Branching Random Walks on Graphs

Chinmoy Dutta * Gopal Pandurangan † Rajmohan Rajaraman * Scott Roche *

Abstract

We study a new distributed randomized information propagation mechanism in networks that we call a *branching random walk (BRW)*. BRW is a generalization of the well-studied "standard" random walk which is a fundamental primitive useful in a wide variety of network applications ranging from token management and load balancing to search, routing, information propagation and gossip. BRW is parameterized by a *branching factor* k. The process starts from an arbitrary node, which is labeled *active* for step 1. For instance, this could be a node that has a piece of data, rumor, or a virus. In a BRW, in any step, each active node chooses k random neighbors to become active for the next step. A node is active for step $t+1$ only if it is chosen by an active node in step t. This results in a branching type process in the underlying network which has interesting properties that are strikingly different from the standard random walk, which is equivalent to BRW with branching factor $k = 1$. Similar to the standard random walk, we focus on the *cover time*, which is the the number of steps for the walk to reach all the nodes and the *partial cover time*, which is the number of steps needed for the walk to reach at least a constant fraction of the nodes.

We derive almost-tight bounds on cover time and partial cover time in *expander* graphs, an important class of graphs that arise in many distributed network applications, especially in the modeling and construction of peer-to-peer and overlay networks. A main result of this paper is that the time needed by a BRW for partial coverage in an n-node bounded-degree expander is $O(\log n)$ (even with branching factor 2, assuming sufficiently large expansion) and for full coverage is $O(\log^2 n)$ with high probability. Since the cover time of standard random walk is $\Theta(n \log n)$ in an expander, this shows that BRW gives an exponential speedup.

Keywords: Random Walks, Branching Process, Expanders, Information Spreading, Gossip, Coverage Time.

Regular Presentation

*College of Computer and Information Science, Northeastern University, Boston MA 02115, USA. E-mail: {chinmoy, rraj, str}@ccs.neu.edu. Supported in part by NSF grant CNS-0915985.

†Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371 and Department of Computer Science, Brown University, Providence, RI 02912, USA. E-mail: gopalpandurangan@gmail.com. Supported in part by the following grants: Nanyang Technological University grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 grant MOE2010-T2-2-082, US NSF grant CCF-1023166, and a grant from the US-Israel Binational Science Foundation (BSF).

1 Introduction

A well-studied randomized information propagation process in networks is the (standard) *random walk*. In a random walk, in each step, the current node (the one that has the piece of information) chooses a random neighbor to pass the information. Random walk is a fundamental primitive useful in a wide variety of network applications ranging from token management and load balancing to search, routing, information propagation, gathering, and gossip (e.g.,[10, 11, 6, 35] and the references therein). Random walks are local and lightweight and require little index or state maintenance which make them especially attractive to self-organizing dynamic networks such as Internet overlay and ad hoc wireless networks [35]. An important parameter of interest in a random walk is the *cover time* — the (expected) number of rounds needed till the walk visits all the nodes in the network. Random walks are communication efficient in the sense that there is only a single "active" node holding the information at any step; thus only constant work (i.e., communication or number of messages transmitted) is performed per round in a random walk. However, the price is the cover time can be quite high in general — $\Theta(n^3)$ in the worst case (see e.g., [29]). In fact, even in *expander networks*, an important class of graphs which have good connectivity properties and arise in a number of network applications (see e.g., [25]), the cover time is polynomially large — $\Theta(n \log n)$ [29]. Hence, several recent works have addressed the issue of speeding up the cover time of random walks [1, 4, 12]. In many of these works the main approach to speed up is by slightly modifying the random walks — e.g., visiting additional (constant) number of neighbors of the current node, while proceeding with the random walk as usual. The typical speedup given by these approaches is not very large, the cover time remains still polynomial. In particular, in expander graphs the speed up is by a logarithmic factor [4]. This raises the question whether we can speed up random walks significantly (at least in important classes such as expanders), by modifying the process. This is one main motivation of the current paper.

In this paper, we study a new distributed randomized information propagation mechanism in networks that we call a *branching random walk (BRW)*. BRW is a generalization of the (standard) random walk, and is parameterized by a *branching factor* k. The process starts from an arbitrary node, which is labeled *active* for step 1. For instance, this could be a node that has a piece of data, rumor, or a virus. In a BRW, in any step, each active node chooses k random neighbors to become active for the next step. Note that a node is active for step t only if it is chosen by an active node in step $t-1$. This results in a branching type process in the underlying network which has interesting properties that are strikingly different from the standard random walk, which is equivalent to BRW with branching factor $k = 1$. Similar to the standard random walk, we focus on the *cover time*, which is the the number of steps for the walk to reach all the nodes and the *δ-cover time*, which is the number of steps needed for the walk to reach at least a δ fraction of the nodes.

We derive almost-tight bounds on cover time and partial cover time in *expander* graphs. Expanders are a very important class of graphs that have applications in various areas of computer science —networks, crypography, derandomization, complexity and coding theory etc. (e.g., see [25] for a survey). For example, in distributed computing and networks, they have been used for censorship resistant networks [18, 17], fault tolerant networks [32], analyzing information spreading in networks [24], and efficient (Byzantine) agreement and leader election algorithms [13, 33, 28, 27, 3].

A main contribution of this paper is the analysis of the cover time and partial cover time of BRW in a (bounded-degree) expander. We show that the cover time in a n-node expander is $O(\log n)$ (even with branching factor 2, assuming sufficiently large expansion) and the partial cover time is $O(\log^2 n)$ with high probability. Since the cover time of standard random walk is $\Theta(n \log n)$ in an expander, this shows that BRW gives an exponential speedup. This also implies that the total number of messages sent is $O(n \log n)$ for partial coverage and $O(n \log^2 n)$ for full coverage. We note that this is essentially optimal and is within only a logarithmic factor compared to the cover time of the standard random walk (which is BRW with branching factor 1). Thus, increasing the branching factor to just 2 in every time step, yields an exponential speedup compared to branching factor 1, while not increasing the total message complexity by too much.

We believe that our BRW results can also be generalized to understand the time taken for an epidemic process in an SIS-type model to spread in a network [21, 26, 34]. By varying the branching factor and the time that a node remains infected, the process can also be viewed as a generalized rumor spreading model, with applications in both epidemiology and information dissemination.

1.1 Our results

We analyze the partial and full cover times of branching random walks on bounded-degree regular expanders. We say that a graph is an (α, δ)-expander if the number of neighbors of every vertex set S of vertices of size at most δn is at least $\alpha|S|$. (Note that the neighbors of vertices in S may include vertices in S.)

- We show that for any Δ-regular n-vertex (α, δ)-expander, the k-branching random walk covers at least δn nodes in $O(\log n)$ steps for $k \geq 1 + \ln(2\Delta/(\alpha - 1))$ assuming α is sufficiently large. In particular, for any random regular graph, the 2-branching random walk covers $\Omega(n)$ nodes in $O(\log n)$ steps with high probability.

- We show that the cover time of a k-branching random walk on any bounded-degree regular $(\Omega(n), \alpha)$-expander graph is $O(\log^2 n)$ for $k \geq 1 + \ln(2\Delta/(\alpha - 1))$, assuming α is sufficiently large. In particular, the cover time of the 2-branching random walk on any random regular graph with constant degree is $O(\log^2 n)$.

1.2 Related work

The study of random walks on graphs has a rich history, and we refer the reader to [29, 31] for a survey. A classic result of Aleliunas shows that the cover time of a random walk on an undirected n-vertex m-edge graph is at most $2nm$.

With the rapidly increasing interest in diffusion processes in large-scale networks and the gossiping paradigm, there have been a number of studies on speeding up random walks. One of the earliest studies is due to Adler et al [1], who studied a process on the hypercube in which in each round a vertex is chosen uniformly at random and covered; if the chosen vertex was already covered, then an uncovered neighbor of the vertex is chosen uniformly at random and covered. For any d-regular graph, Dimitrov and Plaxton showed that a similar process achieves a cover time of $O(n + (n \log n)/d)$ [12]. For expander graphs, Berenbrink et al showed a simple variant of the standard random walk that achieves a linear cover time [4].

It is instructive to compare BRW with other mechanisms to speed up random walks as well as with gossip-based rumor spreading mechanisms. Perhaps the most related mechanism is that of parallel random walks which was first studied in [5] for the special case where the starting nodes are drawn from the stationary distribution, and in [2] for arbitrary starting nodes. Nearly-tight results on the speedup of cover time as a function of the number of parallel walks have been obtained by [15] for several graph classes including the cycle, d-dimensional meshes, hypercube, and expanders. (Also see [14] for results on mixing time.) Though BRWs are similar to parallel random walks in the sense that at any step multiple nodes may be selecting random neighbors, there are significant differences between the two mechanisms. First the cover times of these walks are not comparable. For instance, while k parallel random walks may have a cover time of $\Omega(n^2/\log k)$ for any $k \in [1, n]$ [15], a 2-branching random walk on a line has a cover time of $O(n)$. Second, while the number of active nodes in k parallel random walks is always k, the number of active nodes in any k-branching random walk is continually changing and *may not even be monotonic*. Most importantly, the analysis of cover time of BRWs needs to address several dependencies in the process by which the set of active nodes evolve; we use the machinery of time-inhomogenous Markov chains to obtain the $O(\log^2 n)$ bound on the cover time for bounded-degree expanders [30] (see Section 3).

The works of [10, 11] presented fast distributed algorithms for performing (standard) random walks. The goal is to improve the round complexity of the standard walk — which takes ℓ rounds to do a walk

of length ℓ. The above works present a sublinear time distributed algorithm for performing random walks whose time complexity is sublinear in the length of the walk. In particular, the algorithm of [11] performs a random walk of length ℓ in $\tilde{O}(\sqrt{\ell D})$ rounds with high probability on an undirected network, where D is the diameter of the network. The high-level idea behind the algorithm is to perform several short walks in parallel and then stitch them carefully. However, this speed up comes with a drawback: this the message complexity of the above faster algorithm is much worse compared to the naive sequential walk which takes only ℓ messages. In contrast, we note that the exponential speedup given by BRW over the standard comes only at the cost of a slightly worse message complexity.

Gossip-based information propagation mechanisms have also been used for information (rumor) spreading in distributed networks. In the most typical rumor spreading models, gossip involves either a push step, in which nodes that are aware of a piece of information (being disseminated) pass it to random neighbors, or a pull step, in which nodes that are unaware of the information attempt to extract the information from one of their randomly chosen neighbors, or some combination of the two. In such models, the knowledgeable nodes or the ignorant nodes participate in the dissemination problem in *every* round (step) of the algorithm. The main parameter of interest in many of these analyses is the number of rounds needed till all the nodes in the network get to know the information.

The rumor spreading mechanism that is most closely related to BRWs is the basic push protocol, in which in every step every informed node selects a random neighbor and pushes the information to the neighbor, thus making it informed. Feige et al [16] show that the push process completes in every undirected graph in $O(n \log n)$ steps, with high probability. Since then, the push protocol and its variants have been extensively analyzed both for special graphs, as well as for general graphs in terms of their expansion properties (see e.g., [7, 8, 9, 23, 22, 20, 19]). Again, though BRWs and rumor spreading share the property that multiple nodes are active in a given step, the two mechanisms differ significantly from each other. While the set of active nodes in rumor spreading is monotonically nondecreasing, this is not so in branching random walks, an aspect that makes the analysis challenging especially with regard to full coverage. Furthermore, the message complexity of the push protocol can be substantially different than that of BRWs. For instance, for the star network, the push protocol covers all nodes in $\Theta(n \log n)$ steps with a message complexity of $\Theta(n^2 \log n)$, while the 2-branching random walk has both cover time and message complexity $\Theta(n \log n)$.

2 Partial Cover Time

In this section, we prove the following theorem that bounds the partial cover time for a branching random walk on graphs with sufficient expansion.

Theorem 1. *Let $\delta > 0, \alpha > 0, and \Delta > 2$ be constants. Then if for a constant $k \geq 2$, the equation*

$$\alpha > \frac{\frac{\Delta^2}{e^k} + (k-1)\Delta - \frac{k^2}{2}}{\frac{\Delta}{e^k} + (k-1)\Delta - \frac{k^2}{2}}$$

is satisfied, then a k-branching random walk on any Δ-regular (α, δ)-expander has a δ-cover time of $O(\log n)$ with high probability.

Remark. For $2 \leq k \leq \Delta$, the condition of the lemma is met if

$$\alpha > 1 + \frac{\Delta}{e^k}\left(\frac{\Delta - 1}{\Delta - 2}\right).$$

Since we must assume $\Delta \geq 3$ as one cannot prove fast cover time for $\Delta = 2$ (the case of a line), and we have $\frac{\Delta-1}{\Delta-2} \leq 2$ for $\Delta \geq 3$, the condition of the theorem will be satisfied if

$$\alpha > 1 + \frac{2\Delta}{e^k}.$$

The above gives us the following corollaries.

Corollary 2. *If $\Delta \geq 3$ and $\alpha > 1 + \frac{2\Delta}{e^{\Delta}}$, then a k-branching random walk on a Δ-regular (α, δ)-expander has a δ-cover time of $O(\log n)$ if*

$$k > \ln\left(\frac{2\Delta}{\alpha - 1}\right).$$

Corollary 3. *A 2-branching random walk on a Δ-regular random graph, where $\Delta \geq 3$, covers $\Omega(n)$ nodes in time $O(\log n)$ with high probability.*

Our proof of Theorem 1 shows the stronger result that the number of active nodes itself grows to δn in time $O(\log n)$ with high probability. The proof is divided into three parts: (a) we first show that if the number of active nodes is less than δn at time t, then the expected number of active nodes in time $t+1$ is at least a factor $(1+\nu)$ more where $\nu > 0$ (Lemma 4), (b) using this expectation bound and a martingale argument, we then bound the probability that the number of active nodes do not grow by a constant factor greater than 1 from time t to $t+1$ (Lemma 6), (c) finally, using the bound on the probability of not growing the number of active nodes by a constant factor, we show that the number of active nodes grows to δn in time $O(\log n)$ with high probability (Lemma 7).

We let S_t denote the set of active nodes at time t. Also, for any set S of nodes, we let $N(S)$ denote the set of neighbors of S, i.e., the set of all those nodes which have at least one edge to a node in S. Note that $N(S)$ can intersect S. First we prove growth in expectation.

Lemma 4. *For any $t \geq 0$, if $|S_t| \leq \delta n$, $E[|S_{t+1}|] \geq (1+\nu)|S_t|$ for some constant $\nu > 0$.*

Proof. Let $k =$. Fix a t and suppose $|S_t| \leq \delta n$. It will suffice to show that $\mathrm{E}[|N(S_t) - S_{t+1}|] \leq |N(S_t)| - (1+\nu)|S_t|$. For each vertex $u \in N(S_t)$, let X_u be the indicator variable that is 1 if $u \notin S_{t+1}$ and 0 otherwise. The probability that $X_u = 1$ is $p = \left(1 - \frac{1}{\Delta}\right)^{kd_u}$, where d_u the number of edges of u to vertices in S_t. Thus the expectation of X_u is p. We have

$$E[|N(S_t) - S_{t+1}|] =\leq \sum_{u \in N(S_t)} X_u = \sum_{u \in N(S_t)} \left(1 - \frac{1}{\Delta}\right)^{kd_u} \leq \sum_{u \in N(S_t)} e^{-\frac{kd_u}{\Delta}}.$$

We will like to know for what values of d_u's the expression $e^{-\frac{kd_u}{\Delta}}$ is maximized? We note that $\sum_{u \in N(S_t)} d_u = \Delta|S_t|$. Combining this fact with Lemma 5, it is immediate that the expression will be maximized when for any u, d_u is either Δ or 1, except possibly for one u. Let R_1 be the number of u's for which $d_u = 1$, and R_2 be the number of u's for which $d_u = 1$. We have

$$\begin{aligned} R_1 + R_2 &= |N(S_t)|, \\ R_1 + \Delta R_2 &= \Delta|S_t|. \end{aligned}$$

Solving for R_1 and R_2, we get

$$\begin{aligned} R_1 &= \frac{\Delta}{\Delta - 1}(|N(S_t)| - |S_t|), \\ R_2 &= \frac{1}{\Delta - 1}(\Delta|S_t| - |N(S_t)|). \end{aligned}$$

Thus we have

$$\begin{aligned} E[|N(S_t) - S_{t+1}|] &\leq R_1 e^{-\frac{k}{\Delta}} + R_2 e^{-k} \\ &= \frac{\Delta}{\Delta - 1}(|N(S_t)| - |S_t|)e^{-\frac{k}{\Delta}} + \frac{1}{\Delta - 1}(\Delta|S_t| - |N(S_t)|)e^{-k} \end{aligned}$$

and we want to show that this is at most

$$|N(S_t)| - (1+\nu)|S_t|.$$

Rearranging, we want

$$|N(S_t)|\left(1 - \frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}} + \frac{1}{\Delta-1}e^{-k}\right) + |S_t|\left(\frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}} - \frac{\Delta}{\Delta-1}e^{-k} - 1\right) \geq \nu|S_t|.$$

We know $|N(S_t)| \geq \alpha|S_t|$ since $|S_t| \leq \delta n$. Since $1 - \frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}} + \frac{1}{\Delta-1}e^{-k} > 0$, the above will be true if

$$\alpha\left(1 - \frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}} + \frac{1}{\Delta-1}e^{-k}\right) + \left(\frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}} - \frac{\Delta}{\Delta-1}e^{-k} - 1\right) > 0.$$

Rearranging, we want

$$(\alpha-1)\left(1 - \frac{\Delta}{\Delta-1}e^{-\frac{k}{\Delta}}\right) - \frac{\Delta-\alpha}{\Delta-1}e^{-k} > 0.$$

Since $e^{-\frac{k}{\Delta}} \leq 1 - \frac{k}{\Delta} + \frac{k^2}{2\Delta^2}$, the above condition will be met if

$$(\alpha-1)\left(1 - \frac{\Delta}{\Delta-1}\left(1 - \frac{k}{\Delta} + \frac{k^2}{2\Delta^2}\right)\right) - \frac{\Delta-\alpha}{\Delta-1}e^{-k} > 0,$$

which simplifies to

$$\alpha > \frac{\frac{\Delta^2}{e^k} + (k-1)\Delta - \frac{k^2}{2}}{\frac{\Delta}{e^k} + (k-1)\Delta - \frac{k^2}{2}}.$$

□

We used the following easy lemma in the proof above.

Lemma 5. *Let $c > 0$ and $a, b > 1$ such that $a \leq b$. Then*

$$e^{-c(a-1)} + e^{-c(b+1)} > e^{-ca} + e^{-cb}.$$

Proof.

$$\begin{aligned} e^{-c(a-1)} + e^{-c(b+1)} - (e^{-ca} + e^{-cb}) &= e^{-ca}(e^c - 1) + e^{-c(b+1)}(1 - e^c) \\ &= (e^c - 1)(e^{-ca} - e^{-c(b+1)}) \\ &> 0, \end{aligned}$$

since $c > 0$ and $a < b + 1$. □

Next, we show that the number of active nodes is concentrated near the expectation.

Lemma 6. *For any time t, $\Pr[|S_{t+1}| - E[|S_{t+1}|] \leq -\tau|S_t|] \leq exp(-\frac{\tau^2|S_t|}{2k})$.*

Proof. For any time step t, we arbitrarily order the active nodes in S_t. Then we define random variables (Z_i^j), $1 \leq i \leq |S_t|$, $1 \leq j \leq k$, where Z_i^j indicates which vertex was chosen by the ith node in S_t in its j trial to be active at time $t+1$. (Note that each active node in S_t chooses k neighbors uniformly at random with replacement.)

Let A be the size of S_{t+1}. Then $X_i^j = E[A|Z_1^1, \ldots, Z_1^k, \ldots, Z_{i-1}^1, \ldots, Z_{i-1}^k, Z_i^1, \ldots, Z_i^j]$ is the Doob martingale of A. We have $X_{|S_t|}^k = |S_{t+1}|$. Since $X_j - X_{j-1}$ is bounded by at most 1, Azuma's inequality gives us:

$$\begin{aligned} \Pr[|S_{t+1}| - \mathrm{E}[|S_{t+1}|] \leq -\tau|S_t|] &\leq exp(-\frac{\tau^2|S_t|^2}{2k|S_t|}) \\ &= exp(-\frac{\tau^2|S_t|}{2k}). \end{aligned}$$

□

Lastly, using the bound on probability of growth obtained above, we prove an $O(\log n)$ bound for the δ-cover time.

Lemma 7. *$|S_t| \geq \delta n$ for some $t = O(\log n)$.*

Proof. We want to analyze the change in the number of active nodes as a Markov process. For this, consider a Markov process X_t on the state space $\{1, 2, \ldots n\}$. Setting $\tau = \nu/2$ in Lemma 6, we would like to define the transitions as follows: with probability $1 - exp(-\frac{\nu^2 X_t}{8k})$, $X_{t+1} = (1 + \frac{\nu}{2})X_t$, and with probability $exp(-\frac{\nu^2 X_t}{8k})$, $X_{t+1} = 1$, a conservative over-estimate.

But for technical simplicity, we define our random walk slightly differently. Let C be a sufficiently large constant. It is clear that with some constant probability we can arrive at $X_t = C$ in a constant number of steps from 1. Thus, letting $r = \frac{\nu^2}{8k}$, we define our transitions as follows. Let $X_t = C(1 + \frac{\nu}{2})^i$. Then, with probability $1 - exp(-rC(1 + i\frac{nu}{2}))$, $X_{t+1} = (1 + \frac{\nu}{2})X_t$, and with probability $exp(-rC(1 + i\frac{nu}{2}))$, $X_t = C$. It is not difficult to see that it suffices to prove that this random walk will reach δn in time $O(\log n)$.

We want to show that starting from C, our random walk will reach size δn without going back to 1 with probability greater than $\frac{1}{2}$. We observe that the probability of this not happening is at most

$$e^{-rC} + e^{-rC(1+\frac{\nu}{2})} + e^{-rC(1+2\frac{\nu}{2})} + \cdots \leq e^{-rC}(1 + e^{-rC\frac{\nu}{2}} + e^{-2rC\frac{\nu}{2}} + \ldots) = \frac{e^{-rC}}{1 - e^{-rC\frac{\nu}{2}}} \leq \frac{1}{2}.$$

if C is chosen sufficiently large, depending on the value of r. From here, we can easily achieve our result in time $O(\log^2 n)$ with high probability by repeating the above $O(\log n)$ times. However, we can actually do better and achieve an $O(\log n)$ bound. For this, we view each segment of walk where starting from C we either reach δn (success) or go back to C (failure), whichever is earlier, as a trial. Since probability of a successful trial is at least $\frac{1}{2}$, we know that with high probability, there are at most $b \log n$ failed trials before we have a trial that is successful, for some constant b. Label a step (within a trial) where the walk advances as heads in a coin flip and a step where we go back to C as tails. Each failed trial then consists of some number (possibly zero) of heads flips followed by a single tails flip. Given that we have $b \log n$ failed trials before we have a successful trial, we know that there are $b \log n$ tails before we get a successful trial. We would like to show that with high probability, there are $O(\log n)$ heads in all the failed trials combined as well, which will prove our lemma.

We need to bound the probability of heads conditioned that we are in a failed trial. Now the probability that we get a tails in step i of a failure trial is larger than

$$\frac{e^{-rC(1+i\frac{\nu}{2})}}{\sum_{l=i}^{\infty}\left[\prod_{j=i}^{l-1}(1 - e^{-rC(1+j\frac{\nu}{2})})\right] e^{-rC(1+l\frac{\nu}{2})}} \geq 1 - e^{-rC\frac{\nu}{2}}.$$

Thus the conditional probability of heads within a failure trial is at most $e^{-rC\frac{\nu}{2}}$. Now applying Chernoff bound we get that with high probability, there are $O(\log n)$ heads within all failure trials. □

3 Cover time analysis

In this section, we show that the cover time of a k-branching random walk on constant-degree regular expanders is $O(\log^2 n)$, for any $k \geq 2$. In particular, we show that once the number of active nodes reaches $\Omega(n)$ as we have already established in Section 2, the number of subsequent steps needed to cover every node on a bounded degree regular expander is $O(\log^2 n)$. Our analysis proceeds by considering an alternative process, which we call R_{alt}, whose cover time stochastically dominates that of the k-branching walk starting from a state of δ-coverage.

In R_{alt}, we consider each active vertex as possessing a pebble. In R_{alt}, pebbles continue their walks but no branching occurs. The walks of the pebbles, though random, are not independent from one another. In particular, the transition probabilities of pebbles are modified when more than one pebble hits the same vertex at the same time to resemble collisions of pebbles in the original process. The walk of one pebble, conditioned on the walk of another, can be viewed as a time-inhomogenous Markov process that is only a minor perturbation of the independent walk of a lone pebble on G. Using a well-known result of Mihail [30] that exploits an expansion-like property of the graph, called merging conductance, but is applicable to arbitrary irreducible Markov chains, we show that any given node is visited by a pebble in $\Theta(\log n)$ time, ensuring full coverage of G in $O(\log^2 n)$ time.

In Section 3.1, we define R_{alt}. In Section 3.2, we show that R_{alt} has a cover time of $O(\log^2 n)$ with high probability, implying an $O(\log^2 n)$ bound on the cover time of a k-branching random walk for any $k \geq 2$.

3.1 Description of R_{alt}

For this process, we have the same underlying graph, G, as in our original process. Each of the δn marked nodes has its own individual pebble to start. Arbitrarily index and order the pebbles. This ordering will be used to assign priority when determining the transition probability of a pebble. At each time step, each pebble choses a neighbor to move to according to the following rules: (a) If there is one pebble at a node, the pebble choses a neighbor with probability $\frac{1}{\Delta}$; (b) If there are at least two pebbles at a node, then the two highest priority pebbles each chose a neighbor independently and uniformly at random with probability $\frac{1}{\Delta}$. The remaining pebbles chooses one of the already-selected neighbors independently with probability $\frac{1}{2}$.

Note that if a node has one pebble, behavior of R_{alt} is locally equivalent to random walk over graph G. We can map the original process to R_{alt} as follows. If a node in R_{alt} has two or more pebbles, it behaves exactly identical to an active node in the original process in the next time step (i.e. choosing two neighbors uniformly and independently at random). If R_{alt} has only one pebble, then it does not behave like an active node in the original process, since it only picks one neighbor to transmit its pebble to. In this way, the number of active nodes at the next time step under the original process is a superset of the nodes with pebbles in R_alt at the next time step. In this way, the original process stochastically dominates R_{alt}.

3.2 Analysis of R_{alt}

Theorem 8. *Let G be a bounded-degree Δ-regular expander graph. Let there be δn pebbles on δn vertices of G. Then the cover time of R_{alt} is $O(\log^2 n)$ with high probability.*

Proof. If we can show that each vertex in G is covered with constant probability in $\Theta(\log n)$ steps, then the theorem follows by carrying out $O(\log n)$ phases of $\Theta(\log n)$ steps each. Let E_i be the event that pebble i covers an arbitrary vertex v in s steps. We are interested in $\Pr[\bigcup_i E_i]$, the probability that v is covered by at least one pebble, because we want to show that it is larger than some constant. To prove this, we use a second-order approximation:

$$\Pr[\bigcup_i E_i] \geq \sum_i \Pr[E_i] - \sum_{i \neq j} \Pr[E_i \cap E_j] = \sum_i \Pr[E_i] - \sum_{i \neq j} \Pr[E_i] \Pr[E_j | E_i] \tag{1}$$

Each term $\Pr[E_i]$ is $\leq O(\frac{1}{n})$, since at the end of phase 2, each pebble has a probability of being at any particular node equal to the value of the vertex's component of the stationary distribution vector, $\frac{1}{n}$. From

[2], the probability that an independent walk of pebble i of length $O(\log n)$ deviates from $\frac{1}{n}$ by at most $\frac{2}{n}$. To bound the second-order term we will need to analyze the conditional walk of pebble j given the walk of pebble i, which has already been fixed.

To bound $\Pr[E_j|E_i]$, fix the walk of pebble i. Now consider the walk of j. Clearly if it does not intersect the walk of j in such a way that both i and j are at the same node at the same time, it is just another independent random walk, since we are not including the walks of other pebbles, we can view the edge-selection probability of pebble j at each step of $\frac{1}{\Delta}$ as a marginal probability across all other pebble paths, except for i's. On the other hand, if i and j intersect in both space and time, we need to consider the priorities of the two pebbles. In the worst of all possible cases, i is highly ranked and j is lowly ranked such that by the time we get to j it is at least the third pebble at that node. If this is the case, then j will pick i's edge out of the intersection node with probability $\frac{1}{2}$. With probability $\frac{1}{2}$ it will pick another pebble's edge out of the node. However, since we are calculating without knowledge of the path of any other pebble, we again use the marginal probability of edge selection, $\frac{1}{\Delta}$ for each edge, thus giving us probability of selecting the same edge as i of $\frac{1}{2} + \frac{1}{2\Delta}$ and $\frac{1}{2\Delta}$ for every other edge.

Based on the above discussion, pebble j follows the modified transition matrix $(M+T_{l(i),t})$, where M is the standard transition matrix for G, and $T_{l(i),t}$ is a perturbation matrix depending on the vertex location of i at time t, viewed as the function $l(i)$. The rows $(M+T_{l(i),t})_k$ are identical to M for $k \neq l(i)$. For $k = l(i)$, the entries of the column are $\frac{1}{2} + \frac{1}{2\Delta}$ for one randomly selected entry that is $\frac{1}{\Delta}$ in M, and the rest of the entries are $\frac{1}{2\Delta}$. From any probability distribution z over $V(G)$, it is clear that that after $s = O(\log n)$ steps the probability of pebble j being a vertex v is the corresponding component of the vector $z\prod_{t=1}^{s}(M + T_{l(i),t})$. By Lemma 9 below each component of $z\prod_{t=1}^{s}(M + T_{l(i),t}) < \frac{2}{n}$. Thus,

$$\begin{aligned}
\Pr(\bigcup E_i) &\geq \sum_i P(E_i) - \sum_{i\neq j} P(E_i)P(E_i|E_j) \\
&\geq \epsilon n\frac{1}{n} - \binom{n}{2}\frac{2}{n^2} \\
&\geq \epsilon - \epsilon^2
\end{aligned}$$

and the theorem follows. □

Lemma 9. *For z a probability distribution over $V(G)$, and $(M + T_{l(i),t})$ a family of transition operators on V(G) over a fixed sequence $\{l(i), t\}$ for $t \in \{1, \ldots, s\}$, each component of $z \cdot \prod_{t=1}^{s}(M + T_{l(i),t}) < \frac{2}{n}$.*

Proof. Let $z = u + x$, where u is the uniform distribution over $V(G)$ and x is a vector such that $\sum x_i = 0$. Then $z \cdot \prod_{t=1}^{s}(M + T_{l(i),t}) = x \cdot \prod_{t=1}^{s}(M + T_{l(i),t}) + u \cdot \prod_{t=1}^{s}(M + T_{l(i),t})$.

Lemma 10 proves that the merging conductance of any $M + T_{l(i),t}$ is bounded by some constant away from zero and that therefore we can use Mihail's theorem to show that $x \cdot \prod_{t=1}^{s}(M + T_{l(i),t})$ is component-wise within $O(\frac{1}{n})$ of the stationary distribution of the perturbed matrix within $\Theta(\log n)$ time.

Finally, we want to show that each component of $u \cdot \prod_{t=1}^{s}(M+T_{l(i),t}) < \frac{2}{n}$. Consider our arbitrary path $v_0 \to v_1 \to \ldots \to v_s$. Start a pebble walking from a vertex drawn from the stationary distribution u. Define $p_j(v_i)$ as the probability of a pebble being at vertex i at time step j. We want to show that the probability of being at any vertex in G after s time steps is less than $\frac{2}{n}$. We have that $\forall v, p_0(v) = \frac{1}{n}$. We claim that for all vertices along the fixed path, $p_k(v_k) \leq \frac{2}{n}$, and that $p_k(v) \leq \frac{1}{n}$ for $v \notin \{v_0, \ldots, v_s\}$.

The first claim, $p_k(v_k) \leq \frac{2}{n}$, is proved by induction. We will show that $p_j \leq \frac{1}{n}(1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^j})$.

For the base case, $p_1(v_1) = \frac{1}{n}(\frac{1}{2} + \frac{1}{2}) + \frac{1}{n}(\frac{\Delta-1}{\Delta}) = \frac{1}{n}(2 - \frac{1}{2\Delta}) \leq \frac{2}{n}$. For the inductive case, we have

$$\begin{aligned} p_{j+1}(v_{j+1}) &\leq p_j(v_j)(\frac{1}{2} + \frac{1}{2\Delta}) + \frac{1}{n}\frac{\Delta-1}{\Delta}. \\ &\leq \frac{1}{n}(1 + \frac{1}{2} + \cdots + \frac{1}{2^j})(\frac{1}{2} + \frac{1}{2\Delta}) + \frac{1}{n}\frac{\Delta-1}{\Delta} \\ &= \frac{1}{n}(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{j+1}} + \frac{1}{n} + \frac{1}{\Delta n}(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{j+1}}) - \frac{1}{\Delta n} \\ &\leq \frac{1}{n}(1 + \frac{1}{2} + \cdots + \frac{1}{2^{j+1}}) \end{aligned}$$

The second claim has two cases: the first is that v is a neighbor of vertex along the path, v_j. Then $p_{j+1}(v) \leq \frac{1}{2\Delta}p_j(v_j) + \frac{\Delta-1}{\Delta}\frac{1}{n} \leq \frac{1}{n}$.

Finally, if v is not a neighbor of v_j, then we have $p_{j+1}(v) \leq \frac{1}{n}$ □

Lemma 10. *Let $M + T_{l(i),t}$ be a perturbed random walk on expander G as described in the description of R_{alt} and x be a vector over V such that $\sum_i x_i = 0$. Then for $s = \Theta(\log n)$, each component of $x \cdot \prod_{t=1}^{s}(M + T_{l(i),t})$ is $O(\frac{1}{n})$.*

Proof. The proof of this lemma relies heavily on Theorem 3.2 in [30], which we review and state here. Let P be an irreducible, ergodic Markov process. Note that time-reversibility and even strong aperiodicity are not required. Rather than using just the transition probabilities p_{ij} between vertices i and j, instead consider a weighted transition w_{ij}, where $w_{ij} = \pi_i p_{ij}$, with π_i being the i^{th} component of the stationary distribution of P. For a subset $A \subset V$, consider a property called the merging conductance, defined as:

$$\Phi_P^*(A) = \frac{\sum_{j_1 \in A}\sum_{j_2 \in S-A}\sum_i \frac{w_{j_1 i}w_{j_2 i}}{\pi_i}}{\sum_{i \in A}\pi_i} \tag{2}$$

and define the merging conductance of G to be

$$\Phi_P^*(A) = \min_{A \subset S: \sum_{i \in A}\pi_i \leq \frac{1}{2}} \Phi_P^*(A) \tag{3}$$

Intuitively, the merging conductance can be viewed as a measure of the flow coming into all vertices from both A and $S - A$, for some set A. The higher the merging conductance of a graph, the more well connected it is and evenly distributed the flow is. If we define $\|\vec{x}(t)\| = \sum \frac{(p_i(t)-\pi_i)^2}{\pi_i}$ to be a measure of the distance of a distribution over V, $\vec{p}$ from the stationary distribution, [30] gives us the following theorem, which indicates for a graph with conductance bounded away from zero, convergence to the stationary distribution occurs in logarithmic time.

Theorem 11. *[30, Theorem 3.2]*

$$\|\vec{x}(t)\| \leq (1 - \frac{1}{2}(\Phi_P^*)^2)^t \|\vec{x}(0)\| \tag{4}$$

Any bound on the merging conductance of $M + T_{l(i),t}$ requires an understanding of how much the stationary distribution of the perturbed walk differs from the stationary distribution of the original walk, for which $\pi_i = 1/n$ for all $i \in V$). Consider u to be the one vertex in $M + T_{l(i),t}$ whose transition probabilities are perturbed. Let v be the neighbor that receives the pebble with probability $\frac{1}{2} + \frac{1}{2\Delta}$, and let $u_1, \ldots, u_{\Delta-1}$ be the other neighbors of u that receive the pebble with probability $\frac{1}{2\Delta}$. Clearly π_v is the max of the π_i's. Suppose it were not. Then some other vertex j not in $\{u, v, u_1, \ldots u_{\Delta-1}\}$ has $\pi_j = \pi_{max}$. But since $\pi_j = \pi_{max} = \sum_{k \in N^{-1}(j)} \pi_k p_{kj}$, and $p_{kj} = \frac{1}{\Delta}$, it follows that π_{max} is equal to the mean of the neighboring

π_k's implies $\pi_k = \pi_{max}$ for all $k \in N^{-1}(j)$. We continue this calculation until we reach one of the u_i's, implying that $\pi_{u_i} = \pi_{max}$. However, this is a contradiction, since $\pi_{u_i} = \frac{1}{2d}\pi_u + \frac{d-1}{d}\pi_{max} \leq \pi_{max}$. Thus π_{max} must either be π_u or π_v, and w.l.o.g. we can assume that it is π_v. A similar argument shows that the $u_1, \ldots, u_{\Delta-1}$ take values π_{min}.

Now we bound the spread between π_{min} and π_{max}. Note that $\pi_u = \frac{\Delta-1}{\Delta}\pi_{min} + \frac{1}{\Delta}\pi_{max}$. Then, by elementary algebra, we obtain $\pi_{max} \leq (\Delta+1)\pi_{min}$. Note that the mean value for π over V is $1/n$. Using the bounds calculated above, that means that $\pi_{min} \geq \frac{1}{(\Delta-1)n} > \frac{1}{\Delta n}$ and $\pi_{max} \leq \frac{\Delta-1}{n} < \frac{\Delta}{n}$.

We next lower-bound the number of $\frac{w_{j_1 i} w_{j_2 i}}{\pi_i}$ terms in the calculation of the merging conductance. Let A be the set that produces the minimum merging conductance. Essentially, we need to bound from below the number of vertices that have one neighbor in A and another neighbor in $V - A$. Using vertex expansion α, we have that $N(A)$, the neighborhood of A comprised of neighbors j of each $i \in A$ s.t. $j \in V - A$ has size $|N(A)| \geq \alpha|A|$. Now, we do not know how many of the vertices in $N(A)$ also have a neighbor in $V - A$. However, we can look at the vertex expansion of set $N(A) \bigcup A$. Clearly $|N(N(A) \bigcup A)| \geq \alpha^2|A|$. However, since all the neighbors of A in $V - A$ were captured in $N(A)$, this implies that $N(N(A) \bigcup A)$ must consist only of vertices in $V - A$. From this we can conclude that there is at least one edge from each vertex in $N(N(A) \bigcup A)$ to $N(A)$, and that this edge is incident to a vertex that has at least one edge back to A. Thus there are at least $\alpha^2|A|$ conductance terms in Φ_P^*. Plugging in our worst-case values for π_i and p_{ij}, assuming that $|A| = cn$ for some $c \in [0, 1]$, and that $\sum_{i \in A} = \frac{1}{2}$, we have:

$$\Phi_P^* \geq \alpha^2|A| \frac{\frac{\pi_{min}^2 p_{ij}^2}{\pi_{max}}}{|A|\pi_{max}} = \alpha^2 (\frac{1}{\Delta})^2 (\frac{1}{2\Delta})^2 = \Omega(\frac{1}{\Delta^4}).$$

As long as Δ is a constant integer much smaller than n, the merging conductance Φ_P^* is safely bounded away from zero. Note that we make no attempt to optimize this bound – clearly much tighter bounds are achievable. Therefore, we can treat $(1 - \frac{1}{2}(\Phi_P^*)^2)$ as a constant, $\gamma \in (0, 1]$. The largest value of $\|\vec{x}(0)\|$ occurs of course when starting the walk at one vertex. Thus:

$$\|\vec{x}(0)\| \quad \leq \quad \frac{(1-\pi_{min})^2}{\pi_{min}} + (n-1)\frac{(\pi_{max})^2}{\pi_{min}} = \Delta n(1 - \frac{1}{\Delta n})^2 + \Delta n(n-1)(\frac{\Delta}{n})^2.$$

Hence $\|\vec{x}(0)\|$ is $O(n)$. In order to obtain a value for $\|\vec{x}(s)\|$ that is $O(\frac{1}{n})$ we will need to pick a value of s so that γ^t is $\Omega(\frac{1}{n^2})$:

$$\gamma^s = \frac{A}{n^2} \text{ for some constant } A \Rightarrow n^2\gamma^s = c \Rightarrow 2\log n + s\log\gamma = \log c \Rightarrow s = \frac{\log c - 2\log n}{\log \gamma} \Rightarrow s = \Theta(\log n).$$

Thus for $s = \Theta(\log n)$ we are guaranteed that $\|\vec{x}(0)\| = \Omega(\frac{1}{n})$, and therefore so is each component.

□

4 Conclusion

We introduced a new type of distributed information propagation mechanism, namely the BRW and analyzed the cover time and partial cover time of BRW in expanders, which are used to model and design P2P and overlay networks. We showed that the the cover time and partial cover time are exponentially faster in BRW compared to the standard random walk. Since random walks have extensive applications in networks, we hope BRW will also be useful, with the additional property of faster coverage. There are several interesting open problems regarding BRW that remain to be solved. In general, unlike the standard random walk which has a well-developed theory, we know little about the properties of BRW in general graphs. For example, what is the worst case cover time of BRW, and how does it vary with the branching factor k? It is clear that the cover time is not worse than a standard random walk, but it will be interesting to establish tight asymptotic bounds. Furthermore, it will be interesting to establish and compare the message complexity of BRW with the standard random walk as well as other gossip-based rumor spreading processes.

References

[1] Micah Adler, Eran Halperin, Richard M. Karp, and Vijay V. Vazirani. A stochastic process on the hypercube with applications to peer-to-peer networks. In *STOC*, pages 575–584, 2003.

[2] Noga Alon, Chen Avin, Michal Koucký, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. Many random walks are faster than one. In *SPAA*, pages 119–128, 2008.

[3] John Augustine, Gopal Pandurangan, Peter Robinson, and Eli Upfal. Towards robust and efficient computation in dynamic peer-to-peer networks. In *SODA*, 2012.

[4] Petra Berenbrink, Colin Cooper, Robert Elsässer, Tomasz Radzik, and Thomas Sauerwald. Speeding up random walks with neighborhood exploration. In *SODA*, pages 1422–1435, 2010.

[5] A. Broder. Generating random spanning trees. In *FOCS*, 1989.

[6] Marc Bui, Thibault Bernard, Devan Sohier, and Alain Bui. Random walks in distributed computing: A survey. In *IICS*, pages 1–14, 2004.

[7] Flavio Chierichetti and Silvio Lattanzi andAlessandro Panconesi. Almost tight bounds for rumour spreading with conductance. In *STOC*, pages 399–408, 2010.

[8] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumour spreading and graph conductance. In *SODA*, pages 1657–1663, 2010.

[9] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412(24):2602–2610, 2011.

[10] Atish Das Sarma, Danupon Nanongkai, and Gopal Pandurangan. Fast distributed random walks. In *PODC*, pages 161–170, 2009.

[11] Atish Das Sarma, Danupon Nanongkai, Gopal Pandurangan, and Prasad Tetali. Efficient distributed random walks with applications. In *PODC*, 2010.

[12] Nedialko B. Dimitrov and C. Greg Plaxton. Optimal cover time for a graph-based coupon collector process. In *ICALP*, pages 702–716, 2005.

[13] Cynthia Dwork, David Peleg, Nicholas Pippenger, and Eli Upfal. Fault tolerance in networks of bounded degree. *SIAM J. Comput.*, 17(5):975–988, 1988.

[14] Klim Efremenko and Omer Reingold. How well do random walks parallelize? In *APPROX-RANDOM*, pages 476–489, 2009.

[15] Robert Elsässer and Thomas Sauerwald. Tight bounds for the cover time of multiple random walks. In *ICALP (1)*, pages 415–426, 2009.

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

[17] Amos Fiat and Jared Saia. Censorship resistant peer-to-peer content addressable networks. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '02, pages 94–103, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.

[18] Amos Fiat and Jared Saia. Censorship resistant peer-to-peer networks. *Theory of Computing*, 3(1):1–23, 2007.

[19] Nikolaos Fountoulakis and Konstantinos Panagiotou. Rumor spreading on random regular graphs and expanders. In *APPROX-RANDOM*, pages 560–573, 2010.

[20] Nikolaos Fountoulakis, Konstantinos Panagiotou, and Thomas Sauerwald. Ultra-fast rumor spreading in social networks. In *SODA*, pages 1642–1660, 2012.

[21] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM*, volume 2, pages 1455 – 1466, march 2005.

[22] Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *STACS*, pages 57–68, 2011.

[23] George Giakkoupis and Thomas Sauerwald. Rumor spreading and vertex expansion. In *SODA*, pages 1623–1641, 2012.

[24] Keren Censor Hillel and Hadas Shachnai. Partial information spreading with application to distributed maximum coverage. In *PODC '10: Proceedings of the 28th ACM symposium on Principles of distributed computing*, New York, NY, USA, 2010. ACM.

[25] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(04):439–562, August 2006.

[26] D. A. Kessler. Epidemic Size in the SIS Model of Endemic Infec- Tions. *ArXiv e-prints*, September 2007.

[27] Valerie King and Jared Saia. Breaking the $O(n^2)$ bit barrier: Scalable byzantine agreement with an adaptive adversary. *J. ACM*, 58:18:1–18:24, July 2011.

[28] Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Towards secure and scalable computation in peer-to-peer networks. In *FOCS*, pages 87–98, 2006.

[29] L. Lovász. Random walks on graphs: a survey. In *Combinatorics, Paul Erdös is Eighty*, pages 1–46. 1993.

[30] Milena Mihail. Conductance and convergence of markov chains-a combinatorial treatment of expanders. In *FOCS*, pages 526–531, 1989.

[31] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2004.

[32] Nicholas Pippenger and Geng Lin. Fault-tolerant circuit-switching networks. *SIAM J. Discrete Math.*, 7(1):108–118, 1994.

[33] Eli Upfal. Tolerating a linear number of faults in networks of bounded degree. *Inf. Comput.*, 115(2):312–320, 1994.

[34] Piet Van Mieghem. The n-intertwined SIS epidemic network model. *Computing*, 93:147–169, 2011.

[35] Ming Zhong and Kai Shen. Random walk based node sampling in self-organizing networks. *Operating Systems Review*, 40(3):49–55, 2006.