Bit-complexité des protocoles de gossip

Pierre Fraigniaud, Giakkoupis George

Bit-complexité des protocoles de "gossip"†

Pierre Fraigniaud[1] et George Giakkoupis[1]

[1] *CNRS et Université Paris Diderot. Email :* firstname.lastname@liafa.jussieu.fr

Nous étudions le problème du *gossip* (i.e., diffusion de rumeurs) dans le modèle des appels aléatoires. Considérons n noeuds communiquant en parallèle par étape. A chaque étape, un ensemble (potentiellement vide) de *rumeurs* est généré à chaque noeud, la même rumeur pouvant être générée simultanément par plusieurs noeuds. L'objectif est de diffuser ces rumeurs à tous les noeuds. Pour ce faire, à chaque étape, chaque noeud appelle un autre noeud choisi uniformément aléatoirement parmi l'ensemble de tous les noeuds, et un noeud ne peut alors communiquer qu'avec le noeud qu'il a appelé, et les noeuds qui l'ont potentiellement appelé. Dans ce modèle, Karp et ses co-auteurs [10] ont montré qu'aucun algorithme de gossip ne peut être à la fois optimal en temps (i.e., s'exécuter en $O(\log n)$ étapes) et en volume de communication (i.e., s'exécuter en transmettant au plus $O(n)$ messages). En particulier, ils ont montré que tout algorithme de gossip n'utilisant pas les IDs des noeuds et diffusant toute rumeur en $O(\log n)$ étapes doit échanger $\Omega(n \log\log n)$ messages par rumeur. Karp et ses co-auteurs ont également montré que ce compromis peut être atteint.

Dans cet article, nous étudions le volume de communication estimé en nombre de bits échangés plutôt qu'en nombre de messages. Nous montrons tout d'abord que tout algorithme de gossip n'utilisant pas les IDs des noeuds et diffusant toute rumeur en $O(\log n)$ étapes doit échanger $\Omega(n(b+\log\log n))$ bits pour diffuser une rumeur de b bits. Nous proposons alors un algorithme de gossip n'utilisant pas les IDs des noeuds qui diffuse toute rumeur en $O(\log n)$ étapes, en échangeant $O(n(b+\log\log n \log b))$ bits pour une rumeur de b bits. Ces résultats démontrent que contrairement à ce qu'il peut sembler lorsque l'on mesure le volume de communication en nombre de messages, il est possible d'être optimal en temps (i.e., s'exécuter en $O(\log n)$ étapes) tout en limitant le volume des communication à $O(nb)$ bits par rumeur, sauf pour des rumeurs extrêmement petites, de taille $b \ll \log\log n \log\log\log n$ bits.

1 Introduction

We study the problem of information spreading in a distributed environment where information is exchanged using randomized communication. Suppose n players communicate in parallel rounds, where in each round every player *calls* a randomly selected communication partner. Each player u is allowed to exchange messages during a round only with the player that u called, and with all the (zero or more) players that called u, in that round. This communication model is often referred to as the *random phone-call model* [10]. In every round, zero or more pieces of information, called *rumors*, are generated, and each rumor is placed to one or more players, the *sources* of the rumor. The goal is that each rumor be distributed among all players within a small number of rounds from the round that the rumor was generated, and by using a small amount of communication between players.

A motivating example for this problem is the maintenance of replicated databases, for instance, on name servers in a large corporate network [2]. In such a system, updates are injected at various nodes and at various times, and these updates must be propagated to all nodes in the network. It is desirable that all databases converge to the same content quickly, and with little communication overhead. The motivation for using a randomized communication model is that such a scheme is simple, scalable, and naturally fault tolerant [2, 8].

A simple rumor-spreading algorithm for the random phone-call model is the so-called *push* algorithm. A rumor r is spread as follows : In each round, starting from the round in which r is generated, every *informed* player u (i.e., every player who knows r) forwards r to the player v that u calls in that round ; we say that u

†Les deux auteurs ont reçu le soutien des projets ANR ALADDIN et PROSE, et du projet INRIA GANG. Une version complète de cette note est à paraître dans les actes du 22ème ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Santorini, Greece, June 13-15, 2010.

pushes r to v. The distribution of r is terminated after $\Theta(\log n)$ rounds, at which time all players know r with high probability [9, 11]. The runtime of the push algorithm is asymptotically optimal for the random phone-call model. However, the algorithm suffers from high communication overhead, requiring $\Theta(n \log n)$ transmissions of the rumor. Intuitively, the number of informed players roughly doubles in each round, until a constant fraction of the players is informed ; and in each subsequent round, the number of non-informed players halves. Thus, in the last $\Theta(\log n)$ rounds $\Theta(n)$ players push the rumor in each round.

The *push-pull* rumor-spreading algorithm, proposed in [10], achieves the same time complexity as the push algorithm, with a smaller communication overhead. A rumor r is distributed as follows : In each round from the round when r is generated, every informed player u pushes r to the player that u called in this round, as in the push algorithm, and, in addition, u forwards r to every player v that called u in this round ; we say that r is *pulled* from u to v. In the basic version of this algorithm, where a rumor is assumed to have a single source, the distribution of r is terminated after $\log_3 n + \Theta(\log \log n)$ rounds. By that time, with high probability, all players know r, and r has been transmitted $\Theta(n \log \log n)$ times. The intuition is that the push and pull transmissions roughly triple the number of informed player in each round until a constant fraction of the players is informed, and, from this point on, the pull transmissions shrink the fraction of non-informed players from s_{t-1} to $s_t = s_{t-1}^2$, in each round t. Thus, only $\Theta(\log \log n)$ additional rounds are required after a constant fraction of players is informed. Note that when more than one sources per rumor are possible, the message complexity may be as bad as $\Theta(n \log n)$—e.g., when there are $\Theta(n)$ sources. Another version of the push-pull algorithm, also proposed in [10], uses a more robust termination criterion which detects when a large fraction of the players is informed, and requires $O(\log n)$ rounds and $\Theta(n \log \log n)$ messages, regardless of the number of sources per rumor.

On the lower-bound side, it is known that no decentralized rumor-spreading algorithm for the random phone-call model requiring $O(\log n)$ rounds and $O(n)$ messages can guarantee that a rumor is spread to all players with constant probability [10]. In other words, it is not possible to achieve simultaneously optimality both in terms of the running time and the message complexity in the random phone-call model. Moreover, for the case of *address-oblivious* algorithm, such as the push and push-pull algorithms above, $\Omega(n \log \log n)$ messages are required, regardless of the number of rounds [10]. So, the push-pull protocol is asymptotically optimal among the address-oblivious algorithm in terms of time and message complexity.

In this paper, we investigate the communication complexity of rumor spreading in the random phone-call model, measured in terms of the number of *bits* exchanged between players. The standard approach to measuring the communication complexity has been in terms of messages, counting one message for every quadruplet (r,t,u,v) such that information for rumor r is exchanged in round t between players u and v. In the rumor-spreading algorithms that have been proposed each such exchange of information typically involves the actual rumor r, plus the values of some small counters, such as the age of the rumor. Arguably, for some applications the volume of information exchanged is at least as relevant as the number of messages, and trying to minimize the number of bits exchanged, in addition to the number of messages, is desirable. This is especially true when a large number of rumors are spread simultaneously, or when rumors are large.

Related work. The problem of randomized rumor spreading was introduced in [9], where the runtime of the push algorithm in the random phone-call model was analyzed. (It was later refined in [11]). Randomized rumor spreading in the setting where players correspond to nodes in a graph (other than the complete graph), and in each round a player chooses its communication partner at random *among its graph neighbors*, was first studied in [8]. There, bounds on the runtime of the push algorithm in arbitrary graphs were derived, and the runtime of the same algorithm in the hypercube and in random graphs was analyzed. The runtime and message complexity of randomized rumor spreading in random graphs were also studied in [6, 7], where a push-pull algorithm was analyzed, as well as two variations of it where players can remember their recent connections, or they initiate multiple calls per round. Push-pull algorithms have also been proposed and analyzed for random d-regular graphs [1], and for scale-free graphs [5]. In [3], a quasirandom analogue to the random phone-call model was introduced, where each player has a cyclic list of all the players (or of all its neighbors, in case of rumor spreading in a graph). A player initially calls a player at a random position in her list, but from then on she calls her neighbors in the order of the list. The push algorithm in the quasirandom model performs asymptotically at least as well as in the random model, for all the cases of graphs studied in [8], even when the lists are given by an adversary. Rumor spreading in the quasirandom model was further explored in [4].

2 Precise description of the problem

As said before, we use the communication model known as the random phone-call model [10]. In each round, an adversary generates a (possibly empty) set of rumors, and places each rumor r to a non-empty subset of players, the *sources* of r. A rumor is just a binary string, and any binary string of any size represents a possible rumor ; so, there are exactly 2^b distinct rumors of size b. No limit is imposed on the number of rumors generated in a round. However, we assume that rumors generated in two different rounds t_1, t_2 with $|t_1 - t_2| = O(\log n)$ are distinct (this assumption is made to simplify the exposition of our algorithm, and can be relaxed). If player u calls a player v and a rumor r is transmitted from u to v we say that r is *pushed*, while if r is transmitted from v to u we say that r is *pulled*.

We measure the bit communication complexity of rumor spreading, that is the total number of bits exchanged between players. Specifically, in our rumor-spreading algorithm, each message exchanged is either related to a single rumor, or to a set of rumors of the same size. In the latter case, to count the bits communicated per rumor we divide the size of the message by the size of the set of rumors. For the lower bound, we assume that a set of b-bit rumors are started by a single source at a round t, and that no other rumors are generated. We also assume that the rumors' size b, their source, and round t are known to all players. To count the bits communicated per rumor, we count the total number of bits exchanged between players, from round t until the distribution of rumors finishes, and then divide by the number of rumors.

We focus on the class of *address-oblivious* algorithms. That is, when player u calls player v, u and v do know the id of each other. Of course, they can communicate their ids, but this exchange of information is also counted in the bit communication complexity.

3 Our results

As we saw above, no rumor-spreading algorithm in the random phone-call model can be both time-optimal, requiring $O(\log n)$ rounds, and message-optimal, requiring $O(n)$ messages per rumor. We show that the situation is different when bit communication complexity is considered in place of message complexity. Specifically, we describe an address-oblivious algorithm that requires $O(\log n)$ rounds and $O(n(b + \log\log n \log b))$ bits of communication, in order to distribute a b-bit rumor among all players with high probability. Also, $O(n \log\log n)$ messages per rumor are used. These guarantees hold even when the rumors are generated by an adversary. On the lower-bound side, we establish that any address-oblivious algorithm performing in $O(\log n)$ rounds requires $\Omega(n(b + \log\log n))$ communication bits to spread a b-bits rumor to all players with constant probability. These two results imply that it is possible to get optimal running time $O(\log n)$ rounds, with $O(nb)$ bit communication complexity per rumor, except for very small rumor sizes $b \ll \log\log n \log\log\log n$.

Theorem 1 *There is an address-oblivious algorithm guaranteeing that, with high probability, any rumor is distributed to all players within $O(\log n)$ rounds and with $O(nb + n\log\log n \log b)$ bits of communication, where b is the rumor's size.*

Theorem 2 *For any $b \geq 1$, no address-oblivious algorithm can guarantee that for any rumor of size b, this rumor is distributed to all players within $O(\log n)$ rounds, with constant probability, and $o(nb + n\log\log n)$ bits of communication are used, in expectation.*

4 Sketch of proofs

Our rumor-spreading algorithm can be described as a push-pull algorithm with "concise" feedback. Note that the original push-pull algorithms of [10] require $O(nb \log\log n)$ communication bits per b-bit rumor. (More precisely, for the basic version this complexity holds for one source per rumor, and for the other version the exact complexity is $O(n(b + \log\log\log n)\log\log n)$ bits.) Thus, our algorithm saves a $\log\log n$ factor for large b, and a $b/\log\log b$ factor for small b. Informally, it works as follows. When a player learns a new rumor r, she pushes r in all subsequent rounds, until the 3rd time she pushes the rumor to some player who already knows it (when a rumor is pushed, the recipient informs the sender whether she knew the rumor). These push transmissions guarantee that a constant fraction of the players is informed within roughly $\log n$ rounds, and that r is pushed no more than $4n$ times. Pull transmissions take place only every $\log n/\log\log n$ rounds—there are

$\Theta(\log\log n)$ pull rounds during the lifetime of r. Say u calls v in such a round. Ideally, we would like the set of rumors pulled from u to v to consist of exactly those rumors that u knows and v does not know ; and this should be achieved without communicating more than roughly $nb/\log\log n$ additional bits per b-bit rumor, per pull round. This is a non trivial task, since players do not know the number or size of the rumors currently circulating ; an unbounded number of rumors can be generated in each round, and any b-bit string can be a valid rumor, for any b. Also, the fact that a rumor may have more than one sources precludes "grouping" into a big rumor all the rumors started at the same time by the same player, which would effectively bound by n the number of rumors generated per round. For these reasons simple solutions such as the use of fingerprints to uniquely describe a rumor with fewer bits do not work. To the core of our rumor-spreading algorithm is a simple and efficient scheme with which a player encodes the set of rumors that she knows using roughly $\log b$ bits per b-bit rumor. This scheme is deterministic and allows for some false positives, which however do not hurt our algorithm. Using this scheme, v informs u about the rumors that v already knows, and so u only transmits new rumors to v.

For the lower bound, note that an $\Omega(n\log\log n)$ bound on the number of bits communicated per rumor is immediate from the same bound of [10] on the number of messages. So, we just have to show an $\Omega(nb)$ bound, which seems like a trivial information-theory result. However, a more careful look reveals that this is not the case : Information may be conveyed not just by the content of the messages exchanged, but also by the round in which they are exchanged. Even sending no messages through an established connection also conveys information. In fact, the $\Omega(nb)$ bound no longer holds if we can have more than $O(\log n)$ rounds. The following (impractical) protocol spreads a b-bit rumor using only $O(n\log n\log b)$ bits, within $O(2^b\log n)$ rounds. The push protocol is modified such that, for each rumor r, the size b of r is pushed instead of r, and these push transmissions take place only in rounds t that are equal to r modulo 2^b (where r is viewed as a binary number).

We prove the $\Omega(nb)$ bound in two steps. We first establish the bound for large rumors, using essentially a counting argument. Then we reduce the case of smaller rumors into the previous case, by showing that given an algorithm that spreads small rumors using $O(nb)$ bits, we can devise an algorithm that also spreads large rumors using $O(nb)$ bits. Error-correcting codes are used in this construction. We note that the $\Omega(nb)$ bound holds also for non address-oblivious algorithms.

Références

[1] P. Berenbrink, R. Elsässer, and T. Friedetzky. Efficient randomised broadcasting in random regular networks with applications in peer-to-peer systems. In *27th ACM Symp. on Principles of Dist. Comp. (PODC)*, pp 155–164, 2008.

[2] 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. *6th ACM Symp. on Princip. of Dist. Comp. (PODC)*, pp 1–12, 1987.

[3] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *Proc. 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 773–781, January 20–22 2008.

[4] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading : Expanders, push vs. pull, and robustness. In *Proc. 36th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 366–377, July 5–12 2009.

[5] R. Elsässer. On randomized broadcasting in power law networks. In *Proc. 20th Int. Symp. on Distributed Computing (DISC)*, pages 370–384, September 18–20 2006.

[6] R. Elsässer. On the communication complexity of randomized broadcasting in random-like graphs. In *Proc. 18th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 148–157, July 30–August 20 2006.

[7] R. Elsässer and T. Sauerwald. The power of memory in randomized broadcasting. In *Proc. 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 218–227, January 20–22 2008.

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

[9] A. Frieze and G. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Appl. Math.*, 10 :57–77, 1985.

[10] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. 41st IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 565–574, November 12–14 2000.

[11] B. Pittel. On spreading a rumor. *SIAM J. Appl. Math.*, 47(1) :213–223, 1987.