Theoretical Computer Science 560 (2014) 108–120

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The worst case behavior of randomized gossip protocols ☆

Hervé Baumann [a], Pierre Fraigniaud [a,*], Hovhannes A. Harutyunyan [b], Rémi de Verclos [c]

[a] CNRS and University Paris Diderot, France
[b] Concordia University, Montréal, Canada
[c] ENS Lyon, France

ARTICLE INFO

Article history:
Received 31 October 2012
Accepted 17 August 2014
Available online 23 August 2014

Keywords:
Rumor spreading
Broadcast
Gossip
Random phone-call model

ABSTRACT

This paper considers the quasi-random rumor spreading model introduced by Doerr, Friedrich, and Sauerwald in [10], hereafter referred to as the *list-based* model. Each node is provided with a cyclic list of all its neighbors, and, upon reception of a message, it chooses a random position in its list, and from then on calls its neighbors in the order of the list. This model is known to perform asymptotically at least as well as the random phone-call model, for many network classes. Motivated by potential applications of the list-based model to live streaming, we are interested in its worst case behavior. Our first main result is the design of an $O(m+n\log n)$-time algorithm that, given any n-node m-edge network G, and any source-target pair $s,t\in V(G)$, computes the maximum number of rounds it may take for a rumor to be broadcast from s to t in G, in the list-based model. Hence, the list-based model is computationally easy to tackle in its basic version. The situation is radically different when one is considering variants of the model in which nodes are aware of the status of their neighbors, i.e., are aware of whether or not they have already received the rumor, at any point in time. Indeed, our second main result states that, unless $\mathrm{P}=\mathrm{NP}$, the worst case behavior of the list-based model with the additional feature that every node is perpetually aware of which of its neighbors have already received the rumor cannot be approximated in polynomial time within a $(\frac{1}{n})^{\frac{1}{2}-\epsilon}$ multiplicative factor, for any $\epsilon>0$. As a byproduct of this latter result, we can show that, unless $\mathrm{P}=\mathrm{NP}$, there are no PTAS enabling to approximate the worst case behavior of the list-based model, whenever every node perpetually keeps track of the subset of its neighbors which have sent the rumor to it so far.

1. Introduction

Randomized *rumor spreading*, also known as randomized *gossip*, or *epidemic* protocol, is a simple, scalable and naturally fault-tolerant protocol to disseminate information in networks. It has been proposed for various applications, including, e.g., the maintenance of replicated databases [9], publish-subscribe [8], application-level multicast [4,28], and live streaming [23]. The random *phone-call model* [34], in its *push* variant, captures the essence of such a protocol. Communications proceed in synchronous rounds and, at each round, a node aware of an atomic piece of information selects one of its neighbors

☆ A preliminary version of this paper appeared in the proceedings of the 9th annual conference on Theory and Applications of Models of Computation (TAMC 2012).

* Corresponding author.

http://dx.doi.org/10.1016/j.tcs.2014.08.008

in the network, uniformly at random, and forwards the piece of information to that node. It was noticed in [10] that a quasi-random analogue to the random phone-call model, hereafter referred to as the *list-based model*, where each node is provided with a cyclic list of all its neighbors, chooses a random position in its list, and from then on calls its neighbors in the order of the list, performs asymptotically at least as well as in the random phone-call model. By "performs as well", it is meant that, in expectation, or with a certain probability, the numbers of rounds required for a piece of information initiated at any source node to reach all nodes in the network, are of similar order of magnitude in both models. This holds for a large number of networks classes, even when the lists are given by an adversary.

In fact, results in the literature demonstrate that randomized phone-call rumor spreading and list-based rumor spreading offer performances close to the best that can be achieved for large classes of networks, that is, close to the optimal communication schedule that could be computed by a centralized algorithm aware of the structure of the entire network. For instance, it is known [10,19,25] that, for almost all random graphs in $\mathcal{G}_{n,p}$ with p above the connectivity threshold $\ln n/n$, both protocols perform asymptotically in $O(\log n)$ rounds, with high probability (whp), that is with probability at least $1 - O(1/n^{\alpha})$ for some $\alpha > 0$.

Unfortunately, such a nice behavior of the different variants of randomized rumor spreading is, for many reasons, not sufficient to guarantee a good performance for all kinds of applications. First, the random graph model $\mathcal{G}_{n,p}$ is not necessarily reflecting the structure of real world networks. For arbitrary networks, the behavior of randomized rumor spreading is not precisely known. Upper bounds on the number of rounds required for an information to spread over all nodes are known [6,7,10,11,26,38,39]. However, although these bounds are often tight, they are general and therefore do not necessarily capture the behavior of rumor spreading for each network G. This holds even if one restricts the analysis to networks with given properties such as bounded edge-expansion [38], bounded node-expansion [39], or bounded conductance [6,7,26] (see also [5]). Second, applications like file sharing or data streaming require sending a large number of information pieces — called *packets* or *chunks* or *frames*, depending on the context. In such circumstances, results in expectation, or even whp, may not guarantee that some (or even a constant fraction) of these pieces will not experience high delay, which may result in increasing the packet loss ratio, preventing good reception of files. Indeed, "whp" refers to a certain statistical guarantee expressed as a function of the network size n. Thus, if the number of packets is of a similar or larger order of magnitude than n, then this guarantee may not be sufficient to prevent several "bad events" to occur. Third, applications like audio or video streaming requires to control the jitter, i.e., the difference in time between the reception of two consecutive packets. Again, results on expectation, or even whp, may not allow the designer to calibrate well the size of application buffers so as to make sure that high jitter will not prevent a good reception of the audio or video stream.

It is worth noticing that enforcing higher probability $1 - O(1/n^{\alpha})$ of success for each packet, i.e., increasing α, does not necessarily provide a solution to the above concerns. Indeed, for doing so, one must assume that the analysis of the protocol allows us to establish explicit tradeoffs between the probability of success and the bound on the number of rounds for the specific network, or class of networks, we are dealing with. Moreover, it may happen that explicit tradeoffs are not tight enough, in the sense that increasing the probability of success above the desired threshold may result in a bound on the number of rounds that actually exceeds the worst case behavior of the randomized gossip protocol. For instance, it is proved in [10] that list-based gossip propagates a rumor to all nodes of the hypercube in at most $1542 \cdot \log n$ rounds, with probability at least $1 - 1/n$. This bound on the number of rounds is above the trivial bound $\log^2 n$ on the worst case behavior of the list-based protocol in hypercubes, for all reasonable values of the number of nodes.[1]

So, in order to address the above concerns, this paper tackles the question of how bad randomized rumor spreading can be.

More specifically, given a network G, we are interested in computing the worst case behavior of randomized rumor spreading in G. Since the general version of randomized gossip does not prevent a packet to ping-pong between two adjacent nodes for an arbitrarily large number of rounds, we perform our investigation in the framework of list-based randomized rumor spreading. In this context, the worst case behavior of the protocol is given by a worst case choice for the cyclic lists of neighbors provided to the nodes, combined with a worst case choice for the starting positions in these lists. Let us denote by $t_{\text{sn}}(G, s, t)$ the maximum number of rounds it can take for a packet initiated at source s to reach target t in network G when applying the list-based randomized rumor spreading, and let $t_{\text{sn}}(G) = \max_{s,t \in V(G)} t_{\text{sn}}(G, s, t)$. The index "sn" stands for "skip none", as every node visits its list blindly, sending the information to each of its neighbors, ignoring the fact that it may be aware that some neighbors have already received the information from other nodes. Thus, $t_{\text{sn}}(G)$ is an absolute upper bound on the maximum delay between two consecutive packets spread in the network using the randomized list-based protocol. Of course, this bound captures only delays caused by the protocol itself, and ignores other reasons that may cause delay, like traffic congestion, which are beyond the scope of this paper.

It is easy to come up with networks for which $t_{\text{sn}}(G) = \Omega(n)$ (e.g., complete networks and star networks), and with other networks for which $t_{\text{sn}}(G) = O(\log n)$ (e.g., complete binary trees). However, computing $t_{\text{sn}}(G)$ for an arbitrary network G does not appear as easy (see Section 2.1). The analysis becomes even more tricky when one is dealing with natural optimization of the list-based protocol. For instance, at every round, every node could skip sending a piece of information to neighbors from which it had received the same piece during previous rounds, including the neighbor from which it

[1] Although the main concern of [10] was not necessarily optimizing the constant in front of the $\log n$ factor, we stress that even diminishing this constant by several orders of magnitude would still result in a bound greater than $\log^2 n$ for reasonable values of n.