Bounding the Communication Complexity of Randomized Broadcasting Algorithms in Random-Like Graphs

Robert Elsässer*

University of California, San Diego
La Jolla, CA 92093
relsaess@math.ucsd.edu

Abstract

Broadcasting algorithms have a various range of application in different fields of computer science. In this paper we study the communication complexity of simple randomized broadcasting algorithms in random-like networks. We start our analysis on the classical random graph model, i.e., a graph G_p with n nodes is constructed by letting any two arbitrary nodes be connected with probability p. First, we state some combinatorial results which are necessary for our main study. Then, we consider a modified version of the random phone call model intorduced by Karp et al. [18], and show that the communication complexity of the corresponding broadcasting algorithm is bounded by an asymptotically optimal value in almost all connected random graphs. More precisely, we show that if p exceeds some threshold, then we are able to broadcast any information r in a random graph G_p of size n within $O(\log n)$ steps by using at most $O(n \max\{\log\log n, \log n/\log d\})$ transmissions of r, where $d = pn$ denotes the expected average degree in G_p. This result holds with probability $1 - o(1/n^c)$, where c is a constant, even if n and d are unknown to the nodes of the graph.

The main result of the paper can be extended to other random graph models as well. A slight modification of our algorithm results in asymptotically optimal communication overhead for certain types of the random power law graphs defined in [5] by Chung and Lu. It is worth mentioning that such random power law graphs are often used to model large scale real world networks such as the Internet.

The algorithm we present in this paper is simple, scalable, and robust. It can efficiently handle restricted communication failures and certain changes in the size of the network. In addition, our methods and the auxiliary combinatorial results might be useful for further investigation on this field.

*On leave from the University of Paderborn, 33102 Paderborn, Germany. Partly supported by the German Research Foundation under contract EL-399/1-1.

1 Introduction

Randomized broadcasting algorithms have extensively been studied in various network topologies. Such algorithms naturally provide robusness, simplicity and scalability. As an example, consider the so-called push model [8]: In a graph $G = (V, E)$ we place at some time t an information r on one of the nodes. Then, starting with this round, each *informed* vertex forwards the information to a communication partner over an incident edge selected independently and uniformly at random. It is known that the push algorithm spreads any information within $O(\log n)$ rounds to all nodes of a random graph G_p, with probability $1 - o(1/n)$, whenever p exceeds some threshold value [14]. However, this algorithm generates $\Omega(n \log n)$ transmissions of r. Therefore, some modifications of this scheme have been developed in order to maintain the advantages of randomized broadcasting and to decrease the communication complexity or to support other distributed protocols. These variants are briefly described in Subsections 1.1 and 1.2.

1.1 Models and Motivation

The study of information spreading in large networks have various fields of application in distributed computing. Consider for example the maintenance of replicated databases in name servers in a large corporate network [8]. There are updates injected at various nodes, and these updates must be propagated to all the nodes in the network. In each step, a processor and its neighbor check whether their copies of the database agree, and if not, they perform the necessary updates. In order to be able to let all copies of the database converge to the same contents, efficient broadcasting algorithms have to be developed.

There is an enormous amout of experimental and theoretical study of broadcasting algorithms in various models and on different network topologies. Several (deterministic and randomized) algorithms have been developed and analyzed. In this paper we concentrate only on the efficiency of randomized algorithms, and study their time and communication complexity using a simple communication model.

The advantage of randomized broadcasting is its inherent robustness against several kind of failures and dynamical changes compared to deterministic schemes that either need substantially more time [15] or can tolerate only a relatively small number of faults [20]. Our intention is to develop randomized broadcasting algorithms which have the following properties:

- They can succesfully handle restricted communication failures in the network.
- They are fully adaptive and work correctly if the size or the topology of the network change slightly during the execution of the algorithm.
- The communication complexity they produce is asymptotically minimal.

When using the push algorithm, the effects of node failures are very limited and dynamical changes in the size of the network do not really affect its efficiency. However, as described above, the push algorithm produces a large amount of transmissions.

Several termination mechanisms noticing when a specific information becomes avalible to all nodes so that its transmission can be stopped were investigated. Using simple mechanisms for the push model, it is possible to bound the number of transmissions in a random graph to $O(n \log n)$.

An idea introduced in [8] consists of so called *pull transmissions*, i.e., any (informed or uninformed) node is allowed to call a randomly chosen neighbor, and to send information from the called to the calling node. These kind of transmission makes only sense if new or updated informations occur frecuently in the network so that almost every node places a random call in each round anyway. It was observed in complete graphs that if a constant fraction of the nodes has the information, then whithin $O(\log \log n)$ steps every node of the graph is informed [8, 18]. This implies that in such graphs at most $O(n \log \log n)$ transmissions are needed if the distribution of the information is stopped at the right time.

In [9], we investigated the so called *agent-based broadcasting* in random graphs. In this model, a number of agents is distributed among the nodes of a graph, and jump from one node to another by using the edges between the nodes. In each round, an agent lying on some node chooses an incident edge, uniformly at random, and switches to the adjacent node on the other end of the chosen edge. An information placed initially on one of the nodes is carried by the agents to other vertices in the graph. If an agent visits an informed node, then it becomes informed, and any node visited by an informed agent becomes informed as well. In [9] it has been shown that if n agents are distributed somehow over the nodes of a random graph G_p of size n, then any information is distributed among the nodes of the graph within $O(\log n)$ steps, with probability $1 - o(1/n)$. However, this algorithm requires $\Omega(n \log n)$ transmissions.

In this paper we are interested in randomized broadcasting algorithms with the desired properties particulary on the class of random-like graphs. The theory of random graphs was founded by Erdős and Rényi [11, 12]. They considered the elements in a probability space consisting of graphs of a particular type. The simplest such probability space consists of all graphs with n vertices and m edges, and each such graph G_m is assigned the same probability.

Another random graph model has been introduced by Gilbert in [16], in which a graph G_p is constructed by letting two pairs of vertices be connected independently and with probability p. In this paper we mainly concentrate on this random graph model, however our results also hold for the Erdős-Rényi graphs.

In order to describe large real world networks, some modifications of these random graph models has been considered. In [1, 13] it has been observed that in many real-world networks (such as the web) the degrees of the nodes have a so called power law distribution, i.e., the fraction of vertices with degree d is proportional to $d^{-\beta}$, where $\beta > 1$ is a fixed constant. In [1], the authors suggested to model complex real world networks as follows: Consider a random graph process in which vertices are added to the graph one at a time and joined to a fixed number of earlier vertices, selected with probabilities proportional to their degrees. In [3] it has been shown that the graphs constructed by this method obey a power law with $\beta = 3$.

1.2 Related Work

Most of papers dealing with randomized broadcasting analyze the runtime of the push algorithm in different graph classes. Pittel [22] proved a nice result, which shows that it is possible to broadcast an information within $\log_2(n) + \ln(n) + O(1)$ steps in a complete graph, by using the push algorithm. In [14], Feige et al. determine asymptotic optimal upper bounds for the runtime of this algorithm in random graphs, bounded degree graphs and the hypercube. Kempe et al. considered geometric networks in [19] and proved that any information is spread to nodes at distance t in $O(\ln^{1+\epsilon} t)$ steps.

As described in the previous subsection, the idea of pull transmissions has been introduced in [8]. In [18], Karp et al. combined the push and pull models, and presented a termination mechanism in order to bound the number of total transmissions by $O(n \log \log n)$ in complete graphs. It has also been shown that this result is asymptotically optimal for these kind of algorithms. They also considered communication failures and analyzed the performance of the algorithm in the case when the random connections established in each round follow an arbitrary probability distribution. This algorithm works fully distributed and the nodes are only supposed to have an estimation about the size of the network.

Concerning the field of random graphs most papers deal with the structural properties of these graphs. One of the greatest discovery of Erdős and Rényi was that many important properties of such graphs appear quite suddenly. We say that a typical element of the probability space considered above has a property Q if the probability that a random graph on n vertices has Q tends to 1 as $n \to \infty$. For the graphs G_m it has been shown that several monotone increasing properties posses a threshold function $M_0(n)$, i.e., if m grows faster than $M_0(n)$, then almost every G_m has the mentioned property, and if m grows slower than $M_0(n)$, then almost every G_m fails to have the property [11, 12]. For an excellent survey on Erdős-Rényi graphs see e.g. [2].

Chung and Lu generalized the classical random graph model in the following way: For a sequence $\mathbf{d} = (d_1, \ldots, d_n)$ let $G(\mathbf{d})$ be the graph in which edges are independently assigned to each pair of vertices (i, j) with probability $d_i d_j / \sum_{k=1}^{n} d_k$. They analyzed the connectivity, distances, and eigenvalues of these graphs for certain sequences of $\mathbf{d}$ [5, 6, 7]. It should be noted that by setting $d_1 = \cdots = d_n$, we obtain the classical G_p random graph model.

1.3 Our Results

In this paper we present an adaptive randomized broadcasting algorithm which is able to distribute an information r, placed initially on a node of a random graph G_p, to all nodes in the network within $O(\log n)$ steps by using $O(n \max\{\log\log n, \log n/\log d\})$ transmissions of r. The algorithm does not require any previous knowledge about the size or average degree of the network. It is robust against restricted communication failures or slight changes in the size of the network, and can be adapted to some generalized random graphs with certain degree distributions.

The rest of the paper is organized in three sections. In Section 2 we derive some graph theoretical results needed for the proof of the main theorem. Section 3 is devoted to the anaysis of our algorithm. The last section contains our conclusions and points to some open problems.

2 Auxiliary Combinatorial Results

In this section, we state several combinatorial results necessary for the main analysis. As described in the introduction, we primarily consider the random graph model defined as follows: Given n and p, generate graph G_p with n vertices by letting each pair be an edge with probability p, independently [2]. Here, we assume that $p \geq \frac{\ln^\delta n}{n}$, where $\delta > 2$ is a suitable constant. The choice of p implies that the graph is connected, and every node has degree $pn(1 \pm o(1))$ with high probability[1] (e.g. [2]).

In the next lemma, we deal with the distribution of the neighbors of a node v in G_p, after it has chosen t neighbors, uniformly at random, in t consecutive steps.

Lemma 1 *Let $G_p = (V, E)$ be a random graph with n nodes and assume that $p \geq \frac{\ln^\delta n}{n}$. Let a node $v \in V$ choose in any step $i < t = O(\log n)$ a neighbor w_i, uniformly at random, among all of its neighbors (we do not require that $w_i \neq w_j$ for $i \neq j$). Then, v is connected to all nodes of an arbitrary subset $S \subset V \setminus \{w_1, \ldots, w_t\}$, $|S| = o(n)$, with probability at least $p^t(1 - |S|(1 - o(1))/d))^t$, where $d = pn$ is the average degree in G_p. Moreover, for an arbitrary subset $S' \subset V \setminus \{v, w_1, \ldots, w_t\}$ with $|S'| = \Theta(n)$ it holds that $|N(v) \cap S'| \geq p|S'| - O(\sqrt{p|S'|\log n})$, w.h.p., where $N(v)$ represents the set of neighbors of v.*

Proof: We begin with the case $|S| = 1$. Let $p_{n,p,i}$ denote the probability that an arbitrary node $u \in V$ has degree i in a random graph G_p of size n, and let $S_w = \{w_1, \ldots, w_t\}$. Let w be an arbitrary node in $V \setminus (\{v\} \cap S_w)$. First we consider the distribution of the neighbors of v after one single step. Let A denote the event that v is connected to all nodes of S_w, and A_1 denote the event that v chooses w_1 in step $i = 1$. Then, it holds that

$$\Pr[A_1 \mid A] = p \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1) + (1 - p) \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w|).$$

Here, $\Pr[A_1 \mid A]$ consists of two terms. The first term expresses the fact that with probability p the vertices w and v are connected, and with probability $p_{n-1-|S_w|,p,j}$ the vertex v has exactly j neighbors

[1] When we write "with high probability" or "w.h.p.", we mean with probability at least $1 - o(1/n)$.

in the random graph formed by the vertices of $V \setminus (S_w \cap \{w\})$. Therefore, v chooses w_1 with probability $1/(j + |S_w| + 1)$. The second term handles the case when v and w are not connected. Then, with probability $p_{n-1-|S_w|,p,j}$ the vertex v has exactly j neighbors in the random graph formed by the vertices $V \setminus (S_w \cap \{w\})$, however now v chooses w_1 with probability $1/(j + |S_w|)$.

The conditional probability $p_{v,w} = \Pr[(v, w) \in E \mid A_1 \wedge A]$ satisfies the equality

$$p_{v,w} = \frac{p \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1)}{\Pr[A_1 \mid A]} \tag{1}$$

Define $e_{m,k} = \sum_{j=0}^{m-1-k} p_{m-k,p,j}/(j + k)$. We know that in a random graph G_p with n nodes, an arbitrary node has $pn \pm O(\sqrt{pn \log n})$ neighbors with some very high probability $1 - 1/n^c$, where c can be any constant value. Let $d_{\min,n}$ be $pn - c'\sqrt{pn \log n}$, and $d_{\max,n}$ be $pn + c'\sqrt{pn \log n}$, where c' is a large constant. Then equation (1) leads to the following inequality

$$\begin{aligned}
p_{v,w} &= \frac{pe_{n,|S_w|+1}}{pe_{n,|S_w|+1} + (1-p)\sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1) \cdot (j + |S_w| + 1)/(j + |S_w|)} \\
&\geq \frac{pe_{n,|S_w|+1}}{pe_{n,|S_w|+1} + \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1) \cdot (j + |S_w| + 1)/(j + |S_w|) - pe_{n,|S_w|+1}} \\
&\geq \frac{pe_{n,|S_w|+1}}{1/n^{c-1} + \sum_{j=d_{\min,n-1-|S_w|}}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1) \cdot (j + |S_w| + 1)/(j + |S_w|)} \\
&\geq \frac{pe_{n,|S_w|+1}}{1/n^{c-1} + \sum_{j=d_{\min,n-1-|S_w|}}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1) \cdot (1 + 1/(d_{\min,n-1-|S_w|} + |S_w|))} \\
&\geq \frac{pe_{n,|S_w|+1}}{e_{n-1,|S_w|+1} \cdot (1 + 1/(d_{\min,n-1-|S_w|} + |S_w|)) + 1/n^{c-1}} \geq p(1 - (1 - o(1))/d).
\end{aligned}$$

We use similar techniques to derive the result for some $t = O(\log n)$. Let A_t denote the event that v chooses w_i in step i for any $i \leq t$. Then,

$$\Pr[A_t \mid A] = p \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1)^t + (1-p) \sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w|)^t.$$

Again, in the first term we consider the case when $(v, w) \in E$, and if v has exactly j neighbors in the random graph formed by the vertices $V \setminus (S_w \cup \{w\})$, then the probability that v chooses w_1 in the first step, w_2 in the second step, and generally w_i in the ith step for any $i \leq t$ is $1/(j + |S_w| + 1)$. The second term differs from the first term by the lack of the edge between v and w.

Define $e^{(t)}_{m,k} = \sum_{j=0}^{m-1-k} p_{m-k,p,j}/(j + k)^t$. The conditional probability $p^{(t)}_{v,w} = \Pr[(v, w) \in E \mid A_t]$ satisfies the inequality

$$\begin{aligned}
p^{(t)}_{v,w} &= \frac{pe^{(t)}_{n,|S_w|+1}}{pe^{(t)}_{n,|S_w|+1} + (1-p)\sum_{j=0}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1)^t \cdot (j + |S_w| + 1)^t/(j + |S_w|)^t} \\
&\geq \frac{pe^{(t)}_{n,|S_w|+1}}{1/n^{c-1} + \sum_{j=d_{\min,n-1-|S_w|}}^{n-2-|S_w|} p_{n-1-|S_w|,p,j}/(j + |S_w| + 1)^t \cdot (j + |S_w| + 1)^t/(j + |S_w|)^t} \\
&\geq \frac{pe^{(t)}_{n,|S_w|+1}}{e^{(t)}_{n,|S_w|+1} \cdot (1 + 1/(d_{\min,n-1-|S_w|} + |S_w|))^t + 1/n^{c-1}} \geq p(1 - (1 - o(1))/d)^t.
\end{aligned}$$

Obviously, the occurance of the edges between v and two arbitrary nodes of $V \setminus (S_w \cup \{v\})$ are not independent from each other. Therefore, we compute the probability $\Pr[(v,v') \in E \text{ for all } v' \in S]$ for an arbitrary subset $S \subset V \setminus (S_w \cup \{v\})$ with $l = o(n)$ vertices. Let $S = \{s_1, \dots, s_l\}$. We know that v is connected to all vertices of S in G_p with probability p^l. Then,

$$\begin{aligned}\Pr[A_t] &= p^l \sum_{j=0}^{n-1-|S_w|-l} p_{n-l-|S_w|,p,j}/(j+|S_w|+l)^t \\ &\quad + \sum_{k=0}^{l-1} \binom{l}{k} p^k (1-p)^{l-k} \sum_{j=0}^{n-1-|S_w|-l} p_{n-l-|S_w|,p,j}/(j+|S_w|+k)^t.\end{aligned}$$

Now, if $p_{v,S}^{(t)} = \Pr[(v,v') \in E \text{ for all } v' \in S \mid A_t]$, then

$$\begin{aligned}p_{v,S}^{(t)} &= \frac{p^l \sum_{j=0}^{n-1-|S_w|-l} p_{n-l-|S_w|,p,j}/(j+|S_w|+l)^t}{\Pr[A_t]} \\ &\geq \frac{p^l e_{n,|S_w|+l}^{(t)}}{p^l e_{n,|S_w|+l}^{(t)} + \sum_{k=0}^{l-1} \binom{l}{k} p^k (1-p)^{l-k} \sum_{j=0}^{n-l-|S_w|} p_{n-l-|S_w|,p,j}/(j+|S_w|)^t} \\ &= \frac{p^l e_{n,|S_w|+l}^{(t)}}{p^l e_{n,|S_w|+l}^{(t)} + (1-p^l) \sum_{j=0}^{n-l-|S_w|} p_{n-l-|S_w|,p,j}/(j+|S_w|)^t} \\ &\geq \frac{p^l e_{n,|S_w|+l}^{(t)}}{\sum_{j=0}^{n-l-|S_w|} p_{n-l-|S_w|,p,j}/(j+|S_w|)^t}\end{aligned}$$

Using similar techniques as in the previous case, we obtain $p_{v,S}^{(t)} \geq p^l(1 - l(1+o(1))/(d+l))$.

In order to show the second statement of the lemma we assume that S' is an arbitrary subset of $V \setminus (S_w \cup \{v\})$ with $\Theta(n)$ nodes. Let $S'' = V \setminus (S' \cup S_w \cup \{v\})$, and let $q_{S',j}$ denote the probability that there are exactly j neighbors of v in S'. Then,

$$\Pr[A_t] = \sum_{j=0}^{|S'|} q_{S',j} \sum_{j'=0}^{|S''|} q_{S'',j'}/(|S_w|+j+j')^t.$$

Let $A_{S'}$ be the event that v has less than $d_{\min,|S'|+1} = pn - c'\sqrt{pn \log n}$ neighbors of in S', where c' is a large constant. Then,

$$\Pr[A_{S'} \mid A_t] = \frac{\sum_{j=0}^{d_{\min,|S'|+1}} q_{S',j} \sum_{j'=0}^{|S''|} q_{S'',j'}/(|S_w|+j+j')^t}{\Pr[A_t]}.$$

Since $t = O(\log n)$, it holds that $\Pr[A_t] = (\Omega(1/d))^t = \Omega(1/(n^\iota d_{\min,|S'|+1}^t))$, where ι is a constant. Now, if c' is large enough, then it holds that $\sum_{j=0}^{d_{\min,|S'|+1}} q_{S',j}/(j+|S_w|) \leq \omega(1/(n^{\iota+3}(d_{\min,|S'|+1}+|S_w|)^t))$. Therefore, $\Pr[A_{S'} \mid A_t] \leq o(1/n^2)$ and the lemma follows. □

In the following lemma we consider the propagation of information in a random graph using the push model. We show that if the number of informed vertices $|I(t)|$ at some time t is less than $n/\log^2 n$, then each informed vertex has at most $O(p|I(t)| + \log n)$ neighbors in $I(t)$, w.h.p.

Lemma 2 *Let $G_p = (V, E)$ be a random graph with $n = |V|$ and $p \geq \frac{\ln^\delta n}{n}$. Let an information be placed at some node $v \in V$ at time t. In the succeeding steps, any informed node chooses a neighbor in each step, uniformly at random, and transmits the information to this node until $|I(t)| \leq n/\log^2 n$. Then, for every informed node v it holds that $|N(v) \cap I(t)| = O(p|I(t)| + \log n)$, where $N(v)$ denotes the set of nodes adjacent to v in G_p.*

3 Information Spreading in Random Graphs

In this section, we analyze the behavior of a modified push & pull algorithm on random graphs. Let $G_p = (V, E)$ be a random graph, where we assume that $p = \frac{\log^\delta n}{n}$, where $\delta > 2$. As mentioned in the introduction, even though we consider applications in which informations are constantly generated by almost every node, we focus on the distribution and lifetime of a single information. The modified push & pull algorithm is described in the following paragraphs.

In each round, each node u chooses a communication partner v from V uniformly at random. In some round t, any information can be exchanged in both directions along the edges between two communication partners. Whenever a connection is established between two nodes, each one of them has to decide whether to transmit the specific information to the other node, without knowing if the vertex at the other end of the edge has already received the information prior this step. Concerning the flow of information we distinguish between push and pull transmissions.

The size of information exchanged in any way is not limited and each information exchange between two neighbors in a round is counted as a single transmission. In addition to the information itself, each node sends its node id and a constant number of other messages related to the information. At the beginning, we initialize at each node u a table $T[c_{\max}]$ of constant length $c_{\max}$, for storing at most $c_{\max}$ node ids, the integers *age*, which describes the age of the information, *itime*, i.e. the last time step (known to u) in which a node was newly informed in the system, and a counter $ctr = 0$. The age is incremented in every succeeding round, by each informed node, and distributed together with the information and the integer *itime*. It should be noted that *itime* and *ctr* are local variables and may differ from node to node.

Let r denote the information we consider. During the execution of the algorithm, each node can be in one of the states U (uninformed), A (active), G (going down), or S (sleeping). If a node is in state U, it means that it has not received the information yet. In all other states the node knows r. Every node being in one of the states A or G transmits the information according to the rules described below. If a node u in state U receives the information, then it switches to state A and sets the counter *itime* to the actual age of r. We consider now the nodes in each of the cases described before. Assume that we are able to compute efficiently an estimate of the value $\tau = \log n / \log d$ (the corresponding algorithm is given later in this section). Let u and v be two nodes which are communicating with each other in some step t. Then,

- if u is in state A and receives r from v, then it checks whether the id of v is stored in T. If v's id is not in T, then *ctr* is incremented by 1 and v's id is stored in T. Otherwise u does not modify *ctr* or T. However, in both cases, u checks whether the received *itime* is larger than its own *itime*, and if so, then u updates *itime*. If r is not received from v, then T is cleaned and *ctr* is set to 0. If *ctr* achieves the value $c_{\max}$, then u switches to state G.

- if u is in state G, then it does not longer consider T and *ctr*. However, the received *itime* is still checked and, if necessary, updated. If *age* achieves $itime + \alpha \max\{\ln itime, \tau\}$, where α is a large constant, then u switches to state S.

- if u is in state S, and receives r from some neighbor v along with an *itime* such that $age < itime + \alpha \max\{\ln itime, \tau\}$, then u switches into state G.

Let $I(t)$ denote the set of informed nodes at time t. The set of uninformed nodes is denoted by $H(t) = V \setminus I(t)$.

In the sequel we show that by using this algorithm, broadcasting can be performed very fast, even if the nodes do not have any information about the size of the network. However, we should mention again that pull transmissions only make sense if new pieces of information (or updates) occur frequently so that almost every node places a random call in each round anyway.

In order to show that the algorithm described above is able to spread an information among all nodes of a graph G_p within $O(\log n)$ rounds, and the number of total messages is bounded by $O\left(n \max\left\{\log\log n, \frac{\log n}{\log d}\right\}\right)$, we assume that, as long as $I(t) \leq n/2$, only push transmissions are performed. When $I(t) \geq n/2$, then the information is transmitted only by pull transmissions. These assumptions simplify the proof, and there is only a difference in a constant factor between the runtime or communication complexity in this modified version and the original algorithm. We omit the details due to space limitations.

Lemma 3 *Let $I(t)$ be the set of informed nodes in G_p at time t. Let us assume that $|I(t)| \leq q\log n$, where q is a properly chosen constant value. Then, within $O(\log n)$ steps the number of informed nodes will exceed the value $q\log n$ with probability $1 - o(1/n)$.*

Now we consider the case when $q\ln n \leq |I(t)| \leq n/n^{4\max\{\log n/\log d, \log\log n\}}$.

Lemma 4 *Let $I(t)$ be the set of informed nodes in G_p at time $t = O(\log n)$, and assume that $q\ln n \leq |I(t)| \leq n/2^{4\max\{\log n/\log d, \log\log n\}}$, where q is the constant defined in Lemma 3. We also assume that the number of active nodes $|I_a(t)|$ before step $t+1$ is at least $|I(t)|(1 - O(t/\log^{\delta-1} n))$. Then, a constant c exists such that $|I(t+1)| \geq |I(t)|(1+c)$ and $|I_a(t+1)| \geq |I(t+1)|(1 - O((t+1)/\log^{\delta-1} n))$ with probability $1 - o(1/n^2)$.*

In the following lemma we consider the case when $I(t) \in [n/2^{4\max\{\log n/\log d, \log\log n\}}, n/2]$.

Lemma 5 *Let $I(t)$ be the set of informed nodes in G_p at time $t = O(\log n)$, and assume that $|I(t)| \in [n/2^{4\max\{\frac{\log n}{\log d}, \log\log n\}}, n/2]$. Then, there exists a constant c such that $|I(t+1)| \geq |I(t)|(1+c)$ with probability $1 - o(1/n^2)$. Moreover, most vertices of $I(t)$ are either in state A, or in state G with $itime = \Theta(\log n)$.*

As mentioned above, after informing more than $n/2$ nodes, we only count the pull transmissions in the network. Then, we can state the following lemma.

Lemma 6 *Let $|H(t)| \in [n/\sqrt{d}, n/2]$ be the number of uninformed nodes in G_p at some time t. Then, $|H(t+1)| \leq |H(t)|^2(1+o(1))/n$.*

The next two lemmas deal with the case $|H(t)| \leq n/\sqrt{d}$.

Lemma 7 *Let $|H(t)| \in [q\ln n, n/\sqrt{d}]$ be the number of uninformed nodes in G_p at some time t, where q is the constant defined in Lemma 3. Then, $|H(t+1)| \leq |H(t)|O(\log n/d)$.*

Lemma 8 *Let $|H(t)| \leq q\ln n$ be the number of uninformed nodes in G_p at time t. Then, an arbitrary uninformed node becomes informed with probability $O(\log n/d)$.*

The results of Lemma 3-8 imply that in a random graph G_p with $p \geq \log^\delta n/n$, the algorithm presented at the beginning of this section informs every node of G_p within time $O(\log n)$ by using $O\left(n \max\left\{\log\log n, \frac{\log n}{\log d}\right\}\right)$ transmissions of r. However, we assumed that the nodes are able to determine an estimate of $\log n/\log d$. Now we present a method which allows most nodes to determine the desired estimate without substantially increasing the runtime or the communication complexity.

Let any node w choose a neighbor, uniformly at random, in steps 0 and 1, respectively. Then, we let it compare the neighbors id, chosen in step 1, with the id of the neighbor reached in step 0. If the two ids at some node w are the same, then w sends out a special information r_w. These special messages will perform random walks in the system and some node $w' \in V$ checks how many of these messages are lying on it at some time $t = q' time_{w'}(r)$, where q' is a large constant and $time_{w'}(r)$ denotes the time when w' has got r. Now, if $time_{w'}(r)$ is large enough (i.e., $time_{w'}(r) = \Omega(\log n)$), then any such message lies on w' with probability $1/n(1 \pm o(1/n^2))$ [2, 10]. Combining the results of [10] with [21] (Lemma 2.13), it holds that if $time_{w'}(r)$ is large enough, then some nodes of G_p have an estimate of $\log n / \log d$.

Now we modify the algorithm described at the beginning of this section so that some nodes compute an estimate on $\log n / \log d$ during the algorithm proceeds and broadcast the information to the other nodes. Then, almost all nodes will use this value to transmit the message while being in state G.

Each node performs now three phases. In the first phase, when a node u gets r, then it sets $time_u(r)$ to the current age of r, and the value $time_u(r)$ will never be updated again. Each node u executes the algorithm described at the beginning of this section, whereby we introduce the following small modification: When u switches to state G, then *itime* will never be updated at u again (in the first phase). This modification implies that u cannot switch from state S to state G in this first phase.

For the second phase, define $s_i = 2^i$, $i \in \{0, \dots, q' \log\log n\}$. We call two numbers, j_1 and j_2, $s2$-equivalent, and denote $j_1 \sim_{s_2} j_2$, if an i exists such that $s_i \leq j_1, j_2 \leq s_{i+1}$. Two nodes w' and w'' are called $s2$-equivalent if $time_{w'}(r)$ and $time_{w''}(r)$ are $s2$-equivalent.

Now, each node u checks while being in states A and G (in the first phase), whether another 5 different nodes exist, which are $s4$-equivalent with him, have not been informed by u, and transmit r to u. If u have seen 5 such nodes when it switches to state S, then it checks after $\alpha \cdot itime$ additional steps the number of r_w's on it. Let τ'' be this number. If $\tau'' > \log itime$, then the node switches in step $8\alpha\lfloor time_u(r)/4 \rfloor + h$ to the special state R (if it is not already in this state) and to state A, updates *itime*, sets $\tau' = \tau''$, and starts to transmit r as in the first phase (along with *age* and *itime*), together with τ'. Here, h depends on τ' and will be defined later. If a node being in state R and A switches to state G, then it transmits r as long as $age < itime + \alpha\tau'$. As in the first phase, *itime* is not updated after a node switches to state G, excepting the case described below.

We should mention that a node being in the special state R is, apart from this special state, in one of the states A, G, or S. If a node u not being in state R receives r and some value τ' from a node which is in state R, then u switches to R and A, sets its own τ' to the received τ', and *itime* is updated to the actual time. As *itime* and *ctr*, τ' and τ'' are also local variables which may be different from node to node.

During the second phase, the received τ' is always compared to the own τ' and, if necessary, the own τ' is updated. Moreover, if τ' has to be reset, and the own τ' is not $s2$-equivalent with the received τ', then the own *itime* is reset to the actual time.

As mentioned above, the value of h depends on τ'. Define $h_1 = \lfloor time_u(r)/4 \rfloor$ for a node u. If $\tau' \in [\log h_1 + 1, 2\log h_1]$ then $h = 0$. For $\tau' \in [2\log h_1 + 1, 4\log h_1]$ we set h to $\alpha h_1/\log h_1$, and generally, if $\tau' \in [2\log h_1 + 1, 2^{i+1}\log h_1]$, then $h = i\alpha h_1/\log h_1$. We further assume that the value h_1 is also transmitted in state R and every node keeps only the largest h_1 value ever transmitted to it.

The third phase begins at time $16\alpha h_1$ for every node being in state R. At this time, the nodes being in state R switch to the special state R' and A, and run the algorithm as described at the beginning of this section, i.e., *itime* is updated in state G, if the received τ' is larger than and not $s2$-equivalent with the own τ', **or** the τ's are $s2$-equivalent and the received *itime* is larger than the own *itime*. Now, a node is in state G as long as $age < itime + \tau'$. Here, τ' represents the own τ' value of the node which may be updated according to the rules described above.

In the next lemma we show that any node switching to state R on his own has $itime = \Theta(\log n)$.

Lemma 9 *Any node, which checks the number of r_w's on itself, has $itime = \Theta(\log n)$, w.h.p. Moreover, there exists such a node in G_p (w.h.p.).*

Proof: As in the proof of Lemma 2, we consider two cases. Let us first assume that $p < \sqrt{n}/n$, and let t' be the largest integer such that $|I(t')| < \sqrt[4]{n}$. Then, using the Chernoff bounds [4, 17] as in the proof of Lemma 2, it can be shown that, with probability $1 - o(n^2)$, none of the first $|I(t')|$ nodes will have more than 4 informed neighbors at time t', apart from the vertices informed by the node itself, or the vertex which informed the node.

In the second case let $p \geq \sqrt{n}$, and let t' be defined as before. Then, the probability that a node chooses 4 times an informed neighbor before step t' is $O(1/n^3)$. Thus, with probability $1 - o(1/n^2)$, there does not exist any vertex which is informed before step t' and checks the number of r_w's on itself.

In order to show the second statement of the lemma, let t' be the smallest integer with $I(t') \geq n/2$. Then, with constant probability, a node in $I(t)$ with some $t > t'$ will be contacted in step $t+1$ by an informed node which has not been contacted by this node before. This implies that, with high probability, a constant fraction of the nodes will check the number of r_ws, and the lemma follows. □

The next lemma deals with the distribution of the vertices in state R.

Lemma 10 *There are at most $O(n/\log n)$ nodes in state R which transmit r for more than $O(\log n/\log d)$ steps.*

Proof: For simplicity we assume that any node which checks the number of r_ws on it has the same h_1. Then, two nodes being $s2$-equivalent switch at the same time into state R, if they have not been contacted by R nodes before. Let τ_1 and τ_2 be two integers in the range $[\log\log n, \log n/\log d]$ such that $\tau_1 \not\sim_{s2} \tau_2$. Assume w.l.o.g. that $\tau_1 < \tau_2$. Then, due to the description of the algorithm, when $\Theta(n/\log n)$ nodes are in state R and have $\tau' \sim_{s2} \tau_1$, then at most n^ϵ with $\epsilon < 1$ nodes have $\tau' \sim_{s2} \tau_2$.

Since *itime* is not updated at a node after the node switches into state G (excepting when τ' has to be significantly increased), $n - O(n/\log n)$ nodes switch into state G, for each range of τ' defined above, after being active for at most $O(\tau')$ rounds. Hence, most nodes are transmitting for $\sum_{i=0}^{\log O\left(\frac{\log n}{\log\log n \cdot \log d}\right)} 2^i \log\log n = O(\log n/\log d)$ steps in state R. □

In the next lemma we show that $n - O(n/\sqrt{d})$ informed nodes will simultaneously be in state R' for $\Omega(\log n/\log d)$ consecutive steps, whenever $\log n/\log d > \gamma \log\log n$, where γ is a large constant.

Lemma 11 *Let G_p be a random graph with $p \geq \log^\delta n/n$ and let the three-phase algorithm be performed for this graph. If $\log n/\log d > \gamma \log\log n$, then, with probability $1 - 1/n^{\Omega(1)}$, there are $n - O(n/\sqrt{d})$ informed nodes which transmit r simultaneously for $\Omega(\log n/\log d)$ consecutive steps. However, $n - O(n/\log n)$ nodes transmit r for at most $O(\log n/\log d)$ steps, and all nodes will switch to state G and R' after a total number of $O(\log n)$ steps.*

Proof: Again, we assume for simplicity that all nodes which check the number of r_ws on them have the same h_1 value. Lemma 3-6 imply that the algorithm described at the beginning of this section, in which we allow any node to transmit for $O(\log\log n)$ steps in state G, informs all but $O(n/\log^4 n)$ nodes in G_p, w.h.p. Furthermore, since *itime* is not updated in state G, at least $n - O(n/\log^4 n)$ nodes will transmit for at most $O(\log\log n)$ steps in the first phase.

Let $\tau_{\max}$ be the maximal number of r_ws occuring on a node in the system, and let u be the node with the smallest *itime* checking the number of r_ws lying on itself. Then, with probability $1 - 1/n^{\Omega(1)}$, there is a node u' with $\lfloor time_u(r)/4 \rfloor = \lfloor time_{u'}(r)/4 \rfloor$, which checks the number of r_ws on itself, and sets $\tau'' \geq \tau_{\max}/4$.

Clearly, there are $n - O(n/\log n)$ nodes which set their $\tau' \geq \tau_{\max}/4$ while being in state R. All these nodes switch into state R', and start to transmit r at time $16\alpha h_1$. We can now apply Lemma 5-8, and conclude that after $O(\log n/\log d)$ steps all vertices have r, and are in state R' with $\tau' \geq \tau_{\max}/4$. Then, the vertices will switch to state G after additional $O(\log n/\log d)$ steps.

The description of the algorithm implies that the difference between the first and last *itime*'s occuring in state R' is at most $O(\log n/\log d)$. Since $\tau' = O(\log n/\log d)$ the lemma holds. □

Now we can summarize the results in the following theorem.

Theorem 1 *Let G_p be a random graph with $p \geq \log^\delta n/n$. The three-phase algorithm informs all nodes of the graph within $O(\log n)$ steps, whereby the communication complexity is bounded by $O\left(n \max\left\{\log\log n, \frac{\log n}{\log d}\right\}\right)$, with probability $1 - 1/n^{\Omega(1)}$.*

Combining the techniques of [18] with the methods used in the proof of Lemma 1, it can be shown that the bound of Theorem 1 is asymptotically tight. We omit the details here due to space limitations.

The results of Theorem 1 can easily be generalized to the other traditional random graph model [2]: Given n and m, let each graph with n vertices and m edges occur with probability $\binom{N}{m}^{-1}$, where $N = \binom{n}{2}$. The random variable $G_{n,m}$ represents a graph generated in this way. If $m = n \ln^\delta n$ with $\delta > 1$, then the results described in this section also hold for $G_{n,m}$.

The result of this section can be extended to the case of faulty nodes as well. Let us assume that an adversary can specify up to F node failures during the execution of the algorithm, however, we do not allow the adversary to choose more than a constant fraction of the neighbors of any node in the network. Then, it can be shown that we can inform all but $O(F)$ nodes in the system within $O(\log n)$ steps using at most $O(n \max\{\log\log n, \log n/\log d\})$ transmissions. In order to show this result, we need further combinatorial tools which are omitted here due to space limitations.

The algorithms presented in this paper can also be used to broadcast an information in a slightly modified version of the power law graphs introduced by Chung and Lu in [5]. The graphs we consider are defined in the following way: Let $\mathbf{d} = \{d_1, d_2, \ldots, d_n\}$ be a sequence of degrees in a graph $G(\mathbf{d}) = (V, E)$ with n nodes. Two nodes $i, j \in V$ share an edge with probability $d_i d_j / \sum_{k=1}^n d_k$. We further assume that the degree sequence has the property $d_i = \psi(i + i_0)^{-1/(\beta-1)}$, where $\psi = d_{\min}\frac{\beta-2}{\beta-1} d n^{1/(\beta-1)}$, $i_0 = n\left(\frac{d(\beta-2)}{(d_{\max}-d_{\min})(\beta-1)}\right)^{\beta-1}$, $d_{\max}$ is the maximum vertex degree, $d_{\min}$ is the minimum degree, and $\beta > 2$ is a constant. Then, the degree sequences satisfy a certain power low distribution, i.e., the fraction of vertices with degree $kd_{\min}$ is proportional to $k^{-\beta}$. Setting now $d_{\min} > 2\log n$ we can ensure that the graph is almost always connected.

A slight modification of the algorithms described here broadcast an information to all nodes of a graph $G(\mathbf{d})$ within $O(\log n)$ steps by using at most $O(n \max\{\log\log n, \log n/\log d_{\min}\})$ transmissions, with probability $1 - o(1/n^{\Omega(1)})$. We omit the details due to space limitations.

4 Conclusion

In this paper, we analyzed the performance of randomized broadcasting algorithms in random-like graphs. First we stated some combinatorial results necessary for the proof of the main theorem. Then, we have shown that by using a modification of the random phone call model described in [18], we are able to broadcast a message to all nodes of a random graph G_p within $O(\log n)$ steps by using $O(n \max\{\log\log n, \log n/\log d\})$ transmissions, with probability $1 - 1/n^{\Omega(1)}$, whenever the probability p exceeds a treshold. We have also shown that this result holds even if the nodes do not have any knowledge about the size or degree of the network.

However, our methods only work if the probability $p \geq \log^\delta n/n$, where $\delta > 2$. A careful inspection of our lemmas would also yield the main result for a graph G_p with $p \geq \delta \log n/n$, where δ is, in this case, a very large constant. In the case when $p \geq (2 + \Theta(1))\log n/n$ but $p = O(\log n/n)$, the graph is still connected with high probability, however, we cannot apply the techniques of this paper to achieve the desired result.

References

[1] A. Barabási and R. Albert. Emergence of scaling in random networks. *Sience*, 286, 1999.

[2] B. Bollobas. *Random Graphs*. Academic Press, 1985.

[3] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18:279–290, 2001.

[4] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, 1952.

[5] F.R.K. Chung and L. Lu. Connected components in random graphs with given expected degre sequences. *Annals of Combinatorics*, 6:125–145, 2002.

[6] F.R.K. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Internet Mathematics*, 1:91–114, 2003.

[7] F.R.K. Chung, L. Lu, and V. Vu. Eigenvalues of random power law graphs. *Annals of Combinatorics*, 7:21–33, 2003.

[8] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. of PODC'87*, pages 1–12, 1987.

[9] R. Elsässer, U. Lorenz, and T. Sauerwald. Agent based information handling in large networks. In *Proc. of MFCS'04*, pages 686–698, 2004.

[10] R. Elsässer and B. Monien. Load balancing of unit size tokens and expansion properties of graphs. In *Proc. of SPAA'03*, pages 266–273, 2003.

[11] P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen 6*, pages 290–297, 1959.

[12] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci. 5*, pages 17–61, 1960.

[13] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Comput. Commun. Rev.*, 29:251–263, 1999.

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

[15] L. Gasieniec and A. Pelc. Adaptive broadcasting with faulty nodes. *Parallel Computing*, 22:903–912, 1996.

[16] E.N. Gilbert. Random graphs. *Ann. Math. Statist. 30*, pages 1141–1144, 1959.

[17] T. Hagerup and C. Rüb. A guided tour of chernoff bounds. *Information Processing Letters*, 36(6):305–308, 1990.

[18] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. *Proc. of FOCS'00*, pages 565–574, 2000.

[19] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. *Proc. of STOC'01*, pages 163–172, 2001.

[20] T. Leighton, B. Maggs, and R. Sitamaran. On the fault tolerance of some popular bounded-degree networks. In *Proc. of FOCS'92*, pages 542–552, 1992.

[21] M.D. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California at Berkeley, 1996.

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

Appendix

Proof: [of Lemma 2] If $|I(t)| = O(\log n)$, then the lemma trivially holds. Therefore, we assume that $|I(t)| > q \log n$, where q is a large constant. We know that if $|I(t)| \leq n/\log^2 n$ for some $t \geq 0$, then $|I(t+1)| \geq c|I(t)|$, where $c > 1$ is a constant [14]. We devide now every step into several substeps, where in each substep, we pick one node from $I(t-1)$ and allow only this one node to choose a neighbor, uniformly at random. Clearly, we pick in each substep a different node, and hence step t contains $|I(t-1)|$ substeps.

Let v be the node choosing a neighbor, uniformly at random, in step t and substep i. Assume, v chooses some neighbor $v' \in V \setminus I(t-1)$, which has not been informed yet. Then, due to Lemma 1, each other node $w \in I(t-1)$ is connected to v' with probability at most p. Therefore, if u is an arbitrary vertex informed at some time t', and picked for transmission in substep 1 of every step $t > t'$, then u is connected to an arbitrary node of $I(t) \setminus (I(t') \cup I(u))$ with probability at most p, where $I(u)$ denote the nodes informed by u in some step $t > t'$. Applying now the Chernoff bounds [4, 17], it follows that, with probability $1 - o(1/n^2)$, u has at most $O(p|I(t)| + \log n)$ neighbors in $I(t) \setminus (I(t') \cup I(u))$.

Obviously, there does not exist any difference between the propagation of the information in this substep model and in the traditional push model, in which the informed nodes choose their neighbors simultaneously. We can therefore apply the Markov inequality, and the lemma follows. $\square$

Proof: [of Lemma 3] Let u be the node at which the information r is placed at time 0. Let $T_t(u) = (V', E')$ be defined in the following way: V' contains the nodes informed until time t, and there is an edge between two nodes $u', u'' \in V'$ in $T_t(u)$ if u'' is informed by u' before step $t+1$. If some node gets the information from several nodes simultaneously, then only one of them (chosen randomly) is considered to share an edge in $T_t(u)$ with this node.

We consider now two cases. In the first case, let us assume that $p \leq 1/\sqrt{n}$. Let $I_{u'}(t)$ denote the set of nodes which have been informed by u before time step t. Then, with probability $1 - o(1/n^2)$, at most 3 edges can occur between a node $u' \in I(t)$ and some other nodes of $I(t) \setminus I_{u'}(t)$. Therefore, as long as $I(t) \leq q \log n$, the probability that a node with $|I_{u'}(t)| \leq c_{\max} - 3$ will be stopped by the algorithm to perform push transmissions is $o(1/n^2)$.

We ignore now the probability that a node with less than $c_{\max} - 2$ neighbors in $T_t(u)$ will be stopped by the algorithm. Clearly, $T_t(u)$ has less than $|I(t)|/(c_{\max} - 4)$ nodes with more than $c_{\max} - 3$ neighbors in $I(t)$. Since a node u' with $|I_{u'}(t)| \leq c_{\max} - 3$ propagates the information to some uninformed node with probability $1 - O(1/\log^{\delta} n)$, applying the methods from [14] we can conclude that the number of informed nodes will exceed $q \log n$ within $O(\log n)$ steps.

If $p \geq 1/\sqrt{n}$, then the probability that a node $u' \in I(t)$ chooses a node from $I(t)$ in step $t+1$ is $O(\log n/\sqrt{n})$. Therefore, as long as $I(t) \leq q \log n$, an arbitrary node u' will be stopped to propagate the information by push operations with probability $o(1/n^2)$. Similarly, each node pushes r to some uninformed node with probability $O(\log n/\sqrt{n})$. Thus, there exists a constant c so that $I(t+3) \geq I(t)(1+c)$ with probability $1 - o(1/n^2)$, and the lemma follows. $\square$

Proof: [of Lemma 4] Lemma 2 implies that every node of $I(t)$ has at most $O(p|I(t)| + \log n)$ neighbors in $I(t)$ (w.h.p.). Since there are $|I(t)|(1 - O(1/\log^{\delta-1} n))$ active nodes in $I(t)$, and any of these active nodes chooses an uninformed neighbor with probability $1/\log^{\delta-1} n$, it follows that $|I(t)|(1 - O(t/\log^{\delta-1} n))(1 - (1+o(1))/\log^{\delta-1} n)$ nodes propagate r to some uninformed nodes. Applying the Chernoff bounds [4, 17] as in [14], it can be shown that at least $|I(t+1)| - |I(t)| > |I(t)|/2$ uninformed nodes are informed in step $t+1$.

On the other hand, since any active node chooses an uninformed neighbor with probability $1/\log^{\delta-1} n$, and $|I_a(t)| \leq |I(t)|$, at most $|I(t)|(1+o(1))/\log^{\delta-1} n$ nodes will choose some informed neighbor. Therefore, the number of nodes switching to state G in step t is less than $|I(t)|(1+o(1))/\log^{\delta-1} n$. Since there have been $|I(t)|O(t/\log^{\delta-1} n)$ inactive informed nodes before, and $|I(t+1)| > |I(t)|$, the lemma follows. $\square$

Proof: [of Lemma 5] We assumed at the beginning that we only consider push transmissions until the number of informed nodes does not reach $n/2$. Lemma 4 implies that if $|I(t)| \leq n/2^{4\max\{\log n/\log d, \log\log n\}}$, then $|I(t)|(1-o(1))$ nodes are active. This implies that when $|I(t)| > n/2^{4\max\{\log n/\log d, \log\log n\}}$ for the first time, then most of the vertices are not in state S or U and have $itime = \Omega(\log n)$. Therefore, all these vertices will be transmitting for $\Omega(\max\{\log\log n, \log n/\log d\})$ steps. As in the proof of Lemma 4, we can show that the number of informed nodes is increased by a constant factor in every succeeding step. This implies that within additional $O(\max\{\log\log n, \log n/\log d\})$ steps, $I(t)$ becomes larger than $n/2$ and most informed vertices are either in state A or in state G with $itime = \Theta(\log n)$. $\square$

Proof: [of Lemma 6] Let t_0 be such that $|I(t_0)| \geq n/2$ for the first time and let $H'(t_0)$ denote the set of vertices which already have r, but are either in state S or in state G with $itime < \epsilon \log n$, where $\epsilon < 1$ is a small constant. From Lemma 4 and 5 we know that $|H'(t)| = o(n/2^{4\max\{\log n/\log d, \log\log n\}})$. We may therefore assume that at time t_0 any node informed before the last $4\max\{\log n/\log d, \log\log n\}$ steps is either in state S or in state G with $itime < \epsilon\log n$. We denote the set of these nodes by D_{t_0}. We know that a node $u' \in D_{t_0}$ is connected to some node $u'' \in D_{t_0}$, not already chosen by u' in some step $t' \leq t$ to transmit r, with probability at most p. Due to Lemma 1, u' is connected to some node of $V \setminus D_{t_0}$ with probability $p(1-o(1))$, and it has $p|V \setminus D_{t_0}|(1-o(1))$ neighbors in $V \setminus D_{t_0}$.

Clearly, as long as $|H(t)| + |D_t| > n/\sqrt{d}$, we can use the methods of [18] to show that at most a fraction of $(|H(t)| + |D_t|)(1+o(1/\log n))/n$ of the $H(t)$ uninformed nodes remain uninformed after the $t+1$st step. Since a node of D_t does not switch to G with $itime = \Theta(\log n)$ with nearly the same probability $(|H(t)| + |D_t|)(1+o(1/\log n))/n$, the lemma follows. $\square$

Proof: [of Lemma 7] Lemma 6 implies that when $|H(t)| \leq n/\sqrt{d}$ for the first time, then $|D_t| \leq n/\sqrt{d}$, and therefore any node of D_t has at most $O(\log n)$ neighbors in D_t, w.h.p. Applying now Lemma 1, we conclude that an arbitrary uninformed node remains uninformed with probability $O(\log n/d)$. $\square$

Proof: [of Lemma 8] Lemma 7 implies that if $|H(t)| \leq q\ln n$, then $|D_t| = O(\log n)$. Hence, an uninformed node has all but $O(\log n)$ neighbors being in state A or G with $itime > \epsilon\log n$. This implies that an uninformed node becomes informed in step $t+1$ with probability $O(\log n/d)$. $\square$