Fast Information Spreading in Graphs with Large Weak Conductance

Keren Censor-Hillel* Hadas Shachnai†

Abstract

Gathering data from nodes in a network is at the heart of many distributed applications, most notably, while performing a global task. We consider *information spreading* among n nodes of a network, where each node v has a message $m(v)$ which must be received by all other nodes. The time required for information spreading has been previously upper-bounded with an inverse relationship to the conductance of the underlying communication graph. This implies high running times for graphs with small conductance.

The main contribution of this paper is an information spreading algorithm which overcomes communication bottlenecks and thus achieves fast information spreading for a wide class of graphs, despite their small conductance. As a key tool in our study we use the recently defined concept of *weak conductance*, a generalization of classic graph conductance which measures how well-connected the components of a graph are. Our hybrid algorithm, which alternates between random and deterministic communication phases, exploits the connectivity within components by first applying *partial information spreading*, after which messages are sent across bottlenecks, thus spreading further throughout the network. This yields substantial improvements over the best known running times of algorithms for information spreading on any graph that has a large weak conductance, from polynomial to polylogarithmic number of rounds.

We demonstrate the power of fast information spreading in accomplishing global tasks on the *leader election* problem, which lies at the core of distributed computing. Our results yield an algorithm for leader election that has a scalable running time on graphs with large weak conductance, improving significantly upon previous results.

Keywords: Distributed computing, randomized algorithms, weak conductance, information spreading, leader election

*CSAIL, MIT. Email: ckeren@csail.mit.edu. Supported by the Simons Postdoctoral Fellows Program. Part of this work was done while the author was a Ph.D. student at the Department of Computer Science, Technion, Haifa 32000, Israel and supported in part by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities and by the *Israel Science Foundation* (grant number 953/06).

†Department of Computer Science, Technion, Haifa 32000, Israel. Email: *hadas@cs.technion.ac.il*

1 Introduction

Collecting data of all nodes in a network is required by many distributed applications which perform global tasks. The goal of an *information spreading* algorithm is to distribute the messages sent by each of n nodes in a network to all other nodes. We consider the synchronous push/pull model of communication along with the *transmitter gossip* constraint [19], where each node contacts in each round *one* neighbor to exchange information with (a node can be contacted by multiple neighbors).

Intuitively, the time required for achieving information spreading depends on the structure of the communication graph, or more precisely, on how well-connected it is. The notion of *conductance*, defined by Sinclair [25], gives a measure for the connectivity of a graph. Roughly speaking, the conductance of a graph G, denoted by $\Phi(G)$, is a value in $[0, 1]$: This value is large for graphs that are well-connected (e.g., cliques), and small for graphs that are not (i.e., graphs which have many communication bottlenecks). It has been shown that the time required for information spreading can be bounded from above based on the conductance of the underlying communication graph [3, 8, 9, 19]. In particular, Mosk-Aoyama and Shah [19] show that, for any $\delta \in (0, 1)$, information spreading can be achieved in $O(\frac{\log n+\log \delta^{-1}}{\Phi(G)})$ rounds with probability at least $1 - \delta$. This holds when each node randomly chooses a neighbor to contact in every round.

Some graphs have small conductance, implying that they are not well-connected, and therefore the above approach may require many rounds of communication for information spreading. This lead us to define *weak conductance*, $\Phi_c(G)$, of a graph G [7], which measures connectivity among *subsets* of nodes in the graph, whose sizes depend on the parameter $c \geq 1$. It was shown that a relaxed requirement of *partial information spreading*, where each node needs to receive only some of the messages, can be solved fast, with high probability, in graphs with large weak conductance, although they may have small conductance. As shown in [7], partial information spreading is a sufficient guarantee for some applications.

In this paper we return to the question of achieving *full information spreading*, where each node must receive every message. We present an algorithm that obtains full informa-

tion spreading on connected graphs, and runs fast with high probability on graphs with large weak conductance, independent of their conductance.[1] This widely expands the known family of graphs for which fast information spreading can be guaranteed, since the weak conductance of a graph is always lower bounded by its conductance and is significantly larger for many graphs.

More generally, for graphs with large weak conductance, our algorithm induces fast solutions for tasks which can be solved using full information spreading, such as leader election, achieving consensus and computation of aggregation functions.

It has been long known that the conductance itself is insufficient as a lower bound for information spreading. For example, Feige et al. [13] show that information spreading on the hypercube can be obtained in $O(\log n)$ rounds, despite its small conductance. Our results refine this observation, suggesting the notion of weak conductance as the correct measure for full information spreading.

1.1 Our Contribution. The main contribution of this paper is an algorithm which achieves fast information spreading, with high probability, for graphs with large weak conductance. Formally, for any $c > 1$ and some small $\delta \in (0, 1/(3c))$, our algorithm achieves full information spreading in $O(c(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)} + c))$ rounds with probability at least $1 - 3c\delta$. This yields substantial improvements in the best known running times of algorithms for information spreading, in particular, on graphs that have small conductance but large weak conductance, from polynomial to polylogarithmic number of rounds.[2]

Since the best known running times of algorithms for full information spreading inversely depend on the conductance, which may be small due to communication bottlenecks, a natural direction towards speeding up information spreading is to identify such bottlenecks and choose these links with higher probability, compared to other neighboring links. However, detecting bottlenecks does not seem easy. One approach for separating bottlenecks from other neighbors is to show that a node receives messages from nodes across a bottleneck only with small probability. This seems to reduce to finding lower bounds for information spreading, a direction which has not proved fruitful so far. Instead, we develop an algorithm that does not detect bottlenecks, nor does it formally define their underlying properties, which also appears to be a challenging task. Nonetheless, our algorithm successfully *copes* with bottlenecks and guarantees fast information spreading despite their presence throughout the network.

We propose a hybrid approach for choosing the neighbor to contact in a given round, which interleaves random choices and deterministic ones. As in the case of random choices, selecting neighbors only in a deterministic manner may require a number of rounds (at least) proportional to the degree of the node, which may be large.[3] Our approach combines random and deterministic techniques using a framework where each node carefully maintains a diminishing list of its neighbors to contact deterministically, and alternates between selections from this list and random choices from the set of all neighbors. The lists maintained by the nodes assure that the information spreads across bottlenecks. A main challenge overcome by our algorithm is the tradeoff imposed by managing the lists, namely, inducing a connected subgraph while having scalable sizes that allow contacting each of the neighbors in them within a small number of rounds.

This constitutes our second contribution: obtaining a connected scalable-degree subgraph in a distributed network of unbounded degrees. We believe that finding such subgraphs can be useful in other applications, e.g., in obtaining scalable-degree *spanners* [21, 22] – fundamental subgraphs that preserve distances between nodes up to some stretch.

We demonstrate the power of fast information spreading in accomplishing global tasks on the *leader election* problem, which lies at the core of distributed computing. Our results yield an algorithm for leader election that has a scalable running time on graphs with large weak conductance, improving significantly upon previous results.

1.2 Related Work. Information spreading algorithms have been extensively studied, starting with the work of Demers et al. [10] for replicated database maintenance. Additional research use information spreading for computation of global functions [17, 19].

Communication models vary in different studies. For example, Karp et al. [16] consider the *random phone-call* model, where in each round every node chooses a random node to communicate with, assuming the communication graph is complete. Our results hold for arbitrary communication graphs.

Additional recent work includes the work of Bradonjić et al. [6], who analyze information spreading in random geometric graphs, and the work of Georgiou et al. [15], studying information spreading in asynchronous networks. Sarwate and Dimakis [24] and Boyd et al. [5] study the problem in wireless sensor networks.

The *quasirandom* model for information spreading has been introduced by Doerr et al. [11], and studied subsequently in [12, 14], as an approach to reduce the amount of

[1] We consider an algorithm to be *fast* if it runs in a *scalable* number of rounds, i.e., in $O(\log n)$ or $O(\text{polylog}(n))$ rounds.

[2] Consider, for example, the class of graphs with conductance $O(1/n)$ but constant weak conductance.

[3] Indeed, this is due to the fact that a neighbor adjacent to a bottleneck link may be contacted last.

randomness. In this model each node has a (cyclic) list of its neighbors, in which it chooses randomly a starting position. It then accesses its list sequentially using this starting position. The paper [11] shows that this model behaves essentially the same as the randomized model.

1.3 Organization. The rest of the paper is organized as follows. We give an overview of the notions of conductance, weak conductance, and partial information spreading in Section 2. In Section 3, we present our algorithm for obtaining full information spreading. The analysis of the number of rounds required by our algorithm is given in Section 4. Finally, Section 5 presents the application of our results to the leader election problem, followed by a discussion in Section 6.

2 Preliminaries

The notion of graph conductance was introduced as a measure of how well-connected a graph is. For a given *cut* $(S, V \setminus S)$, define

$$\varphi(S,V) = \frac{\sum_{i\in S, j\in V\setminus S} P_{i,j}}{|S|}, \tag{2.1}$$

where P is the stochastic matrix associated with the communication of the nodes.

The conductance of the graph is then defined to be the minimal such value, taken over all cuts:

$$\Phi(G) = \min_{S\subseteq V, |S|\leq n/2} \varphi(S,V).$$

Notice that the conductance satisfies $0 \leq \Phi(G) \leq 1$, since for every $i \in S$ we have $\sum_{j\in V\setminus S} P_{i,j} \leq \sum_{j\in V} P_{i,j} = 1$.

We refer the reader to [7] for a proof that this definition, which slightly differs from the traditional definition of conductance [25], is equivalent for a symmetric stochastic matrix P; also, it is shown in [7] that such a symmetric matrix P can represent a model of random choices of neighbors that is different from our model (where node i chooses each neighbor with probability $1/d_i$), but this model is dominated by our model, namely, the analysis of the time required for fast information spreading in the symmetric model holds also in our model, which can only do better.

For some applications, *partial information spreading* suffices, namely, the condition that each node receives the information of all other nodes (to which we refer as *full information spreading*) can be relaxed to smaller amounts of information. Formally, for some values $c \geq 1$ and $\delta \in (0,1)$, we require that with probability at least $1-\delta$ every message reaches at least n/c nodes, and every node receives at least n/c messages. An algorithm that satisfies this requirement is called (δ, c)-*spreading*. Indeed, the special case where $c = 1$ corresponds to full information spreading.

When only a relaxed spreading guarantee is required, the concept of *weak conductance* can be used in order to analyze partial information spreading. While conductance provides a measure for the connectivity of the whole graph, weak conductance measures the *best* connectivity among subsets that include each node. Formally, for an integer $c \geq 1$, the weak conductance of a graph $G = (V, E)$ is defined as:

$$\Phi_c(G) = \min_{i\in V} \left\{ \max_{\substack{V_i\subseteq V,\\ i\in V_i,\\ |V_i|\geq \frac{n}{c}}} \left\{ \min_{\substack{S\subseteq V_i,\\ |S|\leq \frac{|V_i|}{2}}} \varphi(S,V_i) \right\} \right\},$$

where $\varphi(S,V)$ is defined in (2.1). Indeed, in the special case where $c = 1$, the weak conductance of G is equal to its conductance, namely, $\Phi_1(G) = \Phi(G)$. Moreover, this definition implies that the weak conductance of a graph is a monotonically increasing function of c; therefore, the weak conductance of a graph G is at least as large as its conductance.

The following theorem bounds the number of rounds required for (δ, c)-spreading.

THEOREM 2.1. ([7, THEOREM 3]) *For any $\delta \in (0,1)$, the number of rounds required for (δ, c)-spreading is $O\left(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)}\right)$.*

However, we emphasize that the proof of this theorem gives actually a much stronger result. For any vertex $v \in V$, let V_v be the component realizing the definition of the weak conductance. Then it is shown in [7] that the above is a bound on the number of rounds required for every node v to obtain the message $m(u)$ of every $u \in V_v$, and for every u to obtain $m(v)$. We refer the reader to the above paper for the proof, and state the stronger result, which we later use in our analysis.

THEOREM 2.2. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, the number of rounds required for every node v to obtain the message $m(u)$ of every $u \in V_v$, and for every $u \in V_v$ to obtain $m(v)$ is $O\left(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)}\right)$.*

Therefore, for all nodes $v \in V$, the sets V_v will be central to our analysis. These sets depend on the value of c, which is omitted to avoid excessive notation.

We proceed by giving examples of the conductance and weak conductance of different graphs. A *clique* has constant conductance. Its weak conductance is equal to its conductance, since for every node i the best subset V_i is V itself. The conductance of a *path* is $\frac{1}{n}$, while its weak conductance improves only to $\frac{c}{n}$. For these two examples, the weak conductance is in the same order as the conductance for some constant $c \geq 1$.

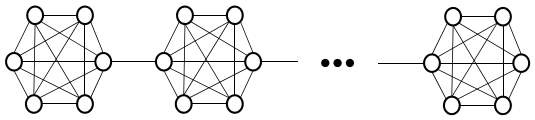

Figure 1: The c-barbell graph is a path of c equal-sized cliques. It is an example of a graph with small conductance and large weak conductance.

The *c-barbell* graph is an example of a graph with very small conductance (for which, to the best of our knowledge, it was not known up to this work how to achieve fast information spreading), but large weak conductance. The c-barbell graph, which is a generalization of the *barbell* graph, consists of a path of c cliques, where each contains n/c nodes (see Figure 1). The c-barbell graph is associated with the transition matrix P for which $P_{i,j} = 1/\left(\frac{n}{c}\right)$ for every two neighbors, $P_{i,i} = 1/\left(\frac{n}{c}\right)$ for every node i that does not connect two cliques, and $P_{i,i} = 0$ for every node i connecting two cliques. While the conductance of this graph is $\left(1/(\frac{n}{c})\right)/\frac{n}{2} = \frac{2c}{n^2}$, the weak conductance is $\left((\frac{n}{2c})(\frac{n}{c} - \frac{n}{2c})\frac{1}{n/c}\right)/\frac{n}{2c} = \frac{1}{2}$. For any constant $c \geq 1$, this implies conductance of $\Theta(1/n^2)$ while the weak conductance is $\Theta(1)$. Indeed, the barbell graph has been studied before [1, 4] as a graph for which information spreading requires a large number of rounds (in [1] the context is random walks, which is closely related, since the path of a message can be viewed as a random walk on the graph).

There are additional families of graphs that have a similar property of small conductance and large weak conductance. Examples include rings of cliques and other structures with c equal-sized well-connected components that are connected by only a few edges.

3 A Fast Information Spreading Algorithm

Our algorithm for obtaining *full information spreading* applies several phases of partial information spreading, interleaved with our deterministic spreading on a scalable subgraph. We emphasize that the only initial information a node has is the size of the network n and the set of its neighbors. No value of c is given to the nodes nor do they aim to obtain partial information spreading; c is only used for analysis. Moreover, the information spread contains no additional headers, only the information $m(v)$ of different nodes.

We consider a synchronous system with n nodes $V = \{v_1, v_2, \ldots, v_n\}$, represented by a graph $G = (V, E)$. In each round r, every node contacts one of its neighbors, as explained below, and exchanges information with it. For the analysis, we consider each round r as a sequence of n events of information exchange, ordered by the ID of the node that initiated the exchange (if both nodes choose each other we consider the node with the smaller ID as the initiator). We number these events by an *event time* t, such that $rn \leq t < (r+1)n$. However, these exchanges occur in parallel, which means that a node sends information it had by the end of the last round, without additional information it may have received in the current round.

For each node v, let $N(v)$ be the set of all neighbors of v. At every event time t, node v maintains a cyclic list $B_t(v)$ of suspected bottlenecks among its neighbors, where $B_0(v)$ is initialized to be $N(v)$, in an arbitrary order. In order to exchange information with its neighbors, each node v alternates between choosing a random neighbor from $N(v)$ and choosing the next neighbor from $B_t(v)$. During this procedure, v removes neighbors from $B_t(v)$ according to the following policy.

Neighbor Removal Policy for Node v: Let u be a node in $N(v)$. At event time t in which v exchanges information with another node, v removes any node u whose message is received for the first time, unless it is received from u itself and v is the initiator of this information exchange.

We emphasize that a node v which removes a node u from $B_t(v)$ can still contact node u if it happens to be its random choice in an even-numbered round r.

Each node v also maintains a buffer $M_t(v)$ of received messages, initialized to consist only of its own message $m(v)$. When v has all n messages it returns the buffer $M_t(v)$ as its output. The pseudocode for a node v appears in Algorithm 1.

Algorithm 1 Full information spreading code for node v.

Initially $M_0(v) = \{m(v)\}$, $B_0(v) = N(v)$, $r = 0$, $t = 0$
1: if r is even
2: $w =$ a random neighbor from $N(v)$
3: else
4: $w =$ the next neighbor from $B_t(v)$
5: Contact w and exchange information
6: Exchange information with every w' that contacts v
// last two lines by order of initiator ID
7: for $i = 1$ to n
8: if $v = v_i$ or v is contacted by $w' = v_i$
9: Add new messages to $M_t(v)$
10: for every node $u \in N(v)$
11: if v receives $m(u)$ for the first time from w and $w \neq u$
12: or v receives $m(u)$ for the first time from w'
13: $B_{t+1}(v) = B_t(v) \setminus \{u\}$
14: $t = t + 1$
15: if $|M_t(v)| = n$ then return $M_t(v)$
16: $r = r + 1$

At every event time t, for every node v we define a partition of $N(v)$ into three sets, as follows.

- $White_t(v) = B_0(v) \setminus B_t(v)$: The set of nodes that have been removed from $B_0(v)$,
- $Black_t(v) = \{u \in N(v) \mid m(u) \in M_t(v)$ and $u \notin White_t(v)\}$: The set of nodes at event time t guaranteed to never be removed from $B_0(v)$,
- $Grey_t(v) = N(v) \setminus (Black_t(v) \cup White_t(v))$: The rest of the nodes, which may or may not be removed in later event times.

We illustrate this partition on the directed graph $G_t = (V, E)$, which is the same as the communication graph, but with colors associated with each edge at event time t: if $u \in White_t(v)$ then (v, u) is colored white, if $u \in Black_t(v)$ then (v, u) is colored black, and otherwise (v, u) is colored grey.

We start the analysis with four simple claims regarding the colors of the edges of G_t. The formal statements appear in the next lemma, whose proof is given in Appendix A.

LEMMA 3.1. *The following four claims hold:*

(i) *In G_0 all edges are colored grey.*

(ii) *For any nodes v and u, if $m(u) \in M_t(v)$, and $u \in N(v)$, then (v, u) is not grey in G_t.*

(iii) *For any event time t, if for some nodes v and u both edges (v, u) and (u, v) change their color at t, then v and u are the pair of nodes that exchange information in this event time.*

(iv) *If (v, u) and (u, v) are both grey, they cannot turn both white at the same time step.*

We are now ready to prove a key lemma in our analysis, which shows that whenever a node v removes a neighbor u from $B_t(v)$, that is, the edge (v, u) is colored white, there is an undirected path of edges between v and u that are guaranteed to never be removed, i.e., a black path.

LEMMA 3.2. *For any event time $t \geq 0$, if (v, u) turns white, then there is a path $v = a_0, a_1, \ldots, a_{\ell-1}, a_\ell = u$ such that for all $0 \leq i \leq \ell - 1$, either (a_i, a_{i+1}) is black or (a_{i+1}, a_i) is black.*

Proof. The proof is by induction on the time step t, where the base case for $t = 0$ is the initial coloring of the graph. By Lemma 3.1 (i), at this time all the edges are colored grey, therefore the lemma holds vacuously. For the induction step, assume that the lemma is true for every edge that turns white in a step $t' < t$. We prove the lemma for any edge (v, u) that turns white at time t. If this happens, then one of the conditions in line 11 or line 12 of the algorithm is satisfied. We distinguish between two cases.

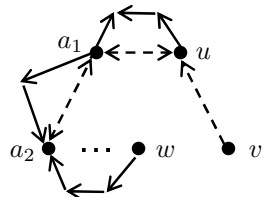

Figure 2: Proof of Lemma 3.2. Dashed arrows represent white edges, while solid arrows represent black edges. If v removes u from $B_t(v)$ by receiving $m(u)$ from w, then there is an undirected black path between u and w.

Case 1: Node v receives $m(u)$ for the first time from node $w \neq u$, and removes node u from $B_t(v)$. The fact that w has the message $m(u)$ at this time implies that $m(u)$ traveled a path $u = a_0, a_1, \ldots, a_{\ell'} = w$ by time $t - 1$. For all $0 \leq i < \ell'$, there has been information exchange between a_i and a_{i+1}, which implies that both nodes have each other's message. Lemma 3.1 (ii) implies that both edges (a_i, a_{i+1}) and (a_i, a_{i+1}) are not grey.

Assume that for some i, both edges are white (if no such i exists then we are done). By Lemma 3.1 (iv), these two edges could not have turned white at the same time step. Let t' be the time step in which the second edge turned white. By the induction hypothesis, at time t' there was an undirected path $a_i = b_{i,0}, b_{i,1}, \ldots, b_{i,\ell_i} = a_{i+1}$ colored black. Going over all such values of i, we have that there is an undirected black path between u and w (see Figure 2).

It remains to show that there is a black path between v and w. Since v and w exchange information, by Lemma 3.1 (ii), both edges (v, w) and (w, v) are colored. Assume that they are both colored white (otherwise we are done). Then (v, w) does not turn white in event time t because v is the initiator. This implies that (v, w) turns white at some time $t' < t$. By the induction hypothesis, there is an undirected black path between v and w at that time, which completes the proof of this case.

Case 2: Node v receives $m(u)$ for the first time from node w' which was the initiator of the information exchange, and removes node u from $B_t(v)$. The proof follows the lines of Case 1 to show that there is an undirected black path between u and w'. It then remains to show that there is a black path between v and w'. A similar argument to that of Case 1, replacing the initiator v with w', proves that there is such a path. ∎

Lemma 3.2 guarantees that after removing elements from the sets $B_t(v)$ of different nodes, the nodes always remain connected by black and grey edges, even though white edges in the communication graph are ignored in line

4 of the algorithm. Again, recall that a node v can still contact a neighbor u it removed from $B_t(v)$, by choosing it in line 2 of the algorithm. Finally, we note that for every round r and node v, only one edge can be colored black by v through all n event times of this round, since a node u joins $Black_t(v)$ only if u is the unique node w with whom v initiated information exchange at event time t.

CLAIM 1. *For every round r and every node v we have that $|Black_{n(r+1)-1}(v)| - |Black_{nr}(v)| \leq 1$.*

4 Analysis

We now claim that there are not *too many* black edges outgoing from any node v. This will guarantee that every neighbor remaining in $B_t(v)$ will be eventually contacted after a *small* number of rounds. The precise measures of these amounts will be defined later.

For the rest of this section, we fix a value $c > 1$ and a value $\delta \in (0, 1/3c)$. However, we emphasize that these values are not used in the algorithm, and are therefore unknown to the nodes. They are only used for the analysis[4], and eventually we will choose a value c that minimizes the number of rounds.

Let $T = O\left(\frac{\log n + \log \delta^{-1}}{\Phi_c(G)}\right)$, the number of rounds obtained in Theorem 2.2. We consider *phases* of the algorithm, each phase consisting of $2T$ rounds. The outline of our analysis is as follows. Recall that, for any $v \in V$, V_v is the component realizing the definition of the weak conductance. Roughly speaking, Theorem 2.2 shows that with high probability after one phase a node v has the messages of all nodes u in its component V_v, since the even-numbered rounds comprise of *regular* information spreading. We then show that after three phases, a node v has the messages of all nodes that are either in its component or in an intersecting component. Finally, we show that after $c(6T + 2c)$ rounds, a node v has the messages of all nodes. This strongly relies on the connectivity argument in Lemma 3.2, and a careful bookkeeping of the number of edges in $B_t(v)$ throughout these phases. In addition, we need to keep track of the probability of failure in every phase.

We begin by using Theorem 2.2 to show that starting from *any* round r_0, after one phase of the algorithm we have spread the information of $M_{nr_0}(v)$ inside the component V_v (instead of just $m(v)$ if $r_0 = 0$). We emphasize that the probability of success is for *all* nodes v to satisfy the requirements.

LEMMA 4.1. *Let r_0 be a round number. After round $r = r_0 + 2T$, with probability at least $1 - \delta$, for every node v we have $\bigcup_{u \in V_v} M_{nr_0}(u) \subseteq M_{nr}(v)$, and $M_{nr_0}(v) \subseteq M_{nr}(u)$ for every $u \in V_v$.*

[4]This is formalized in theorem 4.1.

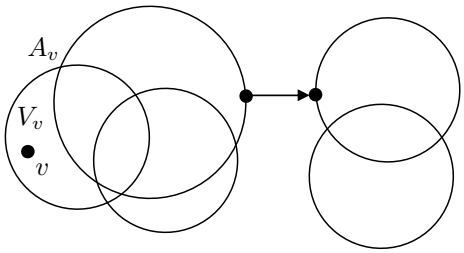

Figure 3: Illustration of a node v, its component V_v, and the set A_v of intersecting components.

Proof. Consider the state of the buffers $M_{nr}(v)$ after $r = r_0 + 2T$ rounds. By Theorem 2.2, and since $2T$ rounds consist of T even-numbered rounds, with probability $1 - \delta$ every node v has all messages $M_{nr_0}(u)$ of every $u \in V_v$, and every $u \in V_v$ has $M_{nr_0}(v)$. ∎

We consider the progress of the algorithm after three phases. For every node v we define the set $I_v = \{u \in V \mid V_u \cap V_v \neq \emptyset\}$ of nodes whose component intersects the component of v. Further, for every node v let $A_v = V_v \cup \left(\bigcup_{u \in I_v} V_u\right)$.

LEMMA 4.2. *Let r_0 be a round number. After round $r = r_0 + 6T$, with probability at least $1 - 3\delta$, for every node v we have $\bigcup_{\{x \in A_v\}} M_{nr_0}(x) \subseteq M_{nr}(v)$.*

Proof. Consider the algorithm after $6T$ rounds. Let v be a node and u a node not in V_v. If $V_v \cap V_u \neq \emptyset$ then there is a node $w \in V_v \cap V_u$. Lemma 4.1 implies that after $4T$ rounds w has all messages $M_{nr_0}(x)$ of nodes x in V_u (one phase for u to receive $M_{nr_0}(x)$ of every $x \in V_u$ and another phase for this information to reach $w \in V_u$). Applying Lemma 4.1 again gives that w spreads these messages to v in $2T$ additional rounds. That is, v has the messages $M_{nr_0}(x)$ of all nodes x in $V_v \cup \left(\bigcup_{u \in I_v} V_u\right)$. All three phases of information spreading need to succeed for the above to happen. A simple union bound on the probability that either fails gives that with probability at least $1 - 3\delta$ all three phases succeed. ∎

After each node v has all the messages of nodes in A_v, it takes only $2c$ rounds for all remaining grey edges to turn white or black. The reason we handled nodes in A_v separately in Lemma 4.2 is that nodes that are outside A_v have components that do not intersect V_v, adding at least n/c new messages to the messages of nodes in V_v. This allows us to claim that soon there are no more grey edges in the graph (see Figure 3).

LEMMA 4.3. *After $r = 6T + 2c$ rounds, with probability at least $1 - 3\delta$, for every node v we have that $\{m(u) \mid u \in N(v)\} \subseteq M_{nr}(v)$.*

Proof. Let S_v^i be the set of nodes u such that v receives the message $m(u)$ after $r_i = 6T + 2i$ rounds. We claim that with probability at least $1 - 3\delta$, for every node v and every i, $1 \leq i \leq c$, after round r_i the buffer $M_{nr_i}(v)$ of messages v receives either contains $m(u)$ of all nodes in $N(v)$, or there are i different nodes $u_1, \ldots, u_i$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq i$, and for every $1 \leq j \leq i$ we have $A_{u_j} \subseteq S_v^i$.

We prove this claim by induction. For the base case $i = 1$, by Lemma 4.2, we have that with probability at least $1 - 3\delta$, after $6T \leq r_1$ rounds each node v has $m(u)$ of all nodes u in A_v, therefore we choose $u_1 = v$.

We next assume that the claim holds up to $i - 1$ and prove it for i. By the induction hypothesis, with probability at least $1 - 3\delta$, for every node v we have after round $r_{i-1} = 6T+2(i-1)$ that there are $i-1$ nodes $u_1, \ldots, u_{i-1}$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq i-1$, and $A_{u_j} \subseteq S_v^{i-1}$ for every $1 \leq j \leq i-1$.

If S_v^{i-1} contains all nodes in $N(v)$ then we are done. Otherwise, there is a node $u \in N(v)$ such that the edge (v, u) is grey at the beginning of round r_i. In the next odd-numbered round v contacts such a node u. Since $u \notin S_v^{i-1}$, by the induction hypothesis we have $u \notin A_{u_j}$ for every $1 \leq j \leq i-1$. This implies that $V_u \cap V_{u_j} = \emptyset$ for every $1 \leq j \leq i-1$. Moreover, by Lemma 4.2, $A_u \subseteq S_v^i$. This completes the proof of our claim. The claim implies that after $6T + 2c$ rounds, either v has $m(u)$ for every $u \in N(v)$, or there are c different nodes $u_1, \ldots, u_c$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq c$, and for every $1 \leq j \leq c$ we have $A_{u_j} \subseteq S_v^c$. In particular, v has the messages of all nodes of the pairwise disjoint sets V_{u_j} for $1 \leq j \leq c$, all of which are of size n/c, which implies that v has all messages. ∎

Having the messages of all components of the neighbors of a node immediately implies no more grey edges. In addition, by Claim 1, we can bound the number of out-going black edges for each node.

COROLLARY 4.1. *After $6T + 2c$ rounds, i.e., for $t = (6T + 2c)n$, with probability at least $1 - 3\delta$, for every node v, we have $B_t(v) = Black_t(v)$, and $|B_t(v)| \leq 6T + 2c$.*

We are now ready to prove our main lemma, which bounds the complexity of the algorithm. Roughly speaking, the argument follows the line of proof of Lemma 4.3, but instead of considering grey edges, it relies on having connectivity among the black edges.

LEMMA 4.4. *With probability at least $1 - 3c\delta$, after at most $r = 2c(6T + 2c)$ rounds, for every node v we have $M_{nr}(v) = \{m(u) \mid u \in V\}$.*

Proof. Let r_0 be a round number, and let S_v^i be the set of nodes u for which $M_{nr_0}(u) \subseteq M_{nr_i}(v)$ after $r_i = r_0 + r'_i$ rounds, where $r'_i = 2i(6T + 2c)$. We use in the proof the following.

CLAIM 2. *With probability at least $1 - 3i\delta$, for every node v and $1 \leq i \leq c$, after round r_i the buffer $M_{nr_i}(v)$ of messages received by v either contains messages of all nodes, or there are i different nodes $u_1, \ldots, u_i$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq i$, and for every $1 \leq j \leq i$ we have $A_{u_j} \subseteq S_v^i$.*

Proof. We prove the claim by induction. For the base case $i = 1$, by Lemma 4.2, with probability at least $1 - 3\delta$, after $r_0 + 6T \leq r_1$ rounds, for every node v we have $\bigcup_{\{x \in A_v\}} M_{nr_0}(x) \subseteq M_{nr_1}(v)$. Therefore, $A_v \subseteq S_v^1$ so we choose $u_1 = v$.

Next, we assume that the claim holds up to $i - 1$ and prove it for i. By the induction hypothesis, with probability at least $1 - 3(i-1)\delta$, for every node v we have after round $r_{i-1} = r_0 + 2(i-1)(6T + 2c)$ that there are $i - 1$ nodes $u_1, \ldots, u_{i-1}$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq i-1$, and $A_{u_j} \subseteq S_v^{i-1}$ for every $1 \leq j \leq i-1$.

If S_v^{i-1} contains all nodes then we are done. Otherwise, by Lemma 3.2 and Corollary 4.1, by this time there is an undirected black path between any two nodes of the graph, and specifically, there is a node $u \notin S_v^{i-1}$ connected by a black edge to some node $w \in S_v^{i-1}$. Since after $6T + 2c$ rounds all nodes v have at most $6T + 2c$ nodes in $B_t(v)$, after at most $6T + 2c$ additional rounds each node contacts each of its black neighbors. Therefore, after round $r' = r_0 + 2(6T + 2c)$ the node w has the messages of all nodes in S_u^1, that is $\bigcup_{x \in S_u^1} M_{nr_0}(x) \subseteq M_{nr'}(w)$.

By the induction hypothesis with r' replacing r_0, after r'_{i-1} additional rounds v has them as well, with another factor of 3δ added to the probability of failure. Formally, $\bigcup_{x \in S_u^1} M_{nr_0}(x) \subseteq M_{nr'}(w) \subseteq M_{nr''}(v)$, where $r'' = r' + r'_{i-1} = r_0 + 2(6T + 2c) + 2(i-1)(6T + 2c) = r_i$.

We now prove that taking $u_i = u$ satisfies the requirements of our claim. Since $u \notin S_v^{i-1}$, by the induction hypothesis we have $u \notin A_{u_j}$ for every $1 \leq j \leq i-1$. This implies that $V_u \cap V_{u_j} = \emptyset$ for every $1 \leq j \leq i-1$. Moreover, $A_u \subseteq S_v^i$ since $A_u \subseteq S_u^1$. This completes the proof. ∎

By Claim 2, after $2c(6T + 2c)$ rounds either v has $m(u)$ for every $u \in N(v)$, or there are c different nodes $u_1, \ldots, u_c$ such that $V_{u_j} \cap V_{u_k} = \emptyset$ for all $1 \leq j < k \leq c$, and for every $1 \leq j \leq c$ we have $A_{u_j} \subseteq S_v^c$. In particular, v has the messages of all nodes of the pairwise disjoint sets V_{u_j} for $1 \leq j \leq c$, all of which are of size n/c. This implies that v has all messages. ∎

Rephrasing Lemma 4.4 gives our main theorem for full information spreading:

THEOREM 4.1. *For every $c > 1$ and every $\delta \in (0, 1/3c)$, Algorithm 1 obtains full information spreading after at most*

$O(c(\frac{\log n+\log \delta^{-1}}{\Phi_c(G)} + c))$ *rounds, with probability at least* $1 - 3c\delta$.

Looking at some specific values of the parameters in the above theorem we get that Algorithm 1 is fast for graphs with scalable weak conductance (for a scalable value of c). For example, if $c = \text{polylog}(n)$ and $\Phi_c = \text{polylog}(n)$, then our algorithm requires a polylogarithmic number of rounds. The probability of failure is $3c\delta$, which is $O(\text{polylog}(n)/n)$ if $\delta = 1/n$, and $O(1/n)$ if c is a constant; in both cases it is $o(1)$.

5 Application: Fast Leader Election

In this section we show how our algorithm for fast information spreading enables to improve the previously known results for the *leader election* problem, in which the nodes of a network need to choose a leader, i.e., agree on the ID of a single node. This fundamental problem allows coordination of processes and symmetry breaking in many distributed applications.

Being a central paradigm in distributed computing, leader election has been widely studied, in different models of communication and under different assumptions, dating back to 1977 [18]. Much effort was invested in the special case of a ring, where the underlying communication graph is a cycle of the n nodes, see, e.g., [2, Chapter 3]. Other research includes work on general graphs, in models that differ from ours. Peleg [20] assumes a node can send information to all its neighbors in a round. Ramanathan et al. [23] obtain logarithmic number of rounds in a model where each round consists of communication between a single node and a set of nodes it chooses among all nodes. In addition to the difference in models, the error probability in their algorithm refers to correctness, i.e., it may be that more than one leader is elected.

In contrast, our algorithm uses full information spreading with the message of each node consisting of its ID, and choosing the node with the maximal (or minimal) ID. With this approach, safety is never compromised, i.e., no two nodes can choose different leaders in any execution. The probability of failure in a certain number of rounds refers to executions in which some nodes do not receive all the required information and therefore do not choose a leader at all. Following Theorem 4.1, electing a leader using our full information spreading algorithm is fast in graphs with large weak conductance. This gives

THEOREM 5.1. *Using Algorithm 1 for full information spreading and choosing the node with the maximal ID, the leader election problem can be solved after at most* $O(c(\frac{\log n+\log \delta^{-1}}{\Phi_c(G)} + c))$ *rounds, with probability at least* $1 - 3c\delta$.

6 Discussion

This paper studies information spreading, presenting a hybrid algorithm, which interleaves random neighbor choices with deterministic ones for exchange of information. Our algorithm is fast on graphs which have large weak conductance. For graphs which also have small conductance, it substantially improves upon the running times of previously known algorithms, from polynomial to *polylogarithmic* number of rounds.

A by-product of our algorithm is the maintenance of a connected scalable-degree subgraph, which we believe will find additional applications. Specifically, it may be possible to obtain scalable-degree spanners with low stretch, by applying similar techniques.

An intriguing open question is whether there is a non-trivial lower bound on the number of rounds required for information spreading as a function of the weak conductance of the underlying graph. Another avenue for future research is to adapt our algorithm to failure-prone environments, as resilience to faults is typically required in practical scenarios.

Finally, we note that our model allows messages of unbounded size. Bounding the size of messages is another direction for further research.

Acknowledgments. The authors thank Hagit Attiya for helpful suggestions and comments on an earlier version of this paper. We also thank Chen Avin and the anonymous referees for useful comments.

References

[1] N. Alon, C. Avin, M. Koucky, G. Kozma, Z. Lotker, and M. R. Tuttle. Many random walks are faster than one. In *Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures (SPAA)*, pages 119–128, 2008.

[2] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics (2nd edition)*. John Wiley Interscience, March 2004.

[3] C. Avin and G. Ercal. Bounds on the mixing time and partial cover of ad-hoc and sensor networks. In *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pages 1–12, 2005.

[4] M. Borokhovich, C. Avin, and Z. Lotker. Tight bounds for algebraic gossip on graphs. In *Proceedings of the 2010 IEEE International Symposium on Information Theory (ISIT)*, pages 1758 –1762, 2010.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508–2530, 2006.

[6] M. Bradonjić, R. Elsässer, T. Friedrich, T. Sauerwald, and A. Stauffer. Efficient broadcast on random geometric graphs. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1412–1421, 2010.

[7] K. Censor Hillel and H. Shachnai. Partial information spreading with application to distributed maximum coverage. In *Proceedings of the 27th ACM symposium on Principles of Distributed Computing (PODC 2010)*.

[8] F. Chierichetti, S. Lattanzi, and A. 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.

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

[10] 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 sixth annual ACM Symposium on Principles of distributed computing (PODC)*, pages 1–12, 1987.

[11] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 773–781, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[12] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading: Expanders, push vs. pull, and robustness. In *36th International Colloquium on Automata, Languages and Programming (ICALP)(1)*, pages 366–377, 2009.

[13] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. In *SIGAL International Symposium on Algorithms*, pages 128–137, 1990.

[14] 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.

[15] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski. On the complexity of asynchronous gossip. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing (PODC)*, pages 135–144, 2008.

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

[17] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 482, Washington, DC, USA, 2003. IEEE Computer Society.

[18] G. L. Lann. Distributed systems - towards a formal approach. In *IFIP Congress*, pages 155–160, 1977.

[19] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing (PODC)*, pages 113–122, New York, NY, USA, 2006. ACM.

[20] D. Peleg. Time-optimal leader election in general networks. *Journal of Parallel and Distributed Computing*, 8(1):96 – 99, 1990.

[21] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[22] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

[23] M. K. Ramanathan, R. A. Ferreira, S. Jagannathan, A. Grama, and W. Szpankowski. Randomized leader election. *Distributed Computing*, 19(5-6):403–418, 2007.

[24] A. D. Sarwate and A. G. Dimakis. The impact of mobility on gossip algorithms. In *28th IEEE International Conference on Computer Communications (INFOCOM)*, pages 2088–2096, 2009.

[25] A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993.

A Proof of Lemma 3.1

For completeness, this section provides full proofs of the claims stated in Lemma 3.1.

Lemma 3.1 [restated] *The following four claims hold:*

(i) *In G_0 all edges are colored grey.*

(ii) *For any nodes v and u, if $m(u) \in M_t(v)$, and $u \in N(v)$, then (v,u) is not grey in G_t.*

(iii) *For any event time t, if for some nodes v and u both edges (v,u) and (u,v) change their color at t, then v and u are the pair of nodes that exchange information in this event time.*

(iv) *If (v,u) and (u,v) are both grey, they cannot turn both white at the same time step.*

Proof. (i) By definition, at event time 0 no edges are colored white, since no node u has been removed from any set $B_0(v)$. Moreover, no buffer $M_0(v)$ contains any message other than $m(v)$, therefore no edge is colored black. This implies that all edges in G_0 are grey.

(ii) If v has the message $m(u)$ by event time t and $u \notin White_t(v)$, then by definition, $u \in Black_t(v)$, that is, (v,u) is black in G_t.

(iii) By the code of Algorithm 1, an edge (v,u) changes its color at event time t only if v received a message at that time. If u is not one of the two nodes that exchange information at event time t then (u,v) cannot change its color at time t.

(iv) By (iii), two neighbors can change the color of the edges connecting them at the same time step only if they exchange information at this time step. Assume, without loss of generality, that v is the initiator of this exchange. Then v does not remove u from $B_t(v)$ at this step because neither of the conditions in lines 11 and 12 are satisfied. ∎