Experimental Analysis of Rumor Spreading in Social Networks

Benjamin Doerr[1], Mahmoud Fouz[2], and Tobias Friedrich[3]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2] Rocket Internet, Dubai, U.A.E.
[3] Friedrich-Schiller-Universität Jena, Germany

Abstract Randomized rumor spreading was recently shown to be a very efficient mechanism to spread information in preferential attachment networks. Most interesting from the algorithm design point of view was the observation that the asymptotic run-time drops when memory is used to avoid re-contacting neighbors within a small number of rounds.
In this experimental investigation, we confirm that a small amount of memory indeed reduces the run-time of the protocol even for small network sizes. We observe that one memory cell per node suffices to reduce the run-time significantly; more memory helps comparably little. Aside from extremely sparse graphs, preferential attachment graphs perform faster than all other graph classes examined. This holds independent of the amount of memory, but preferential attachment graphs benefit the most from the use of memory. We also analyze the influence of the network density and the size of the memory. For the asynchronous version of the rumor spreading protocol, we observe that the theoretically predicted asymptotic advantage of preferential attachment graphs is smaller than expected. There are other topologies which benefit even more from asynchrony.
We complement our findings on artificial network models by the corresponding experiments on crawls of popular online social networks, where again we observe extremely rapid information dissemination and a sizable benefit from using memory and asynchrony.

1 Introduction

Randomized rumor spreading is a class of simple randomized distributed algorithms, all building on the paradigm that nodes of a network contact random neighbors to exchange information. Despite being very simple protocols, they proved to be very efficient both in theoretical investigations [14, 15, 23, 26–32, 36] and in practical applications [19, 33].

In a recent work [22], the authors analyzed the performance of the classical phone call model of Karp et al. [32] on networks following the preferential attachment model suggested by Barabási and Albert [1] to model real-world networks. The model assumes that new vertices attach to already-present vertices with a probability proportional to their degree. The problem of rumor spreading on these networks was first considered by Chierichetti, Lattanzi, and Panconesi [16] who showed that $\mathcal{O}(\log^2 n)$ rounds suffice with high probability. In [22], an asymptotically tight rumor spreading time of $\Theta(\log n)$ was proven, which is the same order of magnitude as for many other network topologies including complete networks, hypercubes and many classical random graph classes. Surprisingly, this run-time drops to the again tight order of $\Theta(\log n / \log \log n)$ when the protocol is modified so that contacting the same neighbor twice in a row is avoided. This observation is important from the viewpoint of algorithm design, since such a mechanism is very simple to implement. However, so far such fine-tuning has rarely led to provably better algorithms.

The aim of this work is to use an experimental investigation in order to (a) better understand the performance of randomized rumor spreading protocols on preferential attachment networks; in the long run, this might help in the design of efficient communication networks; and (b) to better understand the advantage of equipping nodes with a small amount of memory, which is used to avoid contacting a constant number of previous contactees; this is interesting from the viewpoint of algorithm design.

In summary, our main findings are the following. Generally, rumor spreading is very fast in preferential attachment networks, significantly faster than in random-attachment networks and hypercubes (which are much denser), and faster than in complete networks (unless the density is very small).

There is a clearly visible advantage of keeping track of the most-recently-contacted neighbor (using a one-item memory) in preferential attachment networks, particularly if the density is small. There is less to be gained from memory on random attachment networks and almost no gain in complete networks and hypercubes. Additional memory is of some benefit, but not very much.

For communication in social networks in particular, it makes sense to consider an asynchronous version of the rumor spreading protocol with nodes acting at exponentially distributed times (with expectation one). For random graphs with a given expected degree distribution that follows a power law with exponent in $(2, 3)$, Fountoulakis et al. [29] showed very recently that the push-pull protocol becomes much faster in the asynchronous setting. A recent theoretical analysis [23] proves a reduced time of $\mathcal{O}(\sqrt{\log n})$ in preferential attachment graphs and argues that random-attachment graphs, complete graphs and hypercubes keep their $\Theta(\log n)$ times. Our experiments show that the asynchronous model is faster on all graph classes, but a clearly greater advantage for preferential attachment graphs is not visible.

We conducted similar experiments on crawls of the Twitter and Orkut online social networks. Interestingly, we observe an even faster information dissemination than in preferential attachment graphs of corresponding size and density. These experiments also confirm that tracking one neighbor (one-item memory) cell leads to a significant improvement, whereas using additional memory to track more neighbors does not produce significant gains.

Rumor Spreading Protocols

When talking about rumor spreading, in this paper we generally refer to the *random phone call* model introduced by Karp et al. [32]. This is a push-pull protocol, meaning that information is exchanged between initiator and recipient of a call in both directions. A push-protocol with only the caller sending information to the recipient has also been widely discussed in the literature [27, 30], in particular, for the application of making replicated databases consistent [19, 33]. As shown in [16], however, the push protocol has very poor performance in preferential attachment networks.

The random phone call model is a synchronized protocol. In each round, each node of the network calls a random neighbor and both exchange information with each other. If one of the communication partners was informed at the beginning of the round, then both will be at the end of the round. An asynchronous analog of the protocol is discussed in Section 5.

It is interesting to enhance the random phone call model by excluding recently-contacted neighbors. When allowing a memory of size k, each node v chooses his next communication partner uniformly at random from all his neighbors except the previous $\min\{k, \deg(v)-1\}$ contactees. Note that nodes with degree $d(v) \leq k+1$ act as in the quasirandom model of Doerr, Friedrich, and Sauerwald [20] with randomly chosen lists.

Network Models

We are mainly interested in the *preferential attachment* (PA) model of Barabási and Albert [1]. The density of the resulting graph is controlled by a single parameter m. The model iteratively adds new vertices, which are connected to m already present vertices with a probability proportional to their degree. See Bollobás, Riordan, Spencer, and Tusnády [8, 9] for a precise description of this random graph model. It can be easily seen that for $m = 1$ the graph is disconnected with high probability. We therefore focus on $m \geq 2$. Under this assumption, the diameter is $\Theta(\log(n)/\log\log n)$ with high probability [8]. Besides various other typical properties of social networks[3, 6, 7, 18, 28], it also has been shown that the degree distribution follows a power law [9].

In addition to the PA model, we shall also include random-attachment networks in our investigation. In this network model, also known as the m-out model [5], each node chooses m other nodes as neighbors uniformly at random; finally, this neighbor relation is made symmetric and multiple edges are removed. Consequently, we obtain a random graph with average degree close to $2m$ and minimum degree at least m. These graphs form a good point of comparison with preferential attachment graphs with density parameter m, where nodes also choose m random neighbors, but according to the preferential attachment paradigm.

Related Work

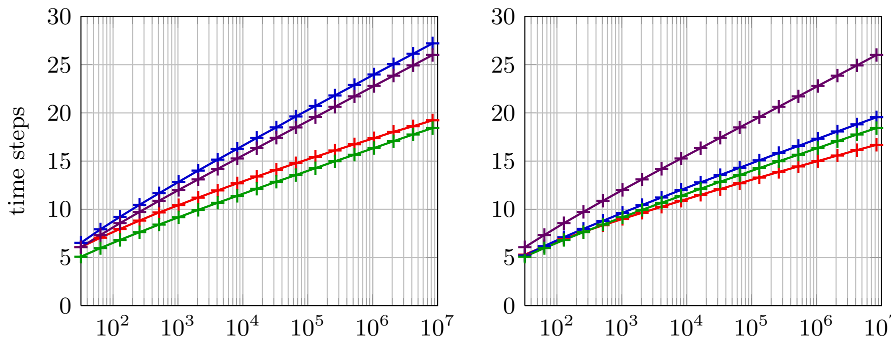
For many network topologies, the random phone call model very quickly distributes a piece of information initially only present at one node to all other nodes. In addition, due to its randomized nature, this process is highly robust against transmission failure. Karp et al. [32] show that in complete networks (any node can talk to any other node), $(1+o(1))\log_3(n)$ rounds suffice to spread a rumor in the whole network. Similarly, Elsässer [25] proved a bound of $\Theta(\log n)$ rounds for Erdős-Rényi random graphs $G_{n,p}$ with $p \geq \text{polylog}(n)/n$. For hypercubes, a $\Theta(\log n)$ bound follows from the $\mathcal{O}(\log n)$ bound of Feige et al. [27] for the push protocol together with the trivial lower bound stemming from the logarithmic diameter of the hypercube.

In a recent paper [22], the authors proved that the random phone call protocol spreads a rumor to all vertices of a preferential attachment graph in $\Theta(\log n)$ rounds as well. This improves over the previous $\mathcal{O}(\log^2 n)$ bound by Chierichetti et al. [16], but falls short of showing that these graphs, which are often used to model social networks, support rumor spreading better than classical network topologies. This is achieved in some sense in [22]. If we slightly alter the protocol such that a node chooses its communication partner uniformly at random from all neighbors excluding the one contacted in the previous round, then the rumor spreading time reduces to $\mathcal{O}(\log n/\log\log n)$, which is a tight bound because it is the diameter of these graphs [8].

Note that excluding previous contactees has almost no effect on classical network topologies. By checking the proofs of the results cited above, we see that also when excluding a constant number of previous contactees, the $\Theta(\log n)$ bound remains valid for complete graphs, hypercubes and random graphs. The quasirandom protocol of Doerr et al. [20] is a way of excluding all previous contactees. It has been investigated only in the push model, where again many known $\Theta(\log n)$ run time bounds have been verified. An experimental investigation [21] revealed that the quasirandom protocol is faster than the independent one, minimally for complete networks, but noticeably for sparser ones like random graphs and hypercubes. Unfortunately, our current results cannot be compared to these, because the latter are based only on push protocols. Baumann et al. [2] observed that the behavior of the quasirandom protocol changes significantly if the nodes known which of their neighbors already received the rumor.

2 Fast Broadcasting in Preferential Attachment Graphs, Influence of Graph Density

The result of [22] shows that rumor spreading in the random phone call model with memory size at least one has an asymptotically faster run-time of $\Theta(\log n/\log\log n)$ in preferential attachment graphs, in contrast to the $\Theta(\log n)$ time observed (i) for the no-memory version on preferential attachment graphs and (ii) regardless of memory on most classical graphs like complete graphs, hypercubes, and random attachment graphs. Since in [22] only asymptotic results



(a) density parameter $m = 2$. **(b)** density parameter $m = 10$.

Figure 1: Comparison of synchronous rumor spreading with one-item memory on preferential attachment graph (——), random-attachment graph (——), complete graph (——), and hypercube (——). The two charts show different density parameters of the preferential and random-attachment graph. The results for complete graphs and hypercubes are equivalent in both charts; they are given for comparison. The x-axis corresponds to the number of vertices $n = 2^5 \dots 2^{23}$. The y-axis corresponds to the run-time to inform all vertices, averaged over 10,000 runs.
For $m = 10$ the preferential attachment graph performs faster than all other graph classes. For not too large ($n \leq 2^{23}$) and very sparse case ($m = 2$) considered, the complete graph is even faster than the preferential attachment graph.

were proven, it is not clear if the proven differences are apparent for reasonable graph sizes. This is the focus of the current section of this paper. We have examined the average time needed to inform all vertices, starting from a random vertex, for different graphs.

In Figure 1, we show the broadcast times observed for complete graphs, hypercubes, and preferential and random attachment graphs with density parameters $m = 2$ and $m = 10$, with one-item memory. We observe that rumor spreading is quite fast in preferential attachment graphs (——), faster than in hypercubes (——) and random-attachment graphs (——) for both density parameters $m = 2$ and $m = 10$, and even faster than in complete graphs (——) for $m = 10$. Hence only the very sparse preferential attachment graphs with $m = 2$ are outperformed by complete graphs for $n \leq 10^7$. As for $n \geq 10^4$ the two last-mentioned charts constantly get closer, we expect that for sufficiently large graphs, information spreading is also faster on sparse preferential attachment graphs than on complete graphs.

We also observed structurally different behavior of the information spreading process on the different graphs. To be precise, let us consider graphs with $n = 10^6$ vertices and $m = 2$, averaged over 10,000 runs. Then on average 57% of the nodes of a random attachment graph are informed with a pull operation (and 43% via push). On the other hand, in preferential attachment graphs 73% of the nodes are informed by a pull operation. Moreover, on average such a pull operation transfers the rumor from a high degree node (with degree 66 on average) to a node with low degree (with degree 3 on average). This matches the structure used in the proofs of [16, 22, 29].

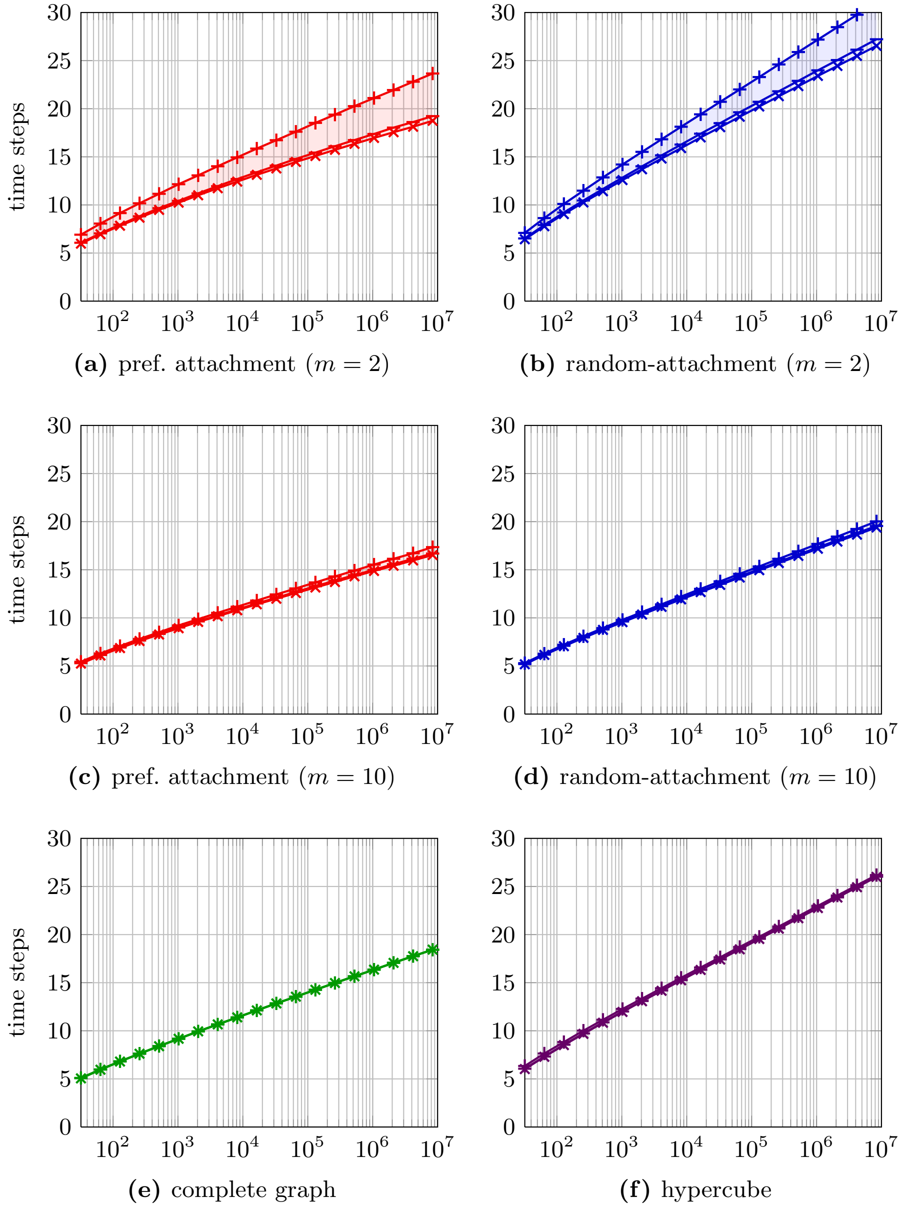
The path by which a piece of information is spread in a preferential attachment graph seems to differ from the typical paths in a random attachment graph. We measured the number of hops a piece of information needed to inform a node and compared this to its graph distance. In general, it is preferable to have a good correlation between the two measures [34]. The graph distance from the source gives a lower bound for the number of hops needed to inform a node. We call the difference between the number of hops needed and the graph distance the *delay*. If the delay is small, the information is spread on nearly-shortest paths. On random attachment graphs we observed that vertices which are less than six steps away from the source have a delay of less than one on average. On preferential attachment graphs, nodes with distance between two and six from the source have on average a delay of four. This shows that on preferential attachment graphs the information is *not* spread via shortest paths, but via detours. This again has been used in the theoretical analyses of [16, 22, 29].

3 The Effect of Short-Term Memory

Perhaps the most surprising finding of [22] is that keeping track of a certain small number of recently-contacted neighbors, and avoiding selecting any of these when randomly choosing the next communication partner, significantly reduces the time needed to inform all nodes of preferential attachment networks. More precisely, it was shown that for the classical random phone call model, this time is $\Theta(\log n)$. If the communication partners are chosen uniformly at random from all neighbors except the one called in the previous round (one-item memory), then this time decreases to $\Theta(\log n/\log\log n)$. Using additional memory to track more than one recent contactee, however, does not yield times better than $\Theta(\log n/\log\log n)$.

In this section, we experimentally investigate this phenomenon. Figure 2 shows the average time needed to inform all nodes. We first discuss the results on preferential attachment graphs with $m = 2$ shown in Figure 2 (a). As expected, we observe a significant improvement between no exclusion (marked with $+$) and exclusion of one neighbor (marked with $-$). In fact, for all graph sizes, one-item memory leads to nodes becoming informed between 14% and 21% faster than no memory. Observing the curves for different graph sizes also suggests that we have a $\Theta(\log n)$ broadcast time in the no-memory case and an $o(\log n)$ time with memory of any non-zero size. We do observe additional but very small improvements if we increase the memory to a size larger than the run-time, that is, when avoiding all previous contactees (marked with $\times$). For the graph sizes considered, the improvement of unbounded memory compared to memory of only one item is around 2%. The advantage of memory for preferential attachment graphs gets smaller for larger m, as shown in Figure 2 (c).

The results on random-attachment graphs are similar, just generally slower. Figure 2 (b) shows that the difference for $m = 2$ between no memory and one-



(a) pref. attachment ($m = 2$)

(b) random-attachment ($m = 2$)

(c) pref. attachment ($m = 10$)

(d) random-attachment ($m = 10$)

(e) complete graph

(f) hypercube

Figure 2: Comparison of synchronous rumor spreading without memory (marked with $+$), one-item memory (marked with $-$), and unbounded memory (marked with $\times$) on different graphs. The x-axis corresponds to the number of vertices $n = 2^5 \dots 2^{23}$. The y-axis corresponds to the run-time to inform all vertices, averaged over 10,000 runs.
The benefit of remembering more than one neighbor is very limited for all graphs. The benefit of one-item memory compared to no memory is the largest for the sparse preferential and random-attachment graphs. The complete graph and hypercube benefit very little from additional memory.

	90% informed	99% informed	100% informed
memory=0	15.74±0.99	16.87±1.00	23.13±2.28
memory=1	15.51±0.98	16.60±1.00	20.97±1.59
memory=2	15.47±0.98	16.55±0.99	20.31±1.30
memory=3	15.45±0.98	16.54±0.99	20.18±1.22
memory=25	15.45±0.97	16.54±0.99	20.11±1.13

Table 1: Comparison of the average time needed to inform a certain fraction of the vertices on the Orkut network depending on the amount of memory. For each combination, the average and standard deviation of 100,000 runs is given. With regard to the time needed to inform all vertices, we observe a large difference between excluding none and excluding the one most recently contacted. If only a 90% or 99% fraction should be informed, the gap is significantly smaller.

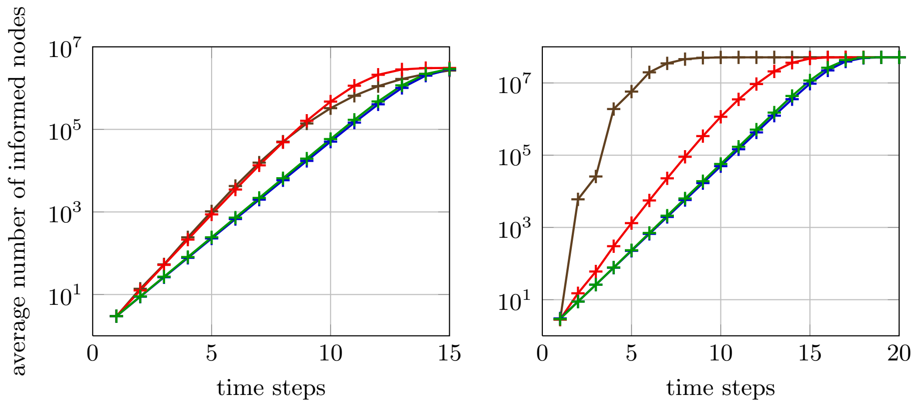
item memory is between 10% and 13%, while the additional improvement of unbounded memory is again around 2%. Theoretical consideration suggests that these gains can be at most by constant factors[4], and our experiments show that this can be at most a small constant.

In contrast, for other network topologies we see little advantage from using memory. For complete graphs, we observe in Figure 2 (e) barely any advantage even with unbounded memory. The difference between no memory and unbounded memory is less than 1% for complete graphs of all sizes. Because of the large vertex degrees, little benefit was expected; however, this is a notable difference from the results of using a pure push protocol without pull. Here, [21] observed at least a small advantage for the quasirandom protocol, which, when used with random lists, is equivalent to random choices with excluded previous contactees. The results of Figure 2 (f) for hypercubes show a similarly small impact of memory. For graphs with more than a few thousand nodes, the difference between no memory and unbounded memory is smaller than 2%.

The benefit of a small amount of memory can also be observed on real-world graphs. We examined the time needed to spread a rumor on a crawl of the Orkut network (for details on the network see Section 4). Table 1 shows a large difference between no memory and one-item memory for the time needed to inform all vertices. It is clearly visible that (a) more memory is of very little benefit and (b) this difference vanishes when considering the time needed to inform only a fraction of the vertices.

In summary, we also observe in experiments that a small amount of memory helps a lot for preferential and random attachment graphs, but much less for classical network topologies like complete graphs and hypercubes.

[4] It is known that these graphs have a diameter of $\Theta(\log n)$, so this is a natural lower bound. On the other hand, with high probability each pair of vertices is connected by a path such that the sum of the degrees of the vertices on the path is at most $\mathcal{O}(\log n)$. Consequently, with probability $1 - o(n^{-1})$, $\mathcal{O}(\log n)$ rounds suffice to transmit a rumor along such a path. This yields an upper bound of $\mathcal{O}(\log n)$ for the broadcast time on random attachment graphs.



(a) Orkut network **(b)** Twitter network

Figure 3: Comparison of synchronous rumor spreading with one-item memory on two real networks (——) with preferential attachment graph (——), random-attachment graph (——), and complete graph (——) of same size and density (where applicable). The Orkut network in (a) has $n = 3,072,441$ vertices and density parameter $m = 38$, the Twitter network in (b) has $n = 51,217,936$ vertices and density parameter $m = 32$.
The Orkut network behaves very similarly to the corresponding preferential attachment graph. The Twitter network is even faster than the corresponding preferential attachment graph. The complete and random-attachment graphs are significantly slower.

4 Real-World Social Networks

Most previous statistics were based on mathematically-defined graph models. To support our claim that news spreads very fast on social networks in general, we have also simulated the rumor spreading process on crawls of the *Twitter* and *Orkut* social networks.

Twitter is a social networking site which allows users to send and read short messages (so-called "tweets") of up to 140 characters. It is currently one of the top ten most visited sites on the Web[5]. We performed our experiments on a snapshot of the Twitter network that was crawled in September 2009 by Cha, Haddadi, Benevenuto, and Gummadi [13], available from [4]. It consists of 51,217,936 nodes and 1,963,263,821 directed edges. By making all edges undirected and considering the largest connected component, we obtained a connected graph with 51,161,011 nodes and 1,613,927,450 undirected edges. The preprocessing step of making all edges undirected might change the network structure, but the resulting network is still a typical social power law network.

Orkut is a social networking site operated by Google Inc. It is one of the top ten most visited websites in India and Brazil[5]. We used the data crawled in October and November 2006 by Mislove, Marcon, Gummadi, Druschel, and

[5] See "Top 500 Sites on the web" at `www.alexa.com`.

Bhattacharjee [35], which can be downloaded from [4]. The crawled graph contains 3,072,441 nodes and 117,185,083 edges. The edges are undirected, since Orkut requires consent from both users before a link between the two is created. At the time of the crawl, new users had to be invited by an existing user; therefore, the graph consists of a single component. The data covers roughly 11% of the total user population. The technical reason for this is that Orkut limits the rate at which a single IP address can download information. As a result, it took more than a month to crawl even this currently available part of the graph.

We chose these online social networks because of the available network data and because we feel that their structure might be similar to that of other real-world social networks. We are aware of the fact that interactions in Twitter and Orkut are more complex than in our simple randomized rumor spreading model.

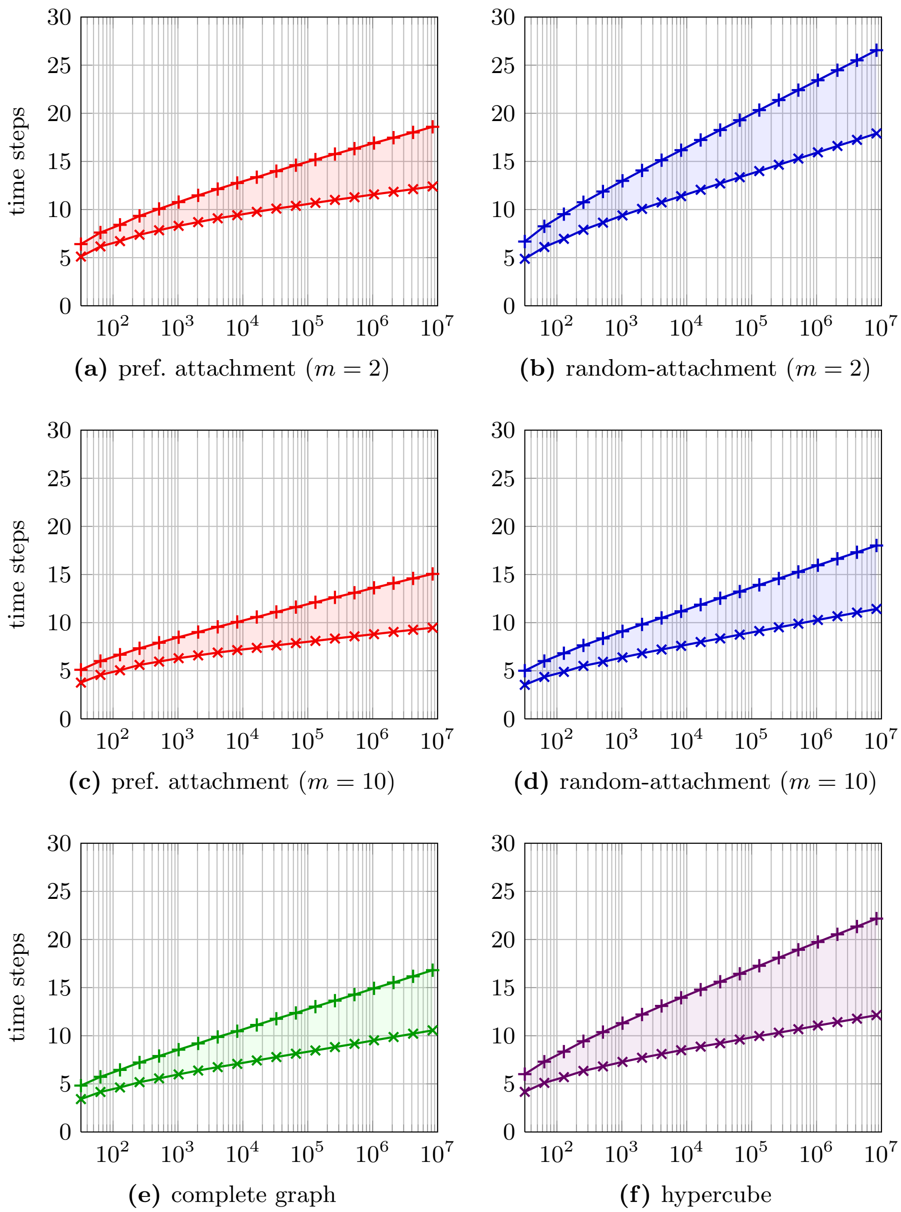
We ran the protocol with one-item memory on these real-world graphs and, for comparison, on preferential attachment, random-attachment and complete graphs with size and density as close as possible to the corresponding values of the real-world graph, that is, $m = 32$ for Twitter network and $m = 38$ for the Orkut network. The numbers shown in Figure 3 are averages of 500 runs[6] for the Twitter network and 100,000 runs for the Orkut network.

Figure 3 shows that news spreads much faster in the real-world networks (——) and the preferential attachment graphs (——) than in the complete (——) and random-attachment graphs (——). Interestingly, rumor spreading in the Orkut network and the comparable preferential attachment graph proceeds very similarly, whereas the Twitter network leads to much faster rumor propagation.

5 Asynchronous Rumor Spreading

So far, we have considered only the synchronized model where all nodes take action simultaneously at discrete time steps. Depending on the circumstances, this assumption may not be plausible. In fact, the assumption of a common centralized time clock contradicts the idea of a distributed self-organized broadcasting protocol [10, 21]. Boyd et al. [10] proposed an *asynchronous time model* with a continuous time line. There, each node has its own clock that ticks at the increments of a rate 1 Poisson process independent from all other clocks, which implies that the time between two ticks is exponentially distributed with parameter 1. In the asynchronous rumor spreading protocol, every node contacts a neighbor whenever its own clock ticks, and both exchange their information. Until last year, rumor spreading in the asynchronous model has received much less attention. Very recently, the authors have studied asynchronous rumor spreading theoretically on preferential attachment graphs [23], while Fountoulakis et al.

[6] The reason for the relatively small number of runs is that the Twitter network has more than one billion edges and we needed more than 50 GB of main memory to process it. A single simulation of the process required a run-time of several hours on a Hewlett Packard DL980 G7 server with eight eight-core Intel Xeon X7560 processors and 2048 GB of main memory.



(a) pref. attachment ($m = 2$)

(b) random-attachment ($m = 2$)

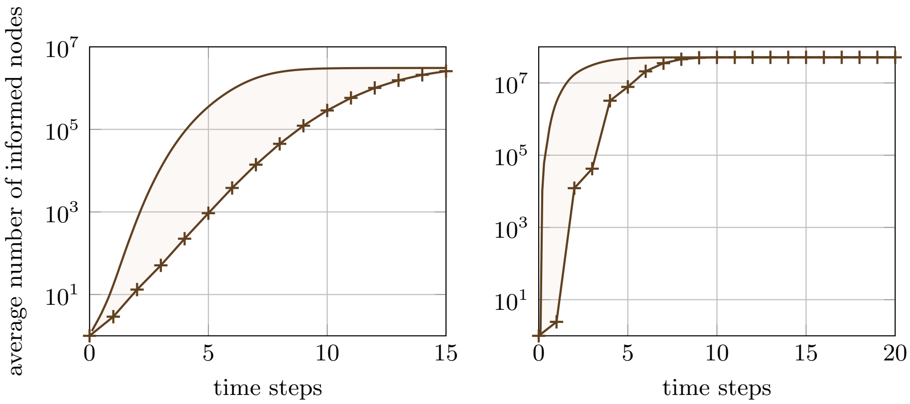
(c) pref. attachment ($m = 10$)

(d) random-attachment ($m = 10$)

(e) complete graph

(f) hypercube

Figure 4: Comparison of the average number of time steps needed to inform 99% of the vertices with synchronous (marked with $+$) and asynchronous (marked with $\times$) rumor spreading without memory on different graphs. The x-axis corresponds to the number of vertices $n = 2^5 \ldots 2^{23}$. The y-axis corresponds to the run-time to inform 99% of the vertices, averaged over 10,000 runs.
The asynchronous protocol spreads information faster than the synchronous protocol on all graphs. The difference is of the same order of magnitude for all graphs.



(a) Orkut network **(b)** Twitter network

Figure 5: Comparison of synchronous (—+—) and asynchronous (——) rumor spreading without memory on two real social networks. The x-axis corresponds to the time steps (in the synchronous setting) or the time (in the asynchronous setting). The y-axis corresponds to the number of informed vertices after this time, averaged over 1000 runs for the Orkut network and 50 runs for the Twitter network.
In both cases, the asynchronous counterparts spread the rumor significantly faster than the synchronous models.

[29] studied it on Chung-Lu random graphs [17] with a given expected degree distribution. Note that Chung-Lu graphs are quite different from preferential attachment graphs, e.g., their average diameter is $\Theta(\log \log n)$ [17], whereas it is $\Theta(\log n / \log \log n)$ [24] for preferential attachment graphs.

It is not surprising that asynchronous rumor spreading can be slow to inform all vertices. Note that it takes $\Theta(\log n)$ time until every node has performed at least one action. For this reason, in Figure 4 we consider times needed to inform 99% of the nodes. Note, however, that the times needed to inform 100% were also lower for the asynchronous model compared to the synchronous one. The charts clearly show a substantial speedup. Interestingly, for $n = 2^{23}$, the speedup for preferential (——) and random-attachment graphs (——) is slightly smaller (48-50% for $m = 2$ and 58-59% for $m = 10$) than for complete graphs (——) and hypercubes (——), which are 59% and 82%, respectively.

These empirical observations for moderately sized graphs are surprising given the theoretical findings on the expected asymptotic behavior. For the preferential graph, it has been shown that the time to inform $n - o(n)$ vertices without memory decreases from $\Theta(\log n)$ for the synchronous model without memory to $\mathcal{O}(\sqrt{\log n})$ for the corresponding asynchronous model [23]. On the other hand, it has been argued that random-attachment graphs, complete graphs and hypercubes keep their $\Theta(\log n)$ times, while our experiments show that the asynchronous model is faster on all graph classes. An asymptotic advantage for preferential attachment graphs is not apparent. We expect that the theoretically proven asymptotic behavior can be observed only for very large graphs. For the real-world social networks Orkut and Twitter, Figure 5 shows that, especially at the beginning, the asynchronous protocol (——) performs much faster than its synchronous counterpart (—+—). (For a comparison between the logarithmically scaled y-axis of Figure 5 (a) and the second row of Table 1, note that after 15 time steps the synchronous protocol only informed 84% of the nodes and the asynchronous protocol informed 99.99%.) This matches well with the theoretical finding that asynchrony speeds up rumor spreading on different models of social networks [23, 29].

6 Discussion

We have empirically studied several classical rumor spreading protocols on different artificial and real-world networks. As theoretically predicted, we observed that in preferential attachment networks rumors spread significantly faster than in all other examined network models. This confirms that the structure of social networks apparently allows spreading news very efficiently. This is remarkable as social networks evolve in a random and decentralized manner and are *not* designed with this purpose in mind.

The experiments also gave a much more detailed picture than possible purely theoretically. It has been demonstrated that in order to design a fast rumor-propagation algorithm on social networks, modeled by preferential attachment graphs, one needs small memory that helps to decide which node to contact next. This again seems to be specific to such networks as memory helps other network topologies much less. We also observed that a surprisingly small amount of memory is sufficient.

While theoretical results for models of social networks predicted a large speed-up when allowing asynchronous communication, we observed that other network topologies can benefit even more. The difference between synchronous and asynchronous propagation is very apparent for the two real-world networks Orkut and Twitter. We also observed that the speed of information spreading is very similar in the Orkut network and a preferential attachment graph of comparable density. Future work should include other rumor spreading protocols (e.g. [2, 11, 12]), more artificial graphs (e.g. [17]), and preferably even larger real-world networks like Facebook, which has close to one billion nodes.

Bibliography

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[2] H. Baumann, P. Fraigniaud, H. A. Harutyunyan, and R. de Verclos. The worst case behavior of randomized gossip. In *9th Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pages 330–345, 2012.

[3] N. Berger, C. Borgs, J. T. Chayes, and A. Saberi. On the spread of viruses on the Internet. In 16*th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 301–310, 2005.

[4] B. Bhattacharjee, P. Druschel, K. Gummadi, et al. Online social networks research at the Max Planck Institute for Software Systems. `socialnetworks.mpi-sws.org`.
[5] T. Bohman and A. M. Frieze. Hamilton cycles in 3-out. *Random Structures & Algorithms*, 35:393–417, 2009.
[6] B. Bollobás and O. Riordan. Robustness and vulnerability of scale-free random graphs. *Internet Mathematics*, 1:1–35, 2003.
[7] B. Bollobás and O. Riordan. Coupling scale-free and classical random graphs. *Internet Mathematics*, 1:215–225, 2003.
[8] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24:5–34, 2004.
[9] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18: 279–290, 2001.
[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52:2508–2530, 2006.
[11] K. Censor-Hillel and H. Shachnai. Fast information spreading in graphs with large weak conductance. In *22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 440–448, 2011.
[12] K. Censor-Hillel, B. Haeupler, J. A. Kelner, and P. Maymounkov. Global computation in a poorly connected world: Fast rumor spreading with no dependence on conductance. In *44th ACM Symposium on Theory of Computing (STOC)*, pages 961–970, 2012.
[13] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in Twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.
[14] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumour spreading and graph conductance. In *21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1657–1663, 2010.
[15] F. Chierichetti, S. Lattanzi, and A. Panconesi. Almost tight bounds for rumour spreading with conductance. In *42nd ACM Symposium on Theory of Computing (STOC)*, pages 399–408, 2010.
[16] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412:2602–2610, 2011.
[17] F. R. K. Chung and L. Lu. The average distance in a random graph with given expected degrees. *Internet Mathematics*, 1:91–113, 2003.
[18] C. Cooper and A. M. Frieze. The cover time of the preferential attachment graph. *Journal of Combinatorial Theory, Series B*, 97:269–290, 2007.
[19] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22:8–32, 1988.
[20] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 773–781, 2008.
[21] B. Doerr, T. Friedrich, M. Künnemann, and T. Sauerwald. Quasirandom rumor spreading: An experimental analysis. In *10th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 145–153, 2009.

[22] B. Doerr, M. Fouz, and T. Friedrich. Social networks spread rumors in sublogarithmic time. In *43rd ACM Symposium on Theory of Computing (STOC)*, pages 21–30, 2011.

[23] B. Doerr, M. Fouz, and T. Friedrich. Asynchronous rumor spreading in preferential attachment graphs. In *13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 307–315, 2012.

[24] S. Dommers, R. van der Hofstad, and G. Hooghiemstray. Diameters in preferential attachment models. *J. of Statistical Physics*, 139:72–107, 2010.

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

[26] R. Elsässer and T. Sauerwald. On the power of memory in randomized broadcasting. In *19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 218–227, 2008.

[27] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Rand. Struct. & Algo.*, 1:447–460, 1990.

[28] A. D. Flaxman, A. M. Frieze, and J. Vera. Adversarial deletion in a scale-free random graph process. *Comb., Probab. & Comput.*, 16:261–270, 2007.

[29] N. Fountoulakis, K. Panagiotou, and T. Sauerwald. Ultra-fast rumor spreading in social networks. In *23rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1642–1660, 2012.

[30] A. M. Frieze and G. R. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10:57–77, 1985.

[31] G. Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 57–68, 2011.

[32] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In 41*st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 565–574, 2000.

[33] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2003.

[34] D. Kempe, J. M. Kleinberg, and A. J. Demers. Spatial gossip and resource location protocols. *J. ACM*, 51(6):943–967, 2004.

[35] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *7th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 29–42, 2007.

[36] B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47:213–223, 1987.