

Peer-to-Peer Networks based on Random Transformations of Connected Regular Undirected Graphs

Peter Mahlmann^{*}
Heinz Nixdorf Institute and
Computer Science Department
University of Paderborn, Germany
mahlmann@upb.de

Christian Schindelhauer^{*}
Heinz Nixdorf Institute and
Computer Science Department
University of Paderborn, Germany
schindel@upb.de

ABSTRACT

We present k -Flipper, a graph transformation algorithm that transforms regular undirected graphs. Given a path of $k+2$ edges it interchanges the end vertices of the path. By definition this operation preserves regularity and connectivity. We show that every regular connected graph can be reached by a series of these operations for all $k \geq 1$. We use a randomized version, called Random k -Flipper, in order to create random regular connected undirected graphs that may serve as a backbone for peer-to-peer networks. We prove for degree $d \in \Omega(\log n)$ that a series of $\mathcal{O}(dn)$ Random k -Flipper operations with $k \in \Theta(d^2 n^2 \log 1/\epsilon)$ transforms any graph into an expander graph with high probability, i.e. $1 - n^{-\Theta(1)}$.

The Random 1-Flipper is symmetric, i.e. the transformation probability from any labeled d -regular graph G to G' is equal to those from G' to G . From this and the reachability property we conclude that in the limit a series of Random 1-Flipper operations converges against a uniform probability distribution over all connected labeled d -regular graphs. For degree $d \in \omega(1)$ growing with the graph size this implies that iteratively applying Random 1-Flipper transforms any given graph into an expander asymptotically almost surely.

We use these operations as a maintenance operation for a peer-to-peer network based on random regular connected graphs that provides high robustness and recovers from degenerate network structures by continuously applying these random graph transformations. For this, we describe how network operations for joining and leaving the network can be designed and how the concurrency of the graph transformations can be handled.

^{*}Partially supported by the DFG-Sonderforschungsbereich 376 and by the EU within 6th Framework Programme under contract 001907 "Dynamically Evolving, Large Scale Information Systems" (DELIS).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SPAA'05, July 18–20, 2005, Las Vegas, Nevada, USA.
Copyright 2005 ACM 1-58113-986-1/05/0007 ...\$5.00.

Categories and Subject Descriptors

G.2.2 [Graph Theory]: Graph Algorithms, Network Problems; E.1 [Data Structures]: Graphs and Networks; C.2.4 [Distributed Systems]; G.3 [Probability and Statistics]: Markov Processes, Stochastic Processes

General Terms

Algorithms, Theory

Keywords

Peer-to-Peer Networks, Random Graphs, Expander Graphs, Distributed Algorithms

1. INTRODUCTION

Random processes in graphs, like random walks, have been successfully studied for a long time [13]. Recently, the interest of random processes transforming graphs into other graphs has grown. Such processes bear the potential for stabilizing distributed networks like peer-to-peer networks. However, changing a graph can be irrevocable if the graph represents a communication network, e.g. if an operation disconnects nodes or sub-graphs from the graph.

For random walks in graphs one is interested whether the corresponding Markov process converges to a stationary distribution and if, how fast and to which distribution. Sometimes the random walks give insight into the graph structure like the PageRank algorithm [17]. For random transformations of graphs one is either interested in maintaining (truly) random graphs or graphs which fulfill certain properties with high probability.

As an example we refer to the JXTA-package of the SUN-Java peer-to-peer network suite.¹ As backbone the network aspires a random graph by a random process. Experimental analysis shows that some parameters (the indegree) of the resulting graph are Pareto distributed what is unlikely for random graphs. Gnutella [6] in its first implementation used an insertion procedure aiming at constructing a robust network structure with small diameter. Experimental analysis shows that random graphs with Pareto degree distribution evolve [8]. In [14] query algorithms based on multiple random walks are presented that resolve queries almost as quickly as the flooding method of Gnutella while

¹See <http://www.jxta.org>

reducing the network traffic by two orders of magnitude in many cases. The authors note that among the various network topologies considered, uniform random graphs yield the best performance. In [9, 10] the reliability of such random networks is affirmed.

In this paper we concentrate on the generation of d -regular undirected connected graphs. There is a number of reasons for this choice. These graphs describe the situation in peer-to-peer networks where peers maintain bidirected connections and both peers are responsible for the link. A uniform degree induces some fairness to the network since each peer stores the same amount of network information. Furthermore, messages passed over a long random walk will pass all peers with equal probability. For an excellent survey on random regular graphs and their properties we refer to [21].

Little is known so far, whether the processes used in practice to maintain random graphs satisfy even minimal standards. We desire the following properties for random transformations of graphs:

Soundness: No transformation maps to graphs which are not in the domain space. For d -regular undirected connected graphs, this means that each operation preserves degree d at every node and there is not even the slightest (small probability) chance to disconnect parts from the graph.

Generality: The random transformation process does not converge to a specific graph. All graphs are *reachable* and in the limit all graphs occur with some non-zero probability. This requirement can be tightened to *uniform generality* where in the limit all graphs occur with the same probability.

Feasibility: The random transformation can be described by a simple (distributed) routine changing only a small number of edges of the graph. Its implementation in a distributed network should be straightforward.

Convergence rate: Only a polynomial number of transformations is necessary to achieve an approximation of the ultimate distribution of all graphs.

Surprisingly, the process used in JXTA which is widely used, has never been analyzed with respect to these features. Identifying such transformations meeting all these requirements gives an approximate solution to the problem of computing a random probability distribution over all graphs of a kind, as being solved by Steger et al. [18] for the problem of generating all d -regular random graphs.

The following random transformation for d -regular undirected graphs without connectivity has been introduced by McKay [15] and is used in [5] (there called “rewiring”).

Definition 1 (Simple Switching) *Choose random edges $\{u, v\}$, $\{u', v'\}$ of the graph G . If $\{u, v'\}$ and $\{u', v\}$ do not exist in G then erase $\{u, v\}$, $\{u', v'\}$ and insert edges $\{u, v'\}$ and $\{u', v\}$.*

In [16] Simple Switching is used to generate random d -regular graphs with $d \in \mathcal{O}(n^{1/3})$ and it takes an expected time of $\mathcal{O}(nd^3)$ per graph to generate a uniform distribution over all d -regular graphs. Simple Switching preserves the degree of each node but does not preserve connectivity.

Its convergence speed is polynomial in the number of nodes which follows from the results of [4, 16].

Simple Switching is feasible if the graph is given as a data structure on a single machine. However, when the graph constitutes an interconnection network of computers, this procedure is no more feasible. As long as all nodes are connected one can choose random edges by performing a random walk with an appropriate length, i.e. the mixing time of the graph. But during these operations Simple Switching may disconnect parts from the network. Then, without extra network connections the network cannot be rejoined anymore. Our point is that feasibility in terms of distributed algorithms implies maintaining connectivity at all stakes.

In this paper we present sound, general, feasible, quickly converging transformations for d -regular random graphs. We introduce a family of graph transformations, namely the k -Flipper operations, for an integer $k \geq 1$ and their randomized versions. Starting with an arbitrary d -regular connected graph we repeatedly apply these operations. Thereby we can guarantee the resulting graph to stay connected and d -regular. Furthermore these operations will turn any graph into an expander and introduce fresh randomness to the graph which is especially helpful in dynamic graphs with a changing node set. For the Random 1-Flipper we can prove uniform generality, i.e. all connected d -regular graphs occur with the same probability in the limit. To our knowledge this gives the first solution to the problem of generating d -regular connected graphs with labeled nodes with uniform probability. As a consequence, expander graphs occur a.a.s. if the degree is non-constant. Note that for the problem of generating regular graphs with nearly uniform probability distribution results have been known for a long time. Jerum and Sinclair [7] give a generator for d -regular graphs on n vertices which approximates the uniform probability distribution by a factor of $1 + \epsilon$ and runs polynomial in n and $\log 1/\epsilon$. This result was improved by Steger and Wormald [18]. We are aware that a.a.s. all d -regular graphs are connected and it is easy to use these algorithms to approximate also d -regular connected graphs. However, their methods cannot be applied in a networking concept.

It is not clear how fast the Random 1-Flipper produces this uniform probability distribution. If we choose $k \in \Theta(d^2 n^2 \log 1/\epsilon)$ we can show the convergence rate of the Random k -Flipper. It turns out that after $\mathcal{O}(dn)$ Random k -Flipper operations an expander graph is established with high probability, i.e. $1 - n^{-\Theta(1)}$.

We use these operations to provide a peer-to-peer network based on random regular connected graphs that provides high robustness and recovers from degenerate network structures by continuously applying these random graph transformations. For this, we discuss how the concurrency of these graph transformations can be solved and how operations for joining and leaving the network can be designed.

2. NOTATIONS

A d -regular undirected graph $G = (V, E)$ is defined by a finite node set $V = \{1, 2, 3, \dots, n\}$ of size n and the edge set $E := \{\{u, v\} : u, v \in V, u \neq v\}$ (of size $dn/2$) such that each node is adjacent to exactly d edges. An undirected graph is connected if for each pair of nodes of V there exists a path using edges of E connecting these nodes. For $v \in V$ we denote the set of nodes neighboring v by $N(v)$. By $\log n := \log_2 n$ we mean the dual logarithm function.

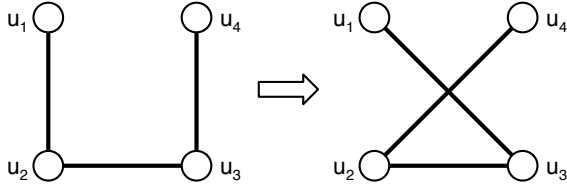


Figure 1: The 1-Flipper operation F_P^1 .

In the following the term “with high probability” (w.h.p.) describes the probability $p \geq 1 - n^{-c}$ and “asymptotically almost surely” (a.a.s) is the probability $p \geq 1 - o(1)$.

3. UNIFORM GENERATION OF REGULAR CONNECTED GRAPHS

In this section we present an elegant method to generate an uniform probability distribution of regular connected graphs in the limit. For this, we introduce the 1-Flipper and its randomized version.

Definition 2 (1-Flipper) Consider a d -regular undirected graph $G = (V, E)$ and four distinct nodes $u_1, u_2, u_3, u_4 \in V$ forming a path $P = (u_1, u_2, u_3, u_4)$ in G . Then, if $\{u_1, u_3\}, \{u_2, u_4\} \notin E$ the 1-Flipper operation F_P^1 transforms graph G to a graph $F_P^1(G) = (V, E')$ with

$$E' := (E \setminus \{\{u_1, u_2\}, \{u_3, u_4\}\}) \cup \{\{u_1, u_3\}, \{u_2, u_4\}\}.$$

Figure 1 illustrates the 1-Flipper operation. We denote $\{u_1, u_2\}, \{u_3, u_4\} \in E$ as *flipping edges* and $\{u_2, u_3\} \in E$ as *hub edge* of the 1-Flipper operation. A randomized version of the 1-Flipper is given by the following algorithm.

Algorithm 1 Random 1-Flipper

Choose random edge $\{u_2, u_3\} \in E$
 Choose random node $u_1 \in N(u_2) \setminus \{u_3\}$
 Choose random node $u_4 \in N(u_3) \setminus \{u_2\}$
if $\{u_1, u_3\}, \{u_2, u_4\} \notin E$ **then**
 $E \leftarrow E \setminus \{\{u_1, u_2\}, \{u_3, u_4\}\}$
 $E \leftarrow E \cup \{\{u_1, u_3\}, \{u_2, u_4\}\}$

We will now analyze this graph transformation. The following lemma shows that the 1-Flipper operation is sound.

Lemma 1 The 1-Flipper operation preserves connectivity and d -regularity.

PROOF. Concerning d -regularity note that each node receives one new edge and loses one of its former edges.

For connectivity consider two nodes u, v and a path P connecting them. This path can be destroyed, if an edge of P is chosen as flipping edge. However all participating nodes of the 1-Flipper operation remain connected so that another path between u and v can be found. \square

A delimiting factor for applying the 1-Flipper operation is the existence of a triangle such that for a hub edge $\{u_2, u_3\}$ nodes v with $\{u_2, v\}, \{u_3, v\} \in E$ exist. Then neither $\{u_2, v\}$ nor $\{u_3, v\}$ can be chosen as flipping edges. Let $\nabla_G(u, v)$ be

the number of triangles in G with $\{u, v\}$ as an edge. Then the following lemma holds:

Lemma 2 For a hub edge $e \in E$ there are $(d - 1 - \nabla_G(e))^2$ possibilities to perform a 1-Flipper operation that changes the edge set E .

PROOF. Assume for a hub edge $e = \{u, v\}$ that edges $\{v, w\}, \{u, w\}$ exist in G , then choosing one of these triangle edges as flipping edges will prevent a change of the edge set. If the flipping edges are not part of such triangles then the 1-Flipper operation will change the edge set. \square

Note that in some graphs for certain hub edges there is no possibility to perform an edge flip at all. In this case all neighbors are connected to both nodes of the hub edge. Then, the 1-Flipper operation has no effect unless another hub edge is chosen. Let $G \xrightarrow{F^1} G'$ denote the predicate that graph G is transformed to G' by a 1-Flipper operation. For the transformation probability between graphs the following lemma holds:

Lemma 3

1. For all $d > 2$ there is a connected d -regular graph G such that $P[G \xrightarrow{F^1} G] \neq 0$.
2. For graphs $G' = F_P^1(G)$ with $P = (u_1, u_2, u_3, u_4)$: $\nabla_G(\{u_2, u_3\}) = \nabla_{G'}(\{u_2, u_3\})$.
3. For graphs $G' = F_P^1(G)$ with $P = (u_1, u_2, u_3, u_4)$: $\nabla_G(\{u_1, u_4\}) = \nabla_{G'}(\{u_1, u_4\})$.
4. For all undirected regular graphs G, G' : $P[G \xrightarrow{F^1} G'] = P[G' \xrightarrow{F^1} G]$.

PROOF.

1. Consider a graph with edges $\{u_1, u_2\}, \{u_1, u_i\}, \{u_2, u_i\}$ for $i \in \{3, \dots, d+2\}$ and an arbitrary set of further edges satisfying the d -regularity. If edge $e = \{u_1, u_2\}$ is chosen as hub edge then $\nabla_G(e) = d - 1$ and from Lemma 2 it follows that there is no possibility to perform a 1-Flipper operation that changes E .
2. This property follows by the definition of the 1-Flipper operation. Note that it is not possible to establish or delete triangles containing the hub edge by definition. Of course other triangles can be created or erased. However they do not count for $\nabla_{G'}(\{u_2, u_3\})$.
3. The proof is analogous to the proof of Lemma 3.2.
4. Note that G and G' differ by exactly four edges connecting four nodes. At least two of these nodes are connected by a hub edge. If exactly two of these nodes are connected by a hub edge e then the probability to transform G to G' is according to Lemma 2 given by $\frac{2}{dn} (1 - \frac{\nabla_G(e)^2}{(d-1)^2})$. From Lemma 3.2 the same probability follows for the opposite direction $G' \xrightarrow{F^1} G$.

If there are hub edges e and e' forming a quadrangle together with the flipping edges the probability to transform G to G' is $\frac{2}{dn} (2 - \frac{\nabla_G(e)^2 + \nabla_G(e')^2}{(d-1)^2})$. Then, the claim follows by applying Lemma 3.2 and Lemma 3.3.

\square

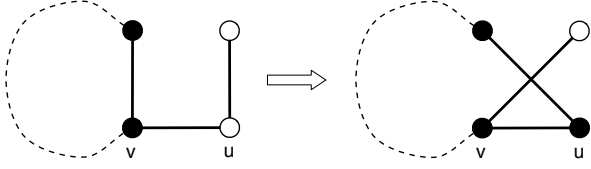


Figure 2: Extending a cycle in G by a 1-Flipper operation with $\{u, v\}$ as hub edge. Nodes of the cycle are depicted black.

We will now prove that the 1-Flipper operation provides generality.

Lemma 4 *For all pairs G, G' of connected d -regular undirected graphs, with $d \geq 2$ and even, there exists a sequence of 1-Flipper operations transforming G into G' .*

PROOF. We will show that any connected d -regular graph $G = (V, E)$ with $V = \{u_1, \dots, u_n\}$ can be transformed into a canonical graph $G_C = (V, E_C)$ with edge set defined to be $E_C = \{\{u_i, u_{(i \bmod n)+1}\}, \dots, \{u_i, u_{((i+d/2) \bmod n)+1}\}\}$ for $1 \leq i \leq n$. From this and the symmetry of the 1-Flipper the lemma follows.

To transform G into G_C we start with making G hamiltonian. Note that every connected graph contains a node disjoint cycle, not necessarily containing all n nodes. We extend this cycle to contain all nodes, thus make G hamiltonian. For this we successively add neighboring nodes to the cycle: Let v denote a node of the cycle and let u be a non-cycle node neighboring v . To add u to the cycle we perform a 1-Flipper operation with $\{u, v\}$ as hub edge. The flipping edges are one of v 's edges on the cycle and an arbitrary edge incident to u different from $\{u, v\}$. This way u is connected to two neighboring nodes on the cycle and thus the cycle can be extended using u (see Figure 2).

It remains to show that no triangles prevent us from applying these 1-Flipper operations. If there exists an edge between u and the end node of the flipping edge lying on the cycle, then u can be incorporated into the cycle without a 1-Flipper operation. For the second flipping edge between u and one of its neighbors we show that u has at least one neighbor which is non-adjacent to v using the d -regularity of G . Node v has already three neighbors: u and two nodes on the cycle. Without v , u has $d - 1$ neighbors which are different from v 's neighbors on the cycle (otherwise we do not have to apply the 1-Flipper operation). Furthermore, these $d - 1$ nodes cannot all be adjacent to v since this would imply degree $d + 2$ for v . So there exist flipping edges such that the 1-Flipper operation described above can always be performed. After at most $n - 3$ applications G is hamiltonian.

Having built the Hamilton cycle we bring the nodes of the cycle into the right ordering such that the cycle represents the edges $\{u_i, u_{(i \bmod n)+1}\}$ of G_C . Note that applying a 1-Flipper to a path (u_1, u_2, u_3, u_4) interchanges the two inner nodes and results in the path (u_1, u_3, u_2, u_4) . If we want to exchange two neighboring nodes u, v of the cycle then we can choose $\{u, v\}$ as hub edge and choose the other cycle edges of u respectively v as flipping edges. This way we can arrange any ordering of the nodes on the cycle.

Again we have to show that no triangles prevent these 1-Flipper operations. Therefore consider a 1-Flipper operation

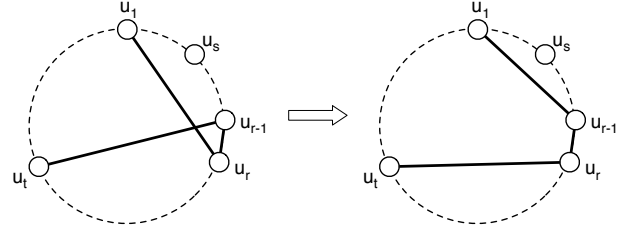


Figure 3: Transforming G into G_C : The endpoint u_r of edge $\{u_1, u_r\}$ is moved in direction of node u_s with a 1-Flipper operation.

applied to four neighboring nodes u_1, u_2, u_3, u_4 of the cycle. This can be done unless edges $\{u_1, u_3\}$ or $\{u_2, u_4\}$ exist. If both of these edges exist we simply redefine the path of the cycle to get the desired ordering. Without restriction of generality we assume that $\{u_2, u_4\}$ is the edge preventing the 1-Flipper operation. In this case we can easily choose another flipping edge instead of $\{u_3, u_4\}$. We need an edge incident to u_3 , not incident to u_2 and not part of a triangle over $\{u_2, u_3\}$. The existence of such an edge can be shown using the d -regularity of G as we did above. So, we can always find flipping edges to perform the desired graph transformations.

Having established the edges $\{u_i, u_{(i \bmod n)+1}\}$ we change the remaining edges incrementally according to G_C . To do this, we restrict ourselves to 1-Flipper operations, where the hub edge is an edge of the cycle and the flipping edges are non-cycle edges. This way the Hamilton cycle remains unchanged. Furthermore we do not use edges which we have already moved according to G_C as flipping edges, i.e. we do not destroy parts of the graph which we already have adjusted.

We start at node u_1 and establish edges $\{u_1, u_3\}, \dots, \{u_1, u_{d/2+1}\}$ in this order. Then we do the same for nodes u_2, \dots, u_n . Here we only show how this can be done for node u_1 . For the remaining nodes this can be done in the same manner. From the outgoing (non-cycle) edges of u_1 we choose the one whose endpoint is reached first when following the Hamilton cycle in direction of increasing node numbers starting from u_1 . Let u_r denote the current and $u_s, s < r$, be the desired endpoint of this edge. We move the endpoint of the edge in decreasing direction of the cycle successively until u_s is reached. The first step is done by applying a 1-Flipper operation to the path (u_1, u_r, u_{r-1}, u_t) , with $\{u_{r-1}, u_t\}$ being a non-cycle and not already adjusted edge. This transformation is illustrated in Figure 3. The remaining steps towards u_s are done using similar 1-Flipper operations.

Once again these 1-Flipper operations can be blocked by triangles. Note that the first flipping edge $\{u_1, u_r\}$ cannot be part of a triangle over $\{u_r, u_{r-1}\}$ since $\{u_1, u_{r-1}\}$ does not exist in G , otherwise it would have been chosen as flipping edge. However node u_{r-1} can have no free flipping edge over the hub edge $\{u_r, u_{r-1}\}$. This can only happen if the edge $\{u_{r-1}, u_{r+1}\}$ exists or there is an already adjusted edge pointing to u_{r-1} , since u_{r-1} cannot be a neighbor of u_1 . Furthermore the neighboring cycle edges can also be blocked by triangles. However there will be a non-blocked hub edge in distance $(d - 2)/2$, as the following lemma shows.

Lemma 5 *In any hamiltonian d -regular graph G with $d > 2$ at most $d - 3$ contiguous edges of the Hamilton cycle can be completely blocked by triangles such that no 1-Flipper using one of these $d - 3$ edges as hub edge and edges lying not on the Hamilton cycle as flipping edges can be applied.*

PROOF. Let $\{u_i, u_{(i \bmod n)+1}\}$, $1 \leq i \leq n$, be the edges forming the Hamilton cycle. Without restriction of generality we regard the neighboring nodes u_1, \dots, u_{d-2} of the Hamilton cycle. To block edge $\{u_1, u_2\}$ u_1 and u_2 have to have $d - 2$ common neighbors neglecting their neighbors on the Hamilton cycle. The same argumentation holds for edge $\{u_2, u_3\}$, so that u_3 has also to be connected to these $d - 2$ nodes. Continuing to edge $\{u_{d-3}, u_{d-2}\}$ this implies that these $d - 2$ neighbors have reached degree d and thus can not have any more neighbors. This implies that edge $\{u_{d-2}, u_{d-1}\}$ can be used as hub edge since there cannot be $d - 2$ triangles over $\{u_{d-2}, u_{d-1}\}$. \square

Having such a non-blocked hub edge in distance at most $(d-2)/2$, we can use it to remove a triangle blocking the original considered hub edge as follows. Assume that there are $d - 2$ neighboring cycle edges $\{u_k, u_{k+1}\}, \{u_{k+1}, u_{k+2}\}, \dots, \{u_{k+(d-3)}, u_{k+(d-2)}\}$ blocked by triangles. From Lemma 5 we know that $\{u_{k-1}, u_k\}$ has two free flipping edges. Applying a 1-Flipper with hub edge $\{u_{k-1}, u_k\}$ and a triangle edge of u_k and an arbitrary free edge of u_{k-1} as flipping edges will remove one of the triangles over $\{u_{k-1}, u_k\}$, thus make $\{u_{k-1}, u_k\}$ non-blocked. This procedure can be repeated until the desired hub edge is non-blocked.

The way described above we can make any desired hub edge non-blocked and thus apply the desired 1-Flipper operations. The situation is slightly different when we already have adjusted the edges of $n - (d - 1)$ nodes. Then there are $d - 2$ hub edges left which we will use (this is due to our construction scheme). According to Lemma 5 alone it is possible that these $d - 2$ edges are blocked. However in our particular case this cannot happen since the endpoints of the blocking triangles would have to lie in parts of G which we have already processed, and this is not possible by definition of G_C . \square (Lemma 4)

Let $G \xrightarrow{i} G'$ denote the predicate that G' is derived from G by applying i Random 1-Flipper operations. Furthermore let $\mathcal{C}_{n,d}$ denote the set of all connected d -regular graphs with n nodes. The following theorem shows that the Random 1-Flipper operation provides uniform generality.

Theorem 1 *Let G_0 be a d -regular connected graph with n nodes and $d > 2$. Then in the limit the Random 1-Flipper operation constructs all connected d -regular labeled graphs with the same probability, i.e.*

$$\lim_{t \rightarrow \infty} P[G_0 \xrightarrow{t} G] = \frac{1}{|\mathcal{C}_{n,d}|}.$$

PROOF. Consider a Markov process over the set of all connected d -regular graphs described by the Random 1-Flipper. From Lemma 3.1 we know that some diagonal entries of the Markov transition matrix are non-zero. From Lemma 3.4 we know that the process is symmetric and therefore double stochastic. Lemma 4 shows that every state of the Markov process can be reached. From this the claim follows by applying essential results of Markov theory. \square

We now give a definition for expander graphs. Expander graphs have a number of advantageous properties, e.g. logarithmic diameter, high vertex connectivity and a small mixing time of random walks.

Definition 3

1. For $S, T \subset V$ denote the set of all edges between S and T by $E(S, T) = \{\{u, v\} | u \in S, v \in T, \{u, v\} \in E\}$.
2. The edge boundary of a set $S \subset V$, denoted by ∂S , is $\partial S = E(S, \bar{S})$ with $\bar{S} = V \setminus S$.
3. A graph $G = (V, E)$ provides expansion $\beta > 0$, or is a β -expander, if for all node sets S with $|S| \leq |V|/2$

$$|\partial S| \geq \beta |S|.$$

Theorem 2 *For $d \in \omega(1)$ a random connected d -regular graph is a $\Theta(d)$ -expander graph a.a.s..*

PROOF. Note that for fixed $d \geq 3$ any random d -regular graph is connected a.a.s.. This follows by independent proofs of [2] and [20] who even prove d -connectivity. Furthermore in [3] it is proved that the isoperimetric number (which is the expansion) of a random regular graph is a.a.s. between $\frac{1}{2}d - \epsilon(d)$ and $\frac{1}{2}d + \epsilon(d)$ where $\epsilon(d) \rightarrow 0$ as $d \rightarrow \infty$.

Since, nearly all random regular graphs are connected and nearly all of them have an expansion of $\Theta(d)$ it follows that nearly all random regular connected graphs have an expansion of $\Theta(d)$. \square

Corollary 1 *For $d > 2$ consider any d -regular connected graph G_0 with n nodes. Then in the limit the Random 1-Flipper operation establishes an expander graph after a sufficiently large number of applications a.a.s..*

As we have seen in this section the Random 1-Flipper constructs expander graphs in the limit. It is an open problem how many 1-Flipper operations are necessary. We conjecture that a polynomial number of operations is sufficient to establish expander graphs with high probability. There is some experimental evidence in favor to this thesis.

4. FAST CONSTRUCTION OF EXPANDER GRAPHS

In this section we present a generalization of the 1-Flipper operation for which we can show a polynomial bound on the convergence speed towards an expander graph. For this we extend the hub edge of the 1-Flipper to a path of k edges leading to following definition.

Definition 4 (k -Flipper) *Consider a d -regular undirected graph $G = (V, E)$ and $k + 3$, $k \in \mathbb{N}$, nodes $u_1, \dots, u_{k+3} \in V$ forming a path $P = (u_1, \dots, u_{k+3})$ in G . We call $\{u_1, u_2\}, \{u_{k+2}, u_{k+3}\} \in E$ flipping edges and the path (u_2, \dots, u_{k+2}) in G the hub path. If the edges $\{u_1, u_{k+2}\}$ and $\{u_2, u_{k+3}\}$ do not exist in E then the k -Flipper operation F_P^k transforms G to $F_P^k(G) = (V, E')$ with*

$$E' := (E \setminus \{\{u_1, u_2\}, \{u_{k+2}, u_{k+3}\}\}) \cup \{\{u_1, u_{k+2}\}, \{u_2, u_{k+3}\}\}.$$

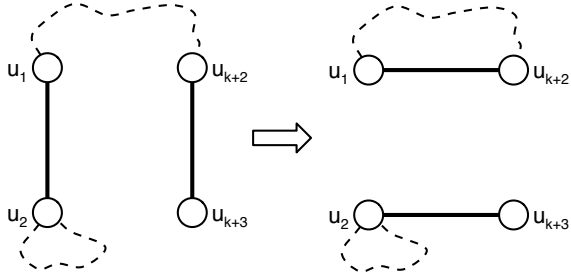


Figure 4: A k -Flipper can disconnect a graph.

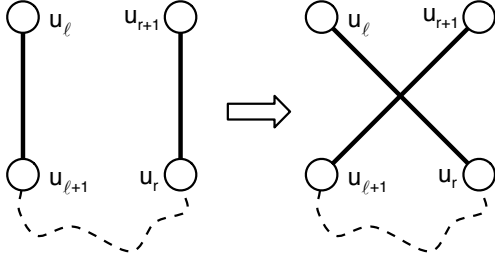


Figure 5: The k -Flipper with truncated hub path.

Unlike as the 1-Flipper, the k -Flipper can disconnect a graph. Figure 4 shows a k -Flipper operation which uses the flipping edge $\{u_1, u_2\}$ twice such that the resulting graph is partitioned into disconnected components. In order to preserve connectivity we have to ensure that there is a hub path without using the flipping edges. On the other hand we do not want to bias the random walk by forbidding to use the first edge or the a priori unknown last edge.

Fortunately, this problem is easy to handle. A simple solution is to truncate the path $P = (u_1, \dots, u_{k+3})$ to a path $P' = (u_\ell, \dots, u_{r+1})$ with $1 \leq \ell < r < k + 3$ such that $\{u_\ell, u_{\ell+1}\} = \{u_1, u_2\}$, $\{u_r, u_{r+1}\} = \{u_{k+2}, u_{k+3}\}$ and $\{u_1, u_2\}, \{u_{k+2}, u_{k+3}\}$ do occur only once in P' (see Figure 5). This observation leads to the following algorithm.

Algorithm 2 Random k -Flipper

```

Choose random node  $u_1 \in V$ 
for  $i \leftarrow 1$  to  $k + 2$  do
  Choose random node  $u_{i+1} \in N(u_i)$ 
for  $i \leftarrow k + 2$  downto  $2$  do
  if  $\{u_i, u_{i+1}\} = \{u_{k+2}, u_{k+3}\}$  then  $r \leftarrow i$ 
for  $i \leftarrow 1$  to  $r$  do
  if  $\{u_i, u_{i+1}\} = \{u_1, u_2\}$  then  $\ell \leftarrow i$ 
if  $r \geq \ell + 2$  and  $\{u_\ell, u_r\}, \{u_{\ell+1}, u_{r+1}\} \notin E$  then
   $E \leftarrow E \setminus \{\{u_\ell, u_{\ell+1}\}, \{u_r, u_{r+1}\}\}$ 
   $E \leftarrow E \cup \{\{u_\ell, u_r\}, \{u_{\ell+1}, u_{r+1}\}\}$ 

```

We continue with the analysis of the Random k -Flipper.

Lemma 6 *The Random k -Flipper operation preserves connectivity and d -regularity.*

PROOF. For d -regularity the same arguments as in Lemma 1 hold. Applying the edge flip to the truncated path $P' = (u_\ell, \dots, u_{r+1})$ it is ensured that the graph will stay connected since no edges of the hub path will be removed. \square

Similar to the 1-Flipper operation, the Random k -Flipper operation provides generality.

Lemma 7 *For all pairs G, G' of connected d -regular undirected graphs there exists a sequence of Random k -Flipper operations transforming G into G' .*

PROOF. Note that a Random k -Flipper operation can be reduced to a 1-Flipper operation if the path at the beginning uses the start edge $k - 1$ times. In Lemma 4 we have proved that this property holds for the 1-Flipper. \square

The Random k -Flipper is not a symmetric graph transformation as the Random 1-Flipper. Therefore, it is not clear if the Random k -Flipper provides uniform generality. Nevertheless, it establishes an expander graph in a polynomial number of rounds which we prove now. For this we start with a bisection of the node set of a graph $G = (V, E)$ into $S \subset V$ and $\bar{S} = V \setminus S$ with $|S| \leq |V|/2$. Let $|V| = n, |S| = m$ and $|E(S, \bar{S})| = q$ be the number of edges of the cut. We are interested in the number q' of edges between S and \bar{S} after applying a Random k -Flipper operation. Now assume that each edge is chosen with uniform probability $\frac{2}{dn}$ as a flipping edge. This assumption is motivated by the fact that a long random walk in the graph will choose flipping edges with uniform probability. Then there are the following cases:

1. Both flipping edges are chosen from $E(S, S)$. Then the Random k -Flipper operation will not increase the number of edges of the cut, i.e. $q' = q$.
2. Both flipping edges are chosen from $E(\bar{S}, \bar{S})$. Again $q' = q$.
3. One flipping edge is in $E(S, S)$ and one is in $E(\bar{S}, \bar{S})$. This will occur with probability $\frac{2(dm-q)(dn-dm-q)}{d^2n^2}$ and two edges are added to the cut, i.e. $q' = q + 2$.
4. One flipping edge is in $E(S, \bar{S})$ and one is in $E(S, S)$. Then $q' = q$.
5. One flipping edge is in $E(S, \bar{S})$ and one is in $E(\bar{S}, \bar{S})$. Then $q' = q$.
6. Both flipping edges are in $E(S, \bar{S})$. This happens with probability $(\frac{2q}{dn})^2$. In this case the number of edges on the cut can be decreased by two or stay the same, i.e. $q' \in \{q - 2, q\}$. However, it is guaranteed from the connectivity property that $q' \geq 1$.

This proves the following lemma.

Lemma 8 *Consider a bisection of a graph $G = (V, E)$ into $S \subset V, \bar{S} = V \setminus S, |S| \leq |V|/2$ and let $|V| = n, |S| = m$ and $|E(S, \bar{S})| = q$. If the flipping edges are chosen with uniform probability then a Random k -Flipper operation transforms q to q' as follows:*

$$P[q' = q - 2] \leq \left(\frac{2q}{dn}\right)^2$$

$$P[q' = q + 2] = 2\left(\frac{m}{n} - \frac{q}{dn}\right)\left(1 - \frac{m}{n} - \frac{q}{dn}\right)$$

So, for a given partition the Random k -Flipper operations describe a random drift towards a state that satisfies an expansion. However, one cannot guarantee a truly uniform choice of the flipping edges. Yet, if the random walk is long enough an approximation can be provided.

Lemma 9 *For $k \in \Theta(d^2 n^2 \log 1/\epsilon)$ the Random k -Flipper chooses the first flipping edge with uniform probability and the second edge with probability $(1 \pm \epsilon) \frac{2}{dn}$ for any $\epsilon > 0$.*

PROOF. Consider the conductance Φ of a d -regular graph defined by

$$\Phi := \min_{S \subset V, S \neq \emptyset} \frac{n|\partial S|}{d|S| \cdot |\bar{S}|}.$$

As a lower bound for the conductance of connected graphs we have

$$\Phi \geq \min_{m \in \{1, \dots, n-1\}} \frac{n}{dm(n-m)} = \frac{4}{dn}.$$

According to Lovász [13] the second eigenvalue λ_2 of the Markov process is bounded by

$$\frac{\Phi^2}{8} \leq 1 - \lambda_2 \leq \Phi,$$

which implies a bound of $\lambda_2 \leq 1 - \frac{2}{d^2 n^2}$. Let $P^t(j)$ denote the probability that a random walk ends at j after t rounds. Then, this implies

$$\left| P^t(j) - \frac{1}{n} \right| \leq \left(1 - \frac{\Phi^2}{8} \right)^t \leq \left(1 - \frac{2}{d^2 n^2} \right)^t.$$

So, after $d^2 n^2 \log 1/\epsilon$ rounds in all graphs this term is smaller than ϵ . This proof follows the ideas presented in [13]. \square

This implies the following Lemma.

Lemma 10 *Consider a bisection of a graph $G = (V, E)$ into the sets $S \subset V$ and $\bar{S} = V \setminus S$, $|S| \leq |V|/2$, with $|V| = n, |S| = m$ and $|E(S, \bar{S})| = q$. Applying a Random k -Flipper operation with $k \in \Theta(d^2 n^2 \log 1/\epsilon)$ and $\epsilon \in (0, \frac{1}{8})$ the number of edges of the cut is changed from q to q' such that*

$$\begin{aligned} P[q' = q - 2] &\leq 4(1 + \epsilon)\alpha^2 \\ P[q' = q + 2] &\geq 2(1 - \epsilon)(\beta - \alpha)(1 - \beta - \alpha) \end{aligned}$$

with $\alpha = \frac{q}{dn}$ and $\beta = \frac{m}{n}$.

PROOF. The proof follows by adapting the approximation bound $(1 \pm \epsilon)$ of Lemma 9 to Lemma 8. \square

Note that for $\alpha \ll \beta$ we observe a random walk with a strong drift over the number of edges. The probability that two edges are added is larger than the probability that two edges are removed. So, we can reduce the analysis to a random walk with a drift and use the following lemma.

Lemma 11 *Consider a random walk on the set of numbers $\{1, \dots, B\}$ with transition probability $2p$ from i to $i + 1$ for $i < B$ and probability p from i to $i - 1$ for $i > 1$. The probability to remain in state i is $1 - 3p$ for $i \in \{2, \dots, B - 1\}$, $1 - 2p$ for state 1 and $1 - p$ for state B .*

For any $c > 0$ there exists c' such that after $c'B/p$ rounds the probability that the random walk ends within the set $\{1, \dots, B - t\}$ is at most $2^{-cB} + 2^{-t+1}$.

PROOF. In the worst case the random walk starts at position 1. The expected distance the random walk upwards within k rounds is pk . If we choose $k = c'B/p$ we can apply Chernoff bounds to prove that with probability $1 - 2^{-cB}$ the position B is reached at least once.

Now consider the stationary distribution, when the Markov process has converged. Let P_t be the probability for state $B - t$. For $t > 1$ we have the recursion

$$P_t = 2pP_{t+1} + (1 - 3p)P_t + pP_{t-1}.$$

This recursion is satisfied for $P_t = \gamma 2^{-t-1}$. Furthermore, this implies: $P_0 = 2P_1 = \gamma/2$. Summing up over all positions and using $P_{B-2} = 2P_{B-1}$ we have $\gamma = \frac{1}{1 - 2^{-B-1}}$ and the probability that the random walk is in the interval $\{1, \dots, B - t\}$ after reaching B is at most 2^{-t+1} , which implies the claim. \square

Now we are able to prove the fast convergence of the Random k -Flipper.

Theorem 3 *If we choose $d \in \Omega(\log n)$ applying $\mathcal{O}(dn)$ Random $\Theta(d^2 n^2 \log 1/\epsilon)$ -Flipper operations transforms any given d -regular connected graph into a connected d -regular graph with expansion $\Theta(d)$ with high probability.*

PROOF. We will prove an expansion of at least $d/16$. For this we consider all sub-sets $S \subset V, S \neq \emptyset$ with $|S| = m \leq n/2$. Now we apply Lemma 10 with a constant choice for ϵ for a bisection where the edge set is at most twice than the expansion, i.e. $q \leq md/8$. Then $\alpha = \frac{q}{dn} \leq \frac{m}{8n} = \frac{1}{8}\beta$ and $\beta \leq \frac{1}{2}$. This implies for $\epsilon = \frac{1}{8}$:

$$\begin{aligned} 2(1 - \epsilon)(\beta - \alpha)(1 - \beta - \alpha) &\geq 2(1 - \epsilon)\frac{7}{8}\beta \left(1 - \frac{9}{8}\beta \right) \\ &\geq \frac{63}{64}(1 - \epsilon)\beta \geq \frac{63}{32}(1 - \epsilon)\beta^2 \geq 126(1 - \epsilon)\alpha^2. \end{aligned}$$

Again let q' denote the number of edges of the cut after applying one Random k -Flipper operation. Then, we have

$$P[q' = q + 2] \geq 94P[q' = q - 2] \text{ and } P[q' = q + 2] \geq \frac{1}{2} \frac{m}{n}.$$

So, for the number of boundary edges we observe a random walk with a drift. The process that we have studied in Lemma 11 gives an upper bound on the number of Random k -Flipper operations for $p = \frac{m}{4n}$ and $B = \frac{md}{4}$ (we spare a factor of two since we walk two steps in a round). Hence, with probability $1 - (2^{-cB} + 2^{-\frac{1}{2}B})$ the number of edges is at least $\frac{md}{16}$ within this cut after $c'B/p \leq c' \frac{md}{4} \frac{4n}{m} \leq c'dn$ rounds of Random k -Flipper operations.

Let $c \geq \frac{1}{2}$. It remains to sum up all error probabilities $2^{-cB} + 2^{-\frac{1}{2}B} \geq 2^{-\frac{1}{2}B} + 2^{-\frac{1}{2}B} \geq 2 \cdot 2^{-\frac{md}{4}}$ of all sub-sets $S \subset V$ of size $m \leq n/2$ for some $d = k \log n$ with $k > 8$.

$$\begin{aligned} 2 \sum_{m=1}^{n/2} \binom{n}{m} 2^{-\frac{1}{4}km \log n} &\leq 2 \sum_{m=1}^{n/2} n^m 2^{-\frac{1}{4}km \log n} \\ &\leq 2 \sum_{m=1}^{n/2} 2^{(1 - \frac{1}{4}k)m \log n} \leq 2 \sum_{m=1}^{n/2} n^{(1 - \frac{1}{4}k)m} \\ &\leq 2 \frac{n}{2} n^{1 - \frac{1}{4}k} \leq n^{-k'} \end{aligned}$$

\square

Possible Improvements

As noted above the Random k -Flipper is not symmetric, i.e. the transition probability from graph G to G' may differ from the transition probability from G' to G . Therefore, it is an open problem if the Random k -Flipper with $k > 1$ provides uniform generality. Nevertheless, slightly modified versions provide symmetry. For this, the random walk needs to avoid to traverse the flipping edges more than once. The Random k -Flipper is allowed to traverse these edges and a sub-path of the random walk will be chosen. This choice causes the break of symmetry. If we avoid visiting nodes more than once we have the *Node Disjoint Random k -Flipper* and if we avoid visiting edges more than once we get the *Edge Disjoint Random k -Flipper*. Both operations have symmetric transition probabilities. However, when using these operations long random walks are not possible, especially not paths of length $\Theta(d^2 n^2 \log 1/\epsilon)$. Furthermore, the proof of Theorem 3 cannot be applied since the random walk used in these modified versions is rather biased. So, the practical use of these symmetric versions is doubtful.

As soon as the expansion property is established one can reduce the length of the random walk of the Random k -Flipper to a polylogarithmic term. Furthermore, it maybe that such a random walk suffices for the whole procedure.

5. PEER-TO-PEER NETWORKS BASED ON RANDOM REGULAR GRAPHS

The operations introduced in the previous sections are particularly suitable for building and maintaining large distributed random networks. Such random networks are the backbone of the JXTA peer-to-peer networking suite for example.

Note that there is a similar approach by Law and Siu [11] with the same goal, yet based on a completely different construction. In a nutshell they build a d -regular random multigraph in which the set of edges is composed of $d/2$ Hamilton cycles of size n . However the probability space produced by their protocol may deviate far from the uniformly distributed space and may lead to graphs with small or no expansion and their operation cannot recover from bad graphs. The Random k -Flipper operations overcome these problems.

From now on we will talk of peers and networks instead of nodes and graphs and discuss the networking aspects like dynamics (joining and leaving peers) and problems arising with the concurrent use of Random k -Flipper operations in a distributed network. In this section we show how to maintain dynamic connected d -regular peer-to-peer networks based on random graphs with expansion property.

For this the Random k -Flipper operations are started distributedly by every peer from time to time and control messages are sent over random paths and neighbors are exchanged. This way continuously fresh randomness is added to the network and by the parallel operations the network quickly converges to an expander graph. Furthermore the network connections are validated automatically by the random walks of the k -Flipper operations.

Unlike as in the graph approach, in distributed dynamic networks it is difficult to guarantee the d -regularity, e.g. if only one edge is missing, then there are only two nodes with degree $d - 1$ while the other nodes have degree d . Finding the partner would involve a search (or a data structure)

for the whole network. Yet the benefit of having exactly d neighbors is rather small if d is at least logarithmic, what is characteristic for peer-to-peer networks. Therefore, we allow nodes to have either d or $d - 1$ neighbors and call such networks $\{d, d - 1\}$ -regular.

5.1 Joining Peers

The easiest way for a peer to get a valid neighborhood without violating the $\{d, d - 1\}$ -regularity of the network is to randomly choose $d/2$ connections defined by d distinct nodes, erase these connections and connect to each of the d nodes: "The peer places itself in the middle of these connections". The distinctness of the d peers of these connections is crucial, since otherwise the new peer would either create multiple connections to some peers or reduce the degree of some peers violating the $\{d, d - 1\}$ -regularity. So, we have to find enough random connections in reasonable time.

We assume a peer u willing to join the network knows at least one peer v within the network. Starting from v , u sends a control message on a random walk. At each node visited u selects a random neighbor v' of v and creates a lock on the connection $\{v, v'\}$ preventing other nodes from selecting this edge. Furthermore these locks prevent Flipper operations from choosing these edges as flipping edges. According to [1] the length of the random walk can be chosen as $\mathcal{O}(d^2 \log d)$. In the expectation this number of hops is sufficient to detect d distinct nodes in any graph by a random walk. If after $\mathcal{O}(d^2 \log d)$ hops not enough random connections could be found the random walk is canceled. This process can fail if either the network is too small or if the network structure is bad (which is unlikely, yet possible). A new peer which could not find enough neighbors can retry finding new neighbors by the same procedure after some time. If the control message has successfully finished the random walk reporting enough neighbors, u is contacted by enough peers of the network and can place itself in the middle of each of the collected connections.

5.2 Leaving Peers

The case of leaving peers can be divided in peers leaving intentionally and peers leaving by some kind of local or network failure. We will consider the case of nodes leaving intentionally first.

It turns out that this case is easy to handle. A leaving peer u successively selects two random peers v and v' of his neighborhood. Then v and v' are informed that u is about to leave. The peers v and v' then connect to each other and remove u from their neighborhood. Next, u removes v and v' from its neighborhood and continues with the next pair of neighbors until all neighbors are processed. This procedure ensures that the network still has degree d or $d - 1$ (provided that the network consists of enough peers).

The case of peers failing unexpectedly is more problematic. First of all, failing peers bear the problem of disconnecting the network. Classical failure analysis in peer-to-peer networks is focused on analyzing the probability that a given peer becomes disconnected [19, 12]. So that we will restrict to this case, too.

Fact 1 *Let G be a random d -regular network of size n with $d \in \Omega(\log n)$. When each peer fails with probability $1/2$, then a single peer will stay connected to the rest of the network with high probability.*

PROOF. To separate a peer from the network all of his $d \in \Omega(\log n)$ neighbors have to fail. This will happen with probability $2^{-d} \leq 2^{-c \log n} = n^{-c}$. \square

Another issue is that a failing peer reduces its neighbors degree by one. As noted above we do not want to fix this degree by choosing pairs since this causes too much communication overhead. Our solution is that a neighbor detects an edge failure while sending control information for the periodically occurring Random k -Flipper operations. If the degree is smaller than $d - 1$ then a peer uses the node joining algorithm to increase its degree by the missing number of edges, i.e. it chooses random connections of the network and places itself in the middle of these connections, thus increasing its own degree by two without changing the degree of other peers.

5.3 Concurrency

When multiple Random k -Flipper operations are applied to a graph concurrently (which is likely when the graph represents a huge network) some additional effort is necessary to guarantee connectivity. In case of the Random 1-Flipper operation there is only a small constant number of nodes involved in intersecting operations. So, these can coordinate their graph transformations in a way that no two intersecting transformations take place at the same time.

However some additional effort is necessary to guarantee connectivity in case of concurrent Random k -Flipper operations without central coordination mechanisms. We start our analysis with the following fact.

Fact 2 *The only way to divide a graph into components with Random k -Flipper operations is to destroy the hub path of a Random k -Flipper operation. Furthermore the only way to remove edges with Random k -Flipper operations is to choose these edges as flipping edges.*

In the following we will consider two concurrent Random k -Flipper operations U and V . Let u_1, \dots, u_i respectively v_1, \dots, v_j denote their truncated random walks according to the Random k -Flipper algorithm (see Algorithm 2).

Lemma 12 *The hub path of a Random k -Flipper operation U cannot be destroyed by another Random k -Flipper operation V if V does not use the flipping edges $\{u_1, u_2\}, \{u_{i-1}, u_i\}$ of U .*

PROOF. When V chooses one or both of its flipping edges to lie on the path (u_2, \dots, u_{i-1}) this will only substitute the edges to be deleted by V by a path. Thus the network will stay connected. \square

Things get more complicated when U and V interfere each other, i.e. U has an flipping edge on the path (v_2, \dots, v_j) and V has a flipping edge on the path (u_2, \dots, u_i) .² This may lead to a disconnected graph as shown in Figure 6.

However the partition of the graph does not necessarily take place since there may be other independent paths between u_1 and u_2 . While the probability of the disconnecting the graph is rather small, we aim at guaranteed connectivity. In our view this is important because on the long run

²Note that we can guarantee $u_1 \neq v_1$ since no peer will start another Random k -Flipper operation before the previous one is finished.

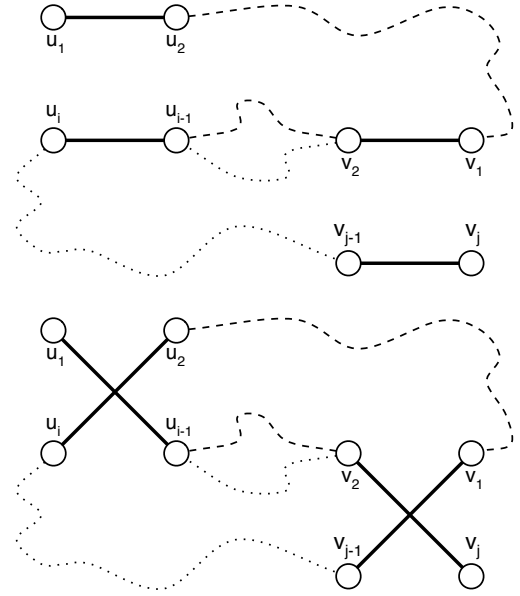


Figure 6: Concurrent Random k -Flipper operations can disconnect a graph. The graph at the top shows random walks performed by two Random k -Flipper operations U (dashed) and V (dotted), which result in the disconnected graph shown beneath.

also small probability events will surely occur – regardless of their small probability.

So, we use the following locking mechanism. Each Random k -Flipper operation U leaves a *stamp* – a representation of its starting edge – at each edge it passes. In addition U keeps track of the edges it has visited during the random walk in a list L . When the random walk is finished, it is checked whether one of the flipping edges $\{u_1, u_2\}, \{u_{i-1}, u_i\}$ has a stamp $s \in L$ on it. In this case another Random k -Flipper with starting edge on U 's random walk has already passed one of U 's flipping edges and U is risking connectivity if the edges are flipped. The solution to this is rather simple. If the last edge $\{u_{i-1}, u_i\}$ has critical stamps on it U can do another random step or choose another edge of the last but one node and check again for stamps in L . Critical stamps on the starting edge $\{u_1, u_2\}$ can be handled similar. If no suitable edges can be found the Random k -Flipper operation is canceled. To prevent the whole graph from getting locked, the stamps should be soft state and be deleted after some constant time t . This implies that also all Random k -Flipper operations have to be finished in time t or they will have to be canceled. Given this stamp mechanism the following lemma holds.

Lemma 13 *Using random walks with stamps as described above, no two concurrent Random k -Flipper operations can interfere with each other.*

PROOF. Again consider two Random k -Flipper operations U and V . For the proof we will regard the path (u_1, \dots, u_i) of U as fixed. Furthermore V has one flipping edge on the path (u_1, \dots, u_i) and thus interferes U . We assume this to be the starting edge $\{v_1, v_2\}$ of V . Note that we can restrict to this case because of symmetry. In order to make U and V

interfere with each other, V has to pass at least one flipping edge of U . When V passes one of these edges it will leave a stamp on it. This will make U to find a stamp representing an edge of the path (u_1, \dots, u_i) on it when checking the flipping edges $\{u_1, u_2\}, \{u_{i-1}, u_i\}$. Thus U will not perform the Random k -Flipper operation. \square

Combining our results the following theorem holds.

Theorem 4 *Using random walks with stamps the network is guaranteed to stay connected while applying concurrent Random k -Flipper operations.*

PROOF. This theorem follows directly from the combination of Lemma 12 and Lemma 13. \square

For peer-to-peer networks it turns out that the Random 1-Flipper is easier to handle than the Random k -Flipper with larger k . Especially if $k \in \Theta(d^2 n^2 \log 1/\epsilon)$ is chosen as in Theorem 3 then the whole network is swamped with stamps of one type. Parallel Random k -Flipper operations may block each other or in the best case are performed sequentially. For the networking point a small choice for k is highly desirable. However, little is known about the convergence rate except for the case of expander graphs. In an expander graph $k \in \mathcal{O}(\log n)$ can be used. There is hope that the parallel use of Random 1-Flipper converges as fast as the sequential use of Random k -Flipper operations since the Random 1-Flipper causes a pointer-jumping effect that boosts the performance. Experimental simulations support this theory.

6. REFERENCES

- [1] G. Barnes and U. Feige. Short random walks on graphs. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 728–737. ACM Press, 1993.
- [2] B. Bollobas. Random graphs. *Combinatorics*, 52:80–102, 1981.
- [3] B. Bollobas. The isoperimetric number of a random graph. *European Journal of Combinatorics*, 9:241–244, 1988.
- [4] C. Cooper, M. Dyer, and C. Greenhill. On markov chains for random regular graphs, 2005. Unpublished Draft.
- [5] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 120–130, Mar. 2004.
- [6] Gnutella. The gnutella protocol specification v0.4.
- [7] M. Jerrum and A. Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science*, 73(1):91–100, 1990.
- [8] M. A. Jovanovic, F. S. Annexstein, and K. A. Berman. Scalability issues in large peer-to-peer networks — a case study of Gnutella. Technical report, University of Cincinnati, 2001.
- [9] M. Kim and M. Medard. Robustness in large-scale random networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2364–2373, Mar. 2004.
- [10] M. Kim, M. Medard, and E. A. M. Torres. On reliability of large-scale random networks. *Submitted to IEEE/ACM Transactions on Networking*, 2005.
- [11] C. Law and K.-Y. Siu. Distributed construction of random expander networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 2133–2143, Mar. 2003.
- [12] D. Liben-Nowell, H. Balakrishnan, and D. Karger. Analysis of the evolution of peer-to-peer systems. In *Proceedings of the twenty-first Annual Symposium on Principles of Distributed Computing (PODC-02)*, pages 233–242, New York, July 21–24 2002. ACM Press.
- [13] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty*, 2:353–398, 1996.
- [14] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*, pages 84–95. ACM Press, 2002.
- [15] B. D. McKay. Subgraphs of random graphs with specified degrees. *Congressus Numerantium*, 33:213–223, 1981.
- [16] B. D. McKay and N. C. Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11:52–67, 1990.
- [17] L. Page. PageRank: Bringing order to the web. Stanford Digital Libraries Working Paper 1997-0072, Stanford University, 1997.
- [18] A. Steger and N. C. Wormald. Generating random regular graphs quickly. *Combinatorics, Probability and Computing*, 8:377–396, 1999.
- [19] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In R. Guerin, editor, *Proceedings of the ACM SIGCOMM 2001 Conference (SIGCOMM-01)*, volume 31, 4 of *Computer Communication Review*, pages 149–160, New York, Aug. 27–31 2001. ACM Press.
- [20] N. C. Wormald. The asymptotic connectivity of labelled regular graphs. *Journal of Combinatorial Theory, Series B*, 31(2):156–167, 1981.
- [21] N. C. Wormald. Models of random regular graphs. In *Surveys in Combinatorics, 1993, Walker (Ed.), London Mathematical Society Lecture Note Series 187*, Cambridge University Press. 1999.