# On the Randomness Requirements of Rumor Spreading

George Giakkoupis*
Université Paris Diderot
ggiak@liafa.jussieu.fr

Philipp Woelfel†
University of Calgary
woelfel@ucalgary.ca

**Abstract**

We investigate the randomness requirements of the classical rumor spreading problem on fully connected graphs with $n$ vertices. In the standard random protocol, where each node that knows the rumor sends it to a randomly chosen neighbor in every round, each node needs $O((\log n)^2)$ random bits in order to spread the rumor in $O(\log n)$ rounds with high probability (w.h.p.). For the simple quasirandom rumor spreading protocol proposed by Doerr, Friedrich, and Sauerwald (2008), $\lceil \log n \rceil$ random bits per node are sufficient. A lower bound by Doerr and Fouz (2009) shows that this is asymptotically tight for a slightly more general class of protocols, the so-called *gate-model*.

In this paper, we consider general rumor spreading protocols. We provide a simple push-protocol that requires only a total of $O(n \log\log n)$ random bits (i.e., on average $O(\log\log n)$ bits per node) in order to spread the rumor in $O(\log n)$ rounds w.h.p. We also investigate the theoretical minimal randomness requirements of efficient rumor spreading. We prove the existence of a (non-uniform) push-protocol for which a *total* of $2\log n + \log\log n + o(\log\log n)$ random bits suffice to spread the rumor in $\log n + \ln n + O(1)$ rounds with probability $1-o(1)$. This is contrasted by a simple time-randomness tradeoff for the class of all rumor spreading protocols, according to which any protocol that uses $\log n - \log\log n - \omega(1)$ random bits requires $\omega(\log n)$ rounds to spread the rumor.

## 1 Introduction

The problem of disseminating information in large networks is a fundamental one with a variety of applications, e.g., in the maintenance of distributed replicated database systems [3, 15]. As a consequence, the problem of broadcasting information has been studied to a large extent, theoretically and experimentally. In order to be useful for a broad range of applications, efficient broadcasting algorithms should be simple, local (nodes need no information about the network topology), and be able to tolerate small changes in the network topology (e.g., due to failures).

The classical algorithm for this problem is the *push-model*, also known as the *fully random rumor spreading algorithm*. The protocol proceeds in rounds; each node of the $n$-node graph can send one message per round. Initially, in round 0, an arbitrary node receives a piece of information, called the *rumor*. That rumor is then spread iteratively to other nodes: In each round every *informed* node (i.e., every node that received the rumor in a previous round) chooses a random neighbor to which it then transmits the rumor.

This fundamental protocol obviously satisfies the desired simplicity and locality properties. Here we focus on results for the complete graph with $n$ vertices, $K_n$. Frieze and Grimmet [18] provided an asymptotically tight analysis of the number of rounds that are needed until every node becomes informed with high probability (w.h.p.).[1] This was improved by Pittel [21], who showed that $\log n + \ln n + O(1)$ rounds suffice with probability $1 - o(1)$.[2]

Clearly, in this protocol each node needs to generate $\lceil \log n \rceil$ random bits in order to select a random neighbor. The analysis of the random process shows that most nodes need to send messages for $\Theta(\log n)$ rounds until all nodes are informed. Therefore, each node has to generate an expected number of $\Theta\big((\log n)^2\big)$ random bits in order to inform all other nodes. Recently, Doerr, Friedrich, and Sauerwald proposed the *quasirandom rumor spreading* algorithm as an alternative that aims at "imitating properties of the classical push model with a much smaller degree of randomness" [7]. In their model, each node needs only to generate one random $\lceil \log n \rceil$-bit string that identifies a start point in some given list of the node's neighbors (e.g., the adjacency list). Starting with that point, the node then informs

*Supported in part by the ANR project PROSE, and the INRIA project GANG.

†Supported by NSERC

[1] We say an event $\mathcal{E}(n)$ occurs with high probability, if there exists a constant $\varepsilon > 0$ such that $\Pr\big(\mathcal{E}(n)\big) = 1 - O(n^{-\varepsilon})$.

[2] "log" denotes the logarithm to base 2 and "ln" the natural logarithm.

its neighbors in the order determined by that list (in a round-robin fashion). Despite the reduced randomness requirements, the protocol is still efficient: Angelopoulos, Doerr, Huber, and Panagiotou [1] as well as Fountoulakis and Huber [16] provide upper and lower bounds for the broadcast time that essentially match the ones of the fully random case.

Doerr and Fouz [4, 5] have considered further reducing the amount of randomness by limiting each node's choice of its start point in its list to a subset of $n/\ell$ nodes that are equidistantly distributed in the node's list. However, they proved a negative result: There exist lists such that for any $\epsilon > 0$ it takes w.h.p. at least $(1-\epsilon)(\log n + \ln n - \log \ell - \ln \ell) + \ell - 1$ rounds until every node is informed. Note that in this so-called *gate-model with randomness parameter* $\ell$ each node needs to generate $\log n - \log \ell + \Theta(1)$ random bits. Hence, in this model it is not possible to spread the rumor to all nodes within $O(\log n)$ rounds, unless each node generates at least $\log n - \log \log n - O(1)$ random bits.

**1.1 Our Contributions.** Since randomness is a sparse resource, we study the problem of reducing the amount of randomness needed for efficient rumor spreading. We restrict ourselves to push-algorithms for the complete graph, where in each round each node can select a neighbor from its adjacency list to which a message is then transmitted. We make the standard assumption that algorithms have no edge connection information available, other than an adjacency list of neighbors in arbitrary order. Moreover, protocols are *anonymous*[3], meaning that a node's decisions do not depend on the node's ID.

The fully random and the quasirandom algorithms are both *oblivious*, in the sense that nodes choose their neighbors without any information gained from incoming messages. While this also precludes information about the number of messages a node received, nodes are aware of the number of rounds that have passed since they received the rumor. In particular, such oblivious algorithms require no information other than the rumor to be transmitted. We prove that any oblivious algorithm, where each node uses at most $b < \log n - 1$ random bits, cannot spread the rumor to all nodes in less than $b + \lfloor n/2^{b+1} \rfloor$ rounds (for carefully chosen adjacency lists). Hence, at least $\log n - \log \log n - O(1)$ random bits are necessary for any oblivious algorithm to spread the rumor within $O(\log n)$ rounds. This generalizes the result in [4] for the gate-model to the class of all oblivious algorithms.

The proof of the above observation reveals that oblivious algorithms cannot work efficiently with low randomness due to a lack of entropy available to nodes. Therefore, a natural idea to improve the randomness requirements is to *share* randomness among nodes. We present a simple modification of the quasirandom protocol, where the rumor is spread to all other nodes within $O(\log n)$ rounds w.h.p., and where the average number of random bits per node is reduced to $O(\log \log n)$. The idea is to proceed in phases. The first phase consists of roughly $\log n - \log \log n$ rounds, in which nodes act exactly as in the quasirandom protocol. After that, nodes switch to a different strategy in order to share randomness with other nodes. Each of the nodes informed so far generates a random *prefix* of $\lceil \log n \rceil - \Theta(\log \log n)$ random bits and continues informing nodes in a quasirandom fashion, but appending the random prefix to its messages. A non-informed node that receives such a random prefix fills it up with a newly chosen random suffix in order to compose a $\lceil \log n \rceil$-bit string. That string then marks the start point in its list, and the node starts to spread the rumor to its neighbors in the quasirandom way. The main difference to the quasirandom algorithm is that in later rounds, nodes do not have to generate $\lceil \log n \rceil$ bits to select the start points in their lists. Rather, most of the entropy for selecting these start points is generated by a small subset of $\Theta(n/\log n)$ nodes that were informed in earlier rounds. This main result demonstrates that a simple protocol can be used to significantly decrease the total amount of randomness without sacrificing efficiency.

This result raises the question about the minimum entropy required for efficient rumor spreading. To answer this question, we show that there is a protocol that distributes the rumor in $\log n + \ln n + O(1)$ rounds with probability $1 - o(1)$, and that needs only $2 \log n + \log \log n + o(\log \log n)$ random bits in total (which are generated by the first node that gets informed). Unfortunately, the proof is existential, and we have no explicit construction of such an extremely low-randomness protocol. However, it is not hard to see that our upper bound is asymptotically optimal: We observe that if the total number of random bits is limited to $\log n - \log \log n - \omega(1)$, then not all nodes can be informed within $O(\log n)$ rounds.

**1.2 Related Work.** The fully random algorithm has been analyzed for various other graph topologies, such as general graphs, bounded-degree graphs, hypergraphs, sufficiently dense random graphs [15], expanders [22],

[3] In the rumor spreading literature, anonymous algorithms are usually called "address-oblivious". We chose a different terminology in order to avoid confusion with the notion of *oblivious algorithms*. Note that our notion of *anonymous nodes* is consistent with that of anonymous processes in the distributed computing literature.

star and Cayley graphs [11, 12], and complete $k$-ary trees [7]. The quasirandom algorithm has been proven to be at least as efficient as the fully random algorithm for most of these types of graphs (see [7, 8] for details). In fact, for some topologies, e.g., sparse random graphs, the quasirandom model is superior to the fully random one. An experimental comparison of the fully random and the quasirandom model was provided in [6].

Besides minimizing the broadcast time and the randomness requirements, one can also optimize the total number of transmissions. This can be achieved by combining the push algorithm with the so-called pull algorithm, where nodes contact random neighbors in order to *receive* (as opposed to *send*) the rumor from them [19, 10, 13, 2]. The problem of minimizing the total communication complexity (i.e., the total number of bits transmitted throughout a run of the protocol) was studied in [17].

The robustness of the fully random algorithm was considered in [14]. The authors showed for all graphs, that if each message transmission is lost with a probability of $1 - p$, then the broadcast time increases by at most $O(1/p)$. The same is true for a variant of the quasirandom algorithm, where recipients send feedback to the sender [8]. Additional robustness results for the quasirandom model on the complete graph can be found in [9].

## 2 Low-Randomness Rumor Spreading

We present a rumor spreading protocol that distributes a rumor to all nodes in $O(\log n)$ rounds w.h.p., using a total number of $O(n \log\log n)$ random bits. Also, no single node generates more than $2 \log n$ random bits.

Our protocol is a modification of the quasirandom rumor spreading protocol [7] that we analyze for the fully connected graph. Sacrificing consistency, but for presentational simplicity, throughout this section $n$ will denote the number of neighbors that each node has—as opposed to the total number of nodes. Thus, we consider the complete graph on $n+1$ nodes, $K_{n+1}$, and we assume that, as in the quasirandom protocol, each node $v$ stores its neighbors in a list $L_v[0 \dots n-1]$ in an arbitrary order.

The protocol proceeds in two phases. The first phase consists of $\log n - \log\log n + O(1)$ rounds, and during these rounds nodes act exactly as in the quasirandom protocol. This phase results in a $\Theta(1/\log n)$ fraction of the nodes being informed w.h.p., and generates $\Theta(n)$ random bits in total ($\lceil \log n \rceil$ random bits generated by each node that gets informed).

In the second phase, nodes switch to a more randomness-efficient strategy. Each of the nodes informed in the first phase generates a random prefix, called a *seed*, of $\log n - \Theta(\log\log n)$ random bits, and continues to spread the rumor in a quasirandom fashion sending the random seed together with the rumor. Every non-informed node that receives a seed appends to it a new random suffix to compose a $\lceil \log n \rceil$-bit string. The node uses this string as the start point in its list, and begins to spread the rumor in the quasirandom way. The second phase lasts for $\Theta(\log n)$ rounds. During these rounds, seeds are distributed to at least a constant fraction of the nodes w.h.p., and these newly-informed nodes inform all the remaining nodes w.h.p. A total number of $O(n \log\log n)$ random bits are generated: $\log n - \Theta(\log\log n)$ bits by each of the $O(n/\log n)$ nodes informed in the first phase, and $O(\log\log n)$ bits by each of the other nodes.

In the next two sections, we describe the two phases of the protocol in more detail and provide their analysis.

**2.1 First Phase.** This phase lasts until the end of round $t_0 + \tau$, where $t_0$ is the round when the rumor is generated, $\tau = \lceil \log \kappa \rceil$, and $\kappa = \Theta(n/\log n)$. To simplify notation, we will assume from now on that the rumor is generated in round $t_0 = 0$; so, the last round of the first phase is round $\tau$. Suppose that node $v$ gets informed in round $t_v \le \tau$. (If $v$ is the source of the rumor, $t_v = 0$.) Then $v$ chooses a position $p_v$ in its list $L_v$ uniformly at random (this requires $\lceil \log n \rceil$ random bits),[4] and in rounds $t_v+1, t_v+2, \dots, \tau$, node $v$ transmits the rumor to nodes $L_v[p_v]$, $L_v[p_v \oplus 1], \dots, L_v[p_v \oplus (\tau - t_v - 1)]$, respectively, where $\oplus$ denotes the addition modulo $n$, i.e., $a \oplus b = (a + b) \bmod n$. Together with the rumor, $v$ transmits also a counter value used to detect the end of the phase: In round 0, the source of the rumor initializes this value to $\kappa + 1$; and, at the beginning of each subsequent round, each informed node decreases its copy of the counter by one, until its value becomes 0—indicating the end of the phase.

Since the number of informed nodes at most doubles in each round, the number of informed nodes at the end of the first phase is at most

$$2^\tau = 2^{\lceil \log \kappa \rceil} < 2\kappa = O(n/\log n). \tag{2.1}$$

Thus, the total number of random bits generated is $O(n)$. Next we show that w.h.p. the number of informed nodes at the end of the phase is also at least $\Omega(n/\log n)$.

[4] In this extended abstract, we ignore the problem of finding a uniformly distributed random value in $\{0, \dots, n-1\}$, if $n$ is not a power of two and only binary random values are available. However, it is easy to accommodate our algorithm and analysis for this case.

LEMMA 2.1. *For any constant $c > 0$, the number of informed nodes at the end of the first phase is at least $3^{-c}\kappa$ with probability $1 - O(n^{-c})$.*

To prove Lemma 2.1 we consider a rumor spreading process that is slightly different than the actual process. The difference is that an informed node may be *inactive*, that is, after it gets informed it does not spread the rumor to other nodes. The rest of the informed nodes, called *active*, behave as in the actual process. In this new processes, at the end of each round $t \leq \tau$, the number of informed nodes is stochastically smaller than the number of informed nodes in the original random process. Therefore, it suffices to show that the lower bound of Lemma 2.1 holds for this modified process.

Whether a node $v$ that gets informed in the first phase will be active or inactive is determined when $v$ chooses the random start point $p_v$ in its list. Note that, at that time, we know exactly to which nodes $v$ will send the rumor to in the remainder of the phase, if $v$ is active. Node $v$ will be inactive if at least one of these nodes also receives the rumor in this phase from an active node that was informed *before* $v$. To be more precise, let $t_u \leq \tau$ be the round in which node $u$ gets informed, and let $\prec$ be the total order defined on the set of all nodes informed in the first phase, where $u \prec v$ if and only if

- $t_u < t_v$, or
- $t_u = t_v$, and $u$ precedes $v$ in some predetermined ordering of all the nodes.

Let $C_v = \{L_v[p_v \oplus i] : i = 0, \ldots, \tau - t_v - 1\}$ be the set of nodes to which $v$ will send the rumor during the first phase, if $v$ is active. Then, $v$ is active if and only if $C_v \cap C_u = \emptyset$ for all nodes $u \prec v$.

For $0 \leq t \leq \tau$, we denote by $I_t$ and $A_t$ the set of informed nodes and the set of active nodes, respectively, at the end of round $t$. Also, $\overline{A_t} = I_t - A_t$ is the set of inactive nodes at that time.

Our analysis studies how $|A_t|$ increases. First we show that, w.h.p., at most a constant number of the nodes that get informed during the first $o(\log n)$ rounds are inactive.

LEMMA 2.2. *For any $t = o(\log n)$ and any constant $k \geq 1$, $\Pr(|\overline{A_t}| \geq k) \leq n^{-k+o(1)}$.*

*Proof.* Let $v_1 \prec v_2 \prec \cdots \prec v_{|I_t|}$ be the informed nodes at the end of round $t$. Further, for $i = 1, \ldots, |I_t|$, let $X_i$ be the 0/1 random variable with $X_i = 1$ if $v_i$ is inactive, and $X_i = 0$ if $v_i$ is active. Recall that $X_i$ is a function of $p_{v_1}, \ldots, p_{v_i}$, and that $X_i = 1$ if and only if at least one of the nodes in $\bigcup_{j<i, X_j=0} C_{v_j}$ belongs also to $C_{v_i} = \{L_v[p_{v_i} \oplus j] : j = 0, \ldots, \tau - t_{v_i} - 1\}$. Since the value of $p_{v_i}$ is chose uniformly at random among the $n$ possible list positions, the probability that $C_{v_i}$ contains a given node is $|C_{v_i}|/n \leq \tau/n$. Also, $\big|\bigcup_{j<i, X_j=0} C_{v_j}\big| \leq i \cdot \tau \leq |I_t| \cdot \tau \leq 2^t\tau$, since $|I_t| \leq 2^t$. Therefore, by the union bound, the probability that $C_{v_i}$ contains any of the nodes in $\bigcup_{j<i, X_j=0} C_{v_j}$ is

$$\Pr(X_i = 1 \,|\, p_{v_1}, \ldots, p_{v_{i-1}}) \leq 2^t\tau^2/n. \tag{2.2}$$

From this, and the fact that $|I_t| \leq 2^t$, it follows that the sum $\sum_{i \leq |I_t|} X_i = |\overline{A_t}|$ is dominated by the binomial random variable $B\big(2^t, 2^t\tau^2/n\big)$. Thus,

$$\begin{aligned}
\Pr(|\overline{A_t}| \geq k) &\leq \Pr\left(B\left(2^t, \frac{2^t\tau^2}{n}\right) \geq k\right) \\
&\leq \binom{2^t}{k} \cdot \left(\frac{2^t\tau^2}{n}\right)^k \leq \frac{(2^t)^k}{k!} \cdot \left(\frac{2^t\tau^2}{n}\right)^k \\
&\leq \frac{1}{n^{k-o(1)}},
\end{aligned}$$

where the second inequality was obtained by using the fact that $\Pr(B(m,q) \geq k) \leq \binom{m}{k} \cdot q^k$; and the last inequality holds because $t = o(\log n)$ and $k$ is a constant. ∎

The next result, Lemma 2.3, is a high-probability lower bound on the rate at which $|A_t|$ increases per round, when $|A_t|$ is sufficiently large. It says that $|A_t|$ essentially doubles in each round. More precisely, w.h.p., $|A_t| \geq (2 - 6q) \cdot |A_{t-1}|$, where

- $q \approx 1/\log^2 n$, for small $t$,
- $q \approx 2(\tau - t) \cdot \kappa/n = \Theta((\tau - t)/\log n)$, for large $t$ (close to $\tau$), and
- $q \approx 2^t\tau^2/n$, for intermediate values of $t$.

LEMMA 2.3. *Let $t \leq \tau$, and*

$$q = \frac{1}{\log^2 n} + \frac{1}{n} \cdot \min\{2(\tau - t) \cdot \kappa,\, 2^t\tau^2\}.$$

*Then*

$$\Pr\Big(|A_t| \geq (2 - 6q) \cdot |A_{t-1}| \;\Big|\; A_1, \ldots, A_{t-1}, |A_{t-1}| = \omega(\log^3 n)\Big) = 1 - n^{-\omega(1)}.$$

*Proof.* Fix all the random choices of the algorithm until the end of round $t-1$ (and, thus, the sets $A_1, \ldots, A_{t-1}$) in such a way that $|A_{t-1}| = \omega(\log^3 n)$. Let $v_1 \prec v_2 \prec \cdots \prec v_{|I_t|-|I_{t-1}|}$ be the nodes that get informed in round $t$. Also, for $i = 1, \ldots, |I_t| - |I_{t-1}|$, let $X_i$ be the 0/1 random variable with $X_i = 1$ if $v_i$ is inactive, and $X_i = 0$

if $v_i$ is active. Similarly to the proof of Lemma 2.2, we have that the probability that $C_{v_i}$ contains a given node is $C_{v_i}/n = (\tau - t)/n$. Also, $\left|\bigcup_{v \prec v_i} C_v\right| \leq \min\{2\kappa,\, 2^t\tau\}$, since, by (2.1), $2\kappa$ is an upper bound on the total number of nodes that get informed in the first phase, and $2^t$ is an upper bound on the number of informed nodes at the end of round $t$. So, similarly to (2.2),

$$\begin{aligned}\Pr(X_i = 1 \,|\, p_{v_1}, \ldots, p_{v_{i-1}}) &\leq \frac{\tau - t}{n} \cdot \min\{2\kappa,\, 2^t\tau\} \\ &\leq \frac{1}{n} \cdot \min\{2(\tau - t)\cdot\kappa,\, 2^t\tau^2\} \\ &= q - \frac{1}{\log^2 n}.\end{aligned}$$

From this, and the observation that $|I_t| - |I_{t-1}| = |A_{t-1}|$, it follows that the number of nodes that get informed in round $t$ and are inactive, that is,

$$\begin{aligned}\sum_{i \leq |I_t| - |I_{t-1}|} X_i &= |\overline{A_t}| - |\overline{A_{t-1}}| \\ &= (|I_t| - |A_t|) - (|I_{t-1}| - |A_{t-1}|) \\ &= 2|A_{t-1}| - |A_t|,\end{aligned}$$

is dominated by the binomial random variable $B\big(|A_{t-1}|,\, q - 1/\log^2 n\big)$. Thus, the probability that $2|A_{t-1}| - |A_t| > 6q|A_{t-1}|$, or equivalently, the probability that $|A_t| < (2 - 6q)\cdot|A_{t-1}|$, is at most

$$\begin{aligned}\Pr\left(B\big(|A_{t-1}|,\, q - 1/\log^2 n\big) > 6q|A_t|\right) &\leq \Pr\big(B(|A_{t-1}|, q) > 6q|A_{t-1}|\big) \\ &\leq 2^{-6q|A_{t-1}|} \\ &= 2^{-\omega(\log n)} = n^{-\omega(1)},\end{aligned}$$

where the second relation was obtained by using Chernoff bounds (Theorem 4.4.3 in [20]); and the second-to-last relation holds because $q \geq 1/\log^2 n$ and $|A_{t-1}| = \omega(\log^3 n)$. ∎

Using the above two lemmata, Lemma 2.1 can be derive as follows.

*Proof of Lemma 2.1.* By Lemma 2.2, applied for $t = 4\log\log n$ and $k = \lfloor c \rfloor + 1$, we obtain that with probability at least $1 - n^{-\lfloor c\rfloor - 1 + o(1)} = 1 - O(n^{-c})$ at most $\lfloor c \rfloor$ of the nodes informed at the end of round $r = 4\log\log n$ are inactive. Thus, with probability $1 - O(n^{-c})$,

$$|A_r| \geq 2^{r - \lfloor c\rfloor},$$

since the number of active nodes doubles during a round, unless some of the nodes informed in that round are inactive. This happens in at most $\lfloor c \rfloor$ of the first $r$ rounds.

Next we apply Lemma 2.3 for each of the rounds $t = r+1, \ldots, \tau$. Suppose that

$$|A_{t-1}| \geq 2^{r - \lfloor c\rfloor} = \omega(\log^3 n), \tag{2.3}$$

which allows us to apply Lemma 2.3. We distinguish two cases.

**Case 1:** If $r + 1 \leq t \leq \log n - 4\log\log n$ then, with probability $1 - n^{-\omega(1)}$,

$$\begin{aligned}|A_t| &\geq \left(2 - \frac{6}{\log^2 n} - \frac{6\cdot 2^t\tau^2}{n}\right)\cdot|A_{t-1}| \\ &\geq \left(2 - \frac{6}{\log^2 n} - \frac{6\cdot(n/\log^4 n)\cdot\tau^2}{n}\right)\cdot|A_{t-1}| \\ &= \big(2 - o\big(\tfrac{1}{\log n}\big)\big)\cdot|A_{t-1}|.\end{aligned}$$

**Case 2:** If $\log n - 4\log\log n \leq t \leq \tau$ then, with probability $1 - n^{-\omega(1)}$,

$$\begin{aligned}|A_t| &\geq \left(2 - \frac{6}{\log^2 n} - \frac{6\cdot 2(\tau - t)\cdot\kappa}{n}\right)\cdot|A_{t-1}| \\ &\geq \left(2 - \frac{6}{\log^2 n} - \frac{6\cdot 2\cdot 4\log\log n\cdot\kappa}{n}\right)\cdot|A_{t-1}| \\ &= \big(2 - o\big(\tfrac{1}{\log\log n}\big)\big)\cdot|A_{t-1}|.\end{aligned}$$

It is now easy to show (inductively) that, if $|A_r| \geq 2^{r-\lfloor c\rfloor}$ and the above inequalities hold for $|A_{r+1}|, \ldots, |A_{t-1}|$, then the sequence $|A_r|, \ldots, |A_{t-1}|$ is non-decreasing, and thus, condition (2.3) holds.

Finally, by the union bound, all the above bounds hold simultaneously with probability at least

$$1 - O(n^{-c}) - (\tau - r)\cdot n^{-\omega(1)} = 1 - O(n^{-c}).$$

So, with this probability,

$$\begin{aligned}|A_\tau| &\geq 2^{r-\lfloor c\rfloor}\cdot(2 - o(1/\log n))^{\log n - 4\log\log n - r} \\ &\qquad\cdot(2 - o(1/\log\log n))^{\tau - (\log n - 4\log\log n)} \\ &= 2^r\cdot 2^{-\lfloor c\rfloor}\cdot 2^{\log n - 4\log\log n - r} \\ &\qquad\cdot(1 - o(1/2\log n))^{\log n - 4\log\log n - r} \\ &\qquad\cdot 2^{\tau - (\log n - 4\log\log n)} \\ &\qquad\cdot(1 - o(1/2\log\log n))^{\tau - (\log n - 4\log\log n)} \\ &\geq 2^\tau\cdot 2^{-\lfloor c\rfloor}\cdot(1 - o(1/\log n))^{\log n} \\ &\qquad\cdot(1 - o(1/\log\log n))^{4\log\log n} \\ &\geq \kappa\cdot 2^{-c}\cdot(1 - o(1)) \\ &\geq 3^{-c}\cdot\kappa,\end{aligned}$$

for all large enough $n$. And since $|I_\tau| \geq |A_\tau|$, the lemma follows. ∎

**2.2 Second Phase.** In this phase, every node that was informed in the first phase generates and distributes a random bit-string along with the rumor. This random bit-string is called a *seed*, and the nodes informed during the first phase are called *seeders*. Seeds are used by nodes not informed in the first phase, to generate the start points in their own lists. More precisely, at the end of the first phase, every seeder generates a seed of length $\ell - \ell^*$, where $\ell = \lceil \log n \rceil$ and $\ell^* = \lceil 3 \log\log n \rceil$.[5] Then, in each subsequent round, the seeder sends the rumor along with the seed to the next node in its list, starting from the current position at the end of the first phase. A seeder stops distributing the rumor (and the seed) after $\Theta(\log n)$ rounds from the beginning of the second phase.

A node that receives a seed and is not a seeder is called a *seed receiver*. Let $s$ be the first seed that seed receiver $u$ receives, and let $t$ be the round when that happens. (If $u$ receives multiple seeds in that round, $s$ can be chosen to be any of them, arbitrarily—but the decision must not depend on the values of the seeds.) Node $u$ then generates a random suffix $x$ of $\ell^*$ bits, and uses the bit-string $s \circ x$ as the start point $p_u$ in its list from which $u$ begins to send the rumor in round $t+1$. Every seed receiver distributes the rumor for $\Theta(\log n)$ rounds. It is possible, that some node $w$ that is neither a seeder nor a seed receiver receives the rumor (but no seed) from some seed receiver. In this case, $w$ does not distribute the rumor, unless it later on receives a seed and thus becomes a seed receiver.

We begin the analysis of this phase by showing that the total number of seed receivers is $\Omega(n)$ w.h.p. Recall that $\kappa = \Theta(n/\log n)$.

LEMMA 2.4. *Suppose that every seeder distributes its seed for $r = \Theta(\log n)$ rounds. Then, for any constant $c > 0$, the total number of seed receivers is at least $(1/34) \cdot \min\{n, 2 \cdot 3^{-c} r\kappa\}$ with probability $1 - O(n^{-c})$.*

*Proof.* Let $v_1, \ldots, v_Z$ be the list of seeders in the order that they were informed (seeders informed in the same round are listed in some predetermined order). Let also $N_i$, for $i \le Z$, be the number of nodes (seed receivers or seeders) that receive a seed from node $v_i$, but not from nodes $v_j$, $j < i$.

First, we show that $N_i \ge r/4$ with constant probability, if $i \le n/2r$. Clearly, $N_i = N_i(p_{v_1}, \ldots, p_{v_i})$, and

$$\mathbf{E}[N_i \,|\, p_{v_1}, \ldots, p_{v_{i-1}}] \ge r \cdot \frac{n - (i-1)\cdot r}{n},$$

because the first $i-1$ seeders send seeds to at most $(i-1)\cdot r$ nodes, thus, the probability that the $k$-th of the $r$ nodes that receive a seed from $v_i$ is not one of those $(i-1)\cdot r$ nodes is at least $\frac{n-(i-1)r}{n}$. Therefore, for $i \le \min\{Z, n/2r\}$, $\mathbf{E}[N_i \,|\, p_{v_1}, \ldots, p_{v_{i-1}}] \ge r/2$, and, by Markov's inequality,

$$\begin{aligned}
&\Pr(N_i \ge r/4 \,|\, p_{v_1}, \ldots, p_{v_{i-1}}) \\
&\quad = 1 - \Pr(N_i < r/4 \,|\, p_{v_1}, \ldots, p_{v_{i-1}}) \\
&\quad = 1 - \Pr(r - N_i > 3r/4 \,|\, p_{v_1}, \ldots, p_{v_{i-1}}) \\
&\quad \ge 1 - \frac{r - r/2}{3r/4} = 1/3. \qquad (2.4)
\end{aligned}$$

Next, we bound w.h.p. the number of seeders, among the first $z \le n/2r$ seeders, for which $N_i \ge r/4$. Let $X_i$ be the 0/1 random variable with $X_i = 1$ if and only if $N_i \ge r/4$ or $i > \min\{Z, n/2r\}$. (Note that $X_i$ is defined for all $i$, not just for $i \le Z$.)

If $i \le \min\{Z, n/2r\}$, then $X_i = X_i(N_i) = X_i(p_{v_1}, \ldots, p_{v_i})$, and, by (2.4), $\mathbf{E}[X_i \,|\, p_{v_1}, \ldots, p_{v_{i-1}}] \ge 1/3$; while if $i > \min\{Z, n/2r\}$, $X_i = 1$. From this we see that, for any $z$, the sum $\sum_{i \le z} X_i$ dominates the binomial random variable $B(z, 1/3)$. And, applying Chernoff bounds (Theorem 4.5 in [20]), we obtain for $z = \omega(\log n)$,

$$\begin{aligned}
\Pr\Big(\sum_{i \le z} X_i \ge z/4\Big) &\ge \Pr\big(B(z, 1/3) \ge z/4\big) \\
&= 1 - n^{-\omega(1)}. \qquad (2.5)
\end{aligned}$$

We can now bound the total number $\sum_{i \le Z} N_i$ of nodes that receive seeds as follows. Let $z = \min\{3^{-c}\kappa, n/2r\} = \Theta(n/\log n)$. Then,

$$\begin{aligned}
&\Pr\Big(\sum_{i \le Z} N_i \ge zr/16\Big) \\
&\qquad \ge \Pr\Big(\Big(\sum_{i \le z} N_i \ge zr/16\Big) \wedge (Z \ge z)\Big) \\
&\qquad \ge \Pr\Big(\Big(\sum_{i \le z} X_i \ge z/4\Big) \wedge (Z \ge 3^{-c}\kappa)\Big),
\end{aligned}$$

where for the last inequality we used the fact that, if $i \le z \le Z$, then $N_i \ge (r/4) \cdot X_i$, since $X_i = 1$ only if $N_i \ge r/4$. Combining the above result with (2.5) and Lemma 2.1 (which says that $\Pr(Z \ge 3^{-c}\kappa) = 1 - O(n^{-c})$), we obtain that

$$\Pr\Big(\sum_{i \le Z} N_i \ge zr/16\Big) = 1 - O(n^{-c}).$$

Therefore, with probability $1 - O(n^{-c})$, at least $zr/16 = \Theta(n)$ nodes receive seeds. And since at most $2\kappa = \Theta(n/\log n)$ of them are seeders, it follows that, with probability $1 - O(n^{-c})$, there are at least $zr/16 - 2\kappa \ge zr/17 = (1/17) \cdot \min\{3^{-c} r\kappa, n/2\}$ seed receivers (where the inequality holds for large enough $n$). ∎

[5] Any constant greater than 2 can be used in place of 3.

We have shown that the total number of seed receivers is $\Omega(n)$ w.h.p. Next, we show that if the number of seed receivers is indeed $\Omega(n)$, then all nodes get informed w.h.p., provided that seed receivers distribute the rumor for a sufficiently large, logarithmic number of rounds.

LEMMA 2.5. *Suppose that every seed receiver distributes the rumor for at least $d = \Theta(\log n)$ rounds. If the total number of seed receivers is at least $\beta n$, for some constant $\beta > 0$, then all nodes get informed with probability $1 - O(n \cdot e^{-\beta d/5})$.*[6]

*Proof.* We say a seeder $v$ *seeds* a seed receiver $u$, if $u$ uses the seed generated by $v$ and sent to $u$ to determine the start point $p_u$ in its list. (Note that if a seed receiver receives seeds from multiple nodes, it will only be seeded by one of those nodes. For our analysis the seeder can be chosen arbitrarily—but the decision must not depend on the values of the seeds.)

The proof of the lemma is based on the following key result.

CLAIM 2.1. *Suppose that seeder $v$ seeds the seed receivers $u_1, \ldots, u_m$. Let $w$ be an arbitrary node, other than the $u_1, \ldots, u_m$. Then, $w$ receives the rumor from at least one of the $v_1, \ldots, v_m$ with probability at least $\big(1 - o(1)\big) \cdot md/4n$.*

From this claim, the lemma follows easily: Fix the set of seeders, and the seed receivers seeded by each seeder, and let $m_i$ be the number of seed receivers seeded by the $i$-th seeder. Let $w$ be an arbitrary node that is not a seeder nor a seed receiver. Since the seeds sent by different seeders are independent, the probability that none of the seeders seeds a seed receiver that sends the rumor to $w$ is at most

$$\begin{aligned}
&\prod_i \Big(1 - \big(1 - o(1)\big) \cdot m_i d/4n\Big) \\
&\qquad \le \prod_i \exp\Big(-\big(1 - o(1)\big) \cdot m_i d/4n\Big) \\
&\qquad = \exp\Big(-\big(1 - o(1)\big) \cdot \sum_i m_i d/4n\Big) \\
&\qquad \le \exp\Big(-\big(1 - o(1)\big) \cdot \beta d/4\Big) \\
&\qquad = O\left(e^{-\beta d/5}\right),
\end{aligned}$$

where the first relation was obtained using the fact that $1 + x \le e^x$; and the second-to-last relation was obtained by using the lemma's assumption that the total number of seed receivers is $\sum_i m_i \ge \beta n$. Hence, by the union bound, with probability at least $1 - O(n \cdot e^{-\beta d/5})$, all nodes $w$ get informed.

It remains to prove Claim 2.1. Recall that each seed is chosen among $2^{\ell - \ell^*}$ many possible seeds, where $\ell = \lceil \log n \rceil$ and $\ell^* = \lceil 3 \log\log n \rceil$; and each suffix is chosen among $2^{\ell^*}$ many possible suffices. We denote the set of all possible seeds by $A$. We say that seed $s$ is *good for node $u_i$*, if there are at least $d/2$ suffixes $x$ such that if $p_{u_i} = s \circ x$ then $u_i$ sends the rumor to $w$. Since each $u_i$ distributes the rumor for at least $d$ rounds, there is at least one good seed for every $u_i$. Thus, the probability that a randomly chosen seed is good for $u_i$ is at least $1/|A| = 1/2^{\ell - \ell^*}$. Also, given that the seed that seeder $v$ chooses is good for $u_i$, the probability that $u_i$ chooses a suffix such that $u_i$ sends the rumor to $w$ is at least

$$q := \frac{d/2}{2^{\ell^*}}.$$

For any seed $s$, let $z_s$ be the number of seed receivers among the $u_1, \ldots, u_m$ for which $s$ is good; i.e.,

$$z_s = |\{i : s \text{ is good for } u_i\}|.$$

Since for each $u_i$ there is at least one good seed,

$$\sum_{s \in A} z_s = \sum_{1 \le i \le m} |\{s : s \text{ is good for } u_i\}| \ge m. \tag{2.6}$$

Let $\mathcal{E}_s$ be the event that seeder $v$ chooses seed $s$, and let $\mathcal{I}$ be the event that at least one of the nodes $u_1, \ldots, u_m$ sends the rumor to $w$. Suppose that $v$ chooses $s$, and that $w$ does not get informed. Then all the $z_s$ seed receivers for which seed $s$ is good have to pick the wrong suffix. Hence,

$$\Pr(\neg\mathcal{I} \,|\, \mathcal{E}_s) \le (1 - q)^{z_s}.$$

Thus,

$$\Pr(\neg\mathcal{I}) = \sum_{s \in A} \Pr(\neg\mathcal{I} \,|\, \mathcal{E}_s) \cdot \Pr(\mathcal{E}_s) \le \frac{1}{|A|} \cdot \sum_{s \in A} (1 - q)^{z_s}.$$

From (2.6), it follows that the sum $\sum_{s \in A} (1 - q)^{z_s}$ is maximized when $z_s = 0$ for all but one seed $s^*$, and $z_{s^*} = m$. Thus,

$$\begin{aligned}
\Pr(\neg\mathcal{I}) &\le \frac{1}{|A|} \cdot \big((|A| - 1) \cdot 1 + 1 \cdot (1 - q)^m\big) \\
&= 1 - \frac{1 - (1 - q)^m}{|A|}.
\end{aligned}$$

We can bound $(1 - q)^m$ as follows. Note that

$$q \cdot m = \frac{d \cdot m/2}{2^{\lceil 3 \log\log n \rceil}} \le \frac{d \cdot m/2}{(\log n)^3} = o(1),$$

[6] Note that the number of seed receivers does not depend on the choice of parameter $d$, because a node informed by a seed receiver can also become a seed receiver, if it is contacted by a seeder later on.

since $d = \Theta(\log n)$, and $m$ is at most equal to the number of rounds for which $v$ distributes its seed, which is $\Theta(\log n)$ rounds. So, using that fact that $(1-\epsilon)^k \leq 1 - k\epsilon + (k\epsilon)^2$, which holds if $k\epsilon \leq 1$, we obtain that $(1-q)^m = 1 - (1-o(1)) \cdot qm$. Thus,

$$\begin{aligned}\Pr(\neg\mathcal{I}) &\leq 1 - (1-o(1)) \cdot \frac{qm}{|A|} \\ &= 1 - (1-o(1)) \cdot \frac{md}{2 \cdot 2^{\ell}}. \\ &\leq 1 - (1-o(1)) \cdot \frac{md}{4n}.\end{aligned}$$

This completes the proof of Claim 2.1, and of Lemma 2.5. ∎

The next statement summarizes the properties of our protocol. Recall that the first phase lasts for $\tau$ rounds, where $\tau = \lceil \log \kappa \rceil = \log n - \log\log n + O(1)$ is a parameter of the protocol. Also, in the second phase, every seeder (i.e., every node informed during the first phase) distributes its seed for $r$ rounds, and every seed receiver distributes the rumor for $d$ rounds, where $r, d = \Theta(\log n)$ are other protocol parameters.

COROLLARY 2.1. *For any constant $c > 0$, there exist parameters $\tau$, $r$, $d$, such that the protocol informs all nodes in $O(\log n)$ rounds with probability $1 - O(n^{-c})$, and uses a total number of $3n \log\log n + O(n)$ random bits.*

*Proof.* For any choice of the parameters $\kappa$ and $r$ such that $\kappa = \Theta(n/\log n)$ and $r = \Theta(\log n)$, we show that the required guarantees hold if we choose $d$ such that

$$d \geq \frac{170 \cdot (c+1)}{\min\{1,\, 2 \cdot 3^{-c} r\kappa/n\}} \cdot \ln n. \tag{2.7}$$

Since $r\kappa/n = \Theta(1)$, the quantity by which $\ln n$ is multiplied in the above formula is bounded by a constant.

Clearly, the protocol runs for at most $\tau + r + d = O(\log n)$ rounds.

The total number of random bits used is at most

$$4\kappa\lceil \log n \rceil + 3n\lceil \log\log n \rceil = 3n \log\log n + O(n),$$

because each of the at most $2^{\tau} \leq 2\kappa$ nodes that get informed in the first phase generates $\lceil \log n \rceil$ random bits to choose the start point in its list, and another $\lceil \log n \rceil - \lceil 3 \log\log n \rceil$ random bits to choose its seed; and each of the at most $n$ seed receivers generates $\lceil 3 \log\log n \rceil$ random bits.

Finally, we bound the probability that all nodes get informed as follows. From Lemma 2.4, it follows that, with probability $1 - O(n^{-c})$, the total number of seed receivers is at least $\beta n$, where

$$\beta := (1/34) \cdot \min\{1,\, 2 \cdot 3^{-c} r\kappa/n\}.$$

And, by Lemma 2.5, given that there are at least $\beta n$ seed receivers, all nodes get informed with probability $1 - O(n \cdot e^{-\beta d/5})$. Thus, all nodes get informed with probability at least

$$1 - O(n^{-c}) - O(n \cdot e^{-\beta d/5}).$$

By Inequality (2.7) and the definition of $\beta$, we obtain that $n \cdot e^{-\beta d/5} \leq n^{-c}$. Hence, the probability above is $1 - O(n^{-c})$. ∎

## 3 Bounds on the Minimal Randomness Requirements

In this section we study the theoretically minimal amount of randomness that is necessary to spread the rumor in $O(\log n)$ rounds to all nodes.

We consider the complete graph $K_n$, with vertices $1, \ldots, n$. The *input* for the rumor spreading problem is a pair $(S, \mathcal{L})$, where $S$ is the source of the rumor and $\mathcal{L} = (L_1, \ldots, L_n)$ is a sequence of adjacency lists. Node $i$, $1 \leq i \leq n$, is given list $L_i$, but it has no a priori information about its own or any other adjacency list; in each round the node can only choose an index $j$ and then the rumor is sent to the node stored in $L_i[j]$. We call $L_i[j]$ the *$j$-th neighbor* of node $i$.

**3.1 Oblivious Protocols.** We assume that each node $i$ is equipped with a private random bit-string $R_i$ of length $b$. Node $i$ is *oblivious*, if its decision to which neighbor to send the rumor to depends only on the random string $R_i$ and the number of rounds passed since $i$ received the rumor. A rumor spreading protocol is *oblivious*, if all nodes are oblivious. Note that the fully random protocol, the quasirandom protocol, and the gate-model [4, 5] are all oblivious.

We note that for an oblivious protocol that uses only $o(\log n)$ bits of randomness, the broadcast time is at least $n^{1-o(1)}$. Consider two phases of $r_1$ and $r_2$ rounds, respectively. After $r_1$ rounds, at most $2^{r_1}$ nodes are informed, and these nodes can inform at most $2^{r_1} \cdot r_2$ other nodes. The nodes that get informed during the second phase have at most $r_2$ rounds in which they can send messages. If each of them has only $b$ random bits available, then they can only "address" $2^b \cdot r_2$ positions in their lists. Thus, we can fix all adjacency lists such that nodes that were informed during the second phase, only inform new nodes in $\{1, \ldots, 2^b \cdot r_2\}$. Hence, if $2^{r_1}(1 + r_2) + 2^b \cdot r_2$ is less than $n$, not all nodes can be informed.

Note that in their lower bound proof for the gate-model, Doerr and Fouz [5] also split the random process into two phases. They make the same worst-case assumption, that $2^{r_1}$ nodes get informed during the first phase. Their analysis of the second phase is different

from ours, though, as theirs is targeted towards the gate-model.

THEOREM 3.1. *For any oblivious protocol, where each node uses at most $b < \log n - 1$ random bits, there is an input $(S, \mathcal{L})$, such that the rumor cannot be distributed to all nodes in fewer than $b + \lfloor n/2^{b+1} \rfloor$ rounds.*

*Proof.* Fix an arbitrary source $S$ and some integers $r_1, r_2$ with $r_1 < \log n$ and let $r = r_1 + r_2$. In each round, the number of informed nodes can double at most. Hence, after $r_1 < \log n$ rounds, at most $2^{r_1}$ nodes are informed. Let $S_1$ be the set of these nodes. Further, let $S_2$ be the set of nodes not in $S_1$, that receive the rumor directly from a node in $S_1$ during rounds $r_1 + 1, \ldots, r_1 + r_2$. Clearly, $|S_2| \leq r_2 \cdot |S_1|$, and thus

$$|S_1 \cup S_2| \leq 2^{r_1}(1 + r_2). \tag{3.8}$$

Now note that during the first $r$ rounds, all nodes in $\overline{S_1} = \{1, \ldots, n\} - S_1$ can send messages for at most $r_2$ rounds. Hence, the number of nodes that receive the rumor directly from nodes in $\overline{S_1}$ is bounded by the number of nodes that would receive the rumor if every node sent messages for $r_2$ rounds.

Since each node $i$ acts obliviously and uses a random string of length $b$, its first $r_2$ messages are sent to neighbors $L_i[j]$, where $j$ is an index from a set $J_i$ of size at most $2^b \cdot r_2$. Clearly, we can choose the input so that $L_i[j] \in \{1, \ldots, 2^b \cdot r_2\}$ for all $j \in J_i$. Hence, the first $r_2$ messages by node $i$ can reach only nodes in $\{1, \ldots, 2^b \cdot r_2\}$.

It follows that during the first $r = r_1 + r_2$ rounds, only nodes in $S_1 \cup S_2 \cup \{1, \ldots, 2^b \cdot r_2\}$ receive the rumor. Hence, by (3.8), the total number of nodes that receive the rumor is bounded by

$$\Delta := 2^{r_1}(1 + r_2) + r_2 \cdot 2^b = 2^{r_1} + r_2(2^{r_1} + 2^b).$$

Now choose $r_1 = b$ and $r_2 = \lfloor n/2^{b+1} \rfloor - 1$. Then

$$\Delta < (r_2 + 1) \cdot 2^{b+1} \leq n.$$

Hence, not all nodes receive the rumor during the first $r_1 + r_2$ rounds. ∎

We conclude that if each node has $\log n - \log\log n - \omega(1)$ random bits available, then $\omega(\log n)$ rounds are necessary to distribute the rumor using an oblivious protocol. Thus, the only way to achieve efficient rumor spreading with $o(\log n)$ randomness requires nodes to acquire additional information from incoming messages.

**3.2 Non-Oblivious Protocols.** While in principle non-oblivious nodes might generate some entropy just by counting the number of incoming messages, it seems difficult to derive any protocol that does not require additional information (in particular some random bits) to be transmitted together with the rumor. In the following, we assume that the amount of communication between nodes can be unbounded. In particular, the first node to receive the rumor can generate a random string and then share it with all other nodes by appending the random string to all messages sent. For simplicity, we also assume that nodes pass the current round number (i.e., the age of the rumor) along with their messages, so that nodes can base their decisions on that.

We show that in such a setting, one $O(\log n)$-bit random string suffices to spread the rumor to all nodes within $O(\log n)$ rounds w.h.p. The idea is the following: We take the classical fully random rumor spreading protocol, and fix the random bit strings used by each node arbitrarily. This way, we obtain a deterministic protocol. Denote $\mathcal{P}$ the set of all deterministic protocols obtained this way. Now we choose $B$ protocols from $\mathcal{P}$ at random to obtain a set $\mathcal{P}'$ of deterministic protocols. It is not hard to prove that for any input $(S, \mathcal{L})$ and large enough $B = n^{O(1)}$, a randomly chosen deterministic protocol in $\mathcal{P}'$ distributes the rumor within $O(\log n)$ rounds to all nodes w.h.p. Thus, we obtain an efficient random protocol, where the first node randomly chooses a protocol $P \in \mathcal{P}'$ and then appends the index of that protocol to each message.

One technical issue arises because in the fully random protocol, each node has access to a *private* random string $R_i$. Therefore, although nodes are anonymous, each node $i$ implicitly uses its ID $i$ to access its random bits. When we simulate one of the deterministic protocols, a node that receives a message telling the node to run protocol $P$, cannot conclude how to act, because it does not have access to its ID. Therefore, as a first step, we show that any private-coin protocol can be simulated by a public-coin protocol, where all nodes have access to the same random string, and do not implicitly use their IDs.

**3.3 Public- versus Private-Coin Protocols.** In a private-coin protocol, each node $i$ bases its decision (i.e., for which index $j$ it sends its next message to node $L_i[j]$) on the current round number, the history of all messages $i$ has received so far, and a private value $R_i$ that is chosen uniformly at random from a countable domain $D$. (All random values $R_1, \ldots, R_n$ are chosen independently.) However, nodes cannot use their IDs for anything else other than to access their private random string. Thus, if two nodes $i \neq j$ receive the same random string $R_i = R_j$, then they have to act identically. In a public-coin protocol, all "private" coin-

flips show the same value. The *randomness* of a rumor spreading protocol is the entropy of the random vector $(R_1, \ldots, R_n)$.

In the following we formally define private- and public-coin protocols. A *private-coin* rumor spreading protocol is a function

$$P : H \times D \times \mathbb{N} \to \{\perp\} \cup \{1, \ldots, n\} \times \{0,1\}^*,$$

where $H$ is the set of all finite lists whose elements are (multi-)sets of binary strings; and $D$ is some countable domain. We require that $P(h, \cdot, \cdot) = (\perp, \cdot)$ whenever all the elements of list $h$ are empty sets. Intuitively, if $P(h, s, r) = (j, m)$ then a node with history $h$ of received messages and random string $s$, sends message $m$ to its $j$-th neighbor in round $r$ if $j \neq \perp$. More precisely, the semantics of $P$ is the following: Before the protocol starts, for each node $i \in \{1, \ldots, n\}$ a private random string $R_i \in D$ is chosen uniformly and independently at random. At the beginning of the protocol (i.e., in round 0), node 1 receives the initial message 1 (all other nodes receive no messages). Suppose that $M_k$ is the (multi-) set of messages that node $i$ received in round $k$,[7] for $k = 0, \ldots, r$, and let $(j, m) = P(\langle M_1, \ldots, M_r \rangle, R_i, r+1)$. If $j = \perp$, then in round $r+1$ node $i$ sends no message, otherwise it sends message $m$ to node $L_i[j]$.

A *public-coin* rumor spreading protocol $P$ is defined as above, except that $R_1 = R_2 = \cdots = R_n = R$, where $R \in D$ is a random string used by all nodes.

We say $P$ has *randomness* $b$, if $n \cdot \log |D| = b$ in the case that $P$ is a private-coin protocol $P$, and $\log |D| = b$ in the case it is a public-coin protocol. If $D$ is not a finite set, then $P$ has *unbounded* randomness.

A rumor spreading protocol has *success mode* $(p, r)$, if during a run of the protocol with probability at least $p$ all nodes get informed within $r$ rounds.

LEMMA 3.1. *For every private-coin protocol $P$, there is a public-coin protocol $P'$ with the same success mode as $P$.*

The idea is to add an ID distribution mechanism to protocol $P$, that allows each node to determine a unique ID from a set $\{1, \ldots, z\}$, based on the first message it receives. Nodes can then use a large public random string $R \in D^z$ and access a unique portion of that string when making their random decisions.

*Proof of Lemma 3.1* Let $P$ be a private-coin protocol with success mode $(p, r)$, and let $D$ be the domain of the private random strings used by nodes. Let

$$Z = \bigcup_{1 \leq j \leq r} \{0, \ldots, r\}^j \quad \text{and} \quad z = |Z|.$$

Nodes will run protocol $P$, but instead of using private random strings, they have to use one public random string $R = (R_1, \ldots, R_z) \in D^z$. (Since $D$ is countable, so is $D^z$.) The idea is to distribute IDs in $Z$ to the nodes, so that each node can determine a unique ID $i \in Z$ from the first message the node receives. After a node has determined its ID, it runs the protocol $P$ using the random string $R_i$, but adding additional information to its messages in order to allow the receiving nodes to determine their own unique IDs.

Below we show how to achieve that each node which receives a message in the first $r$ rounds, also determines a unique ID in $Z$. Clearly, then for the first $r$ rounds the resulting protocol behaves exactly as $P$; hence it has the same success mode.

The ID of a node $u$ is determined by the ID of the node that sends the first message to $u$ and the round number in which that message is sent. The unique node that receives a message in round 0 (i.e., the node that initially generates the rumor) uses ID 0. Whenever a node with ID $i$ sends the rumor, it appends $i$ to that message. Any node $u$ that receives its first message in round $s$, extracts the ID $i$ of the sender from the message, and from then on uses ID $(i, s)$.

In order to ensure that the IDs are in $Z$, after round $r$ nodes switch to a trivial deterministic protocol for which no additional IDs need to be assigned (i.e., in round $r + r'$ a node sends the rumor to the $(r \oplus r')$-th neighbor in its list). A simple induction on the round number $s$, in which a message is sent that determines the ID of the recipient, shows that IDs are unique: First note that the last component of such an ID has value $s$. Only one ID is generated for $s = 0$, which settles the base case. Now suppose $s > 0$. For the purpose of a contradiction assume that two different nodes $v_1$ and $v_2$ receive identical IDs $(i_1, i_2, \ldots, i_{t-1}, s)$. Then in round $s$ two different nodes $u_1$ and $u_2$ send the same ID $i := (i_1, \ldots, i_{t-1})$. But then $u_1$ and $u_2$ both have ID $i$, contradicting the induction hypothesis for $s' = i_{t-1}$, because ID $i$ was generated in round $s' < s$. ∎

### 3.4 Low-Randomness Public-Coin Protocols.

We now show that any public-coin rumor spreading protocol that uses an arbitrary amount of randomness can be converted into a public-coin rumor spreading protocol that has the same round complexity, only a slightly increased error probability, but very low randomness requirements.

[7] $M_k$ is a multi-set because $i$ may receive messages from more than one nodes in the same round—and two messages may be identical.

LEMMA 3.2. *If there is a public-coin rumor spreading protocol with success mode $(p, r)$ (and possibly unbounded randomness), then for any $p' \in [0, p]$ and $B \in \mathbb{N}$, where*

$$B \geq \frac{2 \cdot (\ln n + n \cdot \ln(n!))}{p \cdot (1 - p'/p)^2},$$

*there exists a public-coin rumor spreading protocol with success mode $(p', r)$ and randomness $\log B$.*

The proof is based on the probabilistic method: Fixing the public random string used by $P$ to some arbitrary value yields a deterministic protocol. We determine a set $\mathcal{P}'$ of $B$ such deterministic protocols by choosing $B$ random strings. In $P'$, the first node simply chooses one of the deterministic protocols in $\mathcal{P}'$ uniformly at random and then simulates it, but adding the description of that protocol to each message. Then, all nodes can follow that deterministic protocol.

*Proof of Lemma 3.2.* Suppose $P$ is a public-coin rumor spreading protocol with success mode $(p, r)$. Let $D$ be the (possibly infinite but countable) domain from which random strings $R$ are chosen for $P$. Let $\mathcal{P}$ be the set of all deterministic protocols obtained from $P$ by fixing the random bit-string $R$.

We now construct a public-coin protocol $P'$ with randomness $b := \log B$ as follows: We determine a set $\mathcal{P}' \subseteq \mathcal{P}$ of size $B$, by independently choosing $B$ random strings $s_1, \ldots, s_B$ from $D$ at random. Now our protocol $P'$ is defined as $P'(h, j, r) = P(h, s_j, r)$, where $j \in \{1, \ldots, B\}$. I.e., if the global random string of protocol $P'$ is $j$, then the nodes act as in protocol $P$ but use the random string $s_j$. Hence, the new protocol uses $D' = \{1, \ldots, B\}$ as the domain for random strings and thus has randomness $b$.

Note that each such random string $s_j$ defines a deterministic protocol $P_j$. We say that the deterministic protocol $P_j$ *succeeds* for the input $(S, \mathcal{L})$, if run on that input, $P_j$ spreads the rumor in at most $r$ rounds to all other nodes.

Now fix an input $(S, \mathcal{L})$. For every $1 \leq j \leq B$, let $Y_j$ be an indicator variable, where $Y_j = 1$ if and only if protocol $P_j$ succeeds on input $(S, \mathcal{L})$. Since each random string $s_j \in D'$ is chosen independently at random among all random strings in $D$, all random variables $Y_j$ are independent and $\mathbf{E}[Y_j] \geq p$. Thus, defining $Y = Y_1 + \cdots + Y_B$ and $\delta = 1 - p'/p$, we obtain from Chernoff bounds

$$\begin{aligned}\Pr\big(Y < B \cdot p'\big) &= \Pr\big(Y < B \cdot p \cdot (1 - \delta)\big) \\ &< \exp\left(-B \cdot p \cdot \delta^2/2\right) \\ &< \exp\left(-B \cdot p(1 - p'/p)^2/2\right).\end{aligned}$$

Now note that there are at most $n \cdot (n!)^n$ inputs ($n$ possibilities to choose the source $S$ and $n!$ possibilities for each of the $n$ adjacency lists.). Thus, by the union bound

$$\begin{aligned}&\Pr\big(P' \text{ has success mode } (p', r)\big) \\ &\qquad > 1 - n \cdot (n!)^n \cdot \exp\left(-B \cdot p(1 - p'/p)^2/2\right).\end{aligned}$$

If the term on the right-hand side is at least 0, then a public-coin protocol with success mode $(p', r)$ and randomness $B$ exists. ∎

Combining Lemmata 3.1 and 3.2, and choosing $p' = 1 - 2\epsilon$, we can summarize:

COROLLARY 3.1. *If there is a private-coin protocol with success mode $(p, r)$, then for any $1 - p \leq \epsilon \leq 1/2$ there is a public-coin protocol with success mode $(1 - 2\epsilon, r)$ and randomness*

$$2\left(\log n + \log \frac{1}{\epsilon}\right) + \log \ln n + 1.$$

*Proof.* Let $p' = 1 - 2\epsilon$. Then

$$\begin{aligned}p \cdot (1 - p'/p)^2 &= (p - p')^2/p \\ &\geq (p - p')^2 \\ &\geq \big((1 - \epsilon) - (1 - 2\epsilon)\big)^2 \\ &= \epsilon^2.\end{aligned}$$

Using the Stirling series, it is not hard to see that

$$\ln n + n \ln(n!) \leq n^2 \ln n,$$

for all positive integers $n$. Hence, we can conclude from Lemma 3.2, that for

$$B \geq 2 \cdot \ln n \cdot \left(\frac{n}{\epsilon}\right)^2,$$

there is a public-coin protocol with error-mode $(p', r)$. ∎

COROLLARY 3.2. *There exists a rumor spreading protocol that with probability $1 - o(1)$ informs every node within $\log n + \ln n + O(1)$ rounds, and that uses at most $2 \log n + \log \log n + o(\log \log n)$ random bits.*

*Proof.* Pittel [21] proved that the fully random rumor spreading protocol has success mode $\big(1 - \delta, r\big)$, where $\delta = o(1)$ and $r = \log n + \ln n + O(1)$. Applying Corollary 3.1 with $\epsilon = \max\{\delta, 1/\log \log n\}$ yields the desired protocol. ∎

It turns out that the upper bound from Corollary 3.2 is optimal up to a constant factor:

THEOREM 3.2. *For any protocol with total randomness at most $b < \log n - 1$, there is an input $(s, \mathcal{L})$, such that the rumor cannot be distributed to all nodes in fewer than $b + \lfloor n/2^{b+1} \rfloor$ rounds. In particular, any protocol with randomness $\log n - \log\log n - \omega(1)$ needs at least $\omega(\log n)$ rounds to broadcast the rumor.*

*Proof.* Suppose a randomized protocol $P$ has randomness $b$. Consider the $B = 2^b$ deterministic protocols $P_1, \ldots, P_B$ obtained by fixing the random string to all possible $B$ values. We can fix the lists of all $n$ nodes in such a way that the following holds for all $1 \leq i \leq B$ and $j \in \mathbb{N}$: In any of the protocols $P_1, \ldots, P_i$, each node sends its first $j$ messages to nodes in $\{1, \ldots, i \cdot j\}$.

Now the claim follows with exactly the same arguments as the ones from the proof of Theorem 3.1 for $i = 2^b$ and $j = r_2$. ∎

## 4 Conclusion

We provided a systematic study of the randomness requirements for efficient rumor spreading. We gave evidence that the broadcast time is at least $n^{1-o(1)}$ if all nodes act obliviously and use only $o(\log n)$ random bits each. However, a simple modification of the quasirandom model demonstrates that if nodes can communicate and thus share random bits, then only $O(\log\log n)$ bits on average per node are sufficient. We also presented an asymptotically tight upper bound of $O(\log n)$ for the total number of random bits that are required to spread the rumor in $O(\log n)$ rounds. An important open problem is to find an explicit or even practical protocol that has such low randomness requirements.

Our explicit protocol has the desired properties of being simple and local. However, it does not seem to be as robust as the fully random or the quasirandom protocol. In particular it is important that nodes know when to switch from the first to the second phase: If too many messages get lost, then that switch could occur too early, and not enough nodes get informed in the first phase (resulting in a lack of seed supply in the second phase). It seems that this problem can be fixed, though, in a modified model such as the one proposed in [8], where nodes receive feedback whether their messages have reached their targets or not. In the case of a message loss, a node could then simply repeat sending its message without decrementing the "round counter". This would ensure that even if an arbitrary (but bounded) number of messages get lost, enough seeds would still show up in the system. We leave it to future research to investigate this issue thoroughly.

**Acknowledgement**

We thank Pierre Fraigniaud for pointing us to some of the questions answered in this contribution, and for helpful discussions in early stages of this work.

## References

[1] S. Angelopoulos, B. Doerr, A. Huber, and K. Panagiotou. Tight bounds for quasi-random rumor spreading. *The Electronic Journal of Combinatorics*, 16(1), 2009.

[2] P. Berenbrink, R. Elsässer, and T. Friedetzky. Efficient randomised broadcasting in random regular networks with applications in peer-to-peer systems. In *Proceedings of the 27th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 155–164, 2008.

[3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 1–12, 1987.

[4] B. Doerr and M. Fouz. A time-randomness tradeoff for quasi-random rumour spreading. *Electronic Notes in Discrete Mathematics*, 34:335–339, 2009.

[5] B. Doerr and M. Fouz. Quasi-random rumor spreading: Reducing randomness can be costly. *CoRR*, abs/1008.0501, 2010.

[6] B. Doerr, T. Friedrich, M. Künnemann, and T. Sauerwald. Quasirandom rumor spreading: An experimental analysis. In *Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 145–153, 2009.

[7] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 773–781, 2008.

[8] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading: Expanders, push vs. pull, and robustness. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 366–377, 2009.

[9] B. Doerr, A. Huber, and A. Levavi. Strong robustness of randomized rumor spreading protocols. *CoRR*, abs/1001.3056, 2010.

[10] R. Elsässer. On the communication complexity of randomized broadcasting in random-like graphs. In *Proceedings of the 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 148–157, 2006.

[11] R. Elsässer, U. Lorenz, and T. Sauerwald. On randomized broadcasting in star graphs. *Discrete Applied Mathematics*, 157(1):126–139, 2009.

[12] R. Elsässer and T. Sauerwald. Broadcasting vs. mixing and information dissemination on Cayley graphs. In *Proceedings of the 24th Symposium on Theoretical*

*Aspects of Computer Science (STACS)*, pages 163–174, 2007.

[13] R. Elsässer and T. Sauerwald. The power of memory in randomized broadcasting. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 218–227, 2008.

[14] R. Elsässer and T. Sauerwald. On the runtime and robustness of randomized broadcasting. *Theoretical Computer Science*, 410(36):3414–3427, 2009.

[15] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1(4):447–460, 1990.

[16] N. Fountoulakis and A. Huber. Quasirandom rumor spreading on the complete graph is as fast as randomized rumor spreading. *SIAM Journal on Discrete Mathematics*, 23(4):1964–1991, 2009.

[17] P. Fraigniaud and G. Giakkoupis. On the bit communication complexity of randomized rumor spreading. In *Proceedings of the 22nd ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 134–143, 2010.

[18] A. Frieze and G. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10:57–77, 1985.

[19] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 565–574, 2000.

[20] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[21] B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, 1987.

[22] T. Sauerwald. On mixing and edge expansion properties in randomized broadcasting. In *Proceedings of the 18th International Symposium on Algorithms and Computation (ISAAC)*, pages 196–207, 2007.