



**HAL**  
open science

## Brief Announcement: Rumor Spreading with Bounded In-Degree

Sebastian Daum, Fabian Kuhn, Yannic Maus

► **To cite this version:**

Sebastian Daum, Fabian Kuhn, Yannic Maus. Brief Announcement: Rumor Spreading with Bounded In-Degree. DISC 2015, Toshimitsu Masuzawa; Koichi Wada, Oct 2015, Tokyo, Japan. hal-01207841

**HAL Id: hal-01207841**

**<https://hal.science/hal-01207841v1>**

Submitted on 1 Oct 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Brief Announcement: Rumor Spreading with Bounded In-Degree <sup>\*</sup>

Sebastian Daum, Fabian Kuhn, Yannic Maus

Dept. of Comp. Science, University of Freiburg, Freiburg, Germany  
daum.sebastian@gmail.com, {kuhn,yannic.maus}@cs.uni-freiburg.de

## 1 Introduction, Model & Motivation

Random gossip (**push** and **pull**) is one of the most studied protocols for disseminating information in a network, e.g., [1, 3]. Classically, in each time unit, every node  $u$  is allowed to contact a single random neighbor  $v$ . If  $u$  knows the data (rumor) to be disseminated, node  $v$  learns it (known as **push**) and if node  $v$  knows the rumor,  $u$  learns it (known as **pull**). While in the classic gossip model, each node is only allowed to contact a single neighbor in each time unit, each node can possibly be successfully contacted by and thus interact with many neighboring nodes. As an extreme case, consider the behavior of random **pull** in a star network where a single center node is connected to  $n - 1$  leaf nodes. In fact, all recent papers which study the time complexity of the random **push-pull** protocol critically rely on the fact that a node can be contacted by many nodes in a single round, e.g., [2]. However, in order to obtain applicable and scalable protocols, ideally, we would like to not only limit the number of interactions each node initiates, but also the number of interactions each node participates in.

We therefore study a weaker variant of the described random **pull** algorithm, which we call **rpull** (stands for restricted **pull**). In each round, every node can still initiate a connection to one uniformly random neighbor. However, if a single node receives several connection requests, only one of these connections is actually established. We consider two versions of how one of these incoming requests is selected. Assume that in a given round some informed node  $v$  receives requests from a set of neighbors  $R_v$ . In the *adversarial* **rpull** protocol, an (adaptive) adversary picks some node  $u \in R_v$  which will then learn the rumor, whereas in the *random* **rpull** protocol, we assume that a uniformly random node  $u \in R_v$  learns the rumor (chosen independently for different nodes and rounds). While the choice of which neighbor a node (actively) contacts with a request is under the control of the protocol, it is not necessarily clear how one of the incoming requests in  $R_v$  is chosen, e.g., it might be determined by the underlying network infrastructure in which case the adversarial model allows to study the worst-case behavior.

---

<sup>\*</sup> A full version of this paper can be found at <http://arxiv.org/abs/1506.00828>

## 2 Contributions

**Separation of Adversarial and Random Pull:** For trees we can show that both forms of **rpull** are asymptotically as fast as **pull** plus an additive term in the order of the degree of the node that initially has the rumor. On general graphs we show an exponential separation between adversarial and random **rpull**.

**Theorem 1.** *There is a graph such that for all source nodes, the random **rpull** protocol informs all nodes of the network in polylogarithmic time, w.h.p., whereas the adversarial **rpull** algorithm requires time  $\Omega(\sqrt{n})$  to even succeed with a constant probability.*

**Comparison of Pull and RPull:** Let  $\delta$  and  $\Delta$  denote the smallest and largest degrees of a given graph  $G$ . In each round of **rpull**, in expectation, each informed node receives at most  $\Delta/\delta$  requests. Hence, if an uninformed node  $u$  sends a request to an informed node,  $u$  should receive the rumor with probability at least  $\Omega(\delta/\Delta)$ . Consequently, intuitively, the slowdown of using random **rpull** instead of the usual **pull** protocol should not be more than roughly  $O(\Delta/\delta)$ .

**Theorem 2.** *For every given instance, if the **pull** algorithm informs all nodes in  $\mathcal{T}$  rounds with probability  $p$ , the random **rpull** algorithm reaches all nodes in time  $O(\mathcal{T} \cdot \frac{\Delta}{\delta} \cdot \log n)$  with probability  $(1 - o(1))p$ . The same result holds when comparing random **push-pull** with random **push-rpull**<sup>1</sup>.*

While the statement is intuitive its proof turns out more involved. Formally, we prove a stronger statement and couple the random processes defined by **pull** and random **rpull** such that for every start configuration, w.h.p., the set of nodes informed after  $O(\frac{\Delta}{\delta} \cdot \log n)$  rounds of random **rpull** is a superset of the set of nodes informed in a single **pull** round. We achieve this by coupling both processes with an intermediate process which is similar to **rpull** but removes dependencies between nodes which request from the same neighbor. Note that there is no coupling of **pull** and **rpull** in the classical sense, i.e., a coupling which does relinquish the *w.h.p.* term.

Furthermore, we show that for such a round-by-round analysis, our bound is tight. That is, there are configurations where  $\Omega(\frac{\Delta}{\delta} \log n)$  random **rpull** rounds are needed to simulate a single **pull** round, w.h.p..

## References

1. A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database management. In *Proc. Symp. on Principles of Dist. Comp. (PODC)*, pages 1–12, 1987.
2. G. Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Proc. Symp. on Theoretical Aspects of Comp. Sc. (STACS)*, pages 57–68, 2011.
3. R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Proc. Symp. on Foundations of Comp. Sc. (FOCS)*, pages 565–574, 2000.

<sup>1</sup> By **push-rpull** we denote the combination of **rpull** with a simultaneous execution of the classic **push** protocol.