Faster Information Gathering in Ad-Hoc Radio Tree Networks

Marek Chrobak[*] Kevin P. Costello[†]

June 28, 2016

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, where each message may contain only one rumor. 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 and neither message is received by that node. Aggregation of rumors is not allowed, by which we mean that each node may transmit at most one rumor in each time 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.

[*] Department of Computer Science, University of California at Riverside, USA. Research supported by NSF grants CCF-1217314 and CCF-1536026.

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

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.

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 senders 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 [5]. To our knowledge, no lower bound for this problem is known, apart from the trivial bound of $\Omega(n)$, which follows from the fact that 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 asymptotically optimal, as explained earlier.

While we assume that all nodes are labelled $0, 1, ..., n-1$ (where n is the number of vertices), 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 [5], 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 – [5] develops an $O(n)$-time algorithm for the model with aggregation, where a message can include any number of rumors. Another model studied in [5], 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 [5].

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 (assuming that all nodes are actually reachable from the source.) The naïve RoundRobin algorithm (see the next section) completes broadcasting in time $O(n^2)$. Following a sequence of papers [6, 17, 2, 3, 20, 11] where this naïve bound was gradually improved, it is now known that broadcasting can be solved in time $O(n \log D \log \log(D\Delta/n))$ [10], where D is the diameter of G and Δ is its maximum in-degree. This nearly matches the lower bound of $\Omega(n \log D)$ from [9]. Randomized algorithms for broadcasting have also been well studied [1, 18, 11].

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. (In gossiping, it is assumed that the network is strongly connected.) 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 $O(n^{4/3} \log^4 n)$ [15] (see [6, 25] for some earlier work). The case when no aggregation is allowed (or with limited aggregation) was studied in [4]. Randomized algorithms for gossiping have also been well studied [11, 19, 7]. Interested readers can find more information about gossiping in the survey paper [14].

Connections to other problems. This research, as well as the earlier work in [5], was motivated by the connections between information gathering in trees and other problems in distributed computing involving shared channels, including gossiping in radio networks and MAC contention resolution.

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. 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 ad-hoc radio networks (see [4, 16], 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 similar problems for arbitrary graphs.

The problem of *contention resolution* for multiple-access channels (MAC) has been widely studied in the literature. (See, for example, [21, 22, 13] and the references therein.) There are in fact myriad of variants of this problem, depending on the characteristics of the communication model. Generally, the instance of the MAC contention resolution problem involves a collection of transmitters connected to a shared channel (e.g. ethernet). 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.

Trees. 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 $v \in X$, we denote by X_v 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 γ-*height* of v as follows. If v is a leaf then the γ-height of v is 0. If v is an internal node then let g be the maximum γ-height of a child of v. If v has fewer than γ children of γ-height equal g then the γ-height of v is g. Otherwise, the γ-height of v is $g + 1$. The γ-height of v will be denoted by $\mathit{height}_\gamma(v)$. In case when several trees are under consideration, to resolve potential ambiguity we will write $\mathit{height}_\gamma(v, \mathcal{T})$ for the

γ-height of v in $\mathcal{T}$. The γ-height of a tree $\mathcal{T}$, denoted $height_\gamma(\mathcal{T})$, is defined as $height_\gamma(r)$, that is the γ-height of its root.

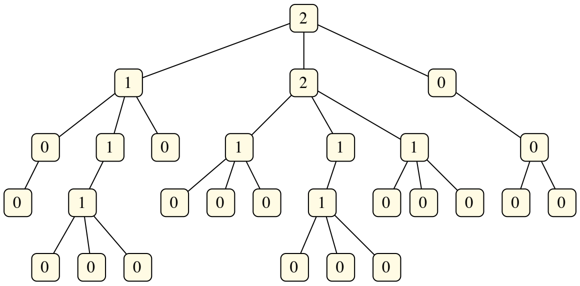

Figure 1: An example showing a tree and the values of 3-heights for all its nodes.

Note that for $\gamma = 1$, the γ-height corresponds to the standard notion of depth for the tree. Its name notwithstanding, the definition of γ-height is meant to capture the "bushiness" of a tree. For example, if $\mathcal{T}$ is just a path then its γ-height is equal 0 for each $\gamma \geq 2$. The concept of γ-height generalizes Strahler numbers [23, 24], introduced in hydrology to measure the branching complexity of streams. 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 [5], and it will play a critical role in our algorithms.

Lemma 1. *Suppose that $\mathcal{T}$ has q leaves, and let $2 \leq \gamma \leq q$. Then $height_\gamma(\mathcal{T}) \leq \log_\gamma q$.*

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

Information gathering protocols in trees. 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, ρ_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 collision occurred, that is if 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 a message from one of its siblings, neither v nor w receive any notification. (In other words, w cannot distinguish collisions between its children's transmissions from background noise.) 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.

Throughout the paper we assume that all nodes know the value of n. This assumption is not essential, because otherwise an information gathering algorithm can employ a doubling trick by trying, sequentially, all powers of 2 as potential values of n. For our algorithms, this will at most double the time to complete gathering (although the algorithm itself will not know that and will iterate the process forever).

Protocol RoundRobin. A simple example of an information gathering protocol is called RoundRobin. In 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.

In our algorithms we will use a variation of this idea that we call SB-RoundRobin (for source-based round-robin). In SB-RoundRobin there are also n steps. In step $l = 0, 1, ..., n-1$, a node v checks if it already has collected a rumor ρ_z that originated from the node z with label $z) = l$. If so, v transmits this rumor, otherwise v is quiet. Note that, similar to RoundRobin, there are no collisions in SB-RoundRobin, since the tree topology implies that a node and its siblings share no rumors in common. The advantage of SB-RoundRobin is that any node v, in a single round (n steps), will transmit all its rumors to its parent.

Strong k-selectors. Let $\bar{S} = (S_0, S_1, ..., S_{m-1})$ be a family of subsets of $[n] = \{0, 1, ..., n-1\}$, the label set. $\bar{S}$ is called a *strong k-selector* if, for each k-element set $A \subseteq [n]$ and each $a \in A$, there is a set S_i such that $S_i \cap A = \{a\}$. As shown in [12, 9], for each k there exists a strong k-selector $\bar{S} = (S_0, S_1, ..., S_{m-1})$ with $m = O(k^2 \log n)$. We will make extensive use of strong k-selectors in our algorithm. 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 jth 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 $O(n\sqrt{\log n})$. This will allow us to gradually introduce some techniques needed to for our faster protocol that will be presented in the next section.

We fix three parameters:

$$K = 2^{\lfloor \sqrt{\log n} \rfloor},$$
$$D = \lceil \log_K n \rceil = O(\sqrt{\log n}), \text{and}$$
$$D' = \lceil \log K^3 \rceil = O(\sqrt{\log n}).$$

(Here and elsewhere all logarithms are assumed to be binary unless otherwise stated). 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 $\mathit{height}_K(\mathcal{T}) \leq D$. Recall that, for a node v of $\mathcal{T}$, $\mathcal{T}_v$ is the subtree of $\mathcal{T}$ rooted at v. We call v *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 $\mathit{height}_2(\mathcal{T}') \leq D'$. 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 also assume a preprocessing step allowing each v to know both the size of its subtree $\mathcal{T}_v$ (in particular, whether it is in $\mathcal{T}'$ or not), its K-height, and, if it is in $\mathcal{T}'$, its 2-height in the subtree $\mathcal{T}'$. We later explain how to implement this preprocessing and how to modify the algorithm to separate the transmitting and receiving steps.

The algorithm consists of two epochs. epoch 1 consists of $D + 1$ stages, each lasting $O(n)$ steps. In this epoch only light vertices participate. The purpose of this epoch is to gather all rumors from $\mathcal{T}$ in $\mathcal{T}'$. Epoch 2 has $D' + 1$ stages and only heavy vertices participate in the computation during this epoch. The purpose of epoch 2 is to deliver all rumors from $\mathcal{T}'$ to r. We describe the computation in the two epochs separately.

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 time step 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 ρ_v.

Epoch 1: light vertices. Let $0 \leq h \leq D$, and let v be a light vertex whose K-height equals h. Then v will be active only during stage h which starts at time $\alpha_h = (C + 1)hn$. This stage is divided into two parts.

In the first part of stage h, that lasts Cn steps, v will transmit according to the strong K-selector $\bar{S}$. Specifically, this part has $\lceil n/K^3 \rceil$ iterations, where each iteration corresponds to a complete execution of $\bar{S}$. At any time, some of the rumors in B_v may be marked; the marking on a rumor indicates that the algorithm has already attempted to transmit it using $\bar{S}$. At the beginning of each iteration, v chooses any rumor $\rho_z \in B_v$ it has not yet marked, then transmits ρ_z in the steps that use sets S_i containing the label of v. This ρ_z is then marked. The definition of strong K-selectors guarantees that if the parent u of v has degree at most K, then ρ_z will be received by u at some point over the course of the selector. However, if u's degree is larger it may not have received ρ_z. 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, that starts at time $\alpha_h + Cn$ and lasts n steps, we simply run a cycle of SB-RoundRobin.

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 $\mathit{height}_K(u) \leq h - 1$. Then at time α_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 $\mathit{height}_K(u) \leq h$ then at time α_{h+1} all rumors from $\mathcal{T}_u$ will arrive in w.

If $\mathit{height}_K(u) \leq h-1$ then the claim holds, immediately from the inductive assumption ($\mathbf{I}_h$). So assume that $\mathit{height}_K(u) = h$. Consider the subtree H rooted at u and containing all descendants of u whose K-height is equal to h. (See Figure 2 for illustration.) By the inductive assumption, at time α_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 α_h.

When the algorithm executes the first part of stage h on H, then each v node in H whose parent is also in H will successfully transmit an unmarked rumor during each pass through the strong 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 sub-tree of H rooted at u.*

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. So this induced sub-tree from the claim is the whole H. As the stage progresses, each parent of a transmitting child will receive a new (and therefore not yet marked) rumor from this child during each iteration of $\bar{S}$. Therefore no "holes" can ever form, that is, if a node has an unmarked rumor then so does its parent. The claim follows.

In particular, node u will receive a new rumor during every iteration of $\bar{S}$, until it has received all rumors from its subtree. Since the tree originally held at most $|\mathcal{T}_u| \le n/K^3$ rumors, u will receive all rumors from its subtree after at most n/K^3 iterations of $\bar{S}$.

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 α_{h+1}, completing the inductive step and the proof that $(\mathbf{I}_{h+1})$ holds.

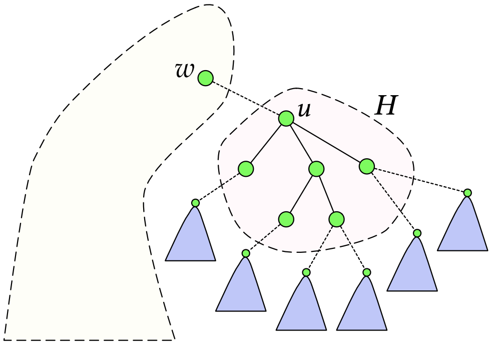

Figure 2: Proving Invariant $(\mathbf{I}_h)$. Dark-shaded subtrees of $\mathcal{T}_u$ consist of light nodes with K-height at most $h-1$. H consists of the descendants of u with K-height equal h.

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 $\mathcal{T}'$.

Epoch 2: heavy vertices. In this epoch we have at most $D'+1$ stages, and only heavy nodes (that is, those in $\mathcal{T}'$) participate in the computation. When the epoch starts, all rumors are already in $\mathcal{T}'$. In stage $D+1+g$ the nodes in $\mathcal{T}'$ whose 2-height is equal g will participate. Similar to

Pseudocode 1 SimpleGather(v)

```
$K = 2^{\lfloor \sqrt{\log n} \rfloor}$, $D = \lceil \log_K n \rceil$
$B_v \leftarrow \{\rho_v\}$                                                    ▷ Initially $v$ has only $\rho_v$
Throughout: all rumors received by $v$ are automatically added to $B_v$ unmarked
if $|T_v| \leq n/K^3$ then                                                   ▷ $v$ is light (epoch 1)
    $h \leftarrow height_K(v, \mathcal{T})$ ;  $\alpha_h \leftarrow (C+1)nh$                         ▷ $v$ participates in stage $h$
    for $i = 0, 1, ..., \lceil n/K^3 \rceil - 1$ do                                       ▷ iteration $i$
        at time $\alpha_h + im$
        if $B_v$ contains an unmarked rumor then                              ▷ Part 1: strong $K$-selector
            choose any unmarked $\rho_z \in B_v$ and mark it
            for $j = 0, 1, ..., m-1$ do
                at time $\alpha_h + im + j$
                if $\mathsf{label}(v) \in S_j$ then Transmit($\rho_z$)
    for $l = 0, 1, ..., n-1$ do                                               ▷ Part 2: SB-RoundRobin
        at time $\alpha_h + Cn + l$
        $z \leftarrow$ node with $\mathsf{label}(z) = l$
        if $\rho_z \in B_v$ then Transmit($\rho_z$)
else                                                                      ▷ $v$ is heavy (epoch 2)
    $g \leftarrow height_2(v, \mathcal{T}')$ ;  $\alpha'_g \leftarrow \alpha_{D+1} + 2ng$                         ▷ $v$ participates in stage $g$
    for $i = 0, 1, ..., n-1$ do                                               ▷ Part 1: all nodes transmit
        at time $\alpha'_g + i$
        if $B_v$ contains an unmarked rumor then
            choose any unmarked $\rho_z \in B_v$ and mark it
            Transmit($\rho_z$)
    for $l = 0, 1, ..., n-1$ do                                               ▷ Part 2: SB-RoundRobin
        at time $\alpha'_g + 2n + l$
        $z \leftarrow$ node with $\mathsf{label}(z) = l$
        if $\rho_z \in B_v$ then Transmit($\rho_z$)
```

the stages of epoch 1, this stage has two parts and the second part executes SB-RoundRobin, as before. The difference is that now, in the first part, instead of using the strong K-selector, each heavy node will transmit at each of the n steps.

We need to show that r will receive all rumors at the end. The argument is similar as for light vertices, 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 $\mathit{height}_2(u, \mathcal{T}') \leq g - 1$. Then at time α'_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 $\mathit{height}_2(u, \mathcal{T}') \leq g$. If $\mathit{height}_2(u, \mathcal{T}') \leq g - 1$, we are done, by the inductive assumption. So we can assume that $\mathit{height}_2(u, \mathcal{T}') = g$. 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 $v \in P$, all rumors from the subtrees of v rooted at its children of 2-height at most $g - 1$ are in v. 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. At the beginning of this section we made some simplifying assumptions. It still remains to explain how to modify our algorithm so that it works even if these assumptions do not hold. These modification are similar to those described in [5], but we include them here for the sake of completeness.

First, we assumed a preprocessing step whereby each v knows certain parameters of its subtree $\mathcal{T}_v$, including the size, its K-height, etc. The justification for this lies in the algorithm from [5] for information gathering in trees with aggregation. Such an algorithm can be modified to compute in linear time any function f such that $f(v)$ is uniquely determined by the values of f on the children of v, so long as the output of f can be transmitted in $O(\log n)$ bits (in particular, any function whose output is an integer at most polynomial in n). The modification is that each node u, when its sends its message (which, in the algorithm from [5] 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 then compute $f(v)$[1].

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

[1] 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.

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 when 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 single 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.

Summarizing this section, we have presented Algorithm SimpleGather that completes information gathering in ad-hoc radio networks with tree topologies in time $O(n\sqrt{\log n})$. 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 consider the same basic model of information gathering in trees as in Section 3, that is, the model does not provide any collision detection and rumor aggregation is not allowed. We show how to refine our $O(n\sqrt{\log n})$ protocol SimpleGather to improve the running time to $O(n \log\log n)$. This is the first main result of our paper, as summarized in the theorem below.

Theorem 1. *The problem of information gathering in ad-hoc networks with tree topology, without rumor aggregation, can be solved in time $O(n \log\log n)$.*

Our protocol that achieves running time $O(n \log\log n)$ will be called FastGather. This 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 high level ideas behind Algorithm FastGather, referring the reader to Section 3 for the implementation of some details.

As before, we use notation $\mathcal{T}$ for the input tree and $n = |\mathcal{T}|$ for the number of vertices in $\mathcal{T}$. We assume that n is sufficiently large, and we will establish some needed lower bounds for n as we work through the proof. 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 ℓ for which $n^{\beta^{-\ell}} \geq \log n$. (Note that L is well defined for sufficiently large n, since β is fixed). We thus have the following exact and asymptotic bounds:

$$\begin{aligned} L &\leq \log_\beta(\log n / \log\log n) & K_L &\geq \log n \\ L &= \Theta(\log\log n) & K_L &= \Theta(\log n). \end{aligned} \tag{1}$$

For $\ell = 1, 2, ..., L$, by $\bar{S}^\ell = (S^\ell_0, S^\ell_1, ..., S^\ell_{m_\ell - 1})$ we denote a strong K_ℓ-selector of size $m_\ell \leq C K_\ell^2 \log n$, 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 $\ell \geq 1$.

Claim 2. *For $\ell = 1, ..., L$, we have $height_{K_\ell}(\mathcal{T}^{(\ell-1)}) \le D$, where $D = 3\beta + 1 = O(1)$.*

First, we verify the claim for $\ell = 1$. $\mathcal{T}^{(0)}$ has at most n leaves, so Lemma 1 implies that

$$height_{K_1}(\mathcal{T}^{(0)}) \le \log_{K_1} n \le \log_{n^{1/\beta}} n = \beta \le D.$$

Consider now some $\ell \ge 2$. The definition of tree $\mathcal{T}^{(\ell-1)}$ implies that it has at most $K_{\ell-1}^3$ leaves. Since $K_{\ell-1} \le 2n^{\beta^{-\ell+1}}$ and $K_\ell \ge n^{\beta^{-\ell}}$, from Lemma 1 we have

$$\begin{aligned} height_{K_\ell}(\mathcal{T}^{(\ell-1)}) &\le \log_{K_\ell}(K_{\ell-1}^3) \\ &\le 3 \log_{n^{\beta^{-\ell}}}(2n^{\beta^{-\ell+1}}) \\ &= 3 \log_{n^{\beta^{-\ell}}} n^{\beta^{-\ell+1}} + 3 \log_{n^{\beta^{-\ell}}} 2 \\ &= 3\beta + \frac{3\beta^\ell}{\log n} \le 3\beta + \frac{3\beta^L}{\log n} \le 3\beta + 1, \end{aligned}$$

where the last inequality holds as long as $3\beta^L \le \log n$, which is true if n is large enough (since, by the first bound in (1), we have $\beta^L \le \log n / \log\log n$). This completes the proof of Claim 2.

As in 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_ℓ-heights, for each $\ell \le L$. After completing the description of the algorithm we will explain how to modify it so that these assumptions will not be needed.

Algorithm FastGather consists of $L+1$ epochs, numbered $1, 2, ..., L+1$. In each epoch $\ell \le L$ only the nodes in $\mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$ participate. At the beginning of epoch ℓ all rumors will be already collected in $\mathcal{T}^{(\ell-1)}$, and the purpose of epoch ℓ is to move all these rumors into $\mathcal{T}^{(\ell)}$. Each of these L epochs will run in time $O(n)$, so their total running time will be $O(nL) = O(n \log\log n)$. In epoch $L+1$, only the nodes in $\mathcal{T}^{(L)}$ participate. At the beginning of this epoch all rumors will be already in $\mathcal{T}^{(L)}$, and when this epoch is complete, all rumors will be collected in r. This epoch will take time $O(n \log\log n)$. Thus the total running time will be also $O(n \log\log n)$. We now provide the complete description of the algorithm.

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

Let $v \in \mathcal{T}^{(\ell-1)} - \mathcal{T}^{(\ell)}$. The computation of v in epoch ℓ consists of $D+1$ identical stages. Each stage $h = 0, 1, ..., D$ starts at time step $\alpha_{\ell,h} = \gamma_\ell + (2C+1)hn$ and lasts $(2C+1)n$ steps.

Stage h itself consists of two parts. The first part starts at time $\alpha_{\ell,h}$ and lasts time $2Cn$. During this part we execute $\lceil n/K_\ell^3 \rceil$ iterations, where each iteration executes the strong K_ℓ-selector $\bar{S}^\ell$. The time needed to complete these iterations is at most

$$\lceil \frac{n}{K_\ell^3} \rceil \cdot CK_\ell^2 \log n \le 2Cn \cdot \frac{\log n}{K_\ell} \le 2Cn \cdot \frac{\log n}{n^{\beta^{-\ell}}} \le 2Cn \cdot \frac{\log n}{n^{\beta^{-L}}} \le 2Cn,$$

where the last inequality follows from the definition of L. Thus all iterations executing the strong selector will complete before time $\alpha_{\ell,h} + 2Cn$. Then v stays idle until time $\alpha_{\ell,h} + 2Cn$, which is when the second part starts. In the second part we run the SB-RoundRobin protocol, which takes n steps. So stage h will complete right before step $\alpha_{\ell,h} + (2C+1)n = \alpha_{\ell,h+1}$. Note that the whole epoch will last $(D+1)(2C+1)n$ steps, as needed.

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 having subtree size less than $n/K_{\ell-1}^3$. Then w will receive all rumors from $\mathcal{T}_u$ before the start time γ_ℓ of epoch ℓ.

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 ℓ. We want to show that ($\mathbf{I}^{\ell+1}$) holds right after epoch ℓ 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 $\mathit{height}_{K_\ell}(u, \mathcal{T}^{(\ell-1)}) \le h - 1$. 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 3, 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_ℓ-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 SB-RoundRobin, will move all rumors from $\mathcal{H}$ to w. The details of the argument here are analogous to that in Section 3.

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

Epoch $L + 1$. Due to the definition of L, we have that $\mathcal{T}^{(L)}$ contains at most $K_L^3 = O(\log^3 n)$ leaves, so its 2-depth is at most $D' = \log(K_L^3) = O(\log\log n)$, 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 SB-RoundRobin.

Let $\alpha'_g = \gamma_L + 2gn$. To prove correctness, it is enough to 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 $\mathit{height}_2(u, \mathcal{T}^{(L)}) \le g - 1$. Then at time α'_g node w has received all rumors from $\mathcal{T}_u$.

The proof of this invariant 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 of $\mathcal{T}$ 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 $O(n \log\log n)$.

It remains to explain that the simplifying assumptions we made at the beginning of this section are not needed. Computing all subtree sizes and all K_ℓ-heights can be done recursively bottom-up, using the linear-time information gathering algorithm from [5] 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_ℓ, and, since we limit book-keeping information in each message up to $O(\log n)$ bits, these values need to be computed separately. Nevertheless, the total pre-computation 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_ℓ-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.

Summarizing, we have proved above that the running time of Algorithm FastGather is $O(n \log\log n)$, thus completing the proof of Theorem 1.

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

In this section we consider a slightly different communication model from that in Sections 3 and 4. We now assume that acknowledgements of successful transmissions are provided to the sender (immediately after the transmission). All the remaining nodes, including the intended recipient, cannot distinguish between collisions and absence of transmissions. The main result of this section, as summarized in the theorem below, is that with this feature it is possible to achieve the optimal running time $O(n)$.

Theorem 2. *The problem of information gathering in ad-hoc networks with tree topology, without rumor aggregation, can be solved in time $O(n)$ if acknowledgments of successful transmissions are provided.*

The overall structure of our $O(n)$-time protocol, called Algorithm LinGather, is similar to Algorithm SimpleGather. It consists of two epochs. The first epoch does not use the acknowledgement feature, and it is in fact identical to epoch 1 in Algorithm SimpleGather, except for a different choice of the parameters. After this epoch, lasting time $O(n)$, all rumors will be collected in the subtree $\mathcal{T}'$ consisting of heavy nodes (for a suitable definition of "heavy").

In the second epoch only the heavy nodes in $\mathcal{T}'$ will participate in the computation, and the objective of this epoch is to move all rumors, already collected in $\mathcal{T}'$, to the root r. The key obstacle to overcome in this epoch is congestion stemming from the fact that, although $\mathcal{T}'$ may be small, its nodes have many rumors to transmit. This congestion means that simply repeatedly applying strong k-selectors is no longer enough. For example, suppose that $\mathcal{T}'$ is a complete binary tree of depth $\log\log n$, with each leaf holding $\frac{n}{\log n}$ rumors. Repeating a strong 2-selector $\frac{n}{\log n}$ times guarantees that each rumor from each leaf will be successfully transmitted to this leaf's parent. But the lower bound in [8] on the size of selectors implies that that this would require time $\Omega(n)$ – no better than SB-RoundRobin, and it would only move all the rumors up a single level of the tree. There are $\log\log n$ levels and the congestion only gets worse as we move up the tree, so this approach will not lead to an $O(n)$-time algorithm.

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, say with ℓ active children, receives at most one rumor per round, it clearly cannot simultaneously be receiving rumors at an average rate greater than $\frac{1}{\ell}$ from each active child individually. With the amortizing selector family, we will be able to achieve a constant fraction of this bound over long time intervals, so long as each child knows (approximately) how many siblings it is competing with.

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.* Using a k-distinguisher a node can determine whether its number of active siblings p is at least k or at most $2k$. While this information is not sufficient to determine the exact relation between p and k, we show how to combine different k-distinguishers to obtain another structure, called a *cardinality estimator*, that will determine the value of p within a factor of 4. Using this estimate, and the earlier described amortizing selector family, a node can quickly transmit its rumors to its parent. This will allow us to gather all rumors from $\mathcal{T}'$ in the root in time $O(n)$.

This section is divided into three parts. In Sections 5.1 and 5.2 we give precise definitions and constructions of our combinatorial structures. Our $O(n)$-time protocol LinGather is described and analyzed in Section 5.3.

5.1 Construction of Amortizing Selectors

We now define the concept of an amortizing selector family $\bar{F}$. 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, $\bar{F} = \{F_i^j\}$. The general idea is that a node with roughly 2^j siblings will use sets $F_0^j, F_1^j, ...$ to determine its transmissions.

Specifically, let $\bar{F} = \{F_i^j\}$, where $i \in \{0, 1, ..., m-1\}$ and $j \in \{0, 1, ..., \kappa\}$, for some integer parameters $m \geq 1$ and $\kappa \in \{1, 2, ..., \lceil \log n \rceil\}$. We say that $\bar{F}$ is *an amortizing selector family with cumulative success rate* ρ if the following statement holds:

(asf) For each $j = 0, 1, ..., \kappa-1$, each subset $A \subseteq [n]$ satisfying $2^{j-1} \leq |A| \leq 2^{j+1}$, and each $a \in A$, there are at least $\rho m/|A|$ distinct indices i for which

$$a \in F_i^j \text{ and } A \cap (F_i^{j-1} \cup F_i^j \cup F_i^{j+1}) = \{a\}.$$

(For $j = 0$ the set F_i^{-1} is defined to be empty.) For a fixed j, we will refer to the collection of sets $\bar{F}^j = F_0^j, F_1^j, ..., F_{m-1}^j$ as a *j-amortizing selector.* Parameter m in the definition of $\bar{F}$ is called the *length* of $\bar{F}$.

In the application to rumour gathering, m can be thought of as the "running time" of $\bar{F}$, 2^j as a node's estimate of its parent's degree, and 2^κ as some bound on the parent's maximum degree (that is, the number of siblings, including v) handled by $\bar{F}$. A node v whose number of siblings is estimated to be about 2^j transmits at time step i if and only if $\mathsf{label}(v) \in F_i^j$. Denoting by A the set of labels of v's siblings, what condition (asf) is then saying is that if $|A| \leq 2^{\kappa-1}$, and if each sibling of v estimates $|A|$ within a factor of 2, then v (as well as each of its siblings) will succeed at rate at least $\rho/|A|$. Thus the parent of v will receive transmissions from its children at overall rate ρ – averaging about one successful transmission per $1/\rho$ steps.

Theorem 3. *There are fixed constants $\rho, C > 0$ such that the following is true: For any n and $\kappa \leq \lceil \log n \rceil$, and any $m \geq Ck^2 \log n$, where $k = 2^\kappa$, there is an amortizing selector family with parameters n, κ, m and cumulative success rate ρ.*

Proof. Let κ, n and m be given such that $m \geq Ck^2 \log n$, where $k = 2^\kappa$, and let C be a constant to be determined later. We form our selector family probabilistically: For each a, i, and j, we independently include a in F_i^j with probability $1/2^{j+1}$.

Observe that it suffices to check the selector property (asf) for the case $|A| = 2^{j+1}$. Indeed, this follows from monotonicity: for any set A with $2^{j-1} \leq |A| \leq 2^{j+1}$, if we replace A by a set A' such that $A \subseteq A'$ and $|A'| = 2^{j+1}$, then for any $a \in A$ and any i satisfying

$$A' \cap (F_i^{j-1} \cup F_i^j \cup F_i^{j+1}) = \{a\},$$

we must also have

$$A \cap (F_i^{j-1} \cup F_i^j \cup F_i^{j+1}) = \{a\}.$$

So if there are at least $\rho' m/|A'|$ distinct indices i satisfying the first equality, then there are at least $\rho m/|A|$ indices satisfying the second equality, for $\rho = \frac{1}{4}\rho'$. In other words, restricting the argument to sets A with $|A| = 2^{j+1}$ can only overestimate the cumulative success rate by a factor of 4.

Now fix some $j \in \{1, 2, \ldots, \kappa - 1\}$, a set $A \subseteq [n]$ with $|A| = 2^{j+1}$ and some $a \in A$, and let the random variable X be the number of indices i for which (asf) holds, that is

$$X = \left|\left\{ i : a \in F_i^j \text{ and } A \cap (F_i^{j-1} \cup F_i^j \cup F_i^{j+1}) = \{a\} \right\}\right|$$

Denote $\ell = 2^{j+1}$. Each label a is included in F^{j-1}, F^j and F^{j+1} with respective probabilities $\frac{2}{\ell}$, $\frac{1}{\ell}$ and $\frac{1}{2\ell}$. Thus the expected value of X is

$$\begin{aligned}\mu_X &= m\Big(\frac{1}{\ell}\Big)\Big(1 - \frac{2}{\ell}\Big)^{\ell-1}\Big(1 - \frac{1}{\ell}\Big)^{\ell-1}\Big(1 - \frac{1}{2\ell}\Big)^{\ell-1} \\ &\geq m\Big(\frac{1}{\ell}\Big)\Big(1 - \frac{2}{\ell}\Big)^{\ell}\Big(1 - \frac{1}{\ell}\Big)^{\ell}\Big(1 - \frac{1}{2\ell}\Big)^{\ell} \geq \frac{2\rho' m}{\ell},\end{aligned} \tag{2}$$

for $\rho' = \frac{1}{2}(\frac{21}{64})^4$, where the last inequality holds because the three expressions $(1-2/\ell)^\ell$, $(1-1/\ell)^\ell$, and $(1-1/2\ell)^\ell$ are increasing functions of ℓ, so they are minimized for $\ell = 4$ (as we assume that $j \geq 1$).

Using the bound (2) on μ_X and applying the Chernoff bound, we get

$$\begin{aligned}\Pr[X \leq \rho' m/\ell] \;&\leq\; \Pr[X \leq \tfrac{1}{2}\mu_X] \\ &\leq\; \mathrm{e}^{-\mu_X/8} \;\leq\; \mathrm{e}^{-\rho' m/4\ell}.\end{aligned}$$

The above derivation assumes $j \geq 1$, but it can be easily extended to $j = 0$; the only difference is that the sets F_i^{-1}, which are empty, do not need to be considered.

We now use the union bound over all choices of ℓ, a, and A. We have at most n choices of a, at most $\log n + 1 \leq 2\log n$ choices for ℓ, and at most $\binom{n}{\ell-1} \leq n^{k-1}$ choices of A, given fixed values of ℓ and a. Thus the probability that our family does not satisfy (asf) with rate ρ', for sets A with cardinality $\ell = 2^{j+1}$, is at most

$$\begin{aligned}n \cdot (2\log n) \cdot n^{k-1} \cdot (\mathrm{e}^{-\rho' m/4\ell}) \;&\leq\; 2n^{2k} \cdot \mathrm{e}^{-C\rho' k \log(n)/4} \\ &=\; 2n^{k(2 - C\rho' \log(\mathrm{e})/4)},\end{aligned}$$

which is smaller than 1 for sufficiently large C. Thus there exists a family $\{F_i^j\}$ that satisfies (asf) with rate ρ' for sets A with cardinality 2^{j+1}. As explained earlier, this implies the existence of an amortizing selector family with cumulative success rate $\rho = \frac{1}{4}\rho'$. □

5.2 Construction of k-Distinguishers and Cardinality Estimators

In this section, we define k-distinguishers and show how to construct them. We then show how we can use k-distinguishers to construct other combinatorial structures called cardinality estimators, which a node can use to estimate the number of its siblings.

Consider a family $\bar{E} = (E_0, E_1, ..., E_{m-1})$ of sets, where $E_j \subseteq [n]$ for each j. For $A \subseteq [n]$ and $a \in A$, define

$$\mathit{hits}_{a,A}(\bar{E}) = |\{j : E_j \cap A = \{a\}\}|,$$

that is $hits_{a,A}(\bar{E})$ is the number of indices j for which E_j intersects A exactly on a. Note that, using this terminology, $\bar{E}$ is a strong k-selector if and only if $hits_{a,A}(\bar{E}) > 0$ for all sets $A \subseteq [n]$ of cardinality at most k and all $a \in A$.

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

(dis1) if $|A| < k$ then $hits_{a,A}(\bar{E}) > \xi$, and

(dis2) if $|A| \geq 2k$ then $hits_{a,A}(\bar{E}) \leq \xi$.

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

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 Y will tell us whether $Y \cap A = \{a\}$ or not. We can then apply this oracle to each set $Y = E_j$ in a k-distinguisher $\bar{E}$, which will give us the cardinality of $hits_{a,A}(\bar{E})$, thus allowing us to extract some information about the cardinality of A: If $|hits_{a,A}(\bar{E})| \leq \xi$ then we know that $|A| \geq k$, and if $|hits_{a,A}(\bar{E})| > \xi$ then we know that $|A| < 2k$.

In our radio network model, if all nodes transmit according to a k-distinguisher $\bar{E}$ (that is, at time j a node v transmits iff $\mathsf{label}(v) \in E_j$), then the acknowledgement of a message received from a parent corresponds exactly to a "yes" response from the hit oracle. This way each node can determine in time $O(k^2 \log n)$ either that its parent has at least k children or that it has fewer than $2k$ children.

What we will show shortly, again by a probabilistic construction, is that not-too-large k-distinguishers exist:

Theorem 4. *For any $n \geq 2$ and $k \in \{1, 2, ..., \lfloor n/2 \rfloor\}$ there exists a k-distinguisher of size $m = O(k^2 \log n)$.*

Proof. 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 $\bar{E} = (E_0, E_1, ..., E_{m-1})$ by independently including each $a \in [n]$ in E_j with probability $1/2k$ ("independently" here means that the mn events of the form "a is contained in E_j" are all mutually independent). Thus, for any set A and $a \in A$, the probability that $E_j \cap A = \{a\}$ is $(1/2k)(1 - 1/2k)^{|A|-1}$, and the expected value of $hits_{a,A}(\bar{E})$ is

$$\mathbf{E}[\, hits_{a,A}(\bar{E}) \,] = m \cdot \frac{1}{2k}\Big(1 - \frac{1}{2k}\Big)^{|A|-1}. \tag{3}$$

We claim that, for a suitable value of ξ, the probability that there exists a set $A \subseteq [n]$ and some $a \in A$ for which $\bar{E}$ does not satisfy both conditions (dis1) and (dis2) is smaller than 1 (and in fact tends to 0). This will be sufficient to show existence of a k-distinguisher with threshold value ξ.

Observe that in order to be a k-distinguisher it is sufficient that $\bar{E}$ satisfies (dis1) for sets A with $|A| = k$ and satisfies (dis2) for sets A with $|A| = 2k$. This is true because the value of $hits_{a,A}(\bar{E})$ is monotone with respect to the inclusion: if $a \in A \subseteq B$ then each set hitting B exactly on a also hits A exactly on a, so $hits_{a,A}(\bar{E}) \geq hits_{a,B}(\bar{E})$.

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 $X_i = |hits_{a,A_i}(\bar{E})|$ and their expected values $\mu_i = \mathrm{Exp}[X_i]$. For any integer $k \geq 1$ we have

$$\begin{aligned} \frac{1}{\mathrm{e}^{1/2}} \;&\leq\; \Big(1 - \frac{1}{2k}\Big)^{k-1}, \text{ and} \\ \frac{1}{\mathrm{e}} \;&\leq\; \Big(1 - \frac{1}{2k}\Big)^{2k-1} \leq \frac{1}{2} \end{aligned}$$

From (3), substituting $m = Ck^2 \log n$, this gives us the corresponding estimates for μ_1 and μ_2:

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

Since $\mathrm{e}^{-1/2} > \frac{1}{2}$, we have $\mu_2 < \mu_1$, so we can choose an $\epsilon \in (0,1)$ and ξ for which

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

Then the probability that $\bar{E}$ violates (dis1) for $A = A_1$ is

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

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 $\bar{E}$ violating (dis2) for $A = A_2$ as follows:

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

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 $\bar{E}$ is not a k-distinguisher is at most

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

where the last inequality holds for sufficiently large C. As explained earlier, this implies existence of a k-distinguisher with threshold value ξ. □

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

$$\widehat{E} \;=\; \bar{E}^0 \bar{E}^1 \ldots \bar{E}^{\lceil \lambda \log n \rceil}$$

of length $m_{\widehat{E}} = \sum_{i=0}^{\lceil \lambda \log n \rceil} O(2^{2i} \log n) = O(n^{2\lambda} \log n)$. We will refer to $\widehat{E}$ as a *cardinality estimator*, because applying our hit oracle to $\bar{E}$ 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 $h_i = \mathit{hits}_{a,A}(\bar{E}^i)$, for all i. If i_0 is the smallest i for which $h_i > \xi_i$, then by definition of our k-distinguishers (for $k = 2^{i_0 - 1}$ and

$k = 2^{i_0}$) we must have $2^{i_0-1} \leq |A| < 2^{i_0+1}$. The value of $i_0 - 1$ will denoted $est_{A,a}(\widehat{E})$. It represents the cardinality estimate of A computed by $\widehat{E}$ for label a.

In our gathering framework, suppose that some subset of children of a node u is transmitting according to $\widehat{E}$, and let A be the set of their labels. If v is one of these children and $a = \mathsf{label}(v)$, then, thanks to the transmission acknowledgement mechanism, this computation will allow v to compute $j = est_{A,a}(\widehat{E})$ for which the cardinality of A is between 2^j and 2^{j+2}, which is exactly what we need to be able to run the amortizing selector.

5.3 Linear-time Protocol

We are now ready to present our linear time algorithm for rumor gathering on trees that uses transmission acknowledgements, thus proving Theorem 2.

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 $0 < \delta < \frac{1}{13}$ be a (fixed) small positive constant, and let $K = \lceil n^\delta \rceil$. We say that $v \in \mathcal{T}$ is *light* if $|T_v| \leq n/K^3$ and we call v *heavy* otherwise. By $\mathcal{T}'$ we denote the subtree of $\mathcal{T}$ induced by the heavy nodes.

We will use three combinatorial structures:

- $\bar{S}$ is a strong K-selector of length $m_{\bar{S}} = O(K^2 \log n)$. As explained in Section 3, such strong K-selectors exist.
- $\bar{F}$ is an amortizing selector family with cumulative success rate ρ, parameter $\kappa = \lceil \log(2K^3) \rceil$ and length $m_{\bar{F}} = CK^7$, where C is a positive constant. If $k = 2^\kappa$ then $k^2 \log n = o(K^7)$, so, from Theorem 3, such $\bar{F}$ exists for any C so long as n is sufficiently large.
- $\widehat{E}$ is a cardinality estimator with parameter $\lambda = 3\delta$ and length $m_{\widehat{E}} \leq m_{\bar{F}} = CK^7$. By the construction in Section 5.2, since $n^{2\lambda} \log n = n^{6\delta} \log n = o(K^7)$, such $\widehat{E}$ exists, as long as C is large enough.

Without loss of generality (see Section 3), in the algorithm we can assume that the nodes can receive and transmit messages at the same time. We also assume that each node v knows the size of its subtree $\mathcal{T}_v$, its K-height, etc.

For a node v, a rumor from $\mathcal{T}_v$ is called *pending at* v if it has been received by v but not by v's parent. Thanks to the transmission acknowledgement feature, v knows which rumors it has already received are pending.

Algorithm LinGather. Our algorithm consists of two epochs. The first epoch is essentially identical to epoch 1 in Algorithm SimpleGather. The objective of this epoch is to move all rumors from $\mathcal{T}$ to $\mathcal{T}'$ in time $O(n)$. In the second epoch only the heavy nodes in $\mathcal{T}'$ will participate in the computation, and its objective is to move all rumors from $\mathcal{T}'$ to r in time $O(n)$. This epoch, which is quite different from our earlier algorithms, will use the amortizing selector families and degree estimators to achieve its goal.

Epoch 1: In this epoch only light nodes participate, and its objective 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.

Let $D = \lceil \log_K n \rceil \leq 1/\delta = O(1)$. 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 equal h participate. Eacah stage h consists of n/K^3 executions of the strong selector $\bar{S}$, followed by an execution of a single round of SB-RoundRobin, taking total time $n/K^3 \cdot m_{\bar{S}} + n = O(n)$. So the entire epoch takes time $(D+1) \cdot O(n) = O(n)$ as well.

Epoch 2: When this epoch starts, all rumors from $\mathcal{T}$ are already gathered in $\mathcal{T}'$, and the objective is to gather them in the root. As before, no node in $\mathcal{T}'$ can have more than $K^3 = O(n^{3\delta})$ heavy children, since each heavy child has at least n/K^3 descendants.

The second epoch is divided into stages, each consisting of $2(m_{\widehat{E}} + m_{\bar{F}})$ steps. A node v will be called *active* in a given stage if at the beginning of the stage it has already received all rumors from its subtree $\mathcal{T}_v$, but still has at least one pending rumor. (It is possible for a node to never become active, if it receives its last rumor within some stage and then finishes transmitting before the beginning of the next stage.)

In any stage, a node $v \in \mathcal{T}' - \{r\}$ behaves as follows:

- In each odd-numbered time step of the stage, if v still has a pending rumor then v transmits such a rumor. (This applies no matter whether v is active or not.)
- In the even-numbered time steps, if v is active in this stage, its computation has two parts:
  - In the first part, v transmits according to the cardinality estimator $\widehat{E}$. (Here, it does not matter what message v transmits.) This will take time $2m_{\widehat{E}}$. Let $j = est_{A,a}(\widehat{E})$, as defined in Section 5.2, for $a = \mathsf{label}(v)$ and A being the set of labels of active siblings of v (inclusive). Thus at this point v knows that its number of active siblings is between 2^j and 2^{j+2}. (Note that different active siblings may compute different estimates.)
  - In the second part, v transmits its pending rumors (if it still has any) using the corresponding $(j+1)$-amortizing selector $\bar{F}^{j+1}$ from the amortizing selector family $\bar{F}$. If v happens to run out of pending rumors, it stops transmitting. This part lasts for $2m_{\bar{F}}$ steps.

The intuition here is that, most of the time, the cardinality estimations from the first part of a stage will remain accurate throughout the stage, allowing the amortizing selectors to make significant progress at each active node. Although no progress is made during the cardinality estimation itself, the estimation takes only a fraction of the time in the epoch (since by construction $m_{\widehat{E}}$ is no larger than $m_{\bar{F}}$), so this does not pose a problem. We now turn to the details of the analysis.

Analysis. Recall that the objective of epoch 1 is to move all rumors from $\mathcal{T}$ to $\mathcal{T}'$. The proof of correctness for epoch 1 is identical to that for Algorithm SimpleGather in Section 3, so we omit it here. The correctness of epoch 1 implies that, for any $u \in \mathcal{T}'$ and any light child x of u, when epoch 2 starts then all rumors from $\mathcal{T}_x$ will be collected in u.

It remains to prove correctness of epoch 2. Our key claim, stated below, is that each node u with subtree size $s = |\mathcal{T}_u|$ will have received all s rumors from $\mathcal{T}_u$ within $O(s)$ steps of the start of the epoch. In order to make the inductive argument work, however, we need a more subtle property, wchich also guarantees that the rumors aggregate at least at a steady rate over time.

(**L**) For any heavy node u with $|\mathcal{T}_u| = s$, and any $j = 1, 2, ..., s$, u will receive at least j rumors in the first $B(2s + j)$ steps of epoch 2, where B is some sufficiently large absolute constant. In particular, u will receive all of the rumors from its subtree no later than in step $3Bs$ of epoch 2.

We will show that invariant (**L**) holds by induction on u's height within $\mathcal{T}'$. If u a leaf of $\mathcal{T}'$ then invariant (**L**) follows from the correctness of epoch 1, because all children of u are light.

Now consider a non-leaf node $u \in \mathcal{T}'$ with $|\mathcal{T}_u| = s$. Let $v_1, v_2, ..., v_\ell$ be the heavy children of u. For each $i = 1, 2, ..., \ell$, let $s_i = |\mathcal{T}_{v_i}|$, and order these children so that $s_1 \geq s_2 \geq \cdots \geq s_\ell \geq n/K^3$. (The last inequality holds because all v_i's are heavy.) When epoch 2 starts, all rumors from the light children of u are already in u, so we only need to estimate how long it takes for the rumors in the subtrees of $v_1, v_2, ..., v_\ell$ to reach u.

In the proof we will treat v_1, a child of u with the largest subtree, separately from all other heavy children $v_2, v_3, ..., v_\ell$. The basic idea is this: Since $s_2, ..., s_\ell$ are each at most $s/2$, the inductive assumption implies that each v_i, for $i \geq 2$, will receive all rumors from its subtree $\mathcal{T}_{v_i}$ no later than at time $3Bs_i \leq \frac{3}{2}Bs$. This leaves us time at least $\frac{1}{2}Bs + Bj$ to push j rumors to u, which can be accomplished using amortized selectors (if multiple nodes transmit simultaneously) or odd-numbered transmissions (if only v_1 transmits).

We introduce the following notation (see Figure 3):

t_1 = the first time step when the nodes $v_2, v_3, ..., v_\ell$ have already received all rumors from their subtrees. By the inductive hypothesis for invariant (**L**) (for nodes of smaller height in $\mathcal{T}'$), we have that $t_1 \leq 3Bs_2 \leq \frac{3}{2}Bs$.

t_2 = the first time step when u has received all rumors from the subtrees of $v_2, v_3, ..., v_\ell$.

q = the number of stages between times t_1 and t_2. Here, we only count *full* stages, namely those that are fully included in the time interval (t_1, t_2).

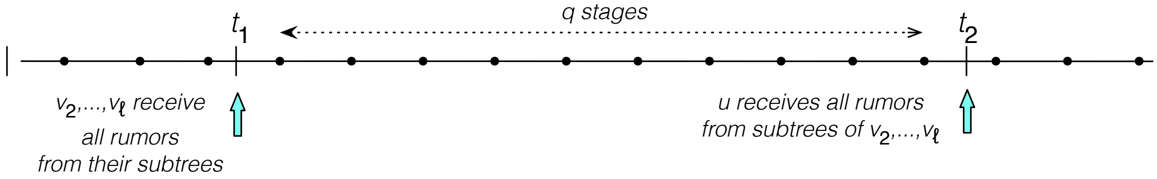
t_1
q stages
t_2
$v_2,...,v_\ell$ receive all rumors from their subtrees
u receives all rumors from subtrees of $v_2,...,v_\ell$

Figure 3: Notation for the analysis of Algorithm LinGather.

In each of the q full stages between times t_1 and t_2 at least one of $v_2, v_3, ..., v_\ell$ must be active. Each of these stages must be of one of the two following types:

Complete: All active v_i's in stages of this type have pending rumors throughout the entire stage. Let $a \geq 1$ be the number of active nodes among $v_2, v_3, ..., v_\ell$ during such stage. Then, by the definition of our amortizing selector family $\bar{F}$, u received at least $\rho a m_{\bar{F}}/(a+1) \geq \frac{1}{2}\rho m_{\bar{F}}$ new rumors from $v_2, v_3, ..., v_\ell$. (We need $a+1$ in the denominator because v_1 may also be active.) Therefore the number of complete stages is at most $2(\sum_{i=2}^{\ell} s_i)/(\rho m_{\bar{F}}) \leq 2s/(\rho m_{\bar{F}})$.

Incomplete: In these stages some active v_i transmits its last rumor from $\mathcal{T}_{v_i}$, and stops transmitting. There are at most K^3 incomplete stages, because $\ell \leq K^3$ and each v_i becomes active at most once (when it has received all rumors from its subtree).

We can thus bound t_2 as follows:

$$\begin{aligned}
t_2 &\le t_1 + 2(m_{\widehat{E}} + m_{\bar{F}})(q+2) \\
&\le \tfrac{3}{2}Bs + 2(m_{\widehat{E}} + m_{\bar{F}}) \cdot [2s/(\rho m_{\bar{F}}) + K^3 + 2] \\
&\le \tfrac{3}{2}Bs + 2(CK^7 + CK^7) \cdot [2s/(CK^7\rho) + 2K^3] & (4) \\
&= \tfrac{3}{2}Bs + 8s/\rho + 8CK^{10} \\
&\le (\tfrac{3}{2}B + 8/\rho + 8C)s & (5) \\
&\le 2Bs. & (6)
\end{aligned}$$

In the above derivation, inequality (4) follows from $m_{\widehat{E}} = CK^7$ and $m_{\bar{F}} \le CK^7$. Inequality (5) is true because $\delta < \frac{1}{13}$ and u is a heavy node, so $K^{10} \le n/K^3 \le s$. Inequality (6) holds because we can take B sufficiently large.

By the above bound, node v_1 is the only node that could still be transmitting after time $2Bs$. In particular, if it has a pending rumor during an odd numbered time step after this point, it successfully transmits this rumor to u. By the inductive assumption, for each $j = 1, 2, ..., s_1$, v_1 will have received at least j rumors by time $B(2s_1 + j) \le B(2s + j - 2)$. So, for $j = 1, 2, ..., s_1$, node u will receive the jth rumor no later than at time $B(2s + j - 2) + 2 \le B(2s + j)$. If $s_1 < j \le s$, then by time $B(2s + j)$ node u will have in fact received all rumors from $\mathcal{T}_u$. This completes the proof of invariant (**L**).

Now, taking $u = r$ in invariant (**L**), we obtain that r will receive all rumors from $\mathcal{T}$ in at most $3Bn$ steps of epoch 2. Since the first epoch also takes linear time, the total running time of Algorithm LinGather is $O(n)$, completing the proof of Theorem 2.

References

[1] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, 1991.

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

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

[4] Malin Christersson, Leszek Gasieniec, and Andrzej Lingas. Gossiping with bounded size messages in ad hoc radio networks. In *Proc. of the 29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 377–389, 2002.

[5] Marek Chrobak, Kevin Costello, Leszek Gasieniec, and Darek R. Kowalski. Information gathering in ad-hoc radio networks with tree topology. In *Proc. of the 8th Int. Conference on Combinatorial Optimization and Applications (COCOA)*, pages 129–145, 2014.

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

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

[8] Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. of the 12th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 709–718, 2001.

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

[10] Artur Czumaj and Peter Davies. Almost optimal deterministic broadcast in radio networks. In *Proc. of the 43rd Int. Colloq. on Automata, Languages and Programming (ICALP)*, 2016. To appear.

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

[12] Paul Erdös, Péter Frankl, and Zoltan 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.

[13] Antonio Fernndez Anta, Miguel A. Mosteiro, and Jorge Ramn Muoz. Unbounded contention resolution in multiple-access channels. *Algorithmica*, 67(3):295–314, 2013.

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

[15] Leszek Gasieniec, Tomasz Radzik, and Qin Xin. Faster deterministic gossiping in directed ad hoc radio networks. In *Proc. of the 9th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 397–407, 2004.

[16] Dariusz R. Kowalski. On selection problem in radio networks. In *Proc. of the 24th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 158–166, 2005.

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

[18] Eyal Kushilevitz and Yishay Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing*, 27(3):702–712, 1998.

[19] Ding Liu and Manoj Prabhakaran. On randomized broadcasting and gossiping in radio networks. In *Proc. of the 8th Int. Computing and Combinatorics Conference (COCOON)*, pages 340–349, 2002.

[20] Gianluca De Marco. Distributed broadcast in unknown radio networks. In *Proc. of the 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 208–217, 2008.

[21] Gianluca De Marco and Dariusz R. Kowalski. Contention resolution in a non-synchronized multiple access channel. In *Proc. of the 27th Int. Symposium on Parallel Distributed Processing (IPDPS)*, pages 525–533, 2013.

[22] Gianluca De Marco and Dariusz R. Kowalski. Fast nonadaptive deterministic algorithm for conflict resolution in a dynamic multiple-access channel. *SIAM Journal on Computing*, 44(3):868–888, 2015.

[23] Arthur N. Strahler. Hypsometric (area-altitude) analysis of erosional topology. *Geological Society of America Bulletin*, 63:1117–1142, 1952.

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

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