# Faster Information Gathering in Ad-Hoc Radio Tree Networks*

Marek Chrobak[1] and Kevin P. Costello[2]

[1] Department of Computer Science, University of California at Riverside
[2] Department of Mathematics, University of California at Riverside

**Abstract.** We study information gathering in ad-hoc radio networks. Initially, each node of the network has a piece of information called a *rumor*, and the overall objective is to gather all these rumors in the designated target node. The ad-hoc property refers to the fact that the topology of the network is unknown when the computation starts. Aggregation of rumors is not allowed, which means that each node may transmit at most one rumor in one step.

We focus on networks with tree topologies, that is we assume that the network is a tree with all edges directed towards the root, but, being ad-hoc, its actual topology is not known. We provide two deterministic algorithms for this problem. For the model that does not assume any collision detection nor acknowledgement mechanisms, we give an $O(n \log \log n)$-time algorithm, improving the previous upper bound of $O(n \log n)$. We also show that this running time can be further reduced to $O(n)$ if the model allows for acknowledgements of successful transmissions.

## 1 Introduction

We study the problem of *information gathering* in ad-hoc radio networks. Initially, each node of the network has a piece of information called a *rumor*, and the objective is to gather all these rumors, as quickly as possible, in the designated target node. The nodes communicate by sending messages via radio transmissions. At any time step, several nodes in the network may transmit. When a node transmits a message, this message is sent immediately to all nodes within its range. When two nodes send their messages to the same node at the same time, a *collision* occurs. Aggregation of rumors is not allowed, which means that each node may transmit at most one rumor in one step.

The network can be naturally modeled by a directed graph, where an edge $(u, v)$ indicates that $v$ is in the range of $u$. The ad-hoc property refers to the fact that the actual topology of the network is unknown when the computation starts. We assume that nodes are labeled by integers $0, 1, ..., n-1$. An information gathering protocol determines a sequence of transmissions of a node, based on its label and on the previously received messages.

**Our results.** In this paper, we focus on ad-hoc networks with tree topologies, that is the underlying ad-hoc network is assumed to be a tree with all edges directed towards the root, although the actual topology of this tree is unknown.

* Research supported by NSF grants CCF-1217314, CCF-1536026, and NSA grant H98230-13-1-0228.

We consider two variants of the problem. In the first one, we do not assume any collision detection or acknowledgment mechanisms, so none of the nodes (in particular neither the sender nor the intended recipient) are notified about a collision after it occurred. In this model, we give a deterministic algorithm that completes information gathering in time $O(n \log\log n)$. Our result significantly improves the previous upper bound of $O(n \log n)$ from [6]. To our knowledge, no lower bound for this problem is known, except for the trivial bound of $\Omega(n)$ (since each rumor must be received by the root in a different time step).

In the second part of the paper, we also consider a variant where acknowledgments of successful transmissions are provided to the sender. All the remaining nodes, though, including the intended recipient, cannot distinguish between collisions and absence of transmissions. Under this assumption, we show that the running time can be improved to $O(n)$, which is again optimal for trivial reasons, up to the implicit constant.

While we assume that all nodes are labelled $0, 1, ..., n-1$ (where $n$ is the number of nodes), our algorithms' asymptotic running times remain the same if the labels are chosen from a larger range $0, 1, ..., N-1$, as long as $N = O(n)$.

**Related work.** The problem of information gathering for trees was introduced in [6], where the model without any collision detection was studied. In addition to the $O(n \log n)$-time algorithm without aggregation – that we improve in this paper – [6] develops an $O(n)$-time algorithm for the model with aggregation, where a message can include any number of rumors. Another model studied in [6], called *fire-and-forward*, requires that a node cannot store any rumors; a rumor received by a node has to be either discarded or immediately forwarded. For fire-and-forward protocols, a tight bound of $\Theta(n^{1.5})$ is given in [6].

The information gathering problem is closely related to two other information dissemination primitives that have been well studied in the literature on ad-hoc radio networks: broadcasting and gossiping. All the work discussed below is for ad-hoc radio networks modeled by arbitrary directed graphs, and without any collision detection capability.

In *broadcasting*, a single rumor from a specified source node has to be delivered to all other nodes in the network. The naïve RoundRobin algorithm (see the next section) completes broadcasting in time $O(n^2)$. Following a sequence of papers [7, 14, 3, 4] where this naïve bound was gradually improved, it is now known that broadcasting can be solved in time $O(n \log n \log\log n)$ [17] or $O(n \log^2 D)$ [10], where $D$ is the diameter of $G$. This nearly matches the lower bound of $\Omega(n \log D)$ from [9]. Randomized algorithms for broadcasting have also been well studied [1, 15, 10].

The *gossiping* problem is an extension of broadcasting, where each node starts with its own rumor, and all rumors need to be delivered to all nodes in the network. The time complexity of deterministic algorithms for gossiping is a major open problem in the theory of ad-hoc radio networks. Obviously, the lower bound of $\Omega(n \log D)$ for broadcasting [9] applies to gossiping as well, but no better lower bound is known. It is also not known whether gossiping can be solved in time $O(n \operatorname{polylog}(n))$ with a deterministic algorithm, even if message aggregation is allowed. The best currently known upper bound is $O(n^{4/3} \log^4 n)$ [13] (see [7, 21] for some earlier work). The case when no aggregation is allowed (or with limited aggregation) was studied in [5]. Randomized algorithms for gossiping have also been well studied [10, 16, 8]. Interested readers can find more information about gossiping in the survey paper [12].

**Connections to other problems.** For arbitrary graphs, assuming aggregation, one can solve the gossiping problem by running an algorithm for information gathering and then broadcasting all rumors (as one message) to all nodes in the network. Thus an $O(n \operatorname{polylog}(n))$-time algorithm for information gathering would resolve in positive the earlier-discussed open question about the complexity of gossiping. Due to this connection, developing an $O(n \operatorname{polylog}(n))$-time algorithm for information gathering on arbitrary graphs is likely to be very difficult, if possible at all.

This research, as well as the earlier work in [6], was motivated mainly by this connection to gossiping. We hope that developing efficient algorithms for trees, or for some other natural special cases, will ultimately lead to some insights helpful in resolving the complexity of the gossiping problem in arbitrary graphs.

Some algorithms for arbitrary ad-hoc networks (see [5], for example) involve constructing a spanning subtree of the network and disseminating information along this subtree. Better algorithms for information gathering on trees may thus be useful in addressing problems for arbitrary graphs.

The problem of information gathering for trees is also related to the *contention resolution problem* in multiple-access channels (MAC). There is a myriad of variants of MAC contention resolution in the literature. (See, for example, [18, 2].) Generally, the instance of the problem involves a collection of transmitters connected to a shared channel, like Ethernet, for example. Some of these transmitters need to send their messages across the channel, and the objective is to design a distributed protocol that will allow them to do that. The information gathering problem for trees is in essence an extension of MAC contention resolution to multi-level hierarchies of channels, where transmitters have unique identifiers, and the structure of this hierarchy is not known.

## 2 Preliminaries

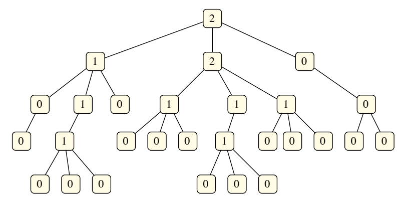
We now provide a formal definition of our model and introduce notation, terminology, and some basic properties used throughout the paper.

**Radio networks with tree topology.** In the paper we focus exclusively on radio networks with tree topologies. Such a network will be represented by a tree $\mathcal{T}$ with root $r$ and with $n = |\mathcal{T}|$ nodes. The edges in $\mathcal{T}$ are directed towards the root, representing the direction of information flow: a node can send messages to its parent, but not to its children. We assume that each node $v \in \mathcal{T}$ is assigned a unique label from $[n] = \{0, 1, ..., n-1\}$, and we denote this label by $\mathsf{label}(v)$.

For a node $v$, by $\deg(v)$ we denote the *degree* of $v$, which is the number of $v$'s children. For any subtree $X$ of $\mathcal{T}$ and a node $v \in X$, $X_v$ denotes the subtree of $X$ rooted at $v$ that consists of all descendants of $v$ in $X$.

For any integer $\gamma = 1, 2, ..., n-1$ and any node $v$ of $\mathcal{T}$ define the *$\gamma$-height* of $v$ as follows. If $v$ is a leaf then the $\gamma$-height of $v$ is 0. If $v$ is an internal node then let $g$ be the maximum $\gamma$-height of a child of $v$. If $v$ has fewer than $\gamma$ children of $\gamma$-height equal $g$ then the $\gamma$-height of $v$ is $g$. Otherwise, the $\gamma$-height of $v$ is $g+1$. The $\gamma$-height of $v$ will be denoted by $\mathit{height}_\gamma(v)$. In case when more than one tree are under consideration, to resolve potential ambiguity we will write $\mathit{height}_\gamma(v, \mathcal{T})$ for the $\gamma$-height of $v$ in $\mathcal{T}$. The $\gamma$-height of a tree $\mathcal{T}$, denoted $\mathit{height}_\gamma(\mathcal{T})$, is defined as $\mathit{height}_\gamma(r)$, that is the $\gamma$-height of its root.

**Fig. 1.** An example showing a tree and the values of 3-heights for all its nodes.



Its name notwithstanding, the definition of $\gamma$-height is meant to capture the "bushiness" of a tree. For example, if $\mathcal{T}$ is a path then its $\gamma$-height equals 0 for each $\gamma \geq 1$. The concept of $\gamma$-height generalizes Strahler numbers [19, 20], introduced in hydrology to measure the size of streams in terms of the complexity of their tributaries. Figure 1 gives an example of a tree and values of 3-heights for all its nodes. The lemma below is a slight refinement of an analogous lemma in [6], and it will play a critical role in our algorithms.

**Lemma 1.** *If $\mathcal{T}$ has $q$ leaves, and $2 \leq \gamma \leq q$, then $\mathit{height}_\gamma(\mathcal{T}) \leq \log_\gamma q$.*

Equivalently, any tree having $\gamma$-height $j$ must have at least $\gamma^j$ leaves. This can be seen by induction on $j$ – if $v$ is a node which is furthest from the root among all nodes of $\gamma$-height $j$, then $v$ by definition has $\gamma$ descendants of $\gamma$-height $j-1$, each of which has $\gamma^{j-1}$ leaf descendants by inductive hypothesis.

**Information gathering protocols.** Each node $v$ of $\mathcal{T}$ has a label (or an identifier) associated with it, and denoted $\mathsf{label}(v)$. When the computation is about to start, each node $v$ has also a piece of information, $\rho_v$, that we call a *rumor*. The computation proceeds in discrete, synchronized time steps, numbered $0, 1, 2, ...$. At any step, $v$ can either be in the *receiving state*, when it listens to radio transmissions from other nodes, or in the *transmitting state*, when it is allowed to transmit. When $v$ transmits at a time $t$, the message from $v$ is sent immediately to its parent in $\mathcal{T}$. As we do not allow rumor aggregation, this message may contain at most one rumor, plus possibly $O(\log n)$ bits of other information. If $w$ is $v$'s parent, $w$ will receive $v$'s message if and only if $w$ is in the receiving state and no other child of $w$ transmitted at time $t$. In Sections 3 and 4 we do not assume any collision detection nor acknowledgement mechanisms, so if $v$'s message collides with one from a sibling, neither $v$ nor $w$ receive any notification. We relax this requirement in Section 5, by assuming that $v$ (and only $v$) will obtain an acknowledgment from $w$ after each successful transmission.

The objective of an information gathering protocol is to deliver all rumors from $\mathcal{T}$ to its root $r$, as quickly as possible. Such a protocol needs to achieve its goal even without the knowledge of the topology of $\mathcal{T}$. More formally, a gathering protocol $\mathcal{A}$ can be defined as a function that, at each time $t$, and for each given node $v$, determines the action of $v$ at time $t$ based only on $v$'s label and the information received by $v$ up to time $t$. The action of $v$ at each time step $t$ involves choosing its state (either receiving or transmitting) and, if it is in the transmitting state, choosing which rumor to transmit.

We will say that $\mathcal{A}$ runs in time $T(n)$ if, for any tree $\mathcal{T}$ and any assignment of labels to its nodes, after at most $T(n)$ steps all rumors are delivered to $r$.

In a simple information gathering protocol called RoundRobin, nodes transmit one at a time, in $n$ rounds, where in each round they transmit in the order $0, 1, ..., n-1$ of their labels. For any node $v$, when it is its turn to transmit, $v$ transmits any rumor from the set of rumors that have been received so far (including its own rumor) but not yet transmitted. In each round, each rumor that is still not in $r$ will get closer to $r$, so after $n^2$ steps all rumors will reach $r$.

**Strong $k$-selectors.** Let $\bar{S} = (S_0, S_1, ..., S_{m-1})$ be a family of subsets of $\{0, 1, ..., n-1\}$. $\bar{S}$ is called a *strong $k$-selector* if, for each $k$-element set $A \subseteq \{0, 1, ..., n-1\}$ and each $a \in A$, there is a set $S_i$ such that $S_i \cap A = \{a\}$. As shown in [11, 9], for each $k$ there exists a strong $k$-selector $\bar{S} = (S_0, S_1, ..., S_m = m-1)$ with $m = O(k^2 \log n)$. We will make extensive use of strong $k$-selectors in our algorithms. At a certain time in the computation our protocols will "run" $\bar{S}$, for an appropriate choice of $k$, by which we mean that it will execute a sequence of $m$ consecutive steps, such that in the $j$th step the nodes from $S_j$ will transmit, while those not in $S_j$ will stay quiet. This will guarantee that, for any node $v$ with at most $k-1$ siblings, there will be at least one step in the execution of $\bar{S}$ where $v$ will transmit but none of its siblings will. Therefore at least one of $v$'s transmissions will be successful.

## 3 An $O(n\sqrt{\log n})$-Time Protocol

We first give a gathering protocol SimpleGather for trees with running time $O(n\sqrt{\log n})$. Our faster protocol will be presented in the next section.

We fix three parameters $K = 2^{\lfloor\sqrt{\log n}\rfloor}$, $D = \lceil\log_K n\rceil = O(\sqrt{\log n})$ and $D' = \lceil\log K^3\rceil = O(\sqrt{\log n})$. We also fix a strong $K$-selector $\bar{S} = (S_0, S_1, ..., S_{m-1})$, where $m \leq CK^2 \log n$, for some integer constant $C$.

By Lemma 1, we have that $height_K(\mathcal{T}) \leq D$. A node $v$ of $\mathcal{T}$ is called *light* if $|\mathcal{T}_v| \leq n/K^3$; otherwise we say that $v$ is *heavy.* Let $\mathcal{T}'$ be the subtree of $\mathcal{T}$ induced by the heavy nodes. By the definition of heavy nodes, $\mathcal{T}'$ has at most $K^3$ leaves, so $height_2(\mathcal{T}') \leq D'$. Also, obviously, $r \in \mathcal{T}'$.

To streamline the description of our algorithm we will allow each node to receive and transmit messages at the same time. We will assume a preprocessing step allowing each $v$ to know both the size of its subtree $\mathcal{T}_v$ and its $K$-height. In particular, $v$ knows whether it is in $\mathcal{T}'$ or not. We will assume that each node $v \in \mathcal{T}'$, also knows its 2-height in the subtree $\mathcal{T}'$. In Appendix A we describe the preprocessing, as well as how to remove the receive/transmit assumption.

A detailed description of Algorithm SimpleGather is given in Pseudocode 1. To distinguish between computation steps (which do not consume time) and communication steps, we use command "**at time** $t$". When the algorithm reaches this command it waits until step time $t$ to continue processing. Each message transmission takes one time step. For each node $v$ we maintain a set $B_v$ of rumors received by $v$, including its own rumor $\rho_v$. The algorithm consists of two epochs, and we describe the computation in each epoch separately.

*Epoch 1: light nodes.* In Epoch 1, only the light nodes participate, and the goal is to move all rumors to $\mathcal{T}'$. This epoch has $D+1$ stages (numbered $h = 0, 1, \ldots, D$), with stage $h$ beginning at time $\alpha_h = (C+1)hn$. A light node $v$ with $K$-height $h$ is only active during stage $h$.

Each stage has two parts. In the first part of stage $h$, $v$ will transmit according to the $K$-selector $\bar{S}$. Specifically, this part has $n/K^3$ iterations, each corresponding to a complete execution of $\bar{S}$. During each iteration, $v$ chooses a single rumor $\rho_z \in B_v$ that it has not yet marked, and transmits $\rho_v$ in each time step corresponding to a set $S_i$ containing the label of $v$.. This $\rho_z$ is then marked, and not chosen again during the first part. Note that if the parent $u$ of $v$ has degree at most $K$, the definition of $K$-selectors guarantees that $\rho_z$ will be received by $u$, but if $u$'s degree is larger it is possible for all transmissions of $\rho_z$ during this stage to be blocked.

Note that the total number of steps required for this part of stage $h$ is $(n/K^3)\cdot m \leq Cn$, so these steps will be completed before the second part of stage $h$ starts.

In the second part, (beginning at time $\alpha_h + Cn$), we simply run the RoundRobin protocol: in the $l$-th step of this part, if $v$ has the rumor of the node with label $l$, then it transmits that rumor.

*Epoch 2: heavy nodes.* This epoch has $D'+1$ stages, with only heavy nodes in $\mathcal{T}'$ participating. When the epoch starts, all rumors are assumed to already be in $\mathcal{T}'$. In stage $g$ the nodes in $\mathcal{T}'$ whose 2-height is equal $g$ will participate. Similar to the stages of epoch 1, each stage runs in time $O(n)$ and has two parts. In the first part, during each time step *every* heavy node holding a rumor it has not yet marked chooses such a rumor, marks it, and transmits it (instead of using a $K$-selector). The second part executes RoundRobin, as before.

The high-level overview of the analysis of this algorithm is that rumors maintain a steady rate of progress towards the root – during stage $h$ of Epoch 1, each rumor either reaches the heavy tree or a node of $K$-height $h+1$; during stage $g$ of Epoch 2 each rumor either reaches the root or a vertex of 2 height $g+1$ in $\mathcal{T}'$. Since the heights are bounded (by Lemma 1), the algorithm will complete in the required time of $O(n\sqrt{\log n})$. We defer the full analysis to Appendix A. In the next section, we will show how to improve this bound to $O(n \log\log n)$.

## 4 A Protocol with Running Time $O(n \log\log n)$

In this section we present our first main result:

**Theorem 1.** *The problem of information gathering on trees, without rumor aggregation, can be solved in time $O(n \log\log n)$.*

**Pseudocode 1** SimpleGather($v$)

```
K = 2^⌊√(log n)⌋, D = ⌈log_K n⌉
B_v ← {ρ_v}                                   ▷ Initially v has only ρ_v
Throughout: all rumors received by v are automatically added to B_v
if |T_v| ≤ n/K^3 then                          ▷ v is light (epoch 1)
    h ← height_K(v, 𝒯) ;  α_h ← (C + 1)nh      ▷ v participates in stage h
    for i = 0, 1, ..., n/K^3 − 1 do            ▷ iteration i
        at time α_h + im
        if B_v has an unmarked rumor then      ▷ Part 1: K-selector
            choose any unmarked ρ_z ∈ B_v and mark it
            for j = 0, 1, ..., m − 1 do
                at time α_h + im + j
                if label(v) ∈ S_j then Transmit(ρ_z)
    for l = 0, 1, ..., n − 1 do                ▷ Part 2: RoundRobin
        at time α_h + Cn + l
        z ← node with label(z) = l
        if ρ_z ∈ B_v then Transmit(ρ_z)
else                                           ▷ v is heavy (epoch 2)
    g ← height_2(v, 𝒯') ;  α'_g ← α_{D+1} + 2ng ▷ v participates in stage g
    for i = 0, 1, ..., n − 1 do                ▷ Part 1: all nodes transmit
        at time α'_g + i
        if B_v contains an unmarked rumor then
            choose any unmarked ρ_z ∈ B_v and mark it
            Transmit(ρ_z)
    for l = 0, 1, ..., n − 1 do                ▷ Part 2: RoundRobin
        at time α'_g + 2n + l
        z ← node with label(z) = l
        if ρ_z ∈ B_v then Transmit(ρ_z)
```

The protocol can be thought of as an iterative application of the idea behind Algorithm SimpleGather from Section 3. We assume that the reader is familiar with Algorithm SimpleGather and its analysis, and in our presentation we will focus on the high level ideas behind Algorithm FastGather, deferring the reader to Section 3 and Appendix B for the implementation of some details.

As before, $\mathcal{T}$ is the input tree with $n$ vertices. We fix some arbitrary integer constant $\beta \geq 2$. For $\ell = 1, 2, ...,$ let $K_\ell = \lceil n^{\beta^{-\ell}} \rceil$. So $K_1 = \lceil n^{1/\beta} \rceil$, the sequence $(K_\ell)_\ell$ is non-increasing, and $\lim_{\ell\to\infty} K_\ell = 2$. Let $L$ be the largest value of $\ell$ for which $n^{\beta^{-\ell}} \geq \log n$. (Note that $L$ is well defined for sufficiently large $n$, since $\beta$ is fixed). Thus $L \leq \log_\beta(\log n/\log\log n)$, $L = \Theta(\log\log n)$, and $\log n \leq K_L = K_{L+1}^\beta < (\log n)^\beta$.

For $\ell = 1, 2, ..., L$, by $\bar{S}^\ell = (S_1^\ell, S_2^\ell, ..., S_{m_\ell}^\ell)$ we denote a strong $K_\ell$-selector of size $m_\ell \leq CK_\ell^2 \log n$, for some integer constant $C$. As discussed in Section 2, such selectors $\bar{S}^\ell$ exist.

Let $\mathcal{T}^{(0)} = \mathcal{T}$, and for each $\ell = 1, 2, ..., L$, let $\mathcal{T}^{(\ell)}$ be the subtree of $\mathcal{T}$ induced by the nodes $v$ with $|\mathcal{T}_v| \geq n/K_\ell^3$. Each tree $\mathcal{T}^{(\ell)}$ is rooted at $r$, and

$\mathcal{T}^{(\ell)} \subseteq \mathcal{T}^{(\ell-1)}$ for $\ell \geq 1$. For $\ell \neq 0$, the definition of $\mathcal{T}^{(\ell)}$ implies also that it has at most $K_\ell^3$ leaves, so, by Lemma 1, its $K_{\ell+1}$-height is at most $\log_{K_{\ell+1}}(K_\ell^3)$. Since $K_\ell \leq 2n^{\beta^{-\ell}}$ and $K_{\ell+1} \geq n^{\beta^{-(\ell+1)}}$, direct calculation gives $\log_{K_{\ell+1}}(K_\ell^3) \leq 3\beta+1$ for sufficiently large $n$. In particular, the $K_{\ell+1}$-height of $\mathcal{T}^{(\ell)}$ is at most $D = 3\beta + 1 = O(1)$.

Similar to the previous section we will make some simplifying assumptions. First, we will assume that all nodes can receive and transmit messages at the same time. Second, we will also assume that each node $v$ knows the size of its subtree $|\mathcal{T}_v|$ and its $K_\ell$-heights, for each $\ell \leq L$. We describe how to remove these assumptions in Appendix B.

Algorithm FastGather consists of $L+1$ epochs, numbered $1, 2, ..., L+1$. For $\ell \leq L$, the goal of epoch $L$ is to move all rumors from $\mathcal{T}^{(L-1)}$ to $\mathcal{T}^{(L)}$ Each of these $L$ epochs will run in time $O(n)$, so their total running time will be $O(nL) = O(n \log\log n)$. The final epoch will move all rumors from $\mathcal{T}^{(L)}$ to the root, also in time $O(n \log\log n)$. We now provide the details.

*Epochs* $\ell = 1, 2, ..., L$. In epoch $\ell$, only the nodes in $\mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$ participate. The computation in this epoch is very similar to the computation of light nodes (in epoch 1) in Algorithm SimpleGather. Epoch $\ell$ starts at time $\gamma_\ell = (D+1)(C+1)(\ell-1)n$ and lasts $(D+1)(C+1)n$ steps.

Let $v \in \mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$. The computation of $v$ in epoch $\ell$ consists of $D+1$ identical stages. Each stage $h = 0, 1, ..., D$ starts at time step $\alpha_{\ell,h} = \gamma_\ell + (C+1)hn$ and lasts $(C+1)n$ steps.

Stage $h$ has two parts. The first part starts at time $\alpha_{\ell,h}$ and lasts time $Cn$. During this part we execute $\lceil n/K_\ell^3 \rceil$ iterations, each iteration consisting of running the strong $K_{\ell+1}$-selector $\bar{S}^\ell$. The time needed to execute these iterations is at most $\lceil n/K_\ell^3 \rceil (CK_{\ell+1}^2 \log n)$, which can be seen by direct calculation to be at most $Cn$.

Thus all iterations executing the strong selector will complete before time $\alpha_{\ell,h} + Cn$. Then $v$ stays idle until time $\alpha_{\ell,h} + Cn$, which is when the second part starts. In the second part we run the RoundRobin protocol, which takes $n$ steps. So stage $h$ will complete right before step $\alpha_{\ell,h} + (C+1)n = \alpha_{\ell,h+1}$.

*Epoch* $L+1$. Due to the definition of $L$, we have that $\mathcal{T}^{(L)}$ contains at most $K_L^3 = O(\log^{3\beta} n)$ leaves, so its 2-depth is $D' = O(\log\log n)$, by Lemma 1. The computation in this epoch is similar to epoch 2 from Algorithm SimpleGather. As before, this epoch consists of $D'+1$ stages, where each stage $g = 0, 1, ..., D'$ has two parts. In the first part, we have $n$ steps in which each node transmits. In the second part, also of length $n$, we run one iteration of RoundRobin.

The high-level analysis of Algorithm FastGather is similar to that of Algorithm SimpleGather: During each stage a rumor's $K$-height increases until it reaches the next level (tree $\mathcal{T}^{(\ell)}$), and in the last epoch its two height increases until it reaches the root. The full analysis is given in Appendix B.

## 5 An $O(n)$-time Protocol with Acknowledgments

In this section we consider a network model where acknowledgments of successful transmissions are provided to the sender. All the remaining nodes, including the intended recipient, cannot distinguish between collisions and absence of transmissions. In this section we present our second main result:

**Theorem 2.** *The problem of information gathering on trees without rumor aggregation can be solved in time $O(n \log \log n)$ if acknowledgments are provided.*

As before, $\mathcal{T}$ is the input tree with $n$ nodes. We will recycle the notions of light and heavy nodes from Section 3, although now we will use slightly different parameters. Let $\delta > 0$ be a small constant, and let $K = \lceil n^\delta \rceil$. We say that $v \in \mathcal{T}$ is *light* if $|T_v| \le n/K^3$ and we call $v$ *heavy* otherwise. By $\mathcal{T}'$ we denote the subtree of $\mathcal{T}$ induced by the heavy nodes.

**Algorithm LinGather.** Our algorithm will consist of two epochs. The first epoch is essentially identical to Epoch 1 in Algorithm SimpleGather, except for a different choice of the parameters. The objective of this epoch is to collect all rumors in $\mathcal{T}'$ in time $O(n)$. In the second epoch, only the heavy nodes in $\mathcal{T}'$ will participate in the computation, and the objective of this epoch is to gather all rumors from $\mathcal{T}'$ in the root $r$. This epoch is quite different from our earlier algorithms and it will use some novel combinatorial structures (obtained via probabilistic constructions) to move all rumors from $\mathcal{T}'$ to $r$ in time $O(n)$.

*Epoch 1:* In this epoch only light nodes will participate, and the objective of Epoch 1 is to move all rumors into $\mathcal{T}'$. In this epoch we will not be taking advantage of the acknowledgement mechanism. As mentioned earlier, except for different choices of parameters, this epoch is essentially identical to Epoch 1 of Algorithm SimpleGather, so we only give a very brief overview here. We use a strong $K$-selector $\bar{S}$ of size $m \le CK^2 \log n$.

Let $D = \lceil \log_K n \rceil \le 1/\delta = O(1)$. By Lemma 1, the $K$-depth of $\mathcal{T}$ is at most $D$. Epoch 1 consists of $D + 1$ stages, where in each stage $h = 0, 1, ..., D$, nodes of $K$-depth $h$ participate. Stage $h$ consists of $n/K^3$ executions of $\bar{S}$, followed by an execution of RoundRobin, taking total time $n/K^3 \cdot m + n = O(n)$. So the entire epoch takes time $(D + 1) \cdot O(n) = O(n)$ as well. The proof of correctness (namely that after this epoch all rumors are in $\mathcal{T}'$) is identical as for Algorithm SimpleGather.

*Epoch 2:* When this epoch starts, all rumors are already gathered in $\mathcal{T}'$, and the objective is to push them further to the root. The key obstacle to be overcome in this epoch is congestion stemming from the fact that nodes have many rumors to transmit. This congestion means that simply repeatedly applying $k$-selectors is no longer enough. For example, if the root has $k$ children, each with $n/k$ rumors, then repeating a $k$-selector $n/k$ times would get all the rumors to the root, but take total time roughly $nk \log n$, which would be too long.

To overcome this obstacle, we introduce two novel tools that will play a critical role in our algorithm. The first tool is a so-called *amortizing selector family.* Since a parent receives at most one rumor per round, if it has $k$ children it clearly cannot simultaneously be receiving rumors at a rate greater than $\frac{1}{k}$ from each child individually. With the amortizing family, we will be able to achieve within a constant fraction of this bound over long time intervals, so long as each child knows (approximately) how many siblings it is competing with.

Similarly to a strong selector, this amortizing family will be a collection of subsets of the underlying label set $[n]$, though now it will be doubly indexed: There will be sets $S_{ij}$ for each $1 \le i \le s$ and each $j \in \{1, 2, 4, 8, \ldots, k\}$ for some parameters $s$ and $k$. We say the family succeeds at cumulative rate $q$ if the following statement is true:

For each $j \in \{1, 2, 4, \ldots, \frac{k}{2}\}$, each subset $A \subseteq \{1, \ldots, n\}$ satisfying $j/2 \le |A| \le 2j$, and each element $v \in A$ there are at least $\frac{q}{|A|}s$ distinct $i$ for which

$$v \in S_{ij} \text{ and } A \cap (S_{i(j/2)} \cup S_{ij} \cup S_{i(2j)}) = \{v\}.$$

In the case $j = 1$ the set $S_{i(j/2)}$ is defined to be empty. Here $s$ can be thought of as the total running time of the selector, and $j$ as a node's estimate of its parent's degree, and $k$ as some bound on the maximum degree handled by the selector. A node fires at time step $i$ if and only if its index is contained in the set $S_{ij}$. What the above statement is then saying that for any subset $A$ of siblings, if $|A|$ is at most $k/2$ and each child estimates $|A|$ within a factor of 2 then each child will transmit at rate at least $\frac{q}{|A|}$.

**Theorem 3.** *There are fixed constants $c, C > 0$ such that the following is true: For any $k$ and $n$ and any $s \ge Ck^2 \log n$, there is an amortizing selector with parameters $n, k, s$ succeeding with cumulative rate $c$.*

This selector will be constructed probabilistically in Appendix C.

Of course, such a family will not be useful unless a node can obtain an accurate estimate of its parent's degree, which will be the focus of our second tool, $k$-distinguishers. As with the amortizing selector, this will be a collection of subsets of the label set $[n]$. Let $\bar{S} = (S_1, S_2, ..., S_m)$, where each $S_j \subseteq [n]$ for each $j$. For $A \subseteq [n]$ and $a \in A$, define $Hits_{a,A}(\bar{S}) = \{j : S_j \cap A = \{a\}\}$, that is $Hits_{a,A}(\bar{S})$ is the collection of indices $j$ for which $S_j$ intersects $A$ exactly on $a$. Note that, using this terminology, $\bar{S}$ is a strong $k$-selector if and only if $Hits_{a,A}(\bar{S}) \neq \emptyset$ for all sets $A \subseteq [n]$ of cardinality at most $k$ and all $a \in A$.

We say that $\bar{S}$ is a *$k$-distinguisher* if there is a *threshold* value $\xi$ (dependent on $k$) such that, for any $A \subseteq [n]$ and $a \in A$, the following conditions hold:

$$\text{if } |A| \le k \text{ then } |Hits_{a,A}(\bar{S})| > \xi, \text{ and if } |A| \ge 2k \text{ then } |Hits_{a,A}(\bar{S})| < \xi.$$

We make no assumptions on what happens for $|A| \in \{k+1, k+2, ..., 2k-1\}$.

The idea is this: consider a fixed $a$, and imagine that we have some set $A$ that contains $a$, but its other elements are not known. Suppose that we also have an access to a *hit oracle* that for any set $S$ will tell us whether $S \cap A = \{a\}$ or not. With this oracle, we can then use a $k$-distinguisher $\bar{S}$ to extract some information about the cardinality of $A$ by calculating the cardinality of $Hits_{a,A}(\bar{S})$. If $|Hits_{a,A}(\bar{S})| \le \xi$ then we know that $|A| > k$, and if $|Hits_{a,A}(\bar{S})| \ge \xi$ then we know that $|A| < 2k$.

What we show in Appendix C, again by a probabilistic argument, is that not-too-large distinguishers exist.

**Theorem 4.** *For any $n \ge 2$ and $1 \le k \le n/2$ there exists a $k$-distinguisher of size $m = O(k^2 \log n)$.*

In our framework, the acknowledgement of a message received from a parent corresponds exactly to such a hit oracle. So if all nodes fire according to such a $k$-distinguisher, each node can determine in time $O(k^2 \log n)$ either that its parent has at least $k$ children or that it has at most $2k$ children.

Now let $\lambda$ be a fixed parameter between 0 and 1. For each $i = 0, 1, ..., \lceil \lambda \log n \rceil$, let $\bar{S}^i$ be a $2^i$-distinguisher of size $O(2^{2i} \log n)$ and with threshold value $\xi_i$. We can then concatenate these $k$-distinguishers to obtain a sequence $\tilde{S}$ of size $\sum_{i=0}^{\lceil \lambda \log n \rceil} O(2^{2i} \log n) = O(n^{2\lambda} \log n)$.

We will refer to $\tilde{S}$ as a *cardinality estimator*, because applying our hit oracle to $\bar{S}$ we can estimate a cardinality of an unknown set within a factor of 4, making $O(n^{2\lambda} \log n)$ hit queries. More specifically, consider again a scenario where we have a fixed $a$ and some unknown set $A$ containing $a$, where $|A| \leq n^{\lambda}$. Using the hit oracle, compute the values $h_i = |Hits_{a,A}(\bar{S}^i)|$, for all $i$. If $i_0$ is the smallest $i$ for which $h_i > \xi_i$, then by definition of our distinguisher we must have $2^{i_0 - 1} < |A| < 2(2^{i_0})$. In our gathering framework, this corresponds to each node in the tree being able to determine in time $O(n^{2\lambda} \log n)$ a value of $j$ (specifically, $i_0 - 1$) such that the number of children of its parent is between $2^j$ and $2^{j+2}$, which is exactly what we need to be able to run the amortizing selector.

For the remainder of this section we will assume the existence of Amortizing Selector Families and Distinguishers, and use them to construct an algorithm which completes Epoch 2 in time $O(n)$.

*The Algorithm for Epoch 2:* For the second epoch, we restrict our attention to the tree $\mathcal{T}'$ of heavy nodes. As before, no parent in this tree can have more than $K^3 = n^{3\delta}$ children, since each child is itself the ancestor of a subtree of size $n/K^3$. We will further assume the existence of a fixed amortizing selector family with parameters $k = 2K^3$ and $s = K^8$, as well as a fixed cardinality estimator with parameter $\lambda = 3\delta$ running in time $D_1 = O\left(n^{6\delta} \log n\right) = O\left(K^6 \log n\right)$.

Our protocol will be divided into stages, each consisting of $2(D_1 + K^8)$ steps. A node will be *active* in a given stage if at the beginning of the stage it has already received all of its rumors, but still has at least one rumor left to transmit (it is possible for a node to never become active, if it receives its last rumor and then finishes transmitting before the beginning of the next stage).

During each odd-numbered time step of a stage, all nodes (active or not) holding at least one rumor they have not yet successfully passed on transmit such a rumor. The even-numbered time steps are themselves divided into two parts. In the first $D_1$ even steps, all active nodes participate in the aforementioned cardinality estimator. At the conclusion of the estimator, each node knows a $j$ such that their parent has between $2^j$ and $2^{j+2}$ active children. Note that active siblings do not necessarily have the same estimate for their family size. For the remainder of the even steps, each active node fires using the corresponding $2^{j+1}$-selector from the amortizing family.

The stages repeat until all rumors have reached the root. Our key claim (which is proven fully in Appendix D) is that the rumors aggregate at least at a steady rate over time – each node with subtree size $m$ in the original tree $\mathcal{T}$ will have received all $m$ rumors within $O(m)$ steps of the start of the epoch.

## References


1. N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *J. Comput. Syst. Sci.*, 43(2):290–298, 1991.
2. A. Fernández Anta, M.A. Mosteiro, and J.R. Mu noz. Unbounded contention resolution in multiple-access channels. *Algorithmica*, 67(3):295–314, 2013.
3. D. Bruschi and M. Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distrib. Comput.*, 10(3):129–135, April 1997.
4. B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in ad hoc radio networks. *Distributed Computing*, 15(1):27–38, 2002.
5. M. Christersson, L. Gasieniec, and A. 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.
6. M. Chrobak, K. Costello, L. Gasieniec, and D. R. Kowalski. Information gathering in ad-hoc radio networks with tree topology. In *8th International Conference on Combinatorial Optimization and Applications (COCOA'14)*, pages 129–145, 2014.
7. M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. *J. Algorithms*, 43(2):177–189, 2002.
8. M. Chrobak, L. Gasieniec, and W. Rytter. A randomized algorithm for gossiping in radio networks. *Networks*, 43(2):119–124, 2004.
9. A. E. F. Clementi, A. Monti, and R. Silvestri. Distributed broadcast in radio networks of unknown topology. *Theor. Comput. Sci.*, 302(1-3):337–364, 2003.
10. A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. *Journal of Algorithms*, 60(2):115 – 143, 2006.
11. P. Erdős, 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.
12. L. Gasieniec. On efficient gossiping in radio networks. In *16th Int. Colloquium on Structural Information and Communication Complexity (SIROCCO'09)*, pages 2–14, 2009.
13. L. Gasieniec, T. Radzik, and Q. Xin. Faster deterministic gossiping in directed ad hoc radio networks. In *Proc. Scandinavian Workshop on Algorithm Theory (SWAT'04)*, pages 397–407, 2004.
14. D. R. Kowalski and A. Pelc. Faster deterministic broadcasting in ad hoc radio networks. *SIAM J. Discrete Math.*, 18(2):332–346, 2004.
15. E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. Comput.*, 27(3):702–712, 1998.
16. D. Liu and M. Prabhakaran. On randomized broadcasting and gossiping in radio networks. In *8th Annual Int. Conference on Computing and Combinatorics (COCOON'02)*, pages 340–349, 2002.
17. G. De Marco. Distributed broadcast in unknown radio networks. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pages 208–217, 2008.
18. G. De Marco and D.R. Kowalski. Contention resolution in a non-synchronized multiple access channel. In *2013 IEEE 27th International Symposium on Parallel Distributed Processing (IPDPS)*, pages 525–533, 2013.
19. A.N. Strahler. Hypsometric (area-altitude) analysis of erosional topology. *Bull. Geol. Soc. Amer.*, 63:1171142, 1952.
20. X.G. Viennot. A Strahler bijection between Dyck paths and planar trees. *Discrete Mathematics*, 246:317329, 2003.
21. Y. Xu. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica*, 36(1):93–96, 2003.

# A Analysis of Algorithm SimpleGather

In this section we provide the analysis of the simplified gathering algorithm from section 3, and in particular show it successfully completes in time $O(n\sqrt{\log n})$ on all trees on $n$ nodes.

*Analysis of Epoch 1 (the light nodes).*

We claim that the following invariant holds for all $h = 0, 1, ..., D$:

($\mathbf{I}_h$) Let $w \in \mathcal{T}$ and let $u$ be a light child of $w$ with $height_K(u) \leq h - 1$. Then at time $\alpha_h$ node $w$ has received all rumors from $\mathcal{T}_u$.

To prove this invariant we proceed by induction on $h$. If $h = 0$ the invariant ($\mathbf{I}_0$) holds vacuously. So suppose that invariant ($\mathbf{I}_h$) holds for some value of $h$. We want to prove that ($\mathbf{I}_{h+1}$) is true when stage $h+1$ starts. We thus need to prove the following claim: if $u$ is a light child of $w$ with $height_K(u) \leq h$ then at time $\alpha_{h+1}$ all rumors from $\mathcal{T}_u$ will arrive in $w$.

If $height_K(u) \leq h - 1$ then the claim holds, immediately from the inductive assumption ($\mathbf{I}_h$). So assume that $height_K(u) = h$. Consider the subtree $H$ rooted at $u$ and containing all descendants of $u$ whose $K$-height is equal to $h$. By the inductive assumption, at time $\alpha_h$ any $w' \in H$ has all rumors from the subtrees rooted at its descendants of $K$-height smaller than $h$, in addition to its own rumor $\rho_{w'}$. Therefore all rumors from $\mathcal{T}_u$ are already in $H$ and each of them has exactly one copy in $H$, because all nodes in $H$ were idle before time $\alpha_h$.

When the algorithm executes the first part of stage $h$ on $H$, then each node $v$ in $H$ whose parent is also in $H$ will successfully transmit an unmarked rumor during each pass through the $K$ selector – indeed, our definition of $H$ guarantees that $v$ has at most $K - 1$ siblings in $H$, so by the definition of strong selector it must succeed at least once. We make the following additional claim:

**Claim 1:** At all times during stage $h$, the collection of nodes in $H$ still holding unmarked rumors forms an induced tree of $H$.

The claim follows from induction: At the beginning of the stage the nodes in $H$ still hold their own original rumor, and it is unmarked since those nodes were idle so far. As the stage progresses, each parent of a transmitting child will receive a new (and therefore not yet marked) rumor during each run through the selector, so no holes can ever form.

In particular, node $u$ will receive a new rumor during every run through the selector until it has received all rumors from its subtree. Since the tree originally held at most $|\mathcal{T}_u| \leq n/K^3$ rumors originally, $u$ must have received all rumors from its subtree after at most $n/K^3$ runs through the selector.

Note that, as $height_K(u) = h$, $u$ will also attempt to transmit its rumors to $w$ during this part, but, since we are not making any assumptions about the degree of $w$, there is no guarantee that $w$ will receive them. This is where the second part of this stage is needed. Since in the second part each rumor is transmitted without collisions, all rumors from $u$ will reach $w$ before time $\alpha_{h+1}$, completing the inductive step and the proof that ($\mathbf{I}_{h+1}$) holds.

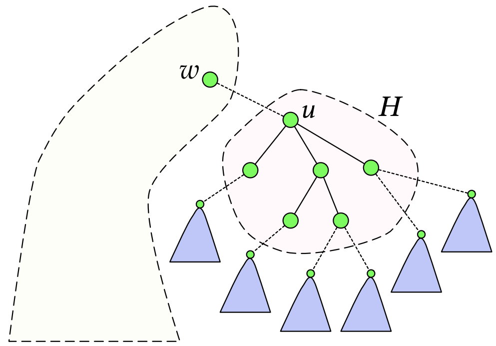



**Fig. 2.** Proving Invariant ($\mathbf{I}_h$). Dark-shaded subtrees of $\mathcal{T}_u$ consist of light nodes with $K$-height at most $h-1$. $H$ consists of the descendants of $u$ with $K$-height equal $h$.

In particular, using Invariant ($\mathbf{I}_h$) for $h = D$, we obtain that after Epoch 1 each heavy node $w$ will have received rumors from the subtrees rooted at all its light children. Therefore at that time all rumors from $\mathcal{T}$ will be already in $\mathcal{T}'$, with each rumor having exactly one copy in $\mathcal{T}'$.

*Analysis of Epoch 2 (the heavy nodes).*
The argument for the heavy nodes is similar as for the light nodes, but with a twist, since we do not use selectors now; instead we have steps when all nodes transmit. In essence, we show that each stage reduces by at least one the 2-depth of the minimum subtree of $\mathcal{T}'$ that contains all rumors.

Specifically, we show that the following invariant holds for all $g = 0, 1, ..., D'$:

($\mathbf{J}_g$) Let $w \in \mathcal{T}'$ and let $u \in \mathcal{T}'$ be a child of $w$ with $height_2(u, \mathcal{T}') \leq g-1$. Then at time $\alpha'_g$ node $w$ has received all rumors from $\mathcal{T}_u$.

We prove invariant ($\mathbf{J}_g$) by induction on $g$. For $g = 0$, ($\mathbf{J}_0$) holds vacuously. Assume that ($\mathbf{J}_g$) holds for some $g$. We claim that ($\mathbf{J}_{g+1}$) holds right after stage $g$.

Choose any child $u$ of $w$ with $height_2(u, \mathcal{T}') \leq g$. If $height_2(u, \mathcal{T}') \leq g-1$, we are done, by the inductive assumption. So we can assume that $height_2(u, \mathcal{T}') = g$. Let $P$ be the subtree of $\mathcal{T}'$ rooted at $u$ and consisting of all descendants of $u$ whose 2-height in $\mathcal{T}'$ is equal $g$. Then $P$ is simply a path. By the inductive assumption, for each $w' \in P$, all rumors from the subtrees of $w'$ rooted at its children of 2-height at most $g-1$ are in $w'$. Thus all rumors from $\mathcal{T}_u$ are already in $P$. All nodes in $P$ participate in stage $g$, but their children outside $P$ do not transmit. Therefore each transmission from any node $x \in P - \{u\}$ during stage $g$ will be successful. Due to pipelining, all rumors from $P$ will reach $u$ after the first part of stage $g$. In the second part, all rumors from $u$ will be successfully sent to $w$. So after stage $g$ all rumors from $\mathcal{T}_u$ will be in $w$, completing the proof that ($\mathbf{J}_{g+1}$) holds.

*Removing simplifying assumptions.* During the description of Algorithm SimpleGather we made some simplifying assumptions. We now describe how to modify our algorithm so that it works even if these assumptions do not hold. These modification are similar to those described in [6], but we include them here for the sake of completeness.

First, we assumed that each node $v$ knows certain parameters of its subtree $\mathcal{T}_v$, including the size, its $K$-height, etc. Any function $f(v)$ that can be computed in a bottom-up manner (that is, $f(v)$ can be determined from the values of $f(u)$, for all children $u$ of $v$), can be computed in time $O(n)$ using the algorithm from [6] for information gathering in trees with aggregation [3]. We modify this algorithm so that each node $u$, when its sends its message (which, in the algorithm from [6] contains all rumors from $\mathcal{T}_u$), it will instead send the value of $f(u)$. A node $v$, after it receives all values of $f(u)$ from each child $u$, will compute $f(v)$.

We also assumed that each node can receive and transmit messages at the same time. We now need to modify the algorithm so that it receives messages only in the receiving state and transmits only in the transmitting state. For the RoundRobin steps this is trivial: a node $v$ is in the transmitting state only if it is scheduled to transmit, otherwise it is in the receiving state. For other steps, we will explain the modification for light and heavy nodes separately.

Consider the computation of the light nodes during the steps when they transmit according to the strong selector. Instead of the strong $K$-selector, we can use the strong $(K+1)$-selector, which will not affect the asymptotic running time. When a node $v$ is scheduled to transmit, it enters the transmitting state, otherwise it is in the receiving state. In the proof, where we argue that the message from $v$ will reach its parent, instead of applying the selector argument to $v$ and its siblings, we apply it to the set of nodes consisting of $v$, its siblings, and its parent, arguing that there will be a step when $v$ is the only node transmitting among its siblings and its parent is in the receiving state.

Finally, consider the computation of the heavy nodes, at steps when all of them transmit. We modify the algorithm so that, in any stage $g$, the iteration (in Line 18) of these steps is preceded by $O(n)$-time preprocessing. Recall that the nodes whose 2-height in $\mathcal{T}'$ is equal $g$ form disjoint paths. We can run a one round of RoundRobin where each node transmits an arbitrary message. This way, each node will know whether it is the first node on one of these paths or not. If a node $x$ is first on some path, say $P$, $x$ sends a message along this path, so that each node $y \in P$ can compute its distance from $x$. Then, in the part where all nodes transmit, we replace each step by two consecutive steps (even and odd), and we use parity to synchronize the computation along these paths: the nodes at even positions are in the receiving state at even steps and in the transmitting state at odd steps, and the nodes at odd positions do the opposite.

---

[3] It needs to be emphasized here that in our model only communication steps contribute to the running time; all calculations are assumed to be instantaneous.

## B Analysis of Algorithm FastGather

In this appendix we give a detailed analysis of Algorithm LinGather, and in particular show that it completes in time $O(n \log\log n)$ on all trees on $n$ nodes.

*Analysis of Epochs* $1, 2, \ldots, L$.

We claim that the algorithm preserves the following invariant for $\ell = 1, 2, ..., L$:

($\mathbf{I}^{\ell}$) Let $w \in \mathcal{T}$ and let $u \in \mathcal{T} - \mathcal{T}^{(\ell-1)}$ be a child of $w$. Then $w$ will receive all rumors from $\mathcal{T}_u$ before time $\gamma_\ell$, that is before epoch $\ell$ starts.

For $\ell = 1$, invariant ($\mathbf{I}^1$) holds vacuously, because $\mathcal{T}^{(0)} = \mathcal{T}$. In the inductive step, assume that ($\mathbf{I}^{\ell}$) holds for some epoch $\ell$. We want to show that ($\mathbf{I}^{\ell+1}$) holds right after epoch $\ell$ ends. In other words, we will show that if $w$ has a child $u \in \mathcal{T} - \mathcal{T}^{(\ell)}$ then $w$ will receive all rumors from $\mathcal{T}_u$ before time $\gamma_{\ell+1}$.

So let $u \in \mathcal{T} - \mathcal{T}^{(\ell)}$. If $u \notin \mathcal{T}^{(\ell-1)}$ then ($\mathbf{I}^{\ell+1}$) holds for $u$, directly from the inductive assumption. We can thus assume that $u \in \mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$.

The argument now is very similar to that for Algorithm SimpleGather in Section 3, when we analyzed epoch 1. For each $h = 0, 1, ..., D$ we prove a refined version of condition ($\mathbf{I}^{\ell+1}$):

($\mathbf{I}_h^{\ell+1}$) Let $w \in \mathcal{T}$ and let $u \in \mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$ be child of $w$ with $height_{K_\ell}(u, \mathcal{T}^{(\ell-1)}) \leq h-1$. Then $w$ will receive all rumors from $\mathcal{T}_u$ before time $\alpha_{\ell,h}$, that is before stage $h$.

The proof is the same as for Invariant ($\mathbf{I}_h$) in Section A, proceeding by induction on $h$. For each fixed $h$ we consider a subtree $\mathcal{H}$ rooted at $u$ and consisting of all descendants of $u$ in $\mathcal{T}^{(\ell-1)}$ whose $K_\ell$-height is at least $h$. By the inductive assumption, at the beginning of stage $h$ all rumors from $\mathcal{T}_u$ are already in $\mathcal{H}$. Then, the executions of $\bar{S}^\ell$, followed by the execution of RoundRobin, will move all rumors from $\mathcal{H}$ to $w$. We omit the details here.

By applying condition ($\mathbf{I}_h^{\ell+1}$) with $h = D$, we obtain that after all stages of epoch $\ell$ are complete, that is at right before time $\gamma_{\ell+1}$, $w$ will receive all rumors from $\mathcal{T}_u$. Thus $\ell$ invariant ($\mathbf{I}^{\ell+1}$) will hold.

*Analysis of Epoch* $L+1$. Let $\alpha'_g = \gamma_L + 2gn$. To prove correctness, we show that the following invariant holds for all stages $g = 0, 1, ..., D'$:

($\mathbf{J}_g$) Let $w \in \mathcal{T}^{(L)}$ and let $u \in \mathcal{T}^{(L)}$ be a child of $w$ with $height_2(u, \mathcal{T}^{(L)}) \leq g-1$. Then at time $\alpha'_g$ node $w$ has received all rumors from $\mathcal{T}_u$.

The proof is identical to the proof of the analogous Invariant ($\mathbf{J}_g$) in Section 3, so we omit it here. Applying Invariant ($\mathbf{J}_g$) with $g = D'$, we conclude that after stage $D'$, the root $r$ will receive all rumors.

As for the running time, we recall that $L = O(\log\log n)$. Each epoch $\ell = 1, 2, ..., L$ has $D + 1 = O(1)$ stages, where each stage takes time $O(n)$, so the execution of the first $L$ epochs will take time $O(n \log\log n)$. Epoch $L + 1$ has $D' + 1 = O(\log\log n)$ stages, each stage consisting of $O(n)$ steps, so this epoch will complete in time $O(n \log \log n)$. We thus have that the overall running time of our protocol is $O(n \log \log n)$.

*Removing the Simplifying Assumptions.* It remains to explain that the simplifying assumptions we made at the beginning of the description of Algorithm FAST-GATHER are not needed. Computing all subtree sizes and all $K_\ell$-heights can be done recursively bottom-up, using the linear-time information gathering algorithm from [6] that uses aggregation. This was explained in Section 3. The difference now is that each node has to compute $L + 1 = O(\log \log n)$ values $K_\ell$, and, since we limit bookkeeping information in each message up to $O(\log n)$ bits, these values need to be computed separately. Nevertheless, the total precomputation time will still be $O(n \log \log n)$.

Removing the assumption that nodes can receive and transmit at the same time can be done in the same way as in Section 3. Roughly, in each epoch $\ell = 1, 2, ..., L$, any node $v \in \mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$ uses a strong $(K_\ell + 1)$-selector (instead of a strong $K_\ell$-selector) to determine whether to be in the receiving or transmitting state. In epoch $L$ the computation (in the steps when all nodes transmit) is synchronized by transmitting a control message along induced paths, and then choosing the receiving or transmitting state according to node parity.

## C Proof of the Existence of the New Combinatorial Structures

In this section we show the existence of distinguishers and amortizing selector families, the two combinatorial tools which were used in our algorithm in Section 5 for information gathering on trees in the model with an acknowledgement mechanism for successful transmissions.

*Proof of the existence of $k$-distinguishers.* We now show a construction of $k$-distinguishers, using a probabilistic argument. Let $m = Ck^2 \log n$, where $C$ is some sufficiently large constant whose value we will determine later. We choose the collection of random sets $\bar{S} = (S_1, S_2, ..., S_m)$, by letting each $S_j$ be formed by independently including each $x \in [n]$ in $S_j$ with probability $1/2k$. Thus, for any set $A$ and $a \in A$, the probability that $S_j \cap A = \{a\}$ is $(1/2k)(1 - 1/2k)^{|A|-1}$, and the expected value of $|Hits_{a,A}(\bar{S})|$ is

$$\mathbf{E}[|Hits_{a,A}(\bar{S})|] = m \cdot \tfrac{1}{2k} \left(1 - \tfrac{1}{2k}\right)^{|A|-1}. \tag{1}$$

Recall that to be a $k$-distinguisher our set needs to satisfy (for suitable $\xi$) the following two properties:

(d1) if $|A| \leq k$ then $|Hits_{a,A}(\bar{S})| > \xi$
(d2) if $|A| \geq 2k$ then $|Hits_{a,A}(\bar{S})| < \xi$.

We claim that, for a suitable value of $\xi$, the probability that there exists a set $A \subseteq [n]$ and some $a \in A$ for which $\bar{S}$ does not satisfy both conditions is smaller than 1 (and in fact tends to 0) This will be sufficient to show an existence of a $k$-distinguisher with threshold value $\xi$.

Observe that in order to be a $k$-distinguisher it is sufficient that $\bar{S}$ satisfies (d1) for sets $A$ with $|A| = k$ and satisfies (d2) for sets $A$ with $|A| = 2k$. This is true because the value of $|Hits_{a,A}(\bar{S})|$ is monotone with respect to the inclusion: if $a \in A \subseteq B$ then $Hits_{a,A}(\bar{S}) \supseteq Hits_{a,B}(\bar{S})$.

Now consider some fixed $a \in [n]$ and two sets $A_1, A_2 \subseteq [n]$ such that $|A_1| = k$, $|A_2| = 2k$ and $a \in A_1 \cap A_2$. For $i = 1, 2$, we consider two corresponding random variables $X_i = |Hits_{a,A_i}(\bar{S})|$ and their expected values $\mu_i = \mathrm{Exp}[X_i]$. For any integer $k \geq 1$ we have

$$\begin{aligned} \tfrac{1}{\mathrm{e}^{1/2}} &\leq (1 - \tfrac{1}{2k})^{k-1} \\ \tfrac{1}{\mathrm{e}} &\leq (1 - \tfrac{1}{2k})^{2k-1} \leq \tfrac{1}{2}. \end{aligned}$$

From (1), substituting $m = Ck^2 \log n$, this gives us the corresponding estimates for $\mu_1$ and $\mu_2$:

$$\begin{aligned} \tfrac{1}{2e^{1/2}} Ck \log n &\leq \mu_1 \\ \tfrac{1}{2\mathrm{e}} Ck \log n &\leq \mu_2 \leq \tfrac{1}{4} Ck \log n \end{aligned}$$

Since $\mathrm{e}^{-1/2} > \frac{1}{2}$, we can choose an $\epsilon \in (0, 1)$ and $\xi$ for which

$$(1 + \epsilon)\mu_2 < \xi < (1 - \epsilon)\mu_1.$$

Thus the probability that $\bar{S}$ violates (d1) for $A = A_1$ is

$$\begin{aligned} \Pr[X_1 \leq \xi] \;\leq\; \Pr[X_1 \leq (1 - \epsilon)\mu_1] \;&\leq\; \mathrm{e}^{-\epsilon^2 \mu_1/2} \\ &\leq\; \mathrm{e}^{-\epsilon^2 \mathrm{e}^{-1/2} Ck \log n/4}, \end{aligned}$$

where in the second inequality we use the Chernoff bound for deviations below the mean. Similarly, using the Chernoff bound for deviations above the mean, we can bound the probability of $\bar{S}$ violating (d2) for $A = A_2$ as follows:

$$\begin{aligned} \Pr[X_2 \geq \xi] \;\leq\; \Pr[X_2 \geq (1 + \epsilon)\mu_2] \;&\leq\; e^{-\epsilon^2 \mu_2/3} \\ &\leq\; e^{-\epsilon^2 \mathrm{e}^{-1} Ck \log n/6}. \end{aligned}$$

To finish off the proof, we apply the union bound. We have at most $n$ choices for $a$, at most $\binom{n}{k-1} \leq n^{k-1} \leq n^{2k-1}$ choices of $A_1$, and at most $\binom{n}{2k-1} \leq n^{2k-1}$ choices of $A_2$. Note also that $\mathrm{e}^{-1/2}/4 > \mathrm{e}^{-1}/6$. Putting it all together, the probability that $\bar{S}$ is not a $k$-distinguisher is at most

$$\begin{aligned} n \cdot \tbinom{n}{k-1} \cdot \Pr[X_1 \leq \xi] + n \cdot \tbinom{n}{2k-1} \cdot \Pr[X_2 \geq \xi] &\leq n^{2k} \cdot (\Pr[X_1 \leq \xi] + \Pr[X_2 \geq \xi]) \\ &\leq 2n^{2k} \cdot e^{-\epsilon^2 \mathrm{e}^{-1} Ck \log n/6} \\ &= 2n^{k(2 - \epsilon^2 \mathrm{e}^{-1} C/6)} < 1, \end{aligned}$$

for $C$ large enough.

*Proof of the existence of amortizing selector families.* Let $k, n, s$ be given such that $s \geq Ck \log n$, where $k$ is a constant to be determined later. We form our selector probabilistically: For each $v, i$, and $j$, we independently include $v$ in $S_{ij}$ with probability $2^{-j}$.

Observe that again by monotonicity it suffices to check the selector property for the case $|A| = 2j$: If we replace $A$ with a larger set $A'$ containing $A$ and satisfying $|A'| \leq 4|A|$, then for any $v \in A$ and any $i$ satisfying

$$A' \cap (S_{i(j/2)} \cup S_{ij} \cup S_{i(2j)}) = \{v\},$$

we also have

$$A \cap (S_{i(j/2)} \cup S_{ij} \cup S_{i(2j)}) = \{v\}.$$

So if there are at least $\frac{cs}{|A'|}$ distinct $i$ satisfying the first equality, there are at least $\frac{(c/4)s}{|A|}$ satisfying the second equality.

Now fix $j \in \{1, 2, 4, \ldots, k\}$, a set $A \subseteq [n]$ with $|A| = 2j$ and some $v \in A$, and let the random variables $X$ and $Y$ be defined by

$$X = |\left\{i : v \in S_{ij} \text{ and } A \cap (S_{i(j/2)} \cup S_{ij} \cup S_{i(2j)}) = \{v\}\right\}|$$

The expected value of $X$ is

$$\mu_X = s\left(\frac{1}{j}\right)\left(1 - \frac{2}{j}\right)^{2j-1}\left(1 - \frac{1}{j}\right)^{2j-1}\left(1 - \frac{1}{2j}\right)^{2j-1}.$$

Utilizing the bound

$$\left(1 - \frac{2}{j}\right)^{2j-1}\left(1 - \frac{1}{j}\right)^{2j-1}\left(1 - \frac{1}{2j}\right)^{2j-1} \geq \frac{1}{e^7},$$

we have

$$\mu_X \geq e^{-7}\frac{s}{j}$$

Now let $c = \frac{1}{4e^7}$. Applying the Chernoff bound, we get

$$\begin{aligned} \Pr[X \leq c\frac{s}{|A|}] \leq \Pr[X \leq \frac{1}{2}\mu_X] &\leq \mathrm{e}^{-\mu_X/8} \\ &\leq \mathrm{e}^{-s/8e^7 j}. \end{aligned}$$

We now use the union bound over all choices of $j$, $v$, and $S$. We have at most $n$ choices of $v$, at most $\log n$ choices for $j$, and at most $\binom{n}{2j-1} \leq n^{2k-1}$ choices of $S$ given $j$ and $v$. Thus the probability that our family is not an amortizing selector is at most

$$n \log n \cdot n^{2k-1} \cdot \left(e^{-s/8e^7 j}\right) \leq 2n^{2k} \log n \cdot \mathrm{e}^{-Ck \log n/8e^7}$$

which is smaller than 1 for sufficiently large $C$. This implies the existence of the amortized selector family.

## D Analysis of the Heavy Epoch of Algorithm LinGather

The key claim in the analysis of this algorithm is that the following invariant holds throughout the epoch for some sufficiently large absolute constant $C$:

($\mathbf{I}^{\ell}$) For any heavy node $v$ such that $v$ has subtree size $m$ in $\mathcal{T}$, and any $0 \leq j \leq m$, that node has received at least $j$ rumors within time $C(2m + j)$ of the beginning of Epoch 2.

In particular, the root has received all of the rumors by time $3Cn$.

We will show this invariant holds by induction on the node's height within $\mathcal{T}'$. If the node is a leaf the statement follows from our analysis of Epoch 1 (the node has received *all* rumors from its subtree by the beginning of epoch 2).

Now assume that a node $u$ with subtree size $m + 1$ has $k$ children within $\mathcal{T}'$, and that those children have subtree sizes $a_1 \geq a_2 \geq \cdots \geq a_k \geq K^3$. Node $u$ may also received some number $a_0$ of messages from non-heavy children (these children, if any, will have already completed their transmissions during the previous epoch).

Let $v$ be a child having subtree size $a_1$ (chosen arbitrarily, if there are two children with maximal subtree size). Let $t_2$ be defined by

$$t_2 = \begin{cases} 3Ca_2 + \frac{3}{c}\left(a_2 + \cdots + a_k\right) + K^{12} & \text{if } k > 1 \\ 0 & \text{if } k = 1 \end{cases}$$

We make the following additional claims.

**Claim 2:** By time $t_2$, all children except possibly $v$ will have completed all of their transmissions.

*Proof.* By inductive hypothesis, all children except $v$ will have received all of their rumors by time $3Ca_2$. During each stage from that point until all non-$v$ nodes complete, one of the following things must be true.

– Some active node transmits the final rumor it is holding, and stops transmitting. This can happen during at most $K^3$ stages, since there are at most $K^3$ children and each child only becomes active once it already has received all rumors from its subtree.
– All active nodes have rumors to transmit throughout the entire stage. If there were $j$ active nodes total during the stage, then by the definition of our amortizing selector family, the parent received at least $c\frac{2K^8}{j}$ rumors from each child during the stage. In particular, it must have received at least

$$c(j-1)\frac{2K^8}{j} \geq cK^8$$

new rumors from children other than $v$.

Combining the two types of stages, the non-$v$ children will have all finished in at most

$$K^3 + \frac{1}{cK^8}(a_2 + \cdots + a_k)$$

complete stages after time $3Ca_2$. Since each stage takes time $2(D_1 + K^8) = (2 + o(1))K^8$, the bound follows.

**Claim 3:** Let $k, m$, and $v$ be as above. By time $2Cm$, all children except possibly $v$ have completed their transmissions.

*Proof.* This is trivial for $k = 1$. For larger $k$ follows from the previous claim, together with the estimate that (for sufficiently large $C$)

$$2Cm \geq 2C(a_1 + \cdots + a_k) \geq 4Ca_2 + 2C(a_3 + \cdots + a_k) \geq \frac{4}{3}(t_2 - K^{12}) \geq \frac{5}{4}t_2$$

Here the middle inequality holds for any $C > 2/c$, while the latter inequality holds since $t_2 \geq a_2 \geq n/K^3 \gg K^{12}$.

By the above claim, node $v$ is the only node that could still be transmitting at time $2Cm$. In particular, if it has a rumor during an odd numbered time step after this point, it successfully transmits. By assumption, $v$ will have received at least $j$ rumors by time $C(2m + j)$ for each $j$. This implies it will successfully broadcast $j$ rumors by time $C(2m + j) + 2$ for each $0 \leq j \leq a_1$.

By time $C[2(m + 1) + j]$, the parent has either received all rumors from all of its children (if $j > a_1$), or at least $j$ rumors from $a_1$ alone (if $j \leq a_1$). Either way, it has received at least $j$ rumors total, and the induction is complete.