Theoretical Computer Science 410 (2009) 3414–3427

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the runtime and robustness of randomized broadcasting ☆

R. Elsässer, T. Sauerwald *

University of Paderborn, Inst. for Comp. Sci., 33102 Paderborn, Germany

ARTICLE INFO

Keywords:
Randomized algorithm
Distributed computing

ABSTRACT

In this paper, we study the following randomized broadcasting protocol. At some time t an information r is placed at one of the nodes of a graph. In the succeeding steps, each informed node chooses one neighbor, independently and uniformly at random, and informs this neighbor by sending a copy of r to it. We begin by developing tight lower and upper bounds on the runtime of the algorithm described above. First, it is shown that on Δ-regular graphs this algorithm requires at least $\log_{2-\frac{1}{\Delta}} n + \log_{\left(\frac{\Delta}{\Delta-1}\right)^{\Delta}} n - o(\log n) \approx 1.69 \log_2 n$ rounds to inform all n nodes. Together with a result of Pittel [B. Pittel, On spreading a rumor, SIAM Journal on Applied Mathematics, 47 (1) (1987) 213–223] this bound implies that the algorithm has the best performance on complete graphs among all regular graphs. For general graphs, we prove a slightly weaker lower bound of $\log_{2-\frac{1}{\Delta}} n + \log_4 n - o(\log n) \approx 1.5 \log_2 n$, where Δ denotes the maximum degree of G. We also prove two general upper bounds, $(1+o(1))n \ln n$ and $\mathcal{O}(n\frac{\Delta}{\delta})$, respectively, where δ denotes the minimum degree.

The second part of this paper is devoted to the analysis of fault-tolerance. We show that if the informed nodes are allowed to fail in some step with probability $1-p$, then the broadcasting time increases by at most a factor $6/p$. As a by-product, we determine the performance of agent based broadcasting in certain graphs and obtain bounds for the runtime of randomized broadcasting on Cartesian products of graphs.

1. Introduction

1.1. Motivation and related work

The study of information spreading in large networks has various fields of application in distributed computing. Consider for example the maintenance of replicated databases on name servers in a large network [5]. 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 content, efficient and fault-tolerant broadcasting algorithms have to be developed.

Another well known example occurs in the analysis of epidemic disease. Often, mathematical studies about infection propagation make the assumption that an infected person spreads the infection equally likely to any member of a population [17], which leads to a complete graph for the underlying network. The question of how fast the disease infects everyone reduces the problem to randomized broadcasting. However, in most of these papers, spreaders are only active in a certain time window, and the question of interest is, whether on certain networks modeling personal contacts, an epidemic outbreak

☆ Parts of the results have been presented at the 17th International Symposium on Algorithms and Computation (ISAAC'06).

* Corresponding author. Tel.: +49 5251 60 6725; fax: +49 5251 60 6697.
E-mail addresses: elsa@upb.de (R. Elsässer), sauerwal@upb.de (T. Sauerwald).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.04.017

occurs. Several threshold theorems involving the basic reproduction number, contact number, and the replacement number have been derived. See e.g. [14] for a collection of results concerning the mathematics of infectious diseases.

There is an enormous amount 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 only concentrate on the efficiency of randomized broadcasting and mainly consider the runtime of the so called *push algorithm* [5] defined as follows. In a graph $G = (V, E)$ of size $n := |V|$, we place at some time t an information r on one of the nodes. Then, in every succeeding time step, each *informed* vertex sends a copy of the information r to one of its neighbors selected independently and uniformly at random.

The advantage of randomized broadcasting is in its inherent robustness against several kinds of failures and dynamical changes compared to deterministic schemes that either need substantially more time [10] or can tolerate only a relatively small number of faults [18]. Most papers dealing with randomized broadcasting analyze the runtime of the push algorithm in different graph classes. Pittel [20] proved that with a certain probability, information is spread to all nodes in a complete graph within $\log_2 n + \ln n + o(\log n)$ steps. Feige et al. [9] determined tight upper bounds of $\mathcal{O}(\Delta(\text{diam} + \log n))$ and $\mathcal{O}(n \log n)$, respectively, for general graphs, where Δ denotes the maximum degree of G. Furthermore it was shown that in random graphs and Hypercubes of size n, all nodes of the graph receive the information within $\mathcal{O}(\log n)$ steps, with high probability.[1]

In [7] we considered the performance of the push algorithm in a Cayley graph known as the Star graph [1]. The d dimensional Star graph S_d has $n = d!$ vertices corresponding to the $d!$ permutations of $(1, 2, \ldots, d)$, and there is an edge between $(x_1, \ldots, x_d)$ and $(y_1, \ldots, y_d)$ iff an index $i \in \{2, \ldots, d\}$ exists such that $x_1 = y_i$, $x_i = y_1$, and $x_j = y_j$ for any $j \neq 1, i$. We have shown in [7] that in these graphs all nodes become informed within $\mathcal{O}(\log n) = \mathcal{O}(d \log d)$ steps by the push algorithm, w.h.p. This result was recently generalized in [8] to a class of Cayley graphs which also contains the Pancake and Transposition graph. For the d dimensional Bubble Sort graph an asymptotically optimal upper bound of $\mathcal{O}(d^2)$ was established. Furthermore, we proved that the runtime of the push model is upper bounded by the mixing time of a certain random walk and an additional logarithmic factor on any graph.

A model related to the push algorithm has been introduced in [5] and is called *pull algorithm*. Here, any (informed or uninformed) node is allowed to call a randomly chosen neighbor, and the information is sent from the called to the calling node. Note that these kinds of transmissions make only sense if new or updated data occur frequently in the network so that every node places a random call in each round anyway.

It was observed in complete graphs of size n that the push algorithm needs at least $\Omega(n \log n)$ transmissions to inform all nodes of the graph, w.h.p. However, in the case of the pull algorithm if a constant fraction of the nodes are informed, then within $\mathcal{O}(\log \log n)$ additional steps every node of this graph is informed as well, w.h.p. [5,16]. This implies that in such graphs at most $\mathcal{O}(n \log \log n)$ transmissions are needed if the distribution of the information is stopped at the right time. Using this fact, Karp et al. [16] combined the push and pull algorithms, and introduced a termination mechanism to bound the number of total transmissions by $\mathcal{O}(n \log \log n)$ in complete graphs. Furthermore they showed that this result is asymptotically optimal among these kinds 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.

In [6], we introduced the so-called *agent based broadcasting model*. In this model, at the beginning n agents are distributed among the nodes and in each of the following steps, these agents jump from one node to another via edges chosen uniformly at random. An information r placed initially on one node is carried by the agents to other vertices. If an agent visits an informed node, then the agent becomes informed, and any node visited by an informed agent becomes informed as well. It was shown that $\mathcal{O}(\log n)$ steps are sufficient to distribute r among all nodes in random graphs. We also considered the performance of this model in bounded degree graphs and compared it to the behavior of the push algorithm on different examples.

1.2. Our results

We present a short overview of the most important new results of this paper and briefly discuss their relationship to previous results. All the following results refer to the push algorithm.

- We prove for Δ-regular graph a lower bound of $\log_{2-\frac{1}{\Delta}} n + \log_{(\frac{\Delta}{\Delta-1})^{\Delta}} n - o(\log n)$. This is matched by the result of Pittel [20], which says that the runtime for complete graphs is $\log_2 n + \ln n \pm o(\log n)$.
- For non-regular graph, we show a slightly weaker lower bound of $\log_{2-\frac{1}{\Delta}} n + \log_4 n - o(\log n)$.
- For general graphs, we prove an upper bound of $(1 + o(1))n \ln n$. This bound is matched by the graph $K_{n-1,1}$ and significantly improves over the upper bound of $12n \log n$ by Feige et al. [9].
- We consider the performance of broadcasting in the presence of failures. If every vertex fails in some step with probability $1-p$ (independently of all other time steps, but *not* necessarily independently of all other vertices), then the broadcasting time increases by a factor of at most $\frac{6}{p}$.

[1] When we write "with high probability" or "w.h.p." we mean with probability at least $1 - n^{-1}$.