

# Information Gathering in Ad-Hoc Radio Networks with Tree Topology

Marek Chrobak<sup>1</sup>(✉), Kevin Costello<sup>2</sup>,  
Leszek Gasieniec<sup>3</sup>, and Darek R. Kowalski<sup>3</sup>

<sup>1</sup> Department of Computer Science,  
University of California at Riverside, Riverside, USA  
[marek@cs.ucr.edu](mailto:marek@cs.ucr.edu)

<sup>2</sup> Department of Mathematics, University of California at Riverside, Riverside, USA

<sup>3</sup> Department of Computer Science, University of Liverpool, Liverpool, UK

**Abstract.** We study 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*. The goal is to deliver all rumors to the root of the tree as quickly as possible. The protocol must complete this task even if the tree topology is unknown. In the deterministic case, assuming that the nodes are labeled with small integers, we give an  $O(n)$ -time protocol that uses unbounded messages, and an  $O(n \log n)$ -time protocol using bounded messages. 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 *collision* occurs, that is if no other in-neighbor of  $u$  transmits at this step. We do not assume any collision detection mechanism, so neither  $u$  nor any other node knows whether a collision occurred.

Another feature of our model is that the topology of  $G$  is unknown. We are interested in distributed protocols, where the computation at a node  $v$  depends only on the label of  $v$  and the information gathered from the received messages.

---

Research supported by grants CCF-1217314 (NSF) and H98230-13-1-0228 (NSA).

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

The two primitives for information dissemination in ad-hoc radio networks that have been 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  $0, 1, \dots, n - 1$ , the fastest known deterministic algorithms for broadcasting run in time  $O(n \log n \log \log n)$  [12] or  $O(n \log^2 D)$  [11], where  $D$  is the diameter of  $G$ . The best lower bound on the running time in this model is  $\Omega(n \log D)$  [10]. (See also [4, 5, 8, 18] for earlier work.) Allowing randomization, broadcasting can be accomplished in time  $O(D \log(n/D) + \log^2 n)$  with high probability [11, 19], even if the nodes are not labelled. This matches the lower bounds in [2, 20].

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. With randomization, gossiping can be solved in expected time  $O(n \log^2 n)$  [11] (see [9, 21] for earlier work), even if the nodes are not labelled. In contrast, for deterministic algorithms, with nodes labelled  $0, 1, \dots, n - 1$ , the fastest known gossiping algorithm runs in time  $O(n^{4/3} \log^4 n)$  [16], following earlier progress in [8, 25]. (See also [15] 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.

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

- (1) We first study deterministic algorithms, under the assumption that the nodes of  $\mathcal{T}$  are labelled  $0, \dots, n - 1$ . In Sect. 4 we examine the model without any bound on the message size, for which we give an optimal,  $O(n)$ -time protocol.
- (2) Next, in Sect. 5, we consider the model with bounded messages, where a message may contain only one rumor, for which we give an  $O(n \log n)$  protocol.
- (3) In Sect. 6 we introduce a more restrictive model of *fire-and-forward* protocols, in which a node can only transmit either 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 a matching lower bound of  $\Omega(n^{1.5})$ .

- (4) We then consider randomized algorithms (Sect. 7), that do not use node labels. In this model, we give an  $O(n \log n)$ -time gathering protocol and we prove a matching lower bound of  $\Omega(n \log n)$ . If the tree is a star, we show that our lower bound is in fact optimal with respect to the leading constant.

Our algorithms for deterministic protocols easily extend to the model with labels drawn from a set  $0, 1, \dots, L$  where  $L = O(n)$ , without affecting the running times. If  $L$  is arbitrary, our algorithms for bounded and unbounded messages can be implemented in time, respectively,  $O(n^2 \log L)$  and  $O(n^2 \log n \log L)$ .

We remark that some protocols for radio networks use forms of information gathering on trees as a sub-routine; see for example [3, 6, 17]. However, these solutions typically focus on undirected graphs, which allow feedback, and on a relaxed variant of information gathering where the goal is to gather only a fraction of rumors in the root. In contrast, we study directed trees without any feedback mechanism, and we require all rumors to be collected at the root.

Due to space limitations, some proofs are omitted in this extended abstract. The missing proofs are provided in [7].

## 2 Preliminaries

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  $[n] = \{0, 1, \dots, n-1\}$ . Denote by  $\text{label}(v)$  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  $\rho_v$ . The objective is to deliver all rumors  $\rho_v$  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  $0, 1, 2, \dots$ . 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  $t$ .

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. 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 tree, denote by  $\mathcal{T}$ , with root  $r$  and with all edges directed towards the root  $r$ .

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  $\rho_v$  of  $v$  is delivered to  $r$  no later than at step  $T(n)$ . In case of randomized protocols, we use the expectation of their running time  $T(n)$ , which is now a random variable, or we show that  $T(n)$  does not exceed a desired time bound with high probability.

We consider three types of gathering protocols. In the model with *unbounded messages* a node can transmit arbitrary information in a single step. In particular, multiple rumors can be aggregated into a single message. In the model with *bounded messages*, no aggregation of rumors is allowed. Each message consists of at most one rumor and  $O(\log n)$  bits of additional information. Our third model is called *fire-and-forward*. In a fire-and-forward protocol, a node can either transmit its own rumor or the rumor received in the previous step, if any. Thus a message originating from a node travels towards the root one hop at a time, until either it vanishes or it successfully reaches the root.

For illustration, consider a protocol ROUNDROBIN (known the literature), where all nodes transmit in a cyclic order, one at a time. The running time is  $O(n^2)$ , because in any consecutive  $n$  steps each rumor will decrease its distance to the root. For information gathering in trees, ROUNDROBIN can be adapted to use only bounded messages. At any round, if a node  $v$  has the rumor  $\rho_u$  with  $\text{label}(u) = t \bmod n$  then  $v$  transmits  $\rho_u$ . Only one child of a node can have  $\rho_u$ , so no collisions will occur, and after at most  $n^2$  steps  $r$  will receive all rumors.

### 3 Some Structure Properties of Trees

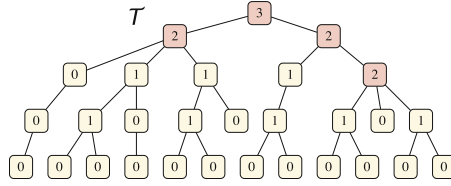
The running times of our algorithms in Sects. 4 and 5 depend on the distribution of high-degree nodes. To capture the structure of this distribution we define the concept of  $\gamma$ -depth which measures how “bushy” the tree is.

**$\gamma$ -Depth of trees.** Let  $\mathcal{T}$  be the given tree network with root  $r$  and  $n$  nodes. Fix an integer  $\gamma$  with  $2 \leq \gamma \leq n - 1$ . We define the  $\gamma$ -height of each node  $v$  of  $\mathcal{T}$ , denoted  $\text{height}_\gamma(v)$ , as follows. If  $v$  is a leaf then  $\text{height}_\gamma(v) = 0$ . If  $v$  is an internal node then let  $g$  be the maximum  $\gamma$ -height of a child of  $v$ . If at least  $\gamma$  children of  $v$  have  $\gamma$ -height equal  $g$  then  $\text{height}_\gamma(v) = g + 1$ ; otherwise  $\text{height}_\gamma(v) = g$ . (For  $\gamma = 2$ , our definition of 2-height is equivalent to Strahler numbers; see [22, 24].) We then define the  $\gamma$ -depth of  $\mathcal{T}$  as  $D_\gamma(\mathcal{T}) = \text{height}_\gamma(r)$ .

In our proofs, we may also consider trees other than the input tree  $\mathcal{T}$ . 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  $v$ . If  $\mathcal{H}$  is any tree and  $v$  is a node of  $\mathcal{H}$  then, to avoid ambiguity, we will write  $\text{height}_\gamma(v, \mathcal{H})$  for the  $\gamma$ -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,  $\text{height}_\gamma(v, \mathcal{H}) \leq \text{height}_\gamma(v)$ .

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.

The lemma below follows by simple induction on the depth of  $\mathcal{T}$  (see [24] for the special case  $\gamma = 2$ ).



**Fig. 1.** An example illustrating the concept of  $\gamma$ -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.

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

We will be particularly interested in subtrees of  $\mathcal{T}$  consisting of the nodes whose  $\gamma$ -height is above a given threshold. Specifically, for  $h = 0, 1, \dots, D_\gamma(\mathcal{T})$ , let  $\mathcal{T}^{\gamma,h}$  be the subtree of  $\mathcal{T}$  induced by the nodes whose  $\gamma$ -height is at least  $h$ . Note that  $\mathcal{T}^{\gamma,h}$  is indeed a subtree of  $\mathcal{T}$  rooted at  $r$ .

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

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

*Proof.* It is sufficient to prove the lemma for the case  $h = 1$ , which can be then extended to arbitrary values of  $h$  by induction. So let  $h = 1$  and  $\mathcal{T}' = \mathcal{T}^{\gamma,1}$ . The proof is by induction on the height of  $v$  in  $\mathcal{T}'$ . If  $v$  is a leaf of  $\mathcal{T}'$  then  $\text{height}_\gamma(v, \mathcal{T}') = 0$ , by definition. All children of  $v$  in  $\mathcal{T}$  must have  $\gamma$ -height equal 0, so  $\text{height}_\gamma(v) = 1$  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 for each child  $u$  of  $v$  in  $\mathcal{T}$ , either  $\text{height}_\gamma(u) = 0$  (that is,  $u \notin \mathcal{T}'$ ) or  $\text{height}_\gamma(u, \mathcal{T}') = \text{height}_\gamma(u) - 1$ . Let  $\text{height}_\gamma(v) = f$ . If  $v$  has a child with  $\gamma$ -height equal  $f$  then there are fewer than  $\gamma$  such children. By induction, these children will have  $\gamma$ -height in  $\mathcal{T}'$  equal  $f - 1$ , and each other child that remains in  $\mathcal{T}'$  has  $\gamma$ -height in  $\mathcal{T}'$  smaller than  $f - 1$ . So  $\text{height}_\gamma(v, \mathcal{T}') = f - 1$ . If all children of  $v$  have  $\gamma$ -height smaller than  $f$  then  $f \geq 2$  (for otherwise  $v$  would have to be a leaf of  $\mathcal{T}'$ ) and  $v$  must have more than  $\gamma$  children with  $\gamma$ -height  $f - 1$ . These children will be in  $\mathcal{T}'$  and will have  $\gamma$ -height in  $\mathcal{T}'$  equal  $f - 2$ . So  $\text{height}_\gamma(v, \mathcal{T}') = f - 1$  in this case as well, completing the proof.

## 4 Deterministic Algorithms with Aggregation

We now prove that using unbounded-size messages we can complete information gathering in time  $O(n)$ , which is optimal even for paths or star graphs.

Since we use unbounded messages, we can assume that each 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 collected rumors. (We can, for example, have each node append its label to its rumor.) We will also assume that each node knows the labels of its children. To acquire this knowledge, we can add a preprocessing phase where nodes with labels  $0, \dots, n - 1$  transmit, one at a time, in this order. Thus after  $n$  steps each node will receive the messages from its children.

**Simple 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.

---

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

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

---

*Analysis.* For a node  $v$ , we say that  $v$  is *dormant* in rounds  $0, \dots, \alpha_v - 1$ ,  $v$  is *active* in rounds  $\alpha_v, \dots, \alpha_v + n - 1$ , and that  $v$  is *retired* in every round thereafter. Since  $v$  will make at least one RR-transmission when it is active,  $v$  will successfully transmit its message 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 argument shows in fact that, at any round, Algorithm UNBDTREE1 satisfies the following invariants: (i) any path 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; and (ii) any dormant node has at least one active descendant.

**Lemma 3.** *Let  $d = D_2(\mathcal{T})$ . For any  $h = 0, 1, \dots, d$  and any node  $v$  with  $\text{height}_2(v) = h$ ,  $v$  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  $\text{height}_2(v) = h$ . To reduce clutter, denote  $\mathcal{Z} = \mathcal{T}^{2,h}$ . From Lemma 2, we have that  $\text{height}_2(v, \mathcal{Z}) = 0$ , which implies that  $\mathcal{Z}_v$  is a path from a leaf of  $\mathcal{Z}$  to  $v$ . Let  $\mathcal{Z}_v = v_1, v_2, \dots, v_q$  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  $v_p, v_{p+1}, \dots, v_q$  that are not on  $\mathcal{Z}_v$  do not transmit, since they are already retired. This implies that for each  $\ell = 0, \dots, q - p - 1$ , 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  $\alpha_r \leq 2nd = O(n \log n)$ , so the overall running time is  $O(n \log n)$ .

**Theorem 1.** *For any tree with  $n$  nodes and any assignment of labels, Algorithm UNBDTREE1 completes information gathering in time  $O(n \log n)$ .*

**Linear-Time Algorithm.** We show how to improve the running time of information gathering in trees to linear time, assuming unbounded 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  $F_0, F_1, \dots, F_{m-1} \subseteq [n]$  of sets such that for any set  $X \subseteq [n]$  with  $|X| \leq k$  and any  $x \in X$ , there is  $j$  for which  $F_j \cap X = \{x\}$ . For any  $k = 1, 2, \dots, n$ , there is a strong  $k$ -selective family with  $m = O(k^2 \log n)$  sets [10, 13].

In essence, the strong  $k$ -selective family is used to speed up information dissemination through low-degree nodes. To achieve linear time, we will interleave the steps using the selective family 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  $\kappa$ -selective family  $F_0, \dots, F_{m-1}$ . Without loss of generality, we assume  $m \leq n$ .

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

In each round  $s = \alpha_v, \alpha_v + 1, \dots, \alpha_v + m - 1$ ,  $v$  transmits in the All-step of round  $s$ , and if  $\text{label}(v) \in F_{s \bmod m}$  then  $v$  also transmits in the Sel-step of round  $s$ . In each round  $s = \alpha_v, \alpha_v + 1, \dots, \alpha_v + n - 1$ , if  $\text{label}(v) = s \bmod n$  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  $0, 1, \dots, \alpha_v - 1$ , *active* in rounds  $\alpha_v, \alpha_v + 1, \dots, \alpha_v + n - 1$ , and retired thereafter. We will also refer to  $v$  as being *semi-retired* in rounds  $\alpha_v + m, \alpha_v + m + 1, \dots, \alpha_v + n - 1$  (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 induction on the depth of  $\mathcal{T}$ , each node will eventually get activated, proving correctness. 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. (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 is to show that Sel- and All-steps disseminate information in linear time through subtrees where all node degrees are less than  $\kappa$ . The process can stall, however, if all active nodes have parents of degree larger than  $\kappa$ . In this case, a complete cycle of ROUNDROBIN will transmit the messages from these nodes to their parents. We show, using Lemma 1, that such stalling can occur at most the total of 3 times. So the overall running time will be still  $O(n)$ .

To formalize this argument, let  $\bar{d} = D_\kappa(\mathcal{T})$ . From Lemma 1, we have  $\bar{d} \leq 3$ . We fix some  $g \in \{0, 1, 2, 3\}$ , a node  $w$  with  $\text{height}_\kappa(w) = g$ , and we let  $\mathcal{Y} = \mathcal{T}_w^{\kappa, g}$ . Thus  $\mathcal{Y}$  consists of the descendants of  $w$  whose  $\kappa$ -height in  $\mathcal{T}$  is exactly  $g$ , or, equivalently (by Lemma 2), the descendants of  $w$  in  $\mathcal{T}^{\kappa, g}$  whose  $\kappa$ -height in  $\mathcal{T}^{\kappa, g}$  is equal 0. So all nodes in  $\mathcal{Y}$  have degree smaller than  $\kappa$ . 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 at most  $O(n)$  rounds after round  $\bar{s}$ .

**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  $h = 0, \dots, d$ , let  $l_h$  be the number of nodes  $u \in \mathcal{Y}$  with  $\text{height}_2(u, \mathcal{Y}) = h$ . 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  $\kappa$ , 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  $O(l_h)$  amortize to  $O(n)$ , and the terms  $O(m)$  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  $\text{height}_2(v, \mathcal{Y}) = h$ . 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  $v_1, v_2, \dots, v_q = v$ , 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 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 (see the invariant (ii)), 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  $v$ .  $\mathcal{Z}_v$  is a path  $v_1, v_2, \dots, v_q = v$  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, no later than in round  $\bar{s} + 2n$ . 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  $\kappa$ -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  $\kappa$  children of  $v_1$  are active in these rounds. This implies that the message of  $u$  will be received by  $v_1$ . 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. 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  $\kappa$ -height equal 0 will get activated in at most  $O(n)$  rounds. For  $g = 1, 2, 3$ , all nodes with  $\kappa$ -height equal  $g$  will activate no later than  $O(n)$  rounds after the last node with  $\kappa$ -height less than  $g$  is activated. This implies that all nodes in  $\mathcal{T}$  will be activated within  $O(n)$  rounds.

**Theorem 2.** *For any tree with  $n$  nodes and any assignment of labels, Algorithm UNBDTREE2 completes information gathering in time  $O(n)$ .*

## 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  $O(n \log n)$ . For simplicity, assume here that we are allowed to receive and transmit at the same time. We later explain how to remove this assumption.

---

**Algorithm BNDTREE.** First, we use 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.

Let  $\ell = \lceil \log n \rceil$ . We divide the computation into  $\ell + 1$  phases. Phase  $h$ , for  $h = 0, \dots, \ell$ , consists of steps  $3nh, \dots, 3n(h+1) - 1$ . In phase  $h$ , only the nodes of 2-height equal  $h$  participate in the computation. Specifically, consider a node  $v$  with  $\text{height}_2(v) = h$ . We have two stages: (**Stage All:**) In each step  $t = 3nh, \dots, 3nh + 2n - 1$ , if  $v$  contains any rumor  $\rho_u$  that it still has not transmitted,  $v$  transmits  $\rho_u$ . (**Stage RR:**) In each step  $t = 3nh + 2n + u$ , for  $u = 0, \dots, n - 1$ , if  $v$  has rumor  $\rho_u$ , then  $v$  transmits  $\rho_u$ . In other steps,  $v$  is in the receiving state.

---

*Analysis.* By Lemma 1, the number of phases is  $O(\log n)$ , so the algorithm makes  $O(n \log n)$  steps. The lemma below will thus complete the analysis.

**Lemma 5.** *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  $\text{height}_2(v) < h$  then  $v$  has all rumors from  $\mathcal{T}_v$ .)*

*Proof.* The proof is by induction on  $h$ . The lemma trivially holds when  $h = 0$ . Assume that the claim holds for some  $h < \ell$ , and consider phase  $h$ . We want to show that 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  $h$  then we are done.

It thus remains to prove that if  $v$  has a child  $u$  with  $\text{height}_2(u) = h$  then right after phase  $h$  all rumors from  $\mathcal{T}_u$  will also be in  $v$ . (Of course, this case applies only if  $\text{height}_2(v) \geq h$ .) The subtree  $\mathcal{T}_u^{2,h}$ , namely the subtree consisting of the descendants of  $u$  with 2-height equal  $h$ , is a path  $\mathcal{P} = u_1, u_2, \dots, u_q = u$ , where  $u_1$  is a leaf of  $\mathcal{T}^{2,h}$ . We show that, thanks to pipelining, all rumors that are in  $\mathcal{P}$  when phase  $h$  starts will reach  $u$  during Stage All.

In phase  $h$ , none of the children of the nodes in  $\mathcal{P}$  transmits, except possibly for the one that is also on  $\mathcal{P}$ . For any step  $3nh + s$ ,  $s = 0, 1, \dots, 2n - 1$ , and for  $i = 1, 2, \dots, q - 1$ , we define  $\phi_{s,i}$  to be the number of rumors in  $u_i$  that are still not transmitted, 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  $s$ . 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  $\Phi_s$  will decrease by 1. For  $i > a_s$ , even if  $u_i$  receives a new rumor from its child, 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,  $\Phi_s$  will decrease by at least 1.

Since  $\Phi_s$  strictly decreases in each step, and its initial value is at most  $q+n \leq 2n$ ,  $\Phi_s$  will become 0 in at most  $2n$  steps. In other words, in  $2n$  steps  $u$  will receive all rumors from  $\mathcal{P}$ , 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.

The assumption that we can transmit and receive at the same time can be eliminated. The idea is that along paths where all nodes have the same 2-height, we can synchronize the computation by having even and odd nodes along this path transmit at different times. See [7] for a complete proof.

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

## 6 Deterministic Fire-and-Forward Protocols

We now consider a very simple type of protocols that we call *fire-and-forward* protocols. For convenience, in this model we allow nodes to receive and transmit messages at the same step. In a fire-and-forward protocol, at any time  $t$ , any node  $v$  can either be idle or make one of two types of transmissions:

*Fire:*  $v$  can transmit its own rumor, or

*Forward:*  $v$  can transmit the rumor received in step  $t - 1$ , if any.

In Sect. 7 we give a randomized fire-and-forward protocol that works in expected time  $O(n \log n)$ . This raises the question whether this running time can be achieved by a deterministic fire-and-forward protocol. (Time  $O(n^2)$  is trivial: release all rumors one at a time, spaced at intervals of length  $n$ .) We now show that this can be improved to  $O(n^{1.5})$  and that this bound is optimal.

The key property of fire-and-forward protocols is that any rumor, once fired, moves up the tree one hop per step, unless either it collides, or is dropped, or it reaches the root. If rumors fired from two nodes collide at all, they will collide at their lowest common ancestor. (We extend the definition of collision to include the situation when a node attempts to fire right after receiving a rumor, when nothing is transmitted). This happens only when the difference in times between these two firings is equal to the difference of their depths in the tree.

Any fire-and-forward protocol  $\mathcal{A}$  can be made *oblivious*, in the sense that the decision whether to fire or not depends only on the label of the node and the current time. Let  $T(n)$  be the running time of  $\mathcal{A}$ . Imagine that we run  $\mathcal{A}$  on the tree  $\mathcal{T}'$  obtained from  $\mathcal{T}$  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  $O(T(n))$  on  $\mathcal{T}'$ . 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.

**An  $O(n^{1.5})$  Upper Bound.** We now present our  $O(n^{1.5})$ -time fire-and-forward protocol. As explained earlier, this protocol should specify a set of firing times for each label, so that for any mapping  $[n] \rightarrow [n]$ , 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 is based on a concept of dispersers, defined below, which are reminiscent of various *rulers* studied in number theory, for example Sidon sequences. The particular construction we give in the paper is, in a sense, a multiple set extension of a Sidon-set construction by Erdős 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  $D_1, \dots, D_m \subseteq [s]$  is called an  $(n, m, s)$ -disperser if for each function  $\delta : \{1, \dots, m\} \rightarrow [n]$  and each  $j$  we have  $D_j + \delta(j) \not\subseteq \bigcup_{i \neq j} (D_i + \delta(i))$ . 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 some firing in  $D_j$  will not collide with firings of other nodes.

**Lemma 6.** *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  $a = 1, 2, \dots, p - 1$  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 Eq. (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

$$x + u \equiv y + v \pmod{p}. \quad (3)$$

With addition and multiplication modulo  $p$ ,  $\mathbb{Z}_p$  is a field. Therefore for any  $x$  and  $y$ , and any  $a \neq b$ , Eqs. (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  $i = 1, 2, \dots, m$ , define  $D_i = \{d_i(x) : x \in [p]\}$ . It is sufficient to show that the sets  $D_1, D_2, \dots, D_m$  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  $i$  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  $i \in \{1, 2, \dots, m\}$ , 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  $D_1, \dots, D_m$  is indeed an  $(n, m, s)$ -disperser.

**Algorithm** MLSDTREE. Let  $D_1, D_2, \dots, D_m$  be the  $(n, m, s)$ -disperser from Lemma 6. We partition all labels (and thus also the corresponding nodes) arbitrarily into batches  $B_1, B_2, \dots, B_l$ , 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  $q = 1, 2, \dots, l$  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 MLSDTREE 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  $j$ th node in batch  $B_q$ . For any batch  $B_q$  and any  $v \in B_q$ , if  $v$  is the  $j$ th 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  $r$ , implying

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

It is easy to remove the assumption that nodes are allowed to receive and transmit at the same time, by incorporating the extended definition of collisions into the construction from Lemma 6. The details will appear in the full paper.

**An  $\Omega(n^{1.5})$  Lower Bound.** We match our lower bound by showing that any fire-and-forward protocol needs time  $\Omega(n^{1.5})$ . The basic idea of the proof is that the firings from the leaves are independent of the computation in the rest of the tree. We use this property to show that for any fire-and-forward protocol  $\mathcal{A}$  with running time  $o(n^{1.5})$  we can construct a caterpillar tree on which at least one leaf will fail. The construction is based on a counting argument and analyzing the structure of the bipartite graphs representing collisions between firings (see [7]).

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

## 7 Randomized Algorithms

**Upper Bound of  $O(n \log n)$ .** We now show a randomized algorithm with expected running time  $O(n \log n)$  that does not use any labels. Our algorithm also does not use any aggregation; each message consists only of one rumor. 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  $n$ . We present the algorithm as a fire-and-forward algorithm (see the previous section). In particular, for simplicity, we assume for now that at each step a node can listen and transmit at the same time.

---

**Algorithm RTREE.** At any time  $t$ , each node  $v \neq r$ , independently of other nodes, decides to fire with probability  $1/n$ . If  $v$  decides to fire and no rumor arrived at  $v$  at step  $t - 1$ ,  $v$  fires. If  $v$  decides not to fire and  $v$  received some rumor in step  $t - 1$ , then  $v$  forwards this rumor in step  $t$ . Otherwise,  $v$  is idle.

---

*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  $\rho_z$  at step  $t$  is at least  $\frac{1}{n}(1 - \frac{1}{n})^{n-1}$ . Further, for different  $t$  these events (receiving  $\rho_z$  by  $r$ ) 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  $v$  decides to fire 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  $\rho_z$  will be received by  $r$  at time  $t$ . (Note that the converse is not true.)

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: (i) At time  $t - \text{depth}(z)$ ,  $z$  decided to fire, and (ii) For each  $u \in \mathcal{T} - \{z, r\}$ ,  $u$  did not decide to fire at time  $t - \text{depth}(u)$ . By the algorithm, all these events are independent, so the probability of this combined event is exactly  $\frac{1}{n}(1 - \frac{1}{n})^{n-1}$ .

By Lemma 7, for any step  $t \geq n$ , the probability of any given rumor  $\rho_z$  reaching  $r$  in step  $t$  is at least as large as the probability of collecting a given coupon in the coupon collector problem. We thus obtain the following theorem.

**Theorem 6.** *Algorithm RTREE has expected running time  $O(n \log n)$ . In fact, it will complete gathering in time  $O(n \log n)$  with probability  $1 - o(1)$ .*

**Lower Bound of  $\Omega(n \log n)$ .** We now show that Algorithm RTREE is within a constant factor of optimal. 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 the tree that is a star graph (consisting of the root with  $n$  children that are also the leaves in the tree). In this tree, 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 the actions of  $v$  at different steps may not be independent. Allowing some dependence, in fact, can help reduce the running time, although only by a constant factor (see Theorem 8).

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  $\{0, 1, \dots, T - 1\}$  representing the sets of transmission times of each node. Each node  $v$  independently picks a subset  $S_v$  according to the distribution, and transmits only at the times in  $S_v$ . The label-less requirement is equivalent to the requirement that the  $S_v$  are identically distributed. Node  $v$  succeeds in transmitting if there is a time  $t$  such that  $t \in S_v$ , but  $t \notin S_w$  for any  $w \neq v$ .

**Theorem 7.** *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  $n$  is large enough and 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 there will almost surely be a rumor that fails to reach the root.*

The proof appears in [7]. We also show in [7] that our lower bound is in fact tight, in the sense that the value of the constant  $c$  in the above theorem is best possible for star graphs.

**Theorem 8.** *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  $T$  with probability  $1 - o(1)$ .*

## References

1. Alon, A., Spencer, J.: *The Probabilistic Method*. Wiley, Hoboken (2008)
2. Alon, N., Bar-Noy, A., Linal, N., Peleg, D.: A lower bound for radio broadcast. *J. Comput. Syst. Sci.* **43**(2), 290–298 (1991)
3. Bar-Yehuda, R., Israeli, A., Itai, A.: Multiple communication in multihop radio networks. *SIAM J. Comput.* **22**(4), 875–887 (1993)
4. Bruschi, D., Del Pinto, M.: Lower bounds for the broadcast problem in mobile radio networks. *Distrib. Comput.* **10**(3), 129–135 (1997)
5. Chlebus, B.S., Gasieniec, L., Gibbons, A., Pelc, A., Rytter, W.: Deterministic broadcasting in ad hoc radio networks. *Distrib. Comput.* **15**(1), 27–38 (2002)
6. Chlebus, B.S., Kowalski, D.R., Radzik, T.: Many-to-many communication in radio networks. *Algorithmica* **54**(1), 118–139 (2009)
7. Chrobak, M., Costello, K., Gasieniec, L., Kowalski, D.R.: Information gathering in ad-hoc radio networks with tree topology (2014). arXiv 1407.1521
8. Chrobak, M., Gasieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. *J. Algorithms* **43**(2), 177–189 (2002)
9. Chrobak, M., Gasieniec, L., Rytter, W.: A randomized algorithm for gossiping in radio networks. *Networks* **43**(2), 119–124 (2004)
10. Clementi, A.E.F., Monti, A., Silvestri, R.: Distributed broadcast in radio networks of unknown topology. *Theor. Comput. Sci.* **302**(1–3), 337–364 (2003)
11. Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. *J. Algorithms* **60**(2), 115–143 (2006)
12. De Marco, G.: Distributed broadcast in unknown radio networks. In: *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pp. 208–217 (2008)
13. Erdős, P., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of  $r$  others. *Isr. J. Math.* **51**(1–2), 79–89 (1985)
14. Erdős, P., Turán, P.: On a problem of Sidon in additive number theory and on some related problems. *J. London Math. Soc.* **16**, 212–215 (1941)
15. Gasieniec, L.: On efficient gossiping in radio networks. In: Kutten, S., Žerovnik, J. (eds.) *SIROCCO 2009*. LNCS, vol. 5869, pp. 2–14. Springer, Heidelberg (2010)
16. Gasieniec, L., Radzik, T., Xin, Q.: Faster deterministic gossiping in directed ad hoc radio networks. In: Hagerup, T., Katajainen, J. (eds.) *SWAT 2004*. LNCS, vol. 3111, pp. 397–407. Springer, Heidelberg (2004)
17. Khabbazzian, M., Kowalski, D.R.: Time-efficient randomized multiple-message broadcast in radio networks. In: *Proceedings of the 30th Annual Symposium on Principles of Distributed Computing, PODC'11*, pp. 373–380 (2011)
18. Kowalski, D.R., Pelc, A.: Faster deterministic broadcasting in ad hoc radio networks. *SIAM J. Discrete Math.* **18**(2), 332–346 (2004)
19. Kowalski, D.R., Pelc, A.: Broadcasting in undirected ad hoc radio networks. *Distrib. Comput.* **18**(1), 43–57 (2005)
20. Kushilevitz, E., Mansour, Y.: An  $\Omega(D \log(N/D))$  lower bound for broadcast in radio networks. *SIAM J. Comput.* **27**(3), 702–712 (1998)
21. Liu, D., Prabhakaran, M.: On randomized broadcasting and gossiping in radio networks. In: Ibarra, O.H., Zhang, L. (eds.) *COCOON 2002*. LNCS, vol. 2387, pp. 340–349. Springer, Heidelberg (2002)
22. Strahler, A.: Hypsometric (area-altitude) analysis of erosional topology. *Bull. Geol. Soc. Amer.* **63**, 117–1142 (1952)



23. Talagrand, M.: Concentration of measure and isoperimetric inequalities in product spaces. *Publ. Math. de l'I.H.E.S.* **81**, 73–205 (1995)
24. Viennot, X.: A Strahler bijection between Dyck paths and planar trees. *Discrete Math.* **246**, 317–329 (2003)
25. An, Y.X.:  $O(n^{1.5})$  deterministic gossiping algorithm for radio networks. *Algorithmica* **36**(1), 93–96 (2003)