g_i = 0$, as follows:

$$\begin{aligned}\mathbf{P}\left(g_i = 0 \mid E_1, \ldots, E_{i-1}\right) &\leq \left(1 - q'_{t_i}\right)^{n - n^{1/6+o(1)}} \\ &= e^{-(n-1)q'_{t_i}\left(1 - n^{-5/6+o(1)}\right)} = e^{-(n-1)q_{t_i}\left(1 - n^{-5/6+o(1)}\right)}.\end{aligned} \tag{8}$$

To bound the probability that g_i is large, we use the union bound. If $g_i > n^{1/6}$, there must be some subset of size $n^{1/6}$, all of whose nodes transmit. (To avoid clutter, here and in the calculations below we write $n^{1/6}$ to mean $\lceil n^{1/6} \rceil$.) So, for a set $J \subseteq V - \{v\}$, letting D_J denote the event that

all nodes in J transmit at time t_i, we have

$$\begin{aligned}
\mathbf{P}\big(g_i > n^{1/6} \mid E_1, \ldots, E_{i-1}\big) &\leq \sum_{|J|=n^{1/6}} \mathbf{P}(D_J) \\
&\leq \binom{n}{n^{1/6}} {q'_{t_i}}^{n^{1/6}} \\
&\leq \binom{n}{n^{1/6}} \left(\frac{(1+o(1))\ln^4 n}{n}\right)^{n^{1/6}} \qquad (9) \\
&\leq \left(\frac{e\ln^4 n}{n^{1/6}}\right)^{n^{1/6}} = e^{-\omega(n^{1/6})}, \qquad (10)
\end{aligned}$$

where inequality (9) follows from $q'_{t_i} \leq \frac{\ln^4 n(1+o(1))}{n}$. Summing the bounds (8) and (10) and taking logarithms, we obtain

$$\begin{aligned}
\ln\left(\mathbf{P}\left(\neg E_i \mid E_1, \ldots, E_{i-1}\right)\right) &\leq \ln\left(e^{-(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right)} + e^{-\omega(n^{1/6})}\right) \\
&\leq -(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right) + \frac{e^{-\omega(n^{1/6})}}{e^{-(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right)}} \qquad (11) \\
&\leq -(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right) + e^{-\omega(n^{1/6})}e^{\ln^4 n} \qquad (12) \\
&= -(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right). \qquad (13)
\end{aligned}$$

We now justify the steps of this derivation. Inequality (11) follows from $\ln(y+z) \leq \ln y + \frac{z}{y}$ for all $y, z > 0$. Inequality (12) follows from $q_{t_i} \leq \frac{\ln^4 n}{n-1}$ (the upper bound from Assumption 2). The absorption in the final step (13) follows from the lower bound in Assumption 2.

We thus have

$$\mathbf{P}\left(E_i \mid E_1, \ldots, E_{i-1}\right) \geq 1 - e^{-(n-1)q_{t_i}\left(1-O(n^{-5/6+o(1)})\right)}. \qquad (14)$$

Assumption 1 implies that the product on the right-hand side of (7) has $n^{o(1)}$ terms. So, substituting the bound (14), for each i, into the right-hand side of (7), we obtain the inequality in Lemma 8.

To complete the proof of Lemma 8, it remains to prove Claim 1.

8.2.1 Proof of Claim 1

The idea of this proof will be to repeatedly apply a series of transformations on the distribution y. Each transformation will in turn decrease the total mass of y on non-singleton subsets, without increasing the failure probability.

Suppose that some $U \subseteq F_v$ with $|U| \geq 2$ has $y_U > 0$. For n large enough, Assumptions 1 and 2 imply that each node $u \neq v$ transmits on average fewer than one time in F_v. This, in turn, implies that $y_\emptyset > y_U$. We now define a new distribution y' as follows: Fix some $t \in U$, and define

$$\begin{aligned}
y'_U &= 0 & y'_{U-\{t\}} &= y_{U-\{t\}} + y_U \\
y'_{\{t\}} &= y_{\{t\}} + y_U & y'_\emptyset &= y_\emptyset - y_U
\end{aligned}$$

with $y'_B = y_B$ for all $B \neq U$. Effectively, what we are doing here is moving mass y_U from each of U and $\emptyset$ to each of $U - \{t\}$ and $\{t\}$. Note that this operation does not change the value of q_t for any

$t \in F_v$. The intuition above is that this separates the transmission times, so this should reduce the failure probability of v. Actually, a stronger monotonicity property is true:

Claim 2. *Fix some node $w \neq v$, and assume that, for each node $u \in V - \{v, w\}$, the transmition set of u within F_v is distributed either according to y or according to y'. Then the probability that v fails if w transmits according to y' is not larger than the probability that v fails if w transmits according to y.*

Proof. It is sufficient to prove the claim for the failure probabilities conditioned on all transmission patterns $F_u \cap F_v$, for $u \in V - \{v, w\}$, having some fixed (but arbitrary) values. This is because, once we prove the claim for such conditional probabilities, the claim for non-conditional probabilities will follow by simple averaging. (In fact, this remains true if the transmission sets within F_v of all $u \notin \{v, w\}$ are arbitrary, not necessarily restricted to y or y'.)

So let us fix all sets $F_u \cap F_v$, for $u \in V - \{v, w\}$. The argument below will be conditioned on these patterns being fixed. From this point on, the proof of the claim essentially comes from direct verification.

Let $S \subseteq F_v$ be the set of times at which v transmits, but no node $u \in V - \{v, w\}$ transmits; in other words, the times when v's transmission would succeed if w did not transmit. Let $\hat{F}_w = F_v \cap F_w$ be the random variable representing the transmission times of w within F_v. Then node v will fail if and only if $S \subseteq \hat{F}_w$.

We now analyze how moving the mass between distributions y and y' affects the probability of v failing, that is the probability of the event that $S \subseteq \hat{F}_w$. It is sufficient to consider only the four sets $\hat{F}_w \in \{U, \{t\}, U - \{t\}, \emptyset\}$, since the probabilities of all other sets $\hat{F}_w$ remain unchanged. We have four cases:

Case 1: $S = \emptyset$. Then v fails for all four choices of $\hat{F}_w$, so moving the mass does not affect the failure probability.

Case 2: $S - U \neq \emptyset$. Then moving the mass again has no impact on the success probability, because v succeeds for all four choices of $\hat{F}_w$.

Case 3: $S \neq \emptyset$, $S \subseteq U$ and $|U| \geq 2$. Then v fails when $\hat{F}_w = U$, but succeeds when $\hat{F}_w = \{t\}$. So we are moving mass from a pair of elements with at least one failure to a pair with at most one failure; therefore we cannot increase the failure probability.

Case 4: $S = \{t\}$. Then the operation moves mass from one failure ($\hat{F}_w = U$) and one success ($\hat{F}_w = \emptyset$) to one failure ($\hat{F}_w = \{t\}$) and one success ($\hat{F}_w = U - \{t\}$), so the failure probability is unchanged.

This completes the proof of Claim 2. □

We now start with all nodes different from v transmitting according to y, and apply Claim 2 in succession to each node $w \neq v$. Observe that at each intermediate step of this process, some nodes $u \in V - \{v, w\}$ will use distribution y while other will use the new distribution y'. At the end, all nodes other than v will transmit according to y'. Since Claim 2 holds at each step, we thus obtain that the probability that v fails when all of the nodes other than v transmit according to y' is no larger then when they all transmit according to y.

Note that y' has the same values of all probabilities q_t as y. Also, in y' each node transmits on average less than one time, which is the only specific property that we needed in order to transform

y into y'. Therefore, if the support of y' still contains a set U with $|U| \geq 2$, we can apply again the same transformation with y' in place of y. We apply these transformations repeatedly, and at each step the sum of the squares of the cardinalities of the sets in the support of y decreases, so this process must terminate after a finite number of steps. In the last step, the resulting distribution y' must only have singletons and the empty set in its support. Thus letting $\tilde{y}$ be this y', we obtain Claim 1.

8.3 Removing Assumptions 1 and 2

So far we have shown that any distribution on transmission times satisfying both Assumptions 1 and 2 will almost surely lead to at least one rumor failing to reach the root. We next consider distributions satisfying Assumption 2, but not necessarily Assumption 1.

For such a distribution, the expected number of times when a given node v transmits is $\mathbf{E}[|F_v|] \leq T \cdot \ln^4 n/n \leq (cn \ln n) \cdot \ln^4 n/n \leq \ln^6 n$ (for n large enough). We now think of the transmission sets F_v as being generated in a two step exposure process. First, we expose the set V' of all nodes v for which $|F_v| \leq \ln^{10} n$, and then the actual transmission sets F_v for all nodes v.

By Markov's inequality, the probability that v transmits at least $\ln^{10} n$ times is $o(1)$. Therefore, letting $n' = |V'|$, we have $n' = (1 - o(1))n$ with probability $1 - o(1)$. We now fix any constant c' such that $c < c' < 1/\ln^2 2$ and we apply the previous analysis to the conditional distribution on V' which, by construction, now satisfies Assumption 1. This will give us that, with probability $1 - o(1)$, there is a node $w \in V'$ that fails to transmit successfully in the first $c'n' \ln n'$ steps, even if the nodes in $V - V'$ do not transmit at all. Of course, including the nodes in $V - V'$ can only make the failure probability larger. Since $n' = (1 - o(1))n$ and $c < c'$, this implies an analogous statement for the original transmission distribution of all nodes in V, namely that with probability $1 - o(1)$ there will be a node $w \in V$ that will fail in the first $cn \ln n$ steps.

It remains to consider distributions which do not noecessarily satisfy Assumption 2. Let L_{low} be the collection of times at which each node transmits with probability at most $\frac{1}{n^3}$, and let L_{high} be the collection of times at which each node transmits with probability at least $\frac{\ln^4 n}{n}$. Of course, $L_{\text{low}} \cap L_{\text{high}} = \emptyset$. Let E_{in} be the event that at least one node transmits successfully in $L_{\text{low}} \cup L_{\text{high}}$, and let E_{out} be the event all nodes transmit successfully in a time not in $L_{\text{low}} \cup L_{\text{high}}$. The total success probability is bounded above by $\mathbf{P}(E_{\text{in}}) + \mathbf{P}(E_{\text{out}})$. By the previous analysis (since Assumption 2 is satisfied once the times in $L_{\text{low}} \cup L_{\text{high}}$ are removed), we have $\mathbf{P}(E_{\text{out}}) = o(1)$. For the other term, we have

$$\begin{aligned}
\mathbf{P}(E_{\text{in}}) &\leq \sum_{t \in L_{\text{low}} \cup L_{\text{high}}} \sum_{v} \mathbf{P}(v \text{ succeeds at time } t) \\
&\leq \sum_{t \in L_{\text{low}}} \sum_{v} \mathbf{P}(v \text{ transmits at time } t) \\
&\qquad + \sum_{t \in L_{\text{high}}} \sum_{v} \mathbf{P}(\text{each node } u \neq v \text{ does not transmit at time } t) \\
&\leq Tn\frac{1}{n^3} + Tn\left(1 - \frac{\ln^4 n}{n}\right)^{n-1} \\
&= \frac{T}{n^2} + (1 + o(1))Tne^{-\ln^4 n} = o(1).
\end{aligned}$$

Putting it all together, the probability that all nodes succeed is $o(1)$. The proof of Theorem 8 is now complete.

8.4 Proof of Theorem 9

We now show that for $c > \frac{1}{\ln^2 2}$ there is a protocol which succeeds on the star graph in time $T = cn \ln n$ with probability approaching 1. The protocol is straightforward: The set $\{0, 1, \ldots, T-1\}$ of possible transmission times is divided into $\frac{T \ln 2}{n} = c \ln 2 \ln n$ disjoint intervals of length $\frac{n}{\ln 2}$ each. (To avoid clutter, we will treat the values $c \ln n$ and $n/\ln 2$ as if they were integral. In reality, they need to be appropriately rounded.) Each node independently and uniformly chooses a single time in each interval at which to transmit.

Fix some arbitrary node $v \neq r$. Under this protocol, the probability that v succeeds in a given interval is

$$\left(1 - \frac{\ln 2}{n}\right)^{n-1} = \frac{1}{2} - o(1),$$

so the probability that v's rumor fails to reach the root in each interval is

$$\left(\frac{1}{2} + o(1)\right)^{c \ln 2 \ln n} = n^{-c \ln^2 2 + o(1)} = o(n^{-1}),$$

since $c > \frac{1}{\ln^2 2}$. This holds for each node $v \neq r$, so we can now apply the union bound, obtaining that with probability $1 - o(1)$ all rumors reach the root.

References

[1] A. Alon and J. Spencer. *The Probabilistic Method, 3rd edition*. Wiley, Hoboken, 2008.

[2] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio broadcast. *J. Comput. Syst. Sci.*, 43(2):290–298, 1991.

[3] Reuven Bar-Yehuda, Amos Israeli, and Alon Itai. Multiple communication in multihop radio networks. *SIAM J. Comput.*, 22(4):875–887, 1993.

[4] Allan Borodin, Yuval Rabani, and Baruch Schieber. Deterministic many-to-many hot potato routing. *IEEE Trans. Parallel Distrib. Syst.*, 8(6):587–596, 1997.

[5] Danilo Bruschi and Massimiliano Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distrib. Comput.*, 10(3):129–135, April 1997.

[6] Bogdan S. Chlebus, Leszek Gasieniec, Alan Gibbons, Andrzej Pelc, and Wojciech Rytter. Deterministic broadcasting in ad hoc radio networks. *Distributed Computing*, 15(1):27–38, 2002.

[7] Bogdan S. Chlebus, Dariusz R. Kowalski, and Tomasz Radzik. Many-to-many communication in radio networks. *Algorithmica*, 54(1):118–139, 2009.

[8] Malin Christersson, Leszek Gasieniec, and Andrzej Lingas. Gossiping with bounded size messages in ad hoc radio networks. In *Automata, Languages and Programming, 29th International Colloquium, ICALP'02*, pages 377–389, 2002.

[9] Marek Chrobak, Leszek Gasieniec, and Wojciech Rytter. Fast broadcasting and gossiping in radio networks. *J. Algorithms*, 43(2):177–189, 2002.

[10] Marek Chrobak, Leszek Gasieniec, and Wojciech Rytter. A randomized algorithm for gossiping in radio networks. *Networks*, 43(2):119–124, 2004.

[11] Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Distributed broadcast in radio networks of unknown topology. *Theor. Comput. Sci.*, 302(1-3):337–364, 2003.

[12] Artur Czumaj and Wojciech Rytter. Broadcasting algorithms in radio networks with unknown topology. *Journal of Algorithms*, 60(2):115 – 143, 2006.

[13] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1–2):79–89, 1985.

[14] Paul Erdös and Pál Turán. On a problem of Sidon in additive number theory and on some related problems. *Journal of London Mathematical Society*, 16:212 – 215, 1941.

[15] Uriel Feige and Prabhakar Raghavan. Exact analysis of hot-potato routing (extended abstract). In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 553–562, 1992.

[16] Leszek Gasieniec. On efficient gossiping in radio networks. In *16th Int. Colloquium on Structural Information and Communication Complexity (SIROCCO'09)*, pages 2–14, 2009.

[17] Leszek Gasieniec and Igor Potapov. Gossiping with unit messages in known radio networks. In *Foundations of Information Technology in the Era of Networking and Mobile Computing, IFIP 17th World Computer Congress - TC1 Stream / 2nd IFIP International Conference on Theoretical Computer Science (TCS'02)*, pages 193–205, 2002.

[18] Leszek Gasieniec, Tomasz Radzik, and Qin Xin. Faster deterministic gossiping in directed ad hoc radio networks. In *Proc. Scandinavian Workshop on Algorithm Theory (SWAT'04)*, pages 397–407, 2004.

[19] Majid Khabbazian and Dariusz R. Kowalski. Time-efficient randomized multiple-message broadcast in radio networks. In *Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '11, pages 373–380, 2011.

[20] Dariusz R. Kowalski and Andrzej Pelc. Faster deterministic broadcasting in ad hoc radio networks. *SIAM J. Discrete Math.*, 18(2):332–346, 2004.

[21] Eyal Kushilevitz and Yishay Mansour. An $\omega(d \log(n/d)$ lower bound for broadcast in radio networks. *SIAM J. Comput.*, 27(3):702–712, 1998.

[22] Ding Liu and Manoj Prabhakaran. On randomized broadcasting and gossiping in radio networks. In *8th Annual Int. Conference on Computing and Combinatorics (COCOON'02)*, pages 340–349, 2002.

[23] Gianluca De Marco. Distributed broadcast in unknown radio networks. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pages 208–217, 2008.

[24] A.N. Strahler. Hypsometric (area-altitude) analysis of erosional topology. *Bull. Geol. Soc. Amer.*, 63:117–1142, 1952.

[25] M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathematiques de l'I.H.E.S.*, 81:73–205, 1995.

[26] X.G. Viennot. A Strahler bijection between Dyck paths and planar trees. *Discrete Mathematics*, 246:317–329, 2003.

[27] Wikipedia. Sidon sequence. http://en.wikipedia.org/wiki/Sidon_sequence.

[28] Wikipedia. Golomb ruler. http://en.wikipedia.org/wiki/Golomb_ruler, 2011.

[29] Wikipedia. Sparse ruler. http://en.wikipedia.org/wiki/Sparse_ruler, 2011.

[30] Ying Xu. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica*, 36(1):93–96, 2003.