Theoretical Computer Science 520 (2014) 27–42

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Randomised broadcasting: Memory vs. randomness ☆

Petra Berenbrink [a], Robert Elsässer [b], Thomas Sauerwald [c,*]

[a] School of Computing Science, Simon Fraser University, Burnaby B.C. V5A 1S6, Canada
[b] Department of Computer Science, University of Salzburg, Austria
[c] Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 7 September 2011
Received in revised form 18 March 2013
Accepted 19 August 2013
Communicated by R. Klasing

Keywords:
Distributed computing
Randomised algorithm
Broadcasting
Expander graphs

ABSTRACT

In this paper we analyse broadcasting in d-regular networks with good expansion properties. For the underlying communication, we consider modifications of the so-called random phone call model. In the standard version of this model, each node is allowed in every step to open a channel to a randomly chosen neighbour, and the channels can be used for bi-directional communication. Then, broadcasting on the graphs mentioned above can be performed in time $O(\log n)$, where n is the size of the network. However, every broadcast algorithm with runtime $O(\log n)$ needs on average $\Omega(\log n/\log d)$ message transmissions per node for random graphs with expected degree d [11].

In this paper we show that it is possible to save significantly on communications if the standard model is modified such that nodes can avoid opening channels to exactly the same neighbours in two consecutive steps. We consider the so-called Rr model where we assume that every node has a cyclic list of all of its neighbours, ordered in a random way. Then, in step i the node communicates with the i-th neighbour from that list. We provide an $O(\log n)$ time algorithm which produces in average $O(\sqrt{\log n})$ transmissions per node in networks with suitably defined expansion properties. Furthermore, we present a related lower bound of $\Omega(\sqrt{\log n/\log\log n})$ for the average number of message transmissions. These results show that by using memory it is possible to reduce the number of transmissions per node by almost a quadratic factor.

1. Introduction

We consider randomised broadcasting in (almost) regular graphs with good expansion properties. In the broadcasting problem, the goal is to spread a message from one vertex to all vertices of a network. Our interest in these graphs is motivated by overlay topologies in peer to peer (P2P) systems. Important topological properties of these networks include good connectivity, high expansion, and small diameter; all these properties are perfectly fulfilled by the graphs considered here. Our aim is to develop time-efficient broadcasting algorithms which produce a minimal number of message transmissions in the graphs described above. Since P2P systems are significant decentralised platforms for sharing data and computing resources, it is very important to provide efficient, simple, and robust broadcasting algorithms for these overlay networks. Minimising the number of transmissions is important in applications such as the maintenance of replicated databases in which broadcasts are necessary to deal with frequent updates in the system [17,21].

In this paper we assume the so-called *phone call model* (see [21]). In this model, each node v may perform the following actions in every step:

☆ An extended abstract of this paper has appeared in the proceedings of LATIN 2010 [4].

* Corresponding author.
E-mail addresses: petra@cs.sfu.ca (P. Berenbrink), elsa@upb.de (R. Elsässer), sauerwal@mpi-inf.mpg.de (T. Sauerwald).

http://dx.doi.org/10.1016/j.tcs.2013.08.011

1) create a new message to be broadcasted,
2) establish a communication channel between v itself and one of its neighbours,
3) transmit messages over incident channels opened by itself or by some of its d neighbours.

At the end of each step, the nodes close all open channels. Note that open channels can be used for bi-directional (push&pull) communications. In the case of `push` transmissions, calling nodes (i.e., the nodes that opened the channels) send their messages to their neighbours. In the case of `pull` transmissions, messages are transmitted from called nodes to the calling ones (we also say that the **called nodes** perform pull transmissions). Note that nodes do not have to send messages over open channels, they can *choose* to do so. For example, if node v opens a channel to w, w does not have to send a message to v. If many distinct messages are to be spread in the network, then the nodes can combine several broadcast messages to larger ones which can be sent over a channel in one time step.

In the standard phone call model it is assumed that nodes open a channel to a randomly chosen neighbour, and the nodes have to decide whether to transmit a specific message over a channel, without knowing if they opened a channel to the corresponding node in earlier steps [21]. In this paper we assume that every node has a cyclic list of all of its neighbours, ordered in a random way. In step i the node opens a channel to the i (modulo d)-th neighbour from that list. This model is called RR *model* (RR its standing for round robin) in the following. The RR rule prevents a node to open a channel to a neighbour for a second time before it opened a channel to all of its neighbours. Hence, the rule helps nodes to communicate with more of their neighbours.

The question we address in this paper is whether remembering the communication partners from earlier rounds helps or not. We give a positive answer to this question and provide further evidence for the power of memory in randomised broadcasting (see [14]). More precisely, we present an algorithm, and show that w.r.t. the average number of transmissions per node this algorithm performs significantly better than any algorithm in the so-called RANDOM[c]-model introduced in [14] (i.e. we achieve an almost quadratic improvement). RANDOM[c] is similar to the standard random phone call model, however, every node may open channels to c different randomly chosen neighbours simultaneously in each step. Our algorithm is *address oblivious*, i.e., the send decisions of the nodes do not depend on the IDs of the nodes to which they open channels in the actual step. However, the nodes are allowed to remember with which nodes they communicated in the steps before [21].

1.1. Related work

There is a huge amount of work considering epidemic type (broadcasting) algorithms on graph models for P2P overlays. Most of these papers deal with the empirical analysis of these algorithms e.g. [22,25]. Due to space constraints, we can only describe here the results which focus on the analytical study of push&pull algorithms.

Runtime. Most randomised broadcasting results analyse the runtime of the push algorithm. For complete graphs of size n, Frieze and Grimmett [18] present an algorithm that broadcasts a message in time $\log_2(n)+\ln(n)+o(\log n)$ with a probability of $1-o(1)$. Later, Pittel [26] shows that (with probability $1-o(1)$) it is possible to broadcast a message in time $\log_2(n)+\ln(n)+f(n)$ [26], where $f(n)$ can be any slow growing function. In [17], Feige et al. determine asymptotically optimal upper bounds for the runtime of the push algorithm on $G(n,p)$ graphs (i.e., traditional Erdős–Rényi random graphs [15,16]), bounded degree graphs, and Hypercubes. In [13] Elsässer and Sauerwald analyse certain Cayley graphs on which the push algorithm performs (asymptotically) optimal. Boyd et al. [5] consider the combined push&pull model in arbitrary graphs of size n, and show that the running time is asymptotically bounded by the mixing time of a corresponding Markov chain plus an $O(\log n)$ value. In [9] Doerr et al. analyse the so-called quasi-random rumor spreading in an adversarial version of the RR model where the order of the lists is assumed to be given by an adversary. However, the nodes choose a random position in their lists to start with communication. They show for hypercubes and $G(n,p)$ graphs that $O(\log n)$ push steps suffice to inform every node, w.h.p.[1] These bounds are similar to the ones in traditional randomised broadcasting (push model). These results have been extended to further graph classes with good expansion properties [10]. Recently, Doerr et al. showed in [8] that by using the RANDOM[2]-model, one can improve the running time of broadcasting on the so-called preferential attachment graph [1] of size n by a factor of $\log\log n$.

Number of transmissions. Karp et al. [21] show that in complete graphs the pull approach is inferior to the push approach, until roughly $n/2$ nodes receive the message, and then the pull approach becomes superior. They present a `push` and `pull` algorithm, together with a termination mechanism, which reduces the number of total transmissions to $O(n\log\log n)$ (w.h.p.), and show that this result is asymptotically optimal. They also consider communication failures and analyse the performance of their method in cases where the connections are established using arbitrary probability distributions.

For sparser graphs it is not possible to get $O(n\log\log n)$ message transmissions together with a broadcast time of $O(\log n)$ in the standard phone call model. In [11] Elsässer considers random $G(n,p)$ graphs, and shows a lower bound of $\Omega(n\log n/\log(pn))$ message transmissions for broadcast algorithms with a runtime of $O(\log n)$. On the positive side, for

[1] W.h.p. or "with high probability" means with probability $1-o(n^{-1})$.