On the Communication Complexity of Randomized Broadcasting in Random-Like Graphs *

Robert Elsässer
Institute for Computer Science
University of Paderborn
33102 Paderborn, Germany
elsa@upb.de

ABSTRACT

Broadcasting algorithms have a various range of applications in different fields of computer science. In this paper we analyze the number of message transmissions generated by efficient randomized broadcasting algorithms in random-like networks. We mainly consider the classical random graph model, i.e., a graph G_p with n nodes in which any two arbitrary nodes are connected with probability p, independently. For these graphs, we present an efficient broadcasting algorithm based on the random phone call model introduced by Karp et al. [21], and show that the total number of message transmissions generated by this algorithm is bounded by an asymptotically optimal value in almost all connected random graphs. More precisely, we show that if $p \geq \log^\delta n/n$ for some constant $\delta > 2$, then we are able to broadcast any information r in a random graph G_p of size n in $O(\log n)$ steps by using at most $O(n \max\{\log \log n, \log n / \log d\})$ transmissions related to r, where $d = pn$ denotes the expected average degree in G_p. We also show that for these kind of graphs there is a a matching lower bound on the number of transmissions generated by any efficient broadcasting algorithm which works within the limits of the random phone call model. Please note that the main result holds with probability $1 - 1/n^{\Omega(1)}$, even if n and d are unknown to the nodes of the graph.

The algorithm we present in this paper is based on a simple communication model [21], is scalable, and robust. It can efficiently handle restricted communication failures and certain changes in the size of the network, and can also be extended to certain types of truncated power law graphs based on the models of [1, 2, 5]. In addition, our methods and results might be useful for further research on this field.

*The research was performed while the author visited the Department of Mathematics at University of California, San Diego. Partly supported by the German Research Foundation under contract EL-399/1-1.

SPAA'06, July 30–August 2, 2006, Cambridge, Massachusetts, USA.

Categories & Subject Descriptors:
F.2.2[Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems;
General Terms: Algorithms, Theory.
Keywords: broadcasting, random graphs.

1. INTRODUCTION

Randomized broadcasting has extensively been studied in various network topologies. Such algorithms naturally provide robustness, simplicity and scalability. As an example, consider the so-called *push model* [10]: In a graph $G = (V, E)$ we place at some time t an information r on one of the nodes. Then, in each succeeding round, any *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 of size n, with probability $1 - o(1/n)$, whenever p exceeds some threshold value [17]. However, this algorithm generates $\Omega(n \log n)$ transmissions of r. Therefore, some modifications of this scheme have been considered in order to decrease the number of message transmissions [10, 21]. These variants are described in the subsections below.

1.1 Models and Motivation

The study of information spreading in large networks has various fields of application in distributed computing. Consider for example the maintenance of replicated databases in name servers in a large corporate network [10]. There are updates injected at various nodes, and these updates must be propagated to all the nodes in the network. In order to let all copies of the database converge to the same content, efficient broadcasting algorithms have to be developed.

There is an enormous amount of experimental and theoretical study of (deterministic and randomized) broadcasting in various models and on different networks. In this paper we only concentrate on randomized broadcasting algorithms, and study their time and communication complexity in a simple communication model. The advantage of randomized broadcasting is its inherent robustness against failures and dynamical changes compared to deterministic schemes that either need substantially more time [18] or can tolerate only a relatively small number of faults [23]. Our intention is to develop time efficient randomized broadcasting algorithms which have the following properties:

- They can successfully handle restricted communication failures in the network.

- They are fully adaptive and work correctly if the size and/or topology of the network change slightly during the execution of the algorithm.
- The number of message transmissions 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 available to all nodes so that its transmission can be stopped were investigated. Using simple mechanisms for the push model, it is possible to restrict the number of transmissions in a random graph of size n to $O(n \log n)$.

An idea introduced in [10] consists of so called *pull transmissions*, i.e., any (informed or uninformed) node is allowed to call a randomly chosen neighbor, and information is sent from the called to the calling node. In this model it may happen that some nodes transmit messages to several neighbors within one step, however, the number of transmissions within one step is bounded by the number of nodes in the graph. These kind of transmission makes only sense if new or updated informations occur frequently in the network so that almost every node has to place a random call in each round anyway. It was observed in complete graphs that after a constant fraction of the nodes is informed, then within $O(\log \log n)$ additional steps every node of a K_n becomes informed as well [10, 21]. This implies that in such graphs at most $O(n \log \log n)$ transmissions are enough, if the distribution of the information is stopped at the right time.

In this paper we are particularly interested in randomized broadcasting algorithms on the class of random-like graphs. The theory of random graphs was founded by Erdős and Rényi [14, 15]. 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 $G_{n,m}$ is assigned the same probability. Another random graph model has been introduced by Gilbert in [19], 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 Erdős-Rényi graphs.

In order to describe large real world networks, some modifications of these random graph models have been considered. In [2, 16] 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. A modified version of the classical random graph model, which approximates such power law graphs, is discussed in Subsection 1.2.

1.2 Related Work

Most papers dealing with randomized broadcasting analyze the runtime of the push algorithm in different graph classes. Pittel [26] proved that it is possible to broadcast an information within $\log_2(n)+\ln(n)+O(1)$ steps in a complete graph of size n, by using the push algorithm. In [17], Feige et al. determined asymptotically 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 [22] and proved that any information is spread to nodes at distance t in $O(\ln^{1+\epsilon} t)$ steps.

In [21], Karp et al. introduced the so called random phone call model by combining push and pull, and presented a termination mechanism for this model, which reduces the number of total transmissions to $O(n \log \log n)$ in complete graphs of size n. It has also been shown that this result is asymptotically optimal among 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, whereby the nodes are supposed to have an estimation on the size of the network. However, we could not use the termination mechanism of [21] for the random graphs G_p, whenever pn is below some threshold and nothing is known to the nodes of the graph.

Most papers on random graphs deal with their structural properties. Please refer to [3] for an excellent survey on the properties of Erdős-Rényi graphs. 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 of size n in which edges are independently assigned to each pair of vertices (i, j) with probability $d_i d_j / \sum_{k=1}^{n} d_k$. Now if $\mathbf{d}$ obeys a power law, then the resulting graph is well suited to model power law graphs. Chung et al. also analyzed the connectivity, distances, and eigenvalues of these graphs for certain sequences of $\mathbf{d}$ [5, 6, 8]. We should mention that most real world networks possess further properties (e.g. exhibit high levels of clustering, cf. [25]) which are unspecific in such generalized random graphs. However, a termination mechanism similar to the one considered in this paper leads to optimal results not only in the graphs $G(\mathbf{d})$, but also in some dynamically constructed random power law graph models (e.g. [1, 2]) [11].

1.3 Our Results

In this paper we present an adaptive randomized broadcasting algorithm which is able, with probability $1-1/n^{\Omega(1)}$, to distribute an information r, placed initially on a node of a random graph G_p of size n and expected average degree $d = pn$, to all nodes of the graph in $O(\log n)$ steps by using $O(n \max\{\log \log n, \frac{\log n}{\log d}\})$ transmissions of r, whenever $p \geq \log^{\delta} n/n$ for some constant $\delta > 2$. This algorithm is robust against restricted communication failures or restricted short time changes in the size of the network, and can be adapted to some generalized random graph models as well. Moreover, since we do not require any previous knowledge about the size or average degree of the network, our algorithm is robust against any kind of long time changes in the size of the network. The results of this paper can be extended to the case $p \geq \delta' \log n/n$, where δ' is a large constant, however, the proofs would be much more complicated, and we only consider therefore the weaker case in this paper.

Our algorithm is based on the so called random phone call model [21], in which every node is allowed (in each time step) to choose a neighbor, uniformly at random, and to establish a communication channel with this neighbor. Then, any node may send/receive messages to/from each neighbor which has established a communication channel with this node in the current step. In this model, even if a node does not know anything about some specific information that has to be transmitted to all nodes in the network, it still establishes communication channels with neighbors to exchange messages. This kind of model makes only sense, if new or updated informations occur frequently in the network so that every node has to transmit some messages in each step anyway. Then, the cost of establishing communication amortizes over all transmissions [21], and we are allowed to consider the number of transmissions of one single message only, since the total number of transmissions will asymptotically be bounded by the number of transmissions produced by all informations occuring in the system [21]. However, if there are only a few informations to be broadcasted to all nodes, then the cost of establishing communication could exceed the total cost for transmissions. Therefore, we assume in this paper that new informations are frequently generated by almost every node, however, we only focus on the distribution and lifetime of a single information.

The rest of the paper is organized in four sections. In Section 2 we concentrate on the case when the nodes are supposed to have an estimation of $\log n/\log d$, and present an algorithm with the properties described above. In Section 3, we improve our algorithm by using an additional trick that enables us to solve the broadcasting problem efficiently even if nothing is known to the nodes about the graph. In Section 4 we discuss the applicability of our methods in some truncated power law networks. Finally, Section 5 contains our conclusions.

2. ALGORITHM WITH PARTIAL KNOWLEDGE

In this section we analyze the behavior of a randomized broadcasting algorithm which is based on the random phone call model introduced in [21]. For a graph $G = (V, E)$, the random phone call model is defined in the following way: In each time step, every node of G chooses a neighbor, uniformly and random, and establishes a communication channel with this neighbor. Any node $u \in V$ is then allowed to send/receive messages to/from all nodes which have established communication chanles with u in the current step. Hereby we assume that all nodes have access to a global clock, and they work in a synchronous environment. Due to the definition of the model, a node may transmit/receive messages to/from several neighbors within one step, however, the number of total messages in one step is bounded by the number of vertices.

We consider the model described above on random graphs G_p of size n and average degree $d = pn$, where $p \geq \frac{\log^{\delta} n}{n}$ for some $\delta > 2$. We assume in this section that every node has an estimation of $\tau = \log n/\log d$ (n and d, however, are still unknown), and present an algorithm which is able, with high probability[1], to broadcast an information r to all nodes of G_p within $O(\log n)$ time steps, whereby the number of transmissions related to r is bounded by $O(n \max\{\log\log n, \frac{\log n}{\log d}\})$.

As mentioned in the introduction, we assume that new informations occur frequently in the network so that almost every node places a call in each round anyway. However, we only focus here on the distribution and lifetime of a single information. The algorithm we describe in the following paragraphs contains several rounds. In each round, whenever a communication channel is established between two nodes, each one of them has to decide what to transmit to the other node, without knowing if the vertex at the other end of the edge has already received some specific information prior to 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.

Let r denote the information we consider and assume w.l.o.g. that r is placed on one of the nodes at time 0. In each step, any node which decides to transmit r, also sends its node ID and a constant number of other messages related to the information (cf. algorithm below).

At the beginning, we initialize at each node u an array $T[c_{\max}]$ with $T[i] = 0$ for any $i = 1, \cdots, c_{\max}$, where $c_{\max}$ is a (large) constant, and the integers $age = 0$, $itime = 0$, and $ctr = 0$. The array T and the integers $itime$, age, and ctr are local variables and may differ from node to node. During the algorithm proceeds, at any node u the array T is used to store at most $c_{\max}$ different node ID's, age denotes the age of r (if u is informed), $itime$ is used to store the most recent time step (known to u) in which a node has been newly informed in the system, and ctr counts the number of consecutive time steps, in which u receives r (from a different node in each step).

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). At the beginning, the node on which r has initially been placed is in state A. All other nodes are in state U. Now, in each step t any node $u \in V$ executes the following procedure:

1. Choose a neighbor, uniformly at random, and call this node to establish a communication channel with it. Furthermore, establish a communication channel with all nodes which call u in this step.

2. If u is in state A or G, then send to all nodes which have established a communication channel with u the message $(r, itime, age, ID(u))$.

3. Receive messages from all neighbors which have established a communication channel with u. Let these messages be denoted by $(r, itime_1, age_1, ID(v_1))$, $\ldots$, $(r, itime_k, age_k, ID(v_k))$ (if any). Then, close all communication channels.

4. Perform the following local computations:

   4.1. If u is in state A,G, or S, and $itime$ is smaller than $\max_{1\leq i\leq k} itime_i$, then set $itime = \max_{1\leq i\leq k} itime_i$. If u is not in state U, then increment age by 1.

   4.2. If u is in state U and there is a neighbor v_i, which transmitted r to u, then switch state of u to A, and set $itime$ and age to $age_i + 1$.

   4.3. If u is in state A, then:

   - if u does not receive r in this step, then set $ctr = 0$ and $T[j] = 0$ for any $j = 1, \ldots, c_{\max}$.
   - if u receives r in this step and there are some $i \in \{1, \ldots, k\}$ such that $ID(v_i) \notin T$, then choose such an i (e.g. uniformly at random),

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

set $T[ctr+1] = ID(v_i)$, and increment ctr by 1. If $ctr = c_{\max}$, then switch state of u to G.

4.4. If u is in state G and age is equal to $itime + \alpha \max\{\log itime, \tau\}$, where α is a large constant, then switch state of u to S.

4.5. If u is in state S and age is smaller than $itime + \alpha \max\{\log itime, \tau\}$, then switch back to state G.

Please note that in this algorithm, the nodes are aware of $\tau = \log n / \log d$ (the modified algorithm for the case in which nothing is known to the nodes is given in the next section).

In the rest of this section we analyze the behavior of the algorithm presented above. First, we state a combinatorial result needed for the main analysis.

For some u, v let $A_{u,v}$ denote the event that u and v are connected by an edge, and let $A_{u,v,l}$ denote the event that u and v share an edge **and** u chooses v in step l (according to the random phone call model described above). In the next lemma, we deal with the distribution of the neighbors of a node u in a graph G_p, after it has chosen t neighbors, uniformly at random, in $t = O(\log n)$ consecutive steps. In particular, we show that the probability of u being connected with some node v, not chosen within these t steps, is not substantially modified after $O(\log n)$ steps.

Lemma 1. *Let $V = \{v_1, \ldots, v_n\}$ be a set of n nodes and let every pair of nodes v_i, v_j be connected with probability p, independently, where $p \geq \log^{\delta} n/n$ for some constant $\delta > 2$. If $t = O(\log n)$, $u, v \in V$, and $A(U_0, U_1, U_2) = \bigwedge_{\substack{0<l\leq t \\ (v_i,v_j,l)\in U_0}} A_{v_i,v_j,l} \bigwedge_{(v_{i'},v_{j'})\in U_1} A_{v_{i'},v_{j'}} \bigwedge_{(v_{i''},v_{j''})\in U_2} \overline{A_{(v_{i''},v_{j''})}}$, for some $U_0 \subset V \times V \times \{0, \ldots, t\}$ and $U_1, U_2 \subset V \times V$, then it holds that*

$$\Pr\left[(u, v) \in E \mid A(U_0, U_1, U_2)\right] = p(1 \pm O(t/d)),$$

for any U_0, U_1, U_2 satisfying the following properties:

- $|U_0 \cap \{(v_i, v_j, l) | v_j \in V\}| = 1$ *for any* $v_i \in V$ *and* $l \in \{0, \ldots, t\}$,
- $|U_1 \cap \{(u, u') | u' \in V\}| = \Omega(d)$ *and* $|U_1 \cap \{(v, v') | v' \in V\}| = \Omega(d)$,
- $(u, v) \notin U_1 \cup U_2$, *and* $(u, v, i) \notin U_0$ *for any* i.

Proof. In our proof, we simplify the condition on U_0 by assuming that only node u is allowed to choose a neighbor, uniformly at random, in each step $1, \ldots, t$. The other nodes do not choose and call any neighbor in these steps. Then, $\{v_i \mid (v_i, v_j, l) \in V \times V \times \{1, \ldots, t\}\} \cap U_0 = \{u\}$. Please note that the proof techniques required for the general case (in which any node chooses a neighbor, uniformly at random, in each step $1, \ldots, t$) are the same as in the proof presented below, however, the presentation becomes more complicated, and therefore we omit the general case here.

Let $S_u = \{v_j \mid (u, v_j, l) \in U_0 \text{ for } 1 \leq l \leq t, \text{ and } (u, v_j) \notin U_1\}$, and denote by $w_1, \ldots, w_t$ the elements of S_u. Let V' denote the set of nodes obtained from V by deleting any node u' with the property $(u, u') \in U_1 \cup U_2$ or $u' \in S_u$. First we consider the distribution of the neighbors of u after one single step. Let A denote the event that u is connected to the nodes of S_u and $(\bigwedge_{(v_{i'},v_{j'})\in U_1} A_{v_{i'},v_{j'}} \bigwedge_{(v_{i''},v_{j''})\in U_2} \overline{A_{(v_{i'},v_{j'})}})$ holds. Let A_1 denote the event that u chooses w_1 in step $l = 1$. If we denote by p_j the probability that u has j neighbors among the nodes of V', then it holds that

$$\begin{aligned}\Pr[A_1 \mid A] &= p \sum_{j=0}^{|V'|-1} \frac{p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \\ &\quad + (1-p) \sum_{j=0}^{|V'|-1} \frac{p_j}{j + |S_u| + |U_1 \cap N(u)|}.\end{aligned}$$

Here, the first term on the right hand side expresses the fact that with probability p the vertices u and v are connected, and with probability p_j the vertex u has exactly j neighbors among the elements of V'. Then, u chooses w_1 with probability $1/(j + |S_u| + |U_1 \cap N(u)| + 1)$, where $|U_1 \cap N(u)|$ denotes the number of edges which are incident to u and are contained in U_1. The second term handles the case when u and v are not connected. Then, with probability p_j the vertex u has exactly j neighbors in V', however now u chooses w_1 with probability $1/(j + |S_u| + |U_1 \cap N(u)|)$.

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

$$p'_{u,v} = \frac{p \sum_{j=0}^{|V'|-1} p_j/(j + |S_u| + |U_1 \cap N(u)| + 1)}{\Pr[A_1 \mid A]}.$$

Let $e' = \sum_{j=0}^{|V'|-1} p_j/(j + |S_u| + |U_1 \cap N(u)| + 1)$. Then it holds that

$$\begin{aligned}\Pr[A_1 \mid A] &= (1-p) \sum_{j=0}^{|V'|-1} \frac{p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \cdot \\ &\quad \cdot \frac{j + |S_u| + |U_1 \cap N(u)| + 1}{j + |S_u| + |U_1 \cap N(u)|} + pe' \\ &\leq pe' + \sum_{j=0}^{|V'|-1} \frac{p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \cdot \\ &\quad \cdot \frac{j + |S_u| + |U_1 \cap N(u)| + 1}{j + |S_u| + |U_1 \cap N(u)|} - pe'\end{aligned}$$

since

$$\begin{aligned}\sum_{j=0}^{|V'|-1} \frac{p \cdot p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \cdot \frac{j + |S_u| + |U_1 \cap N(u)| + 1}{j + |S_u| + |U_1 \cap N(u)|} \\ \geq \sum_{j=0}^{|V'|-1} \frac{p \cdot p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \geq pe'.\end{aligned}$$

Then,

$$\begin{aligned}\Pr[A_1 \mid A] &\leq \sum_{j=0}^{|V'|-1} \frac{p_j}{j + |S_u| + |U_1 \cap N(u)| + 1} \cdot \\ &\quad \cdot \frac{j + |S_u| + |U_1 \cap N(u)| + 1}{j + |S_u| + |U_1 \cap N(u)|} \\ &\leq \sum_{j=0}^{|V'|-1} \frac{p_j(1 + 1/\Omega(d))}{j + |S_u| + |U_1 \cap N(u)| + 1}\end{aligned}$$

and the claim follows.

Similar techniques lead to the result for arbitrary $t = O(\log n)$. The desired upper bound on the conditional probability $\Pr\left[(u, v) \in E \mid A(U_0, U_1, U_2)\right]$ can be obtained in a similar way. $\square$

This lemma implies that even if the occurrence of the edges are not necessarily independent after $t = O(\log n)$ steps, in certain cases (as in the lemmas below) we still can apply some known results which require independency (like the Chernoff bounds [4, 20]) if $\Pr[(u, v) \in E \mid A(U_0, U_1, U_2)]$ is properly approximated by $p(1 \pm O(t/d))$.

Now we are ready to analyze the algorithm presented at the beginning of this section. 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 order to show that the algorithm is able to spread an information among all nodes of a graph G_p within $O(\log n)$ rounds, and the number of total transmissions 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.

In our proofs, *age* and t are considered to be the same at any informed node, since we assumed that r is placed on one of the nodes at time 0. In order to show that the algorithm has the claimed properties, we first prove the following lemmas.

LEMMA 2. *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 $1 - o(1/n)$.*

PROOF. Let u be the node at which the information r is placed at time 0. Let tree $T_t(u) = (V', E')$ be defined in the following way: V' contains the nodes informed by 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 we 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 $t + 1$. 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 switch to state G 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 $T_t(u)$. 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 of [17] we conclude that the number of informed nodes will exceed $q \log n$ within $O(\log n)$ steps, w.h.p.

If $p \geq 1/\sqrt{n}$, then the probability that a node $u' \in I(t)$ chooses a node from $I(t)$ (or is chosen by 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 switch to state G with probability $o(1/n^2)$. Similarly, each node pushes r to some uninformed node with probability $1 - O(\log n/\sqrt{n})$. 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. $\square$

Lemma 2 implies that after $O(\log n)$ rounds, we have $|I(t)| \geq q \log n$. The total number of message transmissions after these rounds does not exceed the value $O(\log^2 n)$.

Now we consider the case when $|I(t)|$ lies between $q \ln n$ and $n/2^{4 \max\{\log n/\log d, \log\log n\}}$.

LEMMA 3. *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 2. We also assume that the number of active nodes $|I_a(t)|$ before step t+1 is at least $|I(t)|(1 - O(\frac{t}{\log^{\delta-1} n}))$. 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 $1 - o(1/n^2)$.*

PROOF. Lemma 1 implies that every node of $I(t)$ has at most $p|I(t)| + O(\sqrt{p|I(t)|\log n} + \log n)$ neighbors in $I(t)$ (w.h.p.). Since there are $|I(t)|(1 - O(t/\log^{\delta-1} n))$ active nodes in $I(t)$, and any of these active nodes chooses an uninformed neighbor with probability $1 - 1/\log^{\delta-1} n$, 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, with probability $1 - o(1/n^3)$. Applying Lemma 1 together with the Chernoff bounds [4, 20] as in [17], it can be shown that with probability $1 - o(1/n^2)$ 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 informed neighbor with probability $1/\log^{\delta-1} n$, and $|I_a(t)| \leq |I(t)|$, at most $\frac{|I(t)|}{\log^{\delta-1} n}(1 + o(1))$ nodes of $I(t)$ will choose some informed neighbor, with probability $1 - o(1/n^2)$. Therefore, the number of nodes switching to state G in step t is less than $|I(t)|(1 + o(1))/\log^{\delta-1} n$, with probability $1 - o(1/n^2)$. Since there have been $|I(t)|O(t/\log^{\delta-1} n)$ inactive informed nodes before (with probability $1 - o(1/n^2)$), and $|I(t+1)| \geq |I(t)|$, the lemma follows. $\square$

Lemma 3 implies that after additional $O(\log n)$ steps, we have $|I(t)| \geq n/2^{4 \max\{\log n/\log d, \log\log n\}}$. Furthermore, there are at most $O(n)$ transmissions during these rounds.

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 4. *Let $I(t)$ be the set of informed nodes in G_p at time $t = O(\log n)$. Assume that $|I(t)|$ is larger than $n/2^{4 \max\{\frac{\log n}{\log d}, \log\log n\}}$ and smaller than $n/2$, and $|I(t)|(1 - o(1))$ nodes of $I(t)$ are either in state A, or in state G with $itime = t_0(1 - o(1))$, where t_0 is the first time step in which $|I(t)|$ is larger than $n/2^{4 \max\{\frac{\log n}{\log d}, \log\log n\}}$. Then, there exists a constant c such that $|I(t + 1)| \geq |I(t)|(1 + c)$ with probability $1 - o(1/n^2)$. Moreover, all vertices informed after step t_0 will transmit for at least $7\alpha/8 \cdot \max\{\frac{\log n}{\log d}, \log\log n\}$ further steps, w.h.p.*

PROOF. We assumed at the beginning that we only consider push transmissions as long as the number of informed nodes does not reach $n/2$. Lemma 3 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, w.h.p. Since $I(t_0)(1 - o(1))$ nodes have been informed in the most recent $o(t_0)$ steps, and $I(t_0)(1 - o(1))$ of these nodes are active at time t_0, $I(t_0)(1 - o(1))$ nodes are in state A and have $itime = t_0(1 - o(1))$. Since $t_0 = \Omega(\log n)$ [17], all these vertices will be transmitting for $\Omega(\max\{\log\log n, \frac{\log n}{\log d}\})$ additional steps. As in the proof of Lemma 3, we can show that, with probability $1 - o(1/n^2)$, the number of informed nodes is increased by a constant factor in every succeeding step. This implies that if α is large enough, then within additional $\alpha/8 \cdot \max\{\log\log n, \log n/\log d\})$ steps, $I(t)$ becomes larger than $n/2$ and all vertices informed after time step t_0 are either in state A or in state G with $itime \geq t_0$, w.h.p. Moreover, all these vertices transmit r for at least $7\alpha/8 \cdot \max\{\log\log n, \log n/\log d\})$ further steps, w.h.p., whenever α is large enough. $\square$

Lemma 4 implies that within $O(\max\{\frac{\log n}{\log d}, \log\log n\})$ rounds we have $|I(t)| \geq n/2$. Moreover, the number of total transmissions is at most $O(n)$ after these rounds.

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 5. *Let $|H(t)| \in [n/\sqrt{d}, n/2]$ be the number of uninformed nodes in G_p at some time $t = O(\log n)$, and assume that there are at most $|H(t)|(1 + O(t/\log^{\delta-1} n))$ informed nodes in state S, or in states A, G with $itime \leq t_0$, where t_0 denotes the time step defined in Lemma 4. Then, at time $t + 1$ it holds that $|H(t + 1)| \leq |H(t)|^2(2 + o(1))/n$, and at most $|H(t + 1)|(1 + O((t + 1)/\log^{\delta-1} n))$ informed nodes are in state S, or in state A, G with $itime \leq t_0$.*

PROOF. Let t_1 be the time step in which $|I(t_1)| \geq n/2$ for the first time and let $H'(t_1)$ denote the set of vertices which already have r, but are either in state S or in state A, G with $itime < t_0$. From Lemma 3 and 4 we know that $|H'(t_1)| = o(n/2^{4\max\{\log n/\log d, \log\log n\}})$. We assume therefore that at time t_1, any node informed before step t_0 is either in state S or has $itime < t_0$. We denote the set of these nodes by D_{t_1}. We know that a node $u' \in D_{t_1}$ is connected to some node $u'' \in D_{t_1}$, not already chosen by u' in some step $t' \leq t_1$, with probability at most $p(1 + o(1))$. Due to Lemma 1, u' has $p|V \setminus D_{t_1}|(1 - o(1))$ neighbors in $V \setminus D_{t_1}$.

Clearly, as long as $|H(t)| + |H'(t)| > n/\sqrt{d}$, we can use the methods of [21] to show that at most a fraction of $(|H(t)| + |H'(t)|)(1 + o(1/\log n))/n$ of the $|H(t)|$ uninformed nodes remain uninformed after the $t+1$st step (or have $itime < t_0$). Since a node of $H'(t)$ does not switch back to G, or does not set $itime \geq t_0$, with nearly the same probability $(|H(t)| + |H'(t)|)(1 + o(1/\log n))/n$, the lemma follows. Moreover, all nodes which have been informed after step t_0 (or have reset $itime$ to some value $\geq t_0$) will transmit for $3\alpha/4 \cdot \max\{\log n/\log d, \log\log n\}$ further steps, w.h.p., whenever α is large enough. $\square$

Lemma 5 implies that after additional $O(\log\log n)$ steps it holds that $|H(t)| \leq n/\sqrt{d}$. The number of transmissions is at most $O(n \log\log n)$.

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

LEMMA 6. *Let $|H(t)| \in [q \ln n, n/\sqrt{d}]$ be the number of uninformed nodes in G_p at some time $t = O(\log n)$, where q is the constant defined in Lemma 2. Then, $|H(t + 1)| \leq |H(t)|O(\sqrt{\log n/d})$, and $|H'(t + 1)| \leq |H(t)|O(\sqrt{\log n/d})$*

PROOF. Lemma 5 implies that when $|H(t)| \leq n/\sqrt{d}$ for the first time, then $|H'(t)| \leq n(1+o(1))/\sqrt{d}$, w.h.p. Lemma 1 implies that any node of $H'(t)$ or $H(t)$ has less than $O(\sqrt{d \log n})$ neighbors in $H'(t)$, w.h.p. Applying now Lemma 1 together with the Chernoff bounds [4, 20], we conclude that an arbitrary uninformed node remains uninformed in some step t with probability $O(\sqrt{\log n/d})$. Similarly, a node of $H'(t)$ does not set $itime \geq t_0$ with probability $O(\sqrt{\log n/d})$. $\square$

Lemmas 6 ensures that after additional $O(\log n/\log d)$ rounds all nodes are informed, w.h.p. The overall communication complexity is $O(n \max\{\log n/\log d, \log\log n\})$. Moreover, if α is large enough, then all nodes will transmit r simultaneously for at least $\frac{\alpha}{4} \cdot \max\{\frac{\log n}{\log d}, \log\log n\}$ further steps, w.h.p. This implies that during these steps, all nodes switch to state G, and since the largest $itime$ occuring in the system is bounded by $t_0+O(\max\{\log n/\log d, \log\log n\})$, all nodes stop transmitting r within $O(\max\{\frac{\log n}{\log d}, \log\log n\})$ additional steps.

Please remember, we assumed that during the algorithm proceeds the nodes are aware of an estimate of $\log n/\log d$. In the following section we present a method which allows most nodes to determine the desired estimate while the algorithm is executed, without substantially increasing the runtime or the number of message transmissions.

3. FULLY ADAPTIVE ALGORITHM

In order to describe the fully adaptive algorithm, first we initialize at any node some additional integers, apart from the array T, and the integers age, $itime$, and ctr. These additional integers at a node u are denoted by $time_u(r)$, τ', τ'', h, h', and $ctr2$. These integers are local variables and may differ from node to node. At the beginning, however, they are all set to 0 at any node.

We also introduce the special states O, R, and R'. During the algorithm proceeds, any node is simultaneously in one of the states U, A, G, or S, **and** in one of the special states O, R, or R'. The special state of a node u is denoted in the rest of this section by $sstate(u)$ and is initialized with O.

In each of the steps 0 and 1, let any node choose a neighbor, uniformly at random, and 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 perform random walks in the system and some node $w' \in V$ checks how many of these messages are lying on it at time $t = \alpha \cdot time_{w'}(r)$, where $time_{w'}(r)$ 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 $(1 \pm o(1/n^2))/n$ [3, 12]. Combining the results of [12] with [24] (Lemma 2.13), it holds that if $time_{w'}(r)$ is large enough, then some nodes of G_p have an estimate of $\log n/\log d$, w.h.p. Please note that in any step there are $O(n/d)$ special messages performing random walks in the system, and therefore the total number of transmissions does not increase substantially in the system.

Now we modify the algorithm described at the beginning of the previous 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.

To describe the algorithm, first define $s_i = 2^i$ for any $i \in \{0, \ldots, \alpha\lceil\log\log n\rceil\}$. Two numbers j_1 and j_2 are called s_2-equivalent, and denoted $j_1 \sim_{s_2} j_2$, if an i exists such that $s_i \leq j_1, j_2 \leq s_{i+1}$. Two nodes w' and w'' are called s_2-equivalent if $time_{w'}(r)$ and $time_{w''}(r)$ are s_2-equivalent.

In the algorithm presented below, any node u may use the subrutine $restate(u, SpecS, v, age, \tilde{\tau})$, where $SpecS$ denotes one of the special states R or R', and $\tilde{\tau}$ is an integer. $restate(u, SpecS, v, age, \tilde{\tau})$ is used when node u has switches its state to $SpecS$, either on its own, or forced by some neighbor v (cf. algorithm below).

$restate(u, SpecS, age, \tilde{\tau})$:

- Switch $sstate(u)$ to $SpecS$ and state of u to A. Furthermore, set $\tau' = \tilde{\tau}$, $itime = age$, $ctr = 0$, and $T[j] = 0$ for any $j = 1, \ldots, c_{\max}$.

Now we are ready to describe the algorithm. At the beginning, the node on which r is placed is in state A. All other nodes are in state U. In each step t, any node $u \in V$ executes the following procedure:

1. Choose a neighbor, uniformly at random, and call this node to establish a communication channel with it. Furthermore, establish a communication channel with all nodes which call u in this step.
2. If u is in state A or G, then send to all nodes which have established a communication channel with u the message $(r, itime, age, ID(u), sstate(u), h', \tau')$.
3. Receive messages from all nodes which have established a communication channel with u in this step. In the following, $(r, itime_1, age_1, ID(v_1), sstate(v_1), h'_1)$, $\ldots$, $(r, itime_k, age_k, ID(v_k), sstate(v_k), h'_k)$ denote the messages (related to r) received in this step, if any. Then, close all communication channels
4. If u is in state A, G, or S, then increment age by 1.
5. Perform the following local computations:

5.1 If $sstate(u) = O$ and $sstate(v_i) = O$ for all $i = 1, \ldots, k$, then consider the following cases:

5.1.1 If u is in state U and there is a neighbor v_i, which transmitted r to u, then switch state of u to A, and set $itime$, age, and $time_u(r)$ to age_i+1. Furthermore, set $h' = 2^{\lfloor \log time_u(r) \rfloor}$.

5.1.2. If u is in state A, then:

- if u does not receive r in step t, then set $ctr = 0$ and $T[j] = 0$ for any j.
- if u receives r in this step and there are some $i \in \{1, \ldots, k\}$ such that $ID(v_i) \notin T$, then choose such an i (e.g. uniformly at random), set $T[ctr + 1] = ID(v_i)$, and increment ctr by 1.
- if $itime < \max_{1 \le i \le k} itime_i$, then set $itime = \max_{1 \le i \le k} itime_i$. If there exists an i such that v_i transmits r to u for the first time, $u \sim_{s_2} v_i$, and v_i has not been informed by u, then increment $ctr2$ by 1. If $ctr = c_{\max}$, then switch state of u to G.

5.1.3. If u is in state G and $age = itime + \alpha \cdot \log itime$, then switch to state S.

5.1.4. If $ctr2 \ge 5$ and $age = \alpha \cdot itime$, then check the number of r_w's on u. If this number is larger than 0, then let τ'' be this number. Otherwise, set $\tau'' = 1$.

5.1.5. If $\tau'' > \log itime$ and $age = 8\alpha \cdot 2^{\lfloor \log time_u(r) \rfloor} + h$, then $restate(u, R, age, \tau'')$. Here h depends on τ'' and $time_u(r)$, and is defined later in this section.

5.1.6. If $\tau'' \ge 1$ and $age = 32\alpha \cdot 2^{\lfloor \log time_u(r) \rfloor}$, then $restate(u, R', age, \log itime)$.

5.2. If $sstate(u) = O$ and there is an i such that $sstate(v_i) \neq O$, then $restate(u, sstate(v_i), age, \tau'_i)$, and set h' to h'_i.

5.3. If $sstate(u) = R$ and $sstate(v_i) \neq R'$ for all $i = 1, \ldots, k$, then consider the following cases:

5.3.1. If there is an i such that τ'_i is larger than τ' of u, and τ'_i is not s_2-equivalent with τ' of u, then $restate(u, R, age, \tau'_i)$.

5.3.2. If u is in state A, then

- if $itime < \max_{1 \le i \le k} itime_i$, then set $itime = \max_{1 \le i \le k} itime_i$.
- if u does not receive r in this step, or there is an i such that $sstate(v_i) \neq R$ or τ'_i is not s_2-equivalent with τ' of u, then set $ctr = 0$ and $T[j] = 0$ for all j.
- otherwise, if there are some $i \in \{1, \ldots, k\}$ such that $ID(v_i) \notin T$, then choose such an i, set $T[ctr + 1] = ID(v_i)$, and increment ctr by 1. If $ctr = c_{\max}$, then switch state of u to G.

5.3.3. If u is in state G and $age > itime + \alpha\tau'$, then switch to state S.

5.3.4. If $\tau'' > \log itime$ and $age = 8\alpha \cdot 2^{\lfloor \log time_u(r) \rfloor} + h$, then $restate(u, R, age, \tau'')$.

5.3.5. If $age = 16\alpha h'$, then $restate(u, R', age, \tau')$.

5.4 If $sstate(u) = R$ and there is an i such that $sstate(v_i) = R'$, then $restate(u, R', age, \tau'_i)$.

5.5 If $sstate(u) = R'$, then consider the following cases:

5.5.1 Let $i_1, \ldots, i_l$ represent the indices for which $sstate(v_{i_j}) = R'$ and τ'_{i_j} is s_2-equivalent with τ' of u. Now, if $itime < \max_{1 \le j \le l} itime_{i_j}$, then set $itime = \max_{1 \le j \le l} itime_{i_j}$. However, if there is an i such that $sstate(v_i) = R'$, τ'_i is larger than τ' of u, and τ'_i is not s_2-equivalent with τ' of u, then $restate(u, R', age, \tau'_i)$.

5.5.2. If state of u is A, then

- if u does not receive r in this step, or there is an i such that $sstate(v_i) \neq R'$ or τ'_i is not s_2-equivalent with τ' of u, then set $ctr = 0$ and $T[j] = 0$ for all j.
- otherwise, if there are some $i \in \{1, \ldots, k\}$ such that $ID(v_i) \notin T$, then choose such an i (e.g. uniformly at random), set $T[ctr+1] = ID(v_i)$, and increment ctr by 1. If $ctr = c_{\max}$, then switch state of u to G.

5.5.3. If state of u is G and $age = itime + \alpha\tau'$, then switch state of u to S.

5.5.4. If state of u is S and $age < itime + \alpha\tau'$, then switch state of u back to G.

As mentioned in the algorithm, the value h depends on τ'' and $time_u(r)$. If $\tau'' \in [\log h' + 1, 2 \log h']$ then $h = 0$. For $\tau'' \in [2 \log h' + 1, 4 \log h']$ set $h = \alpha h' / \log h'$, and generally, if $\tau'' \in [2^i \log h' + 1, 2^{i+1} \log h']$, then $h = i\alpha h' / \log h'$.

The nodes in the algorithm presented above are in one of the special states O, R, or R'. As long as a node is in special state O, it executes a similar procedure to the one presented in the previous section, however *itime* is only checked and updated in state A. The cases, when a node u with $sstate(u) = O$ switches to special state R or R', are described in 5.1.4.-5.1.6., and 5.2. In order to switch to R, u first checks, if it has seen 5 different nodes which are $s2$-equivalent with u, have not been informed by u, and transmit r to u. If this is the case, and at time $\alpha time_u(r)$ there are more than $\log itime$ messages of type r_w lying on u, then u switches to state R at time $8\alpha h' + h$. Otherwise a node can only switch to state R if it receives r from a neighbor being already in state R.

As described in 5.1.6. and 5.2. a node can switch directly to special state R' at time $32\alpha h'$, if it has seen 5 different nodes with the properties described above, but it does not satisfy the other condition which would allow him to switch to special state R; or it receives r from a node being already in special state R'.

If a node u is in special state R, then it executes a procedure similar to the one described in the previous section, however, *itime* is only updated in state A, or if some received τ_i' is larger than and not s_2-equivalent with τ' of u. Moreover, if some received τ_i' is larger than and not s_2-equivalent with τ' of u, or at time $8\alpha h' + h$ it holds that $\tau'' > \log itime$, then the own τ' is also updated and u switches (back) to state A. This implies that most nodes run $\Omega(\log(\log n/\log d))$ times through states A, G, and S, and before switching to state R' (in time step $16\alpha \cdot h'$) their τ' equals $\Omega(\log n/\log d)$.

In state R' the nodes getting the largest τ' values perform the same procedure as described in the previous section, and transmit r to all nodes in the graph.

Now we analyze the runtime and number of transmissions generated by the algorithm described in this section. In our proofs, we only concentrate on the case $\log n/\log d \geq \log\log n$, and assume, as in the previous section, that if $|I(t)| \leq n/2$, then r is only transmitted by push transmissions. If $|I(t)| > n/2$, then r is only transmitted by pull transmissions. First, we show in the next lemma that any node switching to special state R on his own has $itime = \Theta(\log n)$.

LEMMA 7. *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, 20] 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 $p \geq \sqrt{n}/n$, 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_1 be the time step defined in the proof of Lemma 5. (Due to Lemmas 2-4, $I(t)$ will exceed $n/2$ even if *itime* is never updated in state G.) Then, with constant probability, a node in $I(t)$ with some $t > t_1$ 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 on themself, and the lemma follows. ☐

Lemma 7 implies that vertices exist, which will switch to state R, and all these vertices have $itime = \Theta(\log n)$ and $\tau'' = \Omega(\log\log n)$ (please remember that we only analyze the case when $\log n/\log d > \log\log n$).

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

LEMMA 8. *There are at most $O(n/\log n)$ nodes in state R, which transmit r for more than $\omega(\log n/\log d)$ steps, w.h.p.*

PROOF. For simplicity we assume that any node which checks the number of r_w's on itself has the same h'. 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 $\tau_1 < \tau_2$. Furthermore, let U_1 and U_2 denote the set of vertices with $\tau'' = \tau_1$ and $\tau'' = \tau_2$, respectively. Due to the results of [24, 12], $n/\log^{\Omega(\log\log n)} n = |U_1| > |U_2|$. Since the nodes of U_1 set $\tau' \sim_{s_2} \tau_1$ in step $8\alpha h' + \lfloor \log \tau_1 \rfloor \alpha h'/\log h'$, and the nodes of U_2 set $\tau' \sim_{s_2} \tau_2$ in step $8\alpha h' + \lfloor \log \tau_2 \rfloor \alpha h'/\log h'$, we conclude that when $\Theta(n/\log n)$ nodes are in special state R and have $\tau' \sim_{s2} \tau_1$, then at most $n/\log^{\Omega(\log\log n)} n$ nodes have $\tau' \sim_{s_2} \tau_2$. Then, for each $i = \lfloor \log\log\log n \rfloor, \ldots, \lceil \log\log n - \log\log d \rceil$ at least $n - O(n/\log n)$ nodes will switch to state G after being active for at most $O(\tau')$ rounds, where τ' is in the range $[2^i, 2^{i+1}]$.

Since *itime* will not be updated at a node after the node switches to state G (excepting when τ' has to be significantly increased), $n - O(n/\log n)$ nodes are transmitting for at most $\sum_{i=0}^{\log O\left(\frac{\log n}{\log\log n \cdot \log d}\right)} 2^i \log\log n = O(\log n/\log d)$ steps in special state R. ☐

Lemma 8 implies that the number of transmissions is less than $O(n \log n/\log d)$ as long as the vertices are in state R.

In the next lemma we show that, with probability $1 - 1/n^{\Omega(1)}$, at least $n - O(n/\sqrt{d})$ 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 9. *Let G_p be a random graph with $p \geq \log^\delta n/n$ and perform the algorithm presented at the beginning of this section on 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 S and special state R' after a total number of $O(\log n)$ steps, with probability $1 - 1/n^{\Omega(1)}$.*

PROOF. Again, we assume for simplicity that all nodes which check the number of r_w's on them have the same h' value. Lemma 2-5 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_w's occurring on a node in the system, and let u be the node with the smallest $time_u(r)$ checking the number of r_w's lying on itself. Then, with probability $1 - 1/n^{\Omega(1)}$, there is a node u' with $2^{\lfloor \log time_u(r) \rfloor} = 2^{\lfloor \log time_{u'}(r) \rfloor}$, which checks the number of r_w's 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 special state R', and start to transmit r at time $16\alpha h'$. We can now apply Lemmas 4-6, and conclude that after $O(\log n/\log d)$ steps all vertices have r, and are in special state R' with $\tau' \geq \tau_{\max}/4$. Then, if α is large enough, all vertices will have switched to state G within additional $O(\log n/\log d)$ steps.

The arguments above imply that now the difference between the smallest and largest *itime*'s is at most $O(\frac{\log n}{\log d})$. Since $\tau' = O(\log n/\log d)$, we obtain the lemma. □

Now we summarize the results in the following theorem.

THEOREM 1. *Let G_p be a random graph with $p \geq \log^\delta n/n$. The algorithm presented in this section informs all nodes of the graph within $O(\log n)$ steps, whereby the number of message transmissions is bounded by $O(n \max\{\log\log n, \frac{\log n}{\log d}\})$, with probability $1 - 1/n^{\Omega(1)}$.*

Using the techniques of [21] it can be shown that the results of Theorem 1 are asymptotically optimal.

THEOREM 2. *Let G_p be a random graph with $p \geq \log^\delta n/n$ and assume that an information r is placed on one of the nodes at time 0. Furthermore, we assume that in any succeeding step, each node is allowed to choose one neighbor, uniformly at random, and to establish a communication channel with this neighbor. Now, even if every node is allowed to transmit r in both directions along an incident communication channel, then any broadcasting algorithm obeying the rules above needs at least $\Theta(\log n)$ time steps, w.h.p., to spread r to all nodes of G_p. Moreover, any such time efficient broadcasting algorithm produces, with high probability, at least $\Omega(n \max\{\log\log n, \log n/\log d\})$ transmissions of r.*

PROOF. We start the proof with the first statement of the theorem, and show that any algorithm satisfying the assumptions of the theorem requires at least $\Omega(\log n)$ time steps to inform $\sqrt{n}$ nodes of G_p. Obviously, such a broadcasting algorithm has fastest performance, if all informed nodes transmit the information in any time step (we do not consider the number of transmissions at this time). If $I(t)$ denotes the set of informed nodes at time t, then due to [17] there can be at most $|I(t)|$ nodes, which are informed in step $t+1$ by push transmissions. Since any node has $d(1 \pm o(1))$ neighbors in G_p, w.h.p. [3], and each node contacts a neighbor independently and uniformly at random, we apply the Chernoff bounds [4, 20] to conclude that there can be at most $O(\log n)$ different nodes contacting a node $v \in V$ in some step t, with probability $1 - o(1/n^2)$. Therefore, it is sufficient to show that $|I(t+1)| = O(|I(t)|)$ in the case when r is only transmitted by pull transmissions, and $|I(t)| \geq \log^q n$, where q is a large constant.

Now let us analyze the distribution of the edges between $I(t)$ and $H(t)$. If $|I(t)| \leq \sqrt{n}$, then there can be at most $|I(t)|d(1 + o(1))$ edges between $I(t)$ and $H(t)$ (w.h.p.) [3], and let $N(I(t))$ be the set of nodes, which are in $H(t)$ but have at least one neighbor in $I(t)$. Furthermore, let $d_{I(t)}(v_j)$ be the number of neighbors of some node $v_j \in N(I(t))$ in $I(t)$. Then, v_j becomes informed in step $t+1$ with probability $d_{I(t)}(v_j)/d(1 \pm o(1))$ (please note that we consider the spread of information by pull transmissions only). Now, since any uninformed node chooses some neighbor independently, and the expected value $E(|I(t+1)| - |I(t)|) \leq |I(t)|(1+o(1))$, applying the Chernoff bounds [4, 20] we conclude that $|I(t+1)| - |I(t)| = O(|I(t)|)$, whenever $|I(t)| \geq \log^q n$ with q large enough.

In the following paragraphs we analyze the number of transmissions. We allow an algorithm to execute at most $q \log n$ broadcasting steps, where q is a large constant. We show that there is a constant ϵ such that at least $\epsilon n \frac{\log n}{\log d}$ transmissions of r are needed in order to inform all nodes of the graph.

Let $I(t)$ denote the set of informed nodes at time t, and let $I_r(t) \subseteq I(t)$ be the set of nodes transmitting r at time t. Let ϵ' be a small constant, and let U be the set of vertices such that no node of U is informed within the first $\epsilon' n$ vertices, and it contains an 1-factor of size $\Theta(|U|)$. Let U' be the set of vertices belonging to this 1-factor, and let M' be the 1-factor we consider. Now, two vertices $v, v' \in U'$ with $(v, v') \in M'$ call each other (i.e., v chooses v', and v' chooses v) in some step t with probability $\Theta(1/d^2)$. If t_1 denotes the number of steps in which more than $\epsilon' n$ nodes are transmitting r, and t_2 is the number of steps in which at most $\epsilon' n$ nodes are transmitting, then the probability that v and v' choose eachother in every step, in which more than $\epsilon' n$ nodes are transmitting r, is $1/d^{\Theta(2t_1)}$. The probability that in these steps no node will choose v or v' is $\Omega(1)$.

On the other hand, Lemma 1 implies that whenever $|I(t)| - |I_r(t)|$ informed vertices decide not to transmit the information r in step $t+1$, each node of U has at least $p(n - |I_r(t)|) - O(\sqrt{p(n - |I_r(t)|)\log n} + \log n)$ neighbors in $V \setminus I_r(t)$, w.h.p. (Hereby, we used the fact that if a node decides not to transmit, it cannot know that it has no edges to the set $H(t)$, and hence Lemma 1 applies.) Therefore, if $|I_r(t)| \leq \epsilon' n$, nodes v and v' do not establish connection with a vertex transmitting r with some constant probability $p_{\epsilon'}$. Decreasing now ϵ', the probability $p_{\epsilon'}$ can be increased to any constant between 0 and 1.

Summarizing, if $t_1 \leq \epsilon \log n/(\epsilon' \log d)$, and $t_2 \leq q \log n$, where ϵ and ϵ' are two properly chosen constants, then v and v' remain uninformed with some probability $\omega(1/n)$. Since $|U'| = \Theta(n)$, the claim follows.

Similar techniques can also be used to show that the algorithm requires at least $\epsilon \cdot n \log\log n$ transmissions of r, if $d \geq n^{1/\log\log n}$, where ϵ is a small constant. We omit this case here due to space limitations. □

4. APPLICATION TO OTHER CLASSES OF RANDOM GRAPHS

In this section, we discuss the applicability of our algorithms in certain classes of more general random graph models. First, we consider the genralized random graph model of [5]. Then, we briefly extend the analysis to some modified dynamic random power law graph models [1, 2].

In [5], Chung and Lu generalized the classical random graph model to a general model with arbitrary degree distributions: 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 (v_i, v_j) with probability $d_i d_j / \sum_{k=1}^n d_k$. If now the degree distribution $\mathbf{d}$ obeys a power law, then the resulting graph is well suited for modelling power law graphs.

As described in the introduction, the results of the previous sections can be generalized to certain types of truncated power law graphs described by the $G(\mathbf{d})$ random graph model. If $d_{\min}$ is larger than $\log^\delta n$, where $\delta > 2$ is a constant, and the number of vertices with expected degree d_i is proportional to $d_i^{-\beta}$, $\beta > 2$, then we can apply the algorithm presented in Section 2, whenever the nodes are supposed to have an estimate of $\log n / \log d_{\min}$, and obtain for these graphs the same results as in Section 2.

Now, in order to generalize the result of Section 3 to the $G(\mathbf{d})$ model, we have to modify the algorithm. If we applied the algorithm of Section 3 to a graph $G(\mathbf{d})$, in which $d_{\min} \geq \log^\delta n$ and the number of vertices with expected degree d_i is proportional to $d_i^{-\beta}$, $\beta > 2$, then the nodes with largest expected degree would attract most messages of type r_w, and τ'' would significantly exceed the value $\frac{\log n}{\log d_{\min}}$.

We can avoid this effect by letting at the beginning any node w send out a special message r'_w which will perform random walks in the system, too. Then, after a sufficient number of time steps, any such message lies on a node w' with probability $d_{w'} / \sum_{k=1}^n d_k$ [13, 24]. Therefore, all nodes with degree $d_{\min}^{2+\Omega(1)}$ will possess messges of type r'_w, with high probability. Now, if we only count and consider the number of messages of type r_w on the nodes which do not possess any message of type r'_w, then we are able to estimate $\log n / \log d_{\min}$ as in the algorithm described in Section 3. By using this modified algorithm, we obtain the result of Section 3 for these kind of graphs.

The same results can also be extended to certain modifications of the random power law graph models [1, 2] by combining the coupling methods of [7] with the techniques described in the previous sections. However, in order to be able to apply the methods of this paper, the minimal degree in these graphs has to be larger than $\log^\delta n$. We should note that we explicitely use here the assumption that the minimal degree in the graph is bounded by $\log^\delta n$, and we could not generalize the result for the case when this assumption is not fulfilled.

5. CONCLUSION

In this paper, we analyzed the performance of randomized broadcasting algorithms in random-like graphs. We have shown that algorithms exist, which are able to broadcast a message to all nodes of a random graph G_p within $O(\log n)$ steps by using $O(n \max\{\log\log n, \frac{\log n}{\log d}\})$ transmissions, with probability $1 - 1/n^{\Omega(1)}$, whenever $p \geq \log^\delta n/n$, where δ is a large constant. We have also shown that this result holds even if the nodes do not know anything about the size or degree of the graph.

We briefly discussed the applicability of similar algorithms in certain truncated power law graphs, in which the minimal degree is bounded by $\log^\delta n$. However, the problem of whether the results of this paper can be generalized to traditional power law graph models is still open. Nevertheless, we hope that this paper provides insights at more general level, too.

Acknowledgment

The author would like to thank the referees whose helpful comments and observations greatly improved the paper.

6. REFERENCES

[1] W. Aiello, F.R.K. Chung, and L. Lu. Random evolution in massive graphs. In *Proc. of FOCS*, pages 510–519, 2001.

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

[3] B. Bollobás. *Random Graphs*. Academic Press, 1985.

[4] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 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 and L. Lu. Coupling on-line and off-line analyses for random power law graphs,. *Internet Mathematics*, 1:409–461, 2004.

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

[9] F.R.K. Chung, L. Lu, and V. Vu. The spectra of random graphs with given expected degrees. *Proceedings of National Academy of Sciences*, 100:6313–6318, 2003.

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

[11] R. Elsässer. On randomized broadcasting in power law graphs. *Manuscript*, 2006.

[12] 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.

[13] R. Elsässer, B. Monien, and S. Schamberger. Load balancing of indivisible unit size tokens in dynamic and heterogeneous networks. In *Proc. of ESA*, pages 640–651, 2004.

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

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

[16] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *Computer Communication Review*, 29:251–263, 1999.

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

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

[19] E.N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30:1141–1144, 1959.

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

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

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

[23] 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.

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

[25] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

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