Fast and Robust Information Spreading

Keren Censor-Hillel
MIT
ckeren@csail.mit.edu*

George Giakkoupis
INRIA, France
george.giakkoupis@inria.fr

Abstract

Disseminating information in large networks of distributed systems is a fundamental problem. The classical randomized protocol, where in each round every node chooses a random neighbor to exchange information with, is an attractive solution for its simplicity, robustness to failures, and efficiency for many topologies. However, since the running time of this protocol depends on the expansion of the network, there are topologies for which it is very inefficient, requiring a number of rounds significantly larger than the diameter (polynomial in the number of nodes, for networks with constant diameter). Recently, a new generation of randomized protocols have been proposed [3, 4], which provide stronger runtime guarantees achieving a number of rounds close to the diameter. However, it seems that these protocols are less robust, even against modest network changes or failures.

In this paper we present a randomized information spreading algorithm that has runtime comparable to the newer algorithms, namely $O(D \cdot \text{polylog}(n))$ rounds for any network with n nodes and diameter D, and is provably robust for various random failure models, similarly to the classical randomized protocol. Our algorithm relies on solving the Neighbor Exchange Problem [3], where each node must learn the information stored in each of its neighbors. The algorithm is simple and natural, and its main innovation is that nodes choose the neighbor to contact in each round with probability that decreases with the number of messages they have received originating from that neighbor.

*Supported by the Simons Postdoctoral Fellows Program and NSF Award 0939370-CCF.

1 Introduction

A central task in large networks of distributed systems is the task of information spreading, where each node has a piece of information that must reach all other nodes of the network. Every node has to choose which neighbor to contact and exchange information with in each round, with the goal that information spreads to the whole network in as few rounds as possible. Since we assume that nodes do not know the topology of the network in advance, a natural solution is to choose randomly. The classical randomized algorithm [8, 23], where in each round every node chooses a neighbor uniformly and independently at random and the two nodes exchange the information they have, has been studied extensively and found various applications. Surprisingly, this simple local protocol has proven quite efficient for a wide range of topologies. An additional important property of the protocol is that it is naturally robust to various types of network failures.

The number of rounds required by the algorithm that uses uniform random choices depends on how well-connected the network is. More precisely, it depends on the expansion properties of the underlying network graph [5, 6, 20], rather than the diameter of the network, which is the natural lower bound for information spreading. In particular, there are topologies for which the algorithm is very inefficient, requiring a number of rounds significantly larger than the diameter. For example, in the dumbbell graph, where two cliques of $n/2$ nodes are connected with a single edge, a linear number of rounds in needed despite the network having constant diameter.

Recently, a new generation of randomized information spreading protocols have been proposed, which provide stronger runtime guarantees [3,4]. These algorithms achieve fast running times under weaker expansion properties [4], or even running times that are close to the diameter regardless of the expansion of the graph [3]. It seems, however, that these protocols do not demonstrate the robustness of the uniform randomized algorithm. To some degree, they all rely on constructing sparse subgraphs of the network and spreading information on those subgraphs. As a result, it seems that these protocols are sensitive even to modest network changes and failures.

In this paper, we present a simple and natural randomized algorithm whose running time is close to the diameter for all graphs, similarly to the protocols in [3], but is also provably robust for various random failure models, similarly to the uniform randomized algorithm.

Our contribution: We study a basic communication task, the *Neighbor Exchange Problem*, where the goal is for each node to obtain the information of all of its neighbors [3]. Repeating a neighbor exchange algorithm D times, where D is the diameter of the network, solves the problem of information spreading. Our main result is a neighbor exchange algorithm with the following runtime bound that holds for any graph.

Theorem 1. *For any constant $\beta \geq 1$, our algorithm completes neighbor exchange in $O(\log^9 n)$ rounds, with probability $1 - n^{-\beta}$.*

This implies information spreading in $O(D \cdot \log^9 n)$ rounds. Unlike the uniform randomized algorithm, our algorithm uses probabilities that are neither uniform nor fixed throughout the execution. Instead, they are adjusted according to a new simple framework that assigns a smaller probability to a neighbor whose message has been received many times. This gives priority to other neighbors, resulting in a faster running time. While our algorithm is simple and natural, its analysis is non-trivial.

In addition to being fast, our algorithm is robust to failures. Given some probability of independent edge failures for each round, our algorithm works without change, paying only an overhead in the runtime that depends on the failure probability q. Precisely, the running time increases by a factor of at most $O(1/(1-q))$, similarly to the uniform randomized algorithm.

Further, we consider permanent edge and node failures, where each surviving edge or node fails in a round independently with some (small) probability. Under this failure model, we require that a node acquires the messages of all its surviving neighbors that are still connected to it. We show that the performance of our algorithm is not affected by this model.

As will be elaborated in Section 4, other information spreading algorithms that have been shown to be fast [3, 4], can tolerate random failures in the first of the two models above with the use of standard error recovery mechanisms. It is not obvious how these protocols could be made fault-tolerant in the second model with permanent failures.

Related work: Randomized information spreading algorithms have been studied in many papers, starting with the work of Demers et al. [8] for replicated database maintenance.

The *push* version of the uniform randomized algorithm, where information is transmitted only from the node that initiates a connection, was studied first. The runtime of the protocol was analyzed precisely for the complete graph by Frieze and Grimmett [18] and Pittel [28], then for the hypercube by Feige et al. [14] and other similar graphs by Elsässer and Sauerwald [12], and for random graphs by Feige et al. and Fountoulakis et al. [14–16]. The robustness of this protocol was studied by Elsässer and Sauerwald [13].

The *push-pull* version of uniform randomized algorithm, where information is *exchanged* between the pair of communicating nodes similarly to our model, was first studied by Karp et al. [23] for the complete graph. More recently, the protocol was analyzed for preferential attachment graphs by Chierichietti et al. and Doerr et al. [5, 9], and for other graphs modeling social networks by Fountoulakis et al [17].

Additional work considered randomized information spreading algorithms on other families of graphs. Kempe et al. [24, 25] give distance-based bounds for nodes placed with uniform density in R^d. Bradonjić et al. [2] analyze information spreading in random geometric graphs, Georgiou et al. [19] study asynchronous networks, Sarwate and Dimakis [29] and Boyd et al. [1] study the problem in wireless sensor networks, and Pettarin et al. [27] consider sparse mobile networks.

For arbitrary communication graphs, as considered in our work, a sequence of papers bound the running time of the uniform randomized algorithm in terms of the expansion of the graph. Mosk-Aoyama and Shah [26] showed an $O(\log n/\phi)$ bound for regular graphs, where ϕ is the graph conductance. Chierichetti et al. [6, 7] proved a slightly weaker bound that holds for any graph, and [20] improved this bound to $O(\log n/\phi)$ for all graphs. Similar bounds have been shown in terms of the vertex expansion α of the graph [21, 30]. The runtime bound shown for general graphs is roughly $O(\log^2 n/\alpha)$.

Some papers have presented alternative algorithms that deviate from the paradigm of independent random choices. In an influential work [10, 11], Doerr et al. introduced the quasirandom model, in which each node randomly chooses a starting position from its list of neighbors from which it sequentially accesses the list. This approach reduces the amount of randomness needed for the algorithm, without hurting the running time for many graph topologies. Further reduction on randomness can be achieved using hashing [22].

Comparison with Recent Results: In [4], an information spreading algorithm was given, with a runtime that depends on the *weak conductance* of the communication graph, a weaker notion of expansion that is never smaller than the conductance and is larger in many cases. While for many graphs this algorithm improves upon the randomized information spreading algorithm in terms of the number of rounds it requires, it is highly susceptible to failures. The reason for this is that the algorithm implicitly builds a sparse subgraph over which messages are sent. An edge that fails

may disconnect the subgraph, effectively partitioning the communication in the network despite its remaining connected through edges that are not included in the sparse subgraph.

A recent algorithm of [3], greatly improves the running time for information spreading by achieving $O(D + \text{polylog}\, n)$ rounds. However, this algorithm is also vulnerable to failures. Its main ingredient is a neighbor exchange algorithm that requires $O(\log^3 n)$ rounds to complete. Afterwards, it uses this as a subroutine to build an underlying sparse spanner in the graph and send messages along that spanner. As explained for the previous algorithm, if an edge fails then it could disconnect the spanner and prevent information from reaching all nodes, even if the original graph is still connected. Another algorithm can be built from the neighbor exchange procedure by simply repeating it D times, thus obtaining a runtime of $O(D \cdot \log^3 n)$ rounds. However, the neighbor exchange procedure itself cannot tolerate failures, for the following reason. The procedure heavily relies on every pair of nodes sharing knowledge about receiving each other's message. Formally, it uses uniform random choices for the neighbors being contacted, and after every $O(\log^2 n)$ rounds it discards all edges between neighbors that have each other's message. It argues that at least half of the edges are being discarded after each such iteration, requiring $O(\log^3 n)$ rounds in total. To guarantee that if a node u gets the message of a node v then v gets the message of u, a deterministic reversal routine is used, in which at the end of the iteration all steps are repeated in reverse order. This is the Achilles' heel of the algorithm in terms of robustness. If edges can fail independently with probability $1/n^\epsilon$, for $0 < \epsilon < 1$, then there is high probability that some of these reversed paths are broken, implying that the symmetry is not preserved.

2 Neighbor Exchange Algorithm

We denote by m_u the message of u that must be disseminated to all its neighbors. Our algorithm runs in phases with $r = c \cdot \log^3(n)$ rounds in each phase, for some constant c. Each copy of m_u that u sends during a phase t has timestamp t. For every neighbor $v \in N(u)$, node u has a counter $C_{u,v}$ that counts the phases t during which it receives some copy of m_v with timestamp t (i.e., the copy was sent by v during the current phase). Formally, let $C_{u,v}(t)$ denote the value of the counter after phase t. Then, $C_{u,v}(t) = C_{u,v}(t-1) + 1$ if u receives a copy of m_v with timestamp t during phase t, and $C_{u,v}(t) = C_{u,v}(t-1)$ otherwise, with $C_{u,v}(0) = 0$.

For every $t \geq 0$, we define for each node u a total order over its counters $C_{u,v}(t)$ by increasing order, where ties are broken arbitrarily, say by the node IDs. We define $R_{u,v}(t)$ to be the *rank* of v at u after phase t, that is, the place of $C_{u,v}(t)$ in the total order. In every round of phase $t \geq 1$, node u chooses to contact neighbor v with probability $p_{u,v}(t) = 1/(R_{u,v}(t-1) \cdot h_{|N(u)|})$, where h_d is the normalizing constant $\sum_{i=1}^{d} 1/i$, which yields $\sum_{v \in N(u)} p_{u,v}(t) = 1$. We have $\ln(d+1) < h_d \leq \ln(d) + 1$. When a node contacts some other node in a round, the two nodes exchange all the information they had at the beginning of that round. If a node receives multiple copies of a message m_u, it keeps only the copy with the most recent timestamp.

The following theorem states that our algorithm is fast, and is our main result. Section 3 is dedicated to its proof.

Theorem 1. *For any constant $\beta \geq 1$, our algorithm completes neighbor exchange in $O(\log^9 n)$ rounds, with probability $1 - n^{-\beta}$.*

3 Analysis

Before we present the formal proofs, we give an overview of the analysis. By definition of the neighbor exchange problem, the goal of our algorithm is to reach a time where all the counters are positive, implying that every node has received at least one copy of the message of each of its neighbors. To show this, we prove two properties that hold with high probability. The first is that the counters $C_{u,v}$ and $C_{v,u}$ at two neighboring nodes u and v do not differ by more than $\rho = \Theta(\log^3 n)$, unless both counters are greater than $\ell = \Theta(\log^5 n)$ (Corollary 4 of Lemma 3). The second property is that for every pair of neighbors at least one of the counters is larger than ρ after $O(\ell \cdot \log n) = O(\log^6 n)$ phases.

For the second argument, we examine the counters after each phase t by defining a threshold T_t and bounding the number of counters that are below this threshold (Lemma 6). This threshold is initially $T_0 = \ell$, and it decreases by $\delta = \Theta(\ell/\log n)$ whenever a constant fraction of the counters that are below T_t are also above $T_t - \delta$. (The constants in the definition of δ ensure that T_t cannot decrease below ρ.) To compute the rate at which the number of counters below the threshold decreases (either because counters exceed T_t or because they exceed $T_t - \delta$, causing T_t to drop), we argue that a constant fraction of the counters that are below the threshold at the end of a phase increase in the next phase (Lemma 7). It follows that in $O(T_t) = O(\ell)$ phases the number of counters that are below the threshold decreases by a constant fraction (proving Lemma 6).

To prove that a constant fraction of the counters that are below the threshold increase in a phase (Lemma 7), we study the spread of messages along a subgraph of the original graph. We define a virtual coloring procedure that we call Red-Blue Coloring, which colors edges in red while initially all edges are blue. We show that *uniform* random spreading in the blue subgraph (with some imposed failure probability) is dominated by the real algorithm, i.e., the probability for choosing a blue edge is not larger than the probability of choosing it in the real algorithm (Lemma 8). Then, by analyzing the uniform random spreading process, we show that at least a constant fraction of the blue edges have a counter that increases in the next phase (Lemma 9). To prove that, we use a graph decomposition theorem into components of large conductance from [3], and the conductance-based bound of [20]. Last, using a potential argument we show that the counters that are below $T_t - \delta$ in our analysis, and thus most of the counters that are below T_t, are indeed blue in the virtual procedure (Lemma 10). The last two results give that a constant fraction of edges whose counters are below T_t increase in the next phase (completing the proof of Lemma 7).

We are now ready for the formal analysis. As explained, the first key ingredient in our analysis is showing that the counters of two neighbors do not differ by much, with high probability. We will use the following symmetry lemma, the proof of which is the same as that of [6, Lemma 3].

Lemma 2. *The probability that in phase t node u receives a copy of m_v with timestamp t is the same as the probability that v receives a copy of m_u with timestamp t.*

The intuition is that the probability of a message traveling along a path is the same as the probability of traveling along the reverse path. We remark that although the two events in Lemma 2 have the same probability, they are not independent. Also, these probabilities are not the same for different phases t.

The next lemma bounds the difference between the counters of two neighbors at all phases t until the sum of the two counters increases above some threshold λ. Using Lemma 2 it is easy to see that this difference is a martingale, and a direct application of Azuma's Inequality yields a bound on this difference in terms of the number of phases. The next lemma gives a more refined bound, in terms of the values of the counters rather than the number of phases.

Lemma 3. *For every two neighbors $u, v \in V$ and any $\alpha, \lambda > 0$,*

$$\Pr\left[\forall\, t\ \left(C_{u,v}(t) + C_{v,u}(t) \le \lambda\ \rightarrow\ |C_{u,v}(t) - C_{v,u}(t)| \le \alpha\sqrt{\lambda}\right)\right] = 1 - 2\lambda e^{-\alpha^2/2}.$$

Proof. In every phase t, each counter either increases or remains the same. Further, from Lemma 2, the probability that $C_{u,v}$ increases is the same as the probability that $C_{v,u}$ increases. For some ℓ, and for $k = 1, \ldots, \ell$, let t_k be the k-th phase t in which exactly one of the two counters increases, i.e., $C_{u,v}(t) + C_{v,u}(t) = C_{u,v}(t-1) + C_{v,u}(t-1) + 1$. Two observations are in place: First, in phase t_k, each of the two counters $C_{u,v}(t_k)$ and $C_{v,u}(t_k)$ increases with the same probability, $1/2$, thus the sequence $\{X_k\}$ of the differences $X_k = C_{u,v}(t_k) - C_{v,u}(t_k)$ is a random walk on the integer line. By a simple Chernoff bound, it follows that for any k,

$$\Pr[|X_k| > \alpha\sqrt{\lambda}] \le 2e^{-\alpha^2\lambda/(2k)}. \tag{1}$$

The second observation is that $C_{u,v}(t_k) + C_{v,u}(t_k) \ge k$, and thus $C_{u,v}(t) + C_{v,u}(t) \le \lambda$ implies $t \le t_\lambda$. Therefore, the event described in the statement of the lemma supersedes the event: $\forall\, t \in [1..t_\lambda]$ $(|C_{u,v}(t) - C_{v,u}(t)| \le \alpha\sqrt{\lambda})$, which is equivalent to the event: $\forall\, k \in [1..\lambda]$ $(|X_k| \le \alpha\sqrt{\lambda})$. From (1) and the union bound, it follows that the last event holds with probability at least $1 - 2\lambda e^{-\alpha^2/2}$. □

Define ℓ, δ, ρ such that

$$\ell = 3\delta \cdot \log_{4/3} n \qquad \delta = 2\rho \cdot \log_3(3n) \qquad \rho = \sqrt{2 \cdot (\beta+3) \cdot 3\ell \cdot \ln n}.$$

As will be seen later, the number of phases the algorithm requires is $O(\ell \cdot \log n)$, which for a constant β is in the order of $\log^6 n$, implying $O(\log^9 n)$ rounds. The next results follows from Lemma 3. Define the event

$$\mathcal{H}\colon\ \forall\,\{u,v\}\,\forall\, t\,\big(\min\{C_{u,v}(t),\, C_{v,u}(t)\} \le \ell\ \rightarrow\ |C_{u,v}(t) - C_{v,u}(t)| \le \rho\big).$$

Corollary 4. $\Pr[\mathcal{H}] = 1 - o(n^{-\beta})$.

Proof. By applying Lemma 3 for $\lambda = 3\ell$ and $\alpha = \rho/\sqrt{3\ell} = \sqrt{2 \cdot (\beta+3) \cdot \ln n}$, we obtain a lower bound of $1 - 6\ell \cdot n^{-\beta-3}$ on the probability of the event that $|C_{u,v}(t) - C_{v,u}(t)| \le \rho$ for all t for which $C_{u,v}(t) + C_{v,u}(t) \le 3\ell$. This event implies also that $|C_{u,v}(t) - C_{v,u}(t)| \le \rho$ for all t for which $\min\{C_{u,v}(t),\, C_{v,u}(t)\} \le \ell$. This is because otherwise, for the first t for which it does not hold we have that $\max\{C_{u,v}(t),\, C_{v,u}(t)\} = \min\{C_{u,v}(t),\, C_{v,u}(t)\} + \rho + 1$, implying that $C_{u,v}(t) + C_{v,u}(t) = 2\min\{C_{u,v}(t),\, C_{v,u}(t)\} + \rho + 1 \le 2\ell + \rho + 1 < 3\ell$. But if the first event holds then this implies that $|C_{u,v}(t) - C_{v,u}(t)| \le \rho$, which is a contradiction.

Applying now the union bound over all the at most n^2 edges $\{u, v\}$, yields the claim. □

We next consider the values of the counters among all pairs of nodes. Let $E(t, i)$ denote the set of edges $\{u, v\}$ for which $\max\{C_{u,v}(t),\, C_{v,u}(t)\} \le i$. Further, let $M(t, i) = |E(t, i)|$ denote the number of those edges.

We consider the sequence $T_0, T_1, \ldots$ of random variables, where $T_0 = \ell$, and for $t \ge 1$, T_t is defined as follows: Initially let $T \leftarrow T_{i-1}$, and while $M(t, T - \delta) < (3/4) \cdot M(t, T)$ set $T \leftarrow T - \delta$. Then T_t is equal to the final value of T.

Define $E_t = E(t, T_t)$ and $M_t = M(t, T_t) = |E_t|$. Since $M_0 \le n^2$, and M_t decreases by a factor of at least $1/4$ each time T_t decreases by δ, and also T_t does not decrease further after $M_t = 0$, it follows that for all t, we have $T_t \ge \ell - \delta \cdot \log_{4/3} n^2 = \delta \cdot \log_{4/3} n > \rho$.

Observation 5. *For all t, $T_t \ge \rho$.*

The next lemma bounds the rate at which M_t decreases. Its proof, which is described later, constitutes the largest part of our analysis.

Lemma 6. *For $\tau = 4 \cdot (\ell + 1)$, $\mathbf{E}[M_{t+\tau}] \leq (4/5) \cdot \mathbf{E}[M_t]$.*

Combining this lemma with the results presented earlier, we can easily derive Theorem 1. Roughly speaking, once M_t reaches zero, at least one endpoint of each pair of neighbors has the message of the other. Moreover, its corresponding counter is above the threshold, implying that if all the differences between pairs of counters at neighbors are bounded then all neighbors have each other's messages. Lemma 6 shows that M_t decreases fast enough, implying the stated runtime.

Proof of Theorem 1. Applying Lemma 6 repeatedly yields $\mathbf{E}[M_t] \leq (4/5)^{\lfloor t/\tau \rfloor} \cdot M_0$, and thus for

$$t^* = (1 + \beta) \cdot (1 + \log_{5/4} n^2) \cdot \tau,$$

we get $\mathbf{E}[M_{t^*}] \leq n^{-2\beta}$. Markov's inequality then gives $\Pr[M_{t^*} \geq 1] \leq n^{-2\beta}$, and thus

$$\Pr[M_{t^*} = 0] \geq 1 - n^{-2\beta}, \tag{2}$$

i.e., with this probability, we have for very edge $\{u, v\}$ that $\max\{C_{u,v}(t^*), C_{v,u}(t^*)\} \geq T_{t^*} + 1$. Since Observation 5 yields $T_{t^*} \geq \rho$, the last inequality implies $\max\{C_{u,v}(t^*), C_{v,u}(t^*)\} \geq \rho + 1$. Now, if event $\mathcal{H}$ occurs as well, then $|C_{u,v}(t^*) - C_{v,u}(t^*)| \leq \rho$, and thus $\min\{C_{u,v}(t^*), C_{v,u}(t^*)\} \geq 1$. Therefore, if both $M_{t^*} = 0$ and event $\mathcal{H}$ occurs, all nodes have received at least one copy of the message of each of their neighbors by the end of phase $t^* = O(\tau \cdot \log n) = O(\ell \cdot \log n)$. By (2), Corollary 4, and the union bound, this happens with probability $1 - o(n^{-\beta})$. □

In the remainder of the analysis we describe the proof of Lemma 6. The idea is that in order to show that M_t decreases fast enough, we show that a constant fraction of the counters that are below the threshold at some phase increase in the next phase.

Lemma 7. *With probability $1 - 1/n$, at least $3/8$ of the counters $C_{u,v}$, $\{u, v\} \in E_t$, increase in phase $t + 1$.*

Before we describe the proof of Lemma 7, we show how we can use it to prove Lemma 6.

Proof of Lemma 6. For $0 \leq k < \tau$, let $\mathcal{E}_k$ be the event that at least $3/8$ of the counters $C_{u,v}$, $\{u, v\} \in E_{t+k}$, increase in phase $(t + k) + 1$. Further, let $\mathcal{E} = \bigcap_{0 \leq k < \tau} \mathcal{E}_k$. From Lemma 7, we have $\Pr[\mathcal{E}_k] \geq 1 - 1/n$, and by the union bound, $\Pr[\mathcal{E}] \geq 1 - \tau/n$. We will argue below that if event $\mathcal{E}$ occurs then we have $M_{t+\tau} \leq (3/4) \cdot M_t$. Thus, $M_{t+\tau}$ is bounded by $(3/4) \cdot M_t$ with probability $1 - \tau/n$, and since with the remaining probability it is bounded by M_t, we have

$$\mathbf{E}[M_{t+\tau}] \leq (3/4) \cdot \mathbf{E}[M_t] \cdot (1 - \tau/n) + \mathbf{E}[M_t] \cdot (\tau/n) \leq (4/5) \cdot \mathbf{E}[M_t],$$

for large enough n.

To complete the proof it remains to show that $\mathcal{E}$ implies $M_{t+\tau} \leq (3/4) \cdot M_t$. Suppose towards a contradiction that event $\mathcal{E}$ occurs but $M_{t+\tau} > (3/4) \cdot M_t$. We bound the sum $\sum_{\{u,v\} \in E_{t+\tau}} \big(C_{u,v}(t + \tau) + C_{v,u}(t + \tau)\big)$ of the counters for every edge in $E_{t+\tau}$ after phase $t + \tau$. This sum is bounded from below by the difference between the following two quantities:
(i) The sum of the increase in the counters for the edges in E_{t+k} during phase $(t + k) + 1$, for all $0 \leq k < \tau$. Because of events $\mathcal{E}_k$, this is at least $\sum_{0 \leq k < \tau} (3/8) \cdot 2M_{t+k} \geq \tau \cdot (3/4) \cdot M_{t+\tau}$; minus
(ii) The sum of the counters for each $\{u, v\} \in E_t - E_{t+\tau}$ after the first phase t' for which $\{u, v\} \notin E_{t'}$.

This sum is st most $2(M_t - M_{t+\tau}) \cdot (T_t + 1)$.
Thus,

$$\sum_{\{u,v\} \in E_{t+\tau}} \big(C_{u,v}(t+\tau) + C_{v,u}(t+\tau)\big) \geq \tau \cdot (3/4) \cdot M_{t+\tau} - 2(M_t - M_{t+\tau}) \cdot (T_t + 1).$$

Further, the sum on the left side is at most $2M_{t+\tau} \cdot T_{t+\tau} \leq 2M_{t+\tau} \cdot T_t$. Thus

$$2M_{t+\tau} \cdot T_t \geq \tau \cdot (3/4) \cdot M_{t+\tau} - 2(M_t - M_{t+\tau}) \cdot (T_t + 1).$$

Solving for τ and using the assumption that $M_{t+\tau} > (3/4) \cdot M_t$ yields

$$\tau \leq (4/3) \cdot 2(M_t/M_{t+\tau}) \cdot (T_t + 1) < (4/3) \cdot 2(4/3) \cdot (T_t + 1) < 4 \cdot (T_t + 1) \leq 4 \cdot (\ell + 1) \leq \tau,$$

which is a contradiction. □

To prove Lemma 7 we argue about how messages spread on a subgraph of the original graph. We define this subgraph by an edge coloring procedure, which assigns one of the two colors blue and red to each edge. We run this procedure after each phase, and then we study the spread of messages through the blue subgraph during the next phase. We emphasize that this is a virtual procedure that is used only for the analysis and not by the real algorithm.

The coloring after phase t is determined by the following procedure, based on the counter values $C_{u,v}(t)$ and threshold T_t.

Red-Blue Coloring: Initially the edges in E_t are colored blue, and the edges in $E - E_t$ red. Some of the blue edges may turn red later on. For each node u, we have a variable $L(u)$ which contains always the largest value $C_{u,v}(t)$ over all edge $\{u, v\}$ incident to u that are currently blue. We say that the *color-changing condition* holds for u if there is some $l \leq L(u)$ for which more than $3/4$ of the edges $\{u, v\}$ with $C_{u,v}(t) \in [l..L(u)]$ are red. If the color-changing condition holds for some u, then we choose the largest l for which it holds, and switch to red all the blue edges $\{u, v\}$ with $C_{u,v}(t) \in [l..L(u)]$. We repeat this step for all nodes u until the color-changing condition does not hold for any node.

The motivation for changing to red the color of some blue edges of node u in case there are many red edges with smaller counter values at u (and thus smaller ranks) is that we will later argue about the probabilities of node u contacting blue edges alone, and will need the ranks of blue edges to be bounded with proportion to the number of blue edges. This will be made clear later. The reason we may have to repeat this several times is that once an edge $\{u, v\}$ becomes red at node u, it may cause additional edges to turn red at its neighbor v.

Let G_t be the coloring of the graph after phase t. We say that node v is a blue (or red) neighbor of node u if edge $\{u, v\}$ is a blue (resp. red) edge in G_t. Let $B_t(u)$ be the total number of blue neighbors of u in G_t. First, we prove that for each node u, the largest rank of any blue neighbor of u is at most 4 times the number of these neighbors.

Lemma 8. *For every node u and every blue neighbor v of u in G_t, $R_{u,v}(t) \leq 4B_t(u)$.*

Proof. Let v be the blue neighbor of u in G_t with the largest rank $R_{u,v}(t)$. Then v has also the largest counter value $C_{u,v}(t) = L(u)$ among the blue neighbors of u. Assume towards a contradiction that $R_{u,v}(t) > 4B_t(u)$. Then there are more than $3B_t(u)$ red neighbors v' of u that have ranks that are smaller than $R_{u,v}(t)$, and thus have $C_{u,v'}(t) \leq C_{u,v}(t) = L(u)$. But in this case, the color-changing condition holds for u, and thus the Red-Blue Coloring procedure would have colored the edge $\{u, v\}$ red, which is a contradiction. □

Lemma 8 implies that the probability $1/(R_{u,v}(t) \cdot h_{|N(u)|})$ with which u contacts a blue neighbor v in each round of phase $t+1$ is at least $1/(4B_t(u) \cdot h_n)$. We can now show that at least half of the counters for blue edges increase in the next phase, with high probability (note that we have two counters for each blue edge, and we do not require both to increase in the following lemma).

Lemma 9. *With probability $1 - 1/n$, at least half of the counters $C_{u,v}$ for the blue edges $\{u, v\}$ of G_t increase in phase $t + 1$.*

Proof. We couple the algorithm with a process $\mathcal{R}$ in which a node u contacts each of its $B_t(u)$ blue neighbors with probability $1/(4B_t(u) \cdot h_n)$. With the remaining probability $q = 1 - 1/(4 \cdot h_n)$, node u does not contact any neighbor. As follows from Lemma 8, in the real algorithm, u contacts each blue neighbor with probability at least $1/(4B_t(u) \cdot h_n)$, thus we can couple the two processes such that for any two neighbors u and v, if u contacts v in a given round of $\mathcal{R}$ then the same happens in the real algorithm. Therefore, it suffices to prove the lemma using $\mathcal{R}$ instead of the real algorithm.

From the decomposition theorem of [3, Corollary 3.4] we get that at least half of the blue edges of G_t belong to well connected components of conductance $\Omega(1/\log n)$. We observe that the proof of the conductance-based bound of [20] yields an upper bound of $O\left(\frac{1}{1-q} \cdot (\log n)/\phi(S)\right)$ on the number of rounds of process $\mathcal{R}$ that suffice to spread a message from any node in a set $S \subseteq V$, to all other nodes in S with probability $1 - n^{\alpha}$, for any constant α. It follows that there is an edge set $E' \subseteq E$ containing at least half of the blue edges of G_t, such that with probability at least $1 - 1/n$, all pairs of nodes that correspond to these edges get each other's message (timestamped with the current phase number) after $O((4 \cdot h_n) \cdot \log^2 n) = O(\log^3 n)$ rounds. Therefore, for an appropriate constant c, during the $r = c \cdot \log^3 n$ rounds of phase $t + 1$ all pairs of nodes in E' get each other's message, and thus increase their counters, with probability $1 - 1/n$. □

The last piece we need for proving Lemma 7 is that at least $3/4$ of the edges in E_t are blue; we show this next. (Recall that at least $3/4$ of the edges in E_t belong to $E(t, T_t - \delta)$.) Define the event

$$\mathcal{H}_t\colon\ \forall \{u, v\} \left(\min\{C_{u,v}(t),\, C_{v,u}(t)\} \leq \ell \ \rightarrow\ |C_{u,v}(t) - C_{v,u}(t)| \leq \rho \right).$$

Note that the event $\mathcal{H}$ described before Corollary 4 is the same as $\bigcup_t \mathcal{H}_t$.

Lemma 10. *Conditioned on the event $\mathcal{H}_t$, all edges $\{u, v\} \in E(t, T_t - \delta)$ are blue in G_t.*

Proof Sketch. We introduce a potential function that assigns values to a subset of the red edges. Precisely, every red edge $\{u, v\}$ for which there is a blue edge $\{u, v'\}$ such that $C_{u,v}(t) \leq C_{u,v'}(t)$, is assigned a potential value of $\Phi_{u,v} = 3^{(T_t - C_{u,v}(t))/\rho}$. We bound the value of the total potential at the beginning of the Red-Blue Coloring procedure (when all edges in E_t are blue), and then we show that we cannot have a red edge in $E(t, T_t - \delta)$, unless at some point the potential increases above that initial value. The main claim of the proof is that the potential function does not ever increase, which proves the lemma. The full proof appears in the Appendix. □

We can now complete the proof of Lemma 7.

Lemma 7 (repeated). *With probability $1 - 1/n$, at least $3/8$ of the counters $C_{u,v}$, $\{u, v\} \in E_t$, increase in phase $t + 1$.*

Proof of Lemma 7. From the definition of T_t, at least $3/4$ of the edges in E_t are in $E(t, T_t - \delta)$. Thus, Lemma 10 gives that $3/4$ of the edges in E_t are blue in G_t, if event $\mathcal{H}_t$ occurs. Using the bound $\Pr[\mathcal{H}] = 1 - o(n^{-\beta})$ from Corollary 4, we obtain that $3/4$ of the edges in E_t are blue with probability $1 - o(n^{-\beta})$. Further, Lemma 9 says that $1/2$ of the counters for blue edges increase in phase $t+1$ with probability $1-1/n$. (All blue edges of G_t are in E_t.) Therefore, $3/8$ of the counters for edges in E_t increase in phase $t + 1$ with probability $1 - 1/n - (n^{-\beta})$, by the union bound. □

4 Robustness Against Failures

In this section we consider different failure models and show that our algorithm is robust, thus preserving this important property of the uniform randomized algorithm while significantly decreasing the required running time.

Random Transmission Faults: We assume that each edge (link) is faulty independently in each round with some fixed probability $0 < p < 0$. (Different edges can have different failure probabilities, in which case p is an upper bound on these probabilities.) If an edge is faulty in some round then all messages transmitted through that edge in that round are lost. Alternatively, we can assume that messages transmitted thought the same edge in the same round are lost with independent probabilities. More precisely, if in some round two nodes u and v try to exchange information with each other, then the two events that the information transmitted from u to v is lost and that the information from v to u is lost occur independently each with (the same) probability p.

Our algorithm is robust against this failure model. It does not attempt to repeat failed transmissions. In fact, it does not even need to detect failures.

Failures come into play in our analysis in two places. The first is the proof of Lemma 3, which bounds the discrepancy between two counters at the endpoints on an edge. This proof relies on the symmetry of Lemma 2, which states that the probability a message from u reaches v is the same as the probability that a message from v reaches u, in a failure-free phase. The proof of this result, as found, e.g., in [6, Lemma 3], carries through without changes to the failure model above. (Note that the independence of failures across rounds is critical for this result to hold. This independence will not hold for the model of permanent faults we will describe later.) Thus, Lemma 3 still holds.

The second place in our analysis where we must take failures into account is the proof of Lemma 9, which gives a lower bound on the number of counters that increase in a phase. In that proof we argue that the algorithm dominates a process $\mathcal{R}$ in which a node contacts each neighbor with equal probability, and with probability $q = 1-\Theta(1/\log n)$ it does not contact any neighbor. To account for message losses, we just need to change the above probability q to $1-(1-p)\cdot\Theta(1/\log n)$. This yields an increase by a factor of $1/(1-p)$ to the bound we obtain for the length r of a phase. Thus, to cope with message losses it suffices that phases are sufficiently long.[1] This gives:

Theorem 11. *Under the above faults model, for any constant $\beta \geq 1$, our algorithm completes neighbor exchange in $O(\frac{1}{1-p} \cdot \log^9 n)$ rounds, with probability $1 - n^{-\beta}$.*

The algorithm in [4] cannot tolerate asymmetric failures, i.e., that the information sent from node u to node v in some round gets lost, but the information sent from v to u in the same round is transmitted successfully. Thus for the second version of the model we presented above, the algorithm in [4] needs a failure detection mechanism that informs nodes of unsuccessful transmissions.

The Neighbor Exchange algorithm in [3] needs stronger guarantees. In particular, in the reversal phase all messages must be delivered successfully. If a bound on p is known then this can be achieved by repeating each transmission for $\Theta((\log n)/(1-p))$ times, which results in all messages being delivered with high probability.

In contrast, our algorithm has the advantage that it does not require error recovery mechanisms.

[1] In the Appendix we describe a variant of the protocol that does not use phases, and thus eliminates the issue of choosing the phase length.

Random Permanent Faults: We assume that nodes and/or edges are subject to permanent crash faults. In each round, each node (or edge) that has not failed yet fails with probability bounded by $p = 1/n^\epsilon$, for some constant $0 < \epsilon < 1$. We do not require that failures in the same round occur independently. We chose the failure probability p to be small enough that not all nodes (or edges) fail in polylog(n) rounds, and large enough that many failures occur in a round. In this model, it only makes sense to require that each node acquires the messages of all of its non-faulty neighbors that are connected to it via non-faulty edges.

As before, our algorithm tolerates faults in this model without changes. Instead of repeating the majority of the proof, we only sketch the small modifications needed for the analysis to work in this model. Lemma 2 no longer holds. Instead we have that the two probabilities differ by a *factor* of at most $1 + O(p \cdot r^2) = 1 + O((\log^3 n)/n^\epsilon)$. The reason is that, given a failure-free run of a phase in which a message from node u reaches node v via some path (of length at most r), the probability that some node or edge along this path fails during the phase in our model is bounded by $p \cdot r^2$. Since this difference is very small, it can be easily shown that Lemma 3 continues to hold for any $\lambda = \text{polylog}(n)$.

In the proof of Lemma 9, we show that for each edge $\{u, v\}$ in a set E' containing half of the blue edges, nodes u and v receive each other's message w.h.p. (in a failure-free phase). By applying the same argument as above, we can guarantee for our failure model that node u receives v's message with probability at least $1 - p \cdot r^2$. Although this is too weak to yield a high probability bound, it gives that with probability $1/2$ at least half of the counters of the edges in E' increase in the phase. Thus, Lemma 9 still holds if we reduce the probability to $1/2$ and the fraction from half to a quarter. Similarly, Lemma 7 now holds with probability $1/2$ instead of $1 - 1/n$ and for $3/16$ of the counters. Since the goal of Lemma 7 is to facilitate the proof of Lemma 6, which is an expectation result, this weaker version of Lemma 7 suffices for that goal. We just need to increase τ by at most a constant factor to obtain the same decrease in $\mathbf{E}[M_t]$ as before. Thus we have:

Theorem 12. *Under the above faults model, for any constant $\beta \geq 1$, our algorithm completes neighbor exchange in $O(\log^9 n)$ rounds, with probability $1 - n^{-\beta}$.*

It is not clear how the algorithms in [3, 4] can be made fault-tolerant against this model. The problem is that they rely on spreading information through a sparse subgraph, which can become disconnected as several of the nodes (or edges) of this subgraph fail permanently in each round.

5 Discussion

We have shown a randomized information spreading algorithm that is based on a neighbor exchange algorithm which requires $O(\log^9 n)$ rounds to complete and is robust against various random failure models. This highly improves upon the running time for the uniform randomized algorithm, while maintaining high robustness.

The messages sent in our algorithm contain only the addition of timestamps to the initial information that has to be disseminated. Since our neighbor exchange algorithm requires only a polylogarithmic number of rounds to complete, this addition to the message size is negligible and thus comparable to existing algorithms.

An open question arises when not all nodes have information to disseminate. In this case, a node u without initial information still has to create a messages m_u to be disseminated, since our algorithm relies on receiving these messages in order to rank the neighbors and obtain the probabilities for contacting them. We ask whether a better approach can be taken in such a case.

References

[1] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508–2530, June 2006.

[2] Milan Bradonjić, Robert Elsässer, Tobias Friedrich, Thomas Sauerwald, and Alexandre Stauffer. Efficient broadcast on random geometric graphs. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1412–1421, 2010.

[3] Keren Censor-Hillel, Bernhard Haeupler, Jonathan A. Kelner, and Petar Maymounkov. Global computation in a poorly connected world: Fast rumor spreading with no dependence on conductance. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 961–970, 2012.

[4] Keren Censor-Hillel and Hadas Shachnai. Fast information spreading in graphs with large weak conductance. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 440–448, 2011.

[5] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 375–386, 2009.

[6] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Almost tight bounds for rumour spreading with conductance. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 399–408, 2010.

[7] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumour spreading and graph conductance. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1657–1663, 2010.

[8] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 1–12, 1987.

[9] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Social networks spread rumors in sublogarithmic time. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 21–30, 2011.

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

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

[12] Robert Elsässer and Thomas Sauerwald. Broadcasting vs. mixing and information dissemination on Cayley graphs. In *Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 163–174, 2007.

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

[14] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1(4):447–460, 1990.

[15] Nikolaos Fountoulakis, Anna Huber, and Konstantinos Panagiotou. Reliable broadcasting in random networks and the effect of density. In *Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM)*, pages 2552–2560, 2010.

[16] Nikolaos Fountoulakis and Konstantinos Panagiotou. Rumor spreading on random regular graphs and expanders. In *Proceedings of the 14th International Workshop on Randomization and Computation (RANDOM)*, pages 560–573, 2010.

[17] Nikolaos Fountoulakis, Konstantinos Panagiotou, and Thomas Sauerwald. Ultra-fast rumor spreading in social networks. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1642–1660, 2012.

[18] Alan Frieze and Geoffrey Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Appl. Math.*, 10:57–77, 1985.

[19] Chryssis Georgiou, Seth Gilbert, Rachid Guerraoui, and Dariusz R. Kowalski. On the complexity of asynchronous gossip. In *Proceedings of the 27th SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 135–144, 2008.

[20] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 57–68, 2011.

[21] George Giakkoupis and Thomas Sauerwald. Rumor spreading and vertex expansion. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1623–1641, 2012.

[22] George Giakkoupis, Thomas Sauerwald, He Sun, and Philipp Woelfel. Low randomness rumor spreading via hashing. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 314–425, 2012.

[23] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 565, 2000.

[24] David Kempe, Jon Kleinberg, and Alan Demers. Spatial gossip and resource location protocols. In *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC)*, pages 163–172, 2001.

[25] David Kempe and Jon M. Kleinberg. Protocols and impossibility results for gossip-based communication mechanisms. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 471–480, 2002.

[26] Damon Mosk-Aoyama and Devavrat Shah. Computing separable functions via gossip. In *Proceedings of the 25th SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 113–122, 2006.

[27] Alberto Pettarin, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. Tight bounds on information dissemination in sparse mobile networks. In *Proceedings of the 30th SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 355–362, 2011.

[28] Boris Pittel. On spreading a rumor. *SIAM J. Appl. Math.*, 47(1):213–223, 1987.

[29] Anand D. Sarwate and Alexandros G. Dimakis. The impact of mobility on gossip algorithms. *IEEE Transactions on Information Theory*, 58(3):1731–1742, 2012.

[30] Thomas Sauerwald and Alexandre Stauffer. Rumor spreading and vertex expansion on regular graphs. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 462–475, 2011.

APPENDIX

A Proof of Lemma 10

Lemma 10 (repeated). *Conditioned on the event $\mathcal{H}_t$, all edges $\{u,v\} \in E(t, T_t - \delta)$ are blue in G_t.*

Proof. Conditioning on the event $\mathcal{H}_t$ ensures that

$$|C_{u,v}(t) - C_{v,u}(t)| \leq \rho, \quad \text{for all } \{u,v\} \in E_t.$$

We consider all edges as unidirectional, hence we have directions (u,v) and (v,u) for each edge. We introduce a potential function Φ that assigns a value to every unidirectional red edge (u,v) for which $C_{u,v}(t) \leq L(u)$. Observe that for the opposite direction (v,u) of such an edge we have $C_{v,u}(t) > L(v)$, for otherwise the edge would not have been colored red (recall that the Red-Blue Coloring procedure colors both directions of an edge when the color-changing condition holds for at least one of its endpoints). The potential assigned to each (u,v) is

$$\Phi_{u,v} = \begin{cases} 3^{(T_t - C_{u,v}(t))/\rho}, & \text{if } (u,v) \text{ is red and } C_{u,v}(t) \leq L(u); \\ 0, & \text{otherwise.} \end{cases}$$

The total potential is $\Phi = \sum_{(u,v)} \Phi_{u,v}$.

Initially, when all edges in E_t are blue, the total potential is

$$\Phi_{\text{init}} \leq 3n^2,$$

because n^2 is a bound on the number of red unidirectional edges with non-zero potential, and 3 is a bound on their individual potential, because for each such edge (u,v) we have $T_t - C_{u,v}(t) < C_{v,u}(t) - C_{u,v}(t) \leq \rho$. We will show that Φ never increases, and thus $\Phi \leq \Phi_{\text{init}}$. From this, the lemma follows easily: Suppose for contradiction that there is some red edge $\{u,v\} \in E(t, T_t - \delta)$. Consider the step in the Red-Blue Coloring procedure when this edge becomes red, and suppose without loss of generality that the step happens because the color-changing condition holds for node u. Thus, right before that step, there is some red edge $\{u,v'\}$ with $C_{u,v'}(t) \leq C_{u,v}(t) \leq L(u)$. Hence, at that time, (u,v') has potential

$$\Phi_{u,v'} \geq 3^{(T_t - C_{u,v}(t))/\rho} \geq 3^{(T_t - (T_t - \delta))/\rho} = 3^{\delta/\rho} = 3^{2\log_3(3n)} > 3n^2 \geq \Phi_{\text{init}},$$

which is a contradiction.

It remains to show that Φ never increases. We consider the change in potential after each coloring of edges red. Assume a step in the Red-Blue Coloring procedure which happens because the color-changing condition holds for node u. Suppose that before the step we have $L(u) = l_{\text{high}}$ and let l_{low} be the largest l for which the color-changing condition holds for u. Let R (and B) be the set of red (resp. blue) unidirectional edges (u,v) for which $C_{u,v}(t) \in [l_{\text{low}}..l_{\text{high}}]$ before the step. We now argue that the total potential *decrease* in this step is at least

$$\sum_{(u,v) \in R} 3^{(T_t - C_{u,v}(t))/\rho} - \sum_{(u,v) \in B} 3^{(T_t - C_{v,u}(t))/\rho}. \tag{3}$$

Note that in the second sum we have $C_{v,u}(t)$ in the exponent, rather than $C_{u,v}(t)$. The first sum accounts for the potential before the step, of the red unidirectional edges (u,v) for which

$C_{u,v}(t) \leq l_{\text{high}} = L(u)$ before the step and $C_{u,v}(t) \geq l_{\text{low}} > L(u)$ after; this potential is 'lost' in this step. The second sum is an upper bound on the potential 'gain' due to the new red edges: the potential of each $(u, v) \in B$ is zero, as $C_{u,v}(t) \geq l_{\text{low}} > L(u)$ after the step, but the potential of the opposite unidirectional edges (v, u) may be non-zero.

Next we argue that we can map each unidirectional edge $(u, v) \in B$ to 3 distinct unidirectional edges $(u, v') \in R$ with $C_{u,v'}(t) \leq C_{u,v}(t)$ (each $(u, v') \in R$ is mapped to at most one $(u, v) \in B$). Consider all $(u, v) \in B$ sequentially in increasing counter values, and associate with each (u, v) the three edges $(u, v') \in R$ that are still free and have the smallest counter values (ties are broken arbitrarily). Since $|R| \geq 3|B|$, this mapping is valid. We show that it satisfies the requirement that $C_{u,v'}(t) \leq C_{u,v}(t)$ for the edges $(u, v') \in R$ associated with each $(u, v) \in B$: Suppose this is not true. Let $l \in [l_{\text{low}}..l_{\text{high}}]$ be the smallest counter value such that the above condition is violated for some $(u, v) \in B$ with $C_{u,v}(t) = l$. It follows that

$$|\{(u, v') \in R \colon C_{u,v'}(t) \leq l\}| < 3 \cdot |\{(u, v) \in B \colon C_{u,v}(t) \leq l\}|.$$

Combining this with $|R| \geq 3|B|$, we obtain that $l < l_{\text{high}}$ and

$$|\{(u, v') \in R \colon C_{u,v'}(t) > l\}| > 3 \cdot |\{(u, v) \in B \colon C_{u,v}(t) > l\}|.$$

This means that the color-changing condition holds for u for the counter value $l > l_{\text{low}}$, which contradicts the rule that the Red-Blue Coloring procedure chooses the largest possible l for which the color-changing condition holds.

We have just shown that for each $(u, v) \in B$ there are 3 distinct $(u, v') \in R$ with $C_{u,v'}(t) \leq C_{u,v}(t)$. It follows that

$$\sum_{(u,v)\in R} 3^{(T_t - C_{u,v}(t))/\rho} \geq 3 \cdot \sum_{(u,v)\in B} 3^{(T_t - C_{u,v}(t))/\rho} \geq 3 \cdot \sum_{(u,v)\in B} 3^{(T_t - (C_{v,u}(t) - \rho))/\rho} = \sum_{(u,v)\in B} 3^{(T_t - C_{v,u}(t))/\rho},$$

where the second inequality holds because $|C_{u,v}(t) - C_{v,u}(t)| \leq \rho$. From the above relation and (3), we obtain the claim that Φ never increases. □

B A Version of the Algorithm with no Parameters

The algorithm we presented has one parameter, the length $r = c \log^3 n$ of each phase. Next we describe a variant of the algorithm that has no parameters at all. In this variant, we do not divided rounds into phases. The idea is that we interleave the execution of $\log n$ copies of the old algorithm, where the i-th copy has phases of length $r = 2^i$, but all executions use the same messages.

For each *round* t, the copies of $m(u)$ that u sends in that round have timestamp t. As before, if a node $v \neq u$ receives multiple copies of $m(u)$, it keeps the one with the largest timestamp. Each node u, has now $\log n$ counters for each neighbor v, denoted $C_{u,v}[0], \ldots, C_{u,v}[\log n - 1]$. Counter $C_{u,v}[i]$ is updated every 2^i rounds: it increases by one after each round $t = k \cdot 2^i$, for $k \geq 1$, if u has receives at least one copy of $m(v)$ with timestamp s in the range $(k-1) \cdot 2^i < s \leq k \cdot 2^i$; otherwise $C_{u,v}[i]$'s value does not change. The ranks used in round t to determine the neighbor selection probabilities are computed base on the counters $C_{u,v}[t \bmod \log n]$.

Our previous analysis carries through to the new version of the algorithm with just minor modifications. To facilitate the use of the notation we defined in terms of phases, we will assume that the new algorithm has phases of length one, that is, phase t is round t. Briefly, the concentration Lemma 3 holds now for each pair of counters $C_{u,v}[i]$ and $C_{v,i}[i]$, for $0 \leq i < \log n$. To obtain a result similar to Lemma 6, which bounds the shrinking of the set of edges, we use a lemma similar to

Lemma 7: We show that with high probability at least 3/8 of the counters $C_{u,v}[i]$, for $\{u, v\} \in E_t$, increase from round $k2^i$ to $(k+1)2^i + \log n$, for all i for which $2^i \geq c \log^3 n$. Now, however, we do not need to know a bound for c, as it is guaranteed that $2^i \geq c \log^3 n$ will hold for at least some i.