Efficient Dissemination Algorithm for Scale-Free Topologies

Ruijing Hu*, Julien Sopena*, Luciana Arantes*, Pierre Sens*,
* *Laboratoire d'Informatique de Paris 6 (LIP6)*
Université Pierre et Marie Curie, CNRS/INRIA,
Paris, France
Email: firstname.lastname@lip6.fr

Isabelle Demeure†
†*Institut Telecom*
Telecom ParisTech, CNRS
Paris, France
Email: firstname.lastname@telecom-paristech.fr

***Abstract*—This paper presents an efficient dissemination algorithm suitable for scale-free random topologies which model some complex real world networks. In these topologies, some sites, denoted hubs, have many more connections than the others. By exploiting then the dissemination power of hubs, we propose a new gossip algorithm where sites directly connected to hubs do not forward received messages. Our algorithm offers a very high reliability and does not require any input parameter value that informs each site if it is a hub or not. Such an information is deduced by every site during the algorithm execution. Compared to well-known probabilistic gossip algorithms, performance simulation results show that our algorithm presents good performance in terms of message complexity and latency.**

***Keywords*-efficient dissemination algorithm; probabilistic gossip algorithms; large scale-free networks; performance; message complexity; reliability; latency.**

I. Introduction

Information dissemination over P2P networks like Gnutella or social networks like Twitter and Facebook becomes a very critical issue when high reliability, low latency, and low message redundancy are required. These networks are commonly known as scale-free networks since their degree distribution follows a power-law distribution. Furthermore, a minority of the sites (so-called hubs) have a higher degree than the average degree of the network.

A straightforward but inefficient way to disseminate information network wide is pure flooding protocols in which every site of the network relays once its first-time received message to its one-hop neighborhood [24]. However, flooding leads to broadcast storm problem. Probabilistic gossip algorithms mitigate this undesirable phenomenon [9] by reducing the number of edges over which messages are transmitted [5], [8], [11], [19] or by forwarding messages with some probability [14]. Upon reception of a message, a site retransmits it either to a randomly selected subset of neighboring sites, or to all neighbors with some probability. Hence, probabilistic gossip algorithms have emerged as an effective solution to implement highly reliable and scalable broadcast protocols. The topology properties of a network have a strong impact on the efficiency of information dissemination. Therefore, gossip algorithms should be tailored to exploit them. For instance, in the case of scale-free topologies, sites with high degree should retransmit received messages with a higher probability than the others since the former are highly connected site implying that messages will be disseminated faster.

We thus propose in this article a dissemination algorithm suitable for scale-free topologies generated by Barabási-Albert model [3]. This model makes use of preferential attachment which is a basic idea used by many other models [15], [16] for characterizing some real networks. Scale-free topologies are characterized by the presence of sites, denoted **hubs**, that have many more connections than the others.

Our algorithm exploits then as much as possible the potential dissemination power of hubs: it dynamically tries to reduce the set of sites that retransmit received message to these sites. Therefore, in our algorithm non-hub sites, whose number is much greater than the former, do not retransmit messages whenever they are directly connected to a hub. On the other hand, in order to ensure a high reliability, a site which believes that it does not have any hub as a neighbor requires all of its neighbors to become forwarders of received messages, creating thus a path to the closest hub. We show in the article that very few sites must be forwarders. Interestingly, that the deduction of hub-neighbors are performed in a distributed way, based only on sites' local view and exchange of neighbors' knowledge.

Our algorithm is composed of two phases. The first one is responsible for providing, with high probability, the above mentioned hub-neighbor connection requirement while the second one disseminates messages. In the second phase, based on received messages, a site locally deduces the average degree of the network, and if it should behave like a hub or not. Therefore, without any global parameters, but just exploiting processes' local view and information kept by received messages, our algorithm ensures extremely high reliability with only half of message complexity when compared to pure flooding algorithm, as confirmed by some performance evaluation results on top of OMNET++ [1].

The road map of this paper is organized as follows.

Section II gives an overview of scale-free random networks. Section III introduces some performance metrics. Existing probabilistic gossip algorithms are presented in Section IV, while Section V introduces our algorithm. Section VI shows simulation results on OMNET++. Section VII discusses some related work. Finally, Section VIII concludes this work.

II. Scale-Free Random Topology

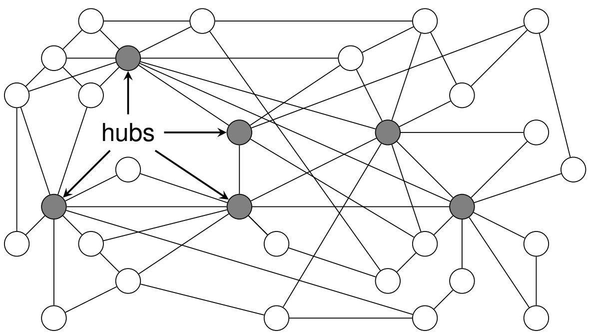


Figure 1. Scale-free network with 30 sites and mean degree=4

In a scale-free network (see Figure 1), the degree distribution follows a power law. It is characterized by the presence of sites, denoted **hubs**, whose number of edges are much higher than the others. The non-hubs sites are denoted **peripheries**.

In the sequel, $\mid l \mid$ denotes the size of set l.

We consider that the dissemination system Π comprises N sites $\{s_1, s_2, \cdots, s_N\}$. The neighborhood of s_i is the set denoted Λ_i and $V_i = \mid \Lambda_i \mid$ denotes the degree of s_i; $P(k)$ represents the probability that a site has a degree equal to k (i.e., the fraction of sites with degree k in the graph) and $\bar{V}$ is the mean degree. Therefore, $\bar{V} = \sum_{k=0}^{N-1} P(k) \cdot k$. Moreover, no message loss is taken into account.

In order to construct scale-free graphs, denoted $\mathcal{S}(N, m)$, we use the Barabási-Albert model [3]: starting from a small *clique* of m_0 ($\ll N$) sites, at every time step a new site is added such that its m ($\leqslant m_0$) edges connect it to m different sites already present in the graph. The probability p that a new site will be connected to an existed site is proportional to the degree of the latter. This is called preferential attachment. Since for $m = 1$ the graph becomes a tree, we consider $m > 1$.

The aforementioned generating process ensures that the graph is connected with power-law degree distribution (see Figure 2) approximately equal to $P(k) = \frac{2m(m+1)}{k(k+1)(k+2)}$ where $k = m, m+1, \cdots, N-1$ and $\bar{V} = 2m$ [27]. Trivially, the

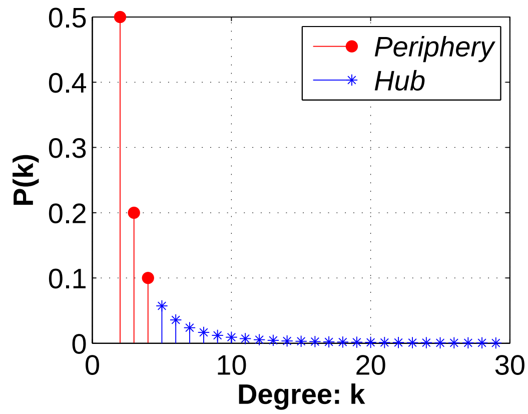


Figure 2. Degree distribution of a scale-free network with m=2

degree distribution does not depend on N which is very large in our system Π. Due to this power distribution, the degree variance is quite high.

Hub and **periphery** sites have degree greater than $2m$ and between m and $2m$ respectively. Hence, the system Π is composed by the set of hubs denoted Π_h and the set of peripheries denoted Π_p. We can deduce that:

$$\mid \Pi_p \mid > 3 \mid \Pi_h \mid$$

. We denote $P_{connect}(hub \mid s_i)$ (resp., $P_{connect}(per \mid s_i)$) the probability that a site s_i connects to a site in Π_h (resp., Π_p).

Lemma 1: Over $\mathcal{S}(N, m)$, $P_{connect}(hub \mid s_i) = P_{connect}(per \mid s_i) = 0.5$

Proof: $\mathcal{S}(N, m)$ generated by Barabási-Albert model is an uncorrelated network described in [32]. In such a network [26], the probability that a site s_i connects to another site s_j can be written as:

$$P(V_j \mid V_i) = \frac{V_j P(V_j)}{\bar{V}}$$

. Thus, the probability that a site s_i connects to a hub is

$$P_{connect}(hub \mid s_i) = \sum_{k=2m+1}^{N-1} \frac{P(k)\, k}{2m} = 0.5$$

, while the probability that the site s_i connects to a periphery is

$$P_{connect}(per \mid s_i) = \sum_{k=m}^{2m} \frac{P(k)\, k}{2m} = 0.5$$

■

III. Performance Metrics

In the context of information dissemination, the following metrics are used in the literature [14], [19], [21] for performance evaluation:

Message Complexity, denoted M: measures the mean number of messages received (or sent, since no message loss is taken into consideration) by each site:

$$M = \frac{\Omega}{N-1} \tag{1}$$

where Ω is the total number of messages exchanged during the dissemination.

Reliability, denoted R: is defined as the percentage of messages generated by a source that are delivered by all sites. A reliability value of 100% is indicative that the algorithm was successful in delivering any given message to all sites (i.e., every site is infected for any given message) ensuring thus *atomicity* similarly to pure flooding algorithms [19].

Latency, denoted L: measures the number of hops required to deliver a message to all recipients, i.e., the number of hops of the longest path among all the shortest paths from the source to all other sites that received the message.

An efficient dissemination algorithm aims at providing high reliability, while minimizing both message complexity and latency.

IV. Gossip Algorithms

Information dissemination in large scale network, is commonly studied on basis of Algorithm 1. Initially, the source sends a message to *all* of its neighbors (lines 2 and 3). A site delivers and retransmits a received message provided it has not previously received it; otherwise the message is discarded.

Algorithm 1: Generic Gossip algorithm

```
1 Broadcast (⟨msg⟩)
2     foreach s_j ∈ Λ_i do
3         Send(⟨msg⟩, s_j)

4 Receive (⟨msg⟩)
5     if msg ∉ msgHistory then
6         Deliver(⟨msg⟩) ;
7         msgHistory ← msgHistory ∪ {⟨msg⟩} ;
8         Gossip(⟨msg⟩,parameters) ;
```

There are four main probabilistic gossip families to implement the retransmission Gossip() procedure, namely (1) Fixed Fanout gossip (*GossipFF*) [19], (2) Probabilistic Edge gossip (*GossipPE*) [29], (3) Probabilistic Broadcast gossip (*GossipPB*) [14], and (4) Degree Dependent gossip (*GossipDD*) [11]. Besides the received message, all these algorithms receive one or more parameters whose value is the same for all sites.

Algorithm 2: Fixed Fanout Gossip (at s_i)

```
9  /* fanout: number of selected
   neighbors                                  */
10 GossipFF (⟨msg⟩,fanout)
11     if fanout ⩾ V_i then
12         toSend ← Λ_i
13     else
14         toSend ← ∅
15         for f = 1 to fanout do
16             random select s_j ∈ Λ_i/toSend
17             toSend ← toSend ⋃ s_j
18     foreach s_j ∈ toSend do
19         Send(⟨msg⟩, s_j)
```

In *GossipFF* (Algorithm 2), site s_i sends msg to a fixed number of sites, denoted $fanout$, in Λ_i, which are randomly selected (lines 15-17). Notice that if $fanout \geqslant V_i$, s_i transmits msg to all its neighbors (lines 11 and 12). Particularly, if $fanout \geqslant \max\{V_1, V_2, \cdots, V_N\}$, Algorithm 2 is a pure flooding algorithm.

For the three following algorithms, `Random()` generates a random number in the interval $[0, 1]$.

Algorithm 3: Probabilistic Edge Gossip (at s_i)

```
20 /* p_e: probability to use an edge     */
21 GossipPE (⟨msg⟩,p_e)
22     foreach s_j ∈ Λ_i do
23         if Random() ⩽ p_e then
24             Send(⟨msg⟩, s_j)
```

In *GossipPE* (Algorithm 3), site s_i randomly chooses those edges over which msg should be transmitted with regard to a fixed probability p_e (see line 23). Note that when $p_e = 1$ for all sites, we obtain the flooding algorithm.

Algorithm 4: Probabilistic Broadcast Gossip (at s_i)

```
25 /* p_v: probability to broadcast       */
26 GossipPB (⟨msg⟩,p_v)
27     if Random() ⩽ p_v then
28         foreach s_j ∈ Λ_i do
29             Send(⟨msg⟩, s_j)
```

Unlike Algorithm 3, in *GossipPB* (Algorithm 4), each site, except the source, diffuses msg to all its neighbors with fixed probability p_v (see line 27). In particular, when $p_v = 1$ this protocol becomes the flooding algorithm.

Algorithm 5: Degree Dependent Gossip (at s_i)

```
30 /* d: degree threshold                                   */
31 /* p_high: retransmission probability
      for high degree sites                                 */
32 /* p_low: retransmission probability for
      low degree sites                                      */
33 GossipDD (⟨msg⟩,d,p_high, p_low)
34     if V_i > d then
35         if Random() ⩽ p_high then
36             foreach s_j ∈ Λ_i do
37                 Send(⟨msg⟩, s_j)
38     else
39         if Random() ⩽ p_low then
40             foreach s_j ∈ Λ_i do
41                 Send(⟨msg⟩, s_j)
```

GossipDD (Algorithm 5) tries to improve the performance of *GossipPB* by separating sites into two sets. Sites with higher degree retransmit msg with a high probability p_{high} whereas lower degree sites retransmit it with a low probability p_{low} with $p_{high} > p_{low}$. The decision about the degree level of a site (high or low) depends on a *threshold* degree d (see line 34).

V. Our Algorithm

The main idea of our gossip algorithm is that only the sites in Π_h (i.e., hubs), whose degree is much higher than those in Π_p (i.e., peripheries), should relay received messages. In this way, since $\mid \Pi_p \mid > 3 \mid \Pi_h \mid$ (see Section II), intuitively half of the message complexity may be reduced compared to flooding algorithm.

Primarily knowing the degree of its one-hop neighbors, each site can deduce, in a distributed way, whether one of its neighbors belongs to Π_h or not. If it is the case, the site never retransmits received messages since it knows that its hub neighbor will do it. On the other hand, if a site believes that is is not directly connected to any hub, all sites between that site and the closest hub must forward every message they receive. We denote such sites *forwarders*. Our algorithm is thus composed of two phases. The first one is responsible for satisfying, with high probability, this hub-neighbor connection requirement over $\mathcal{S}(N, m)$ and the second one for disseminating messages.

Algorithms 6 and 7 respectively describe the above two phases of our gossip algorithm. The variable min_i corresponds to the minimum degree amongst the neighbors of s_i and itself, while max_i corresponds to the maximum degree of s_i's neighbors. Initially, s_i knows max_i and min_i.

Algorithm 6: Hub Connection Algorithm (at s_i)

```
42 /* min_i: min neighbor and its degree
      in local view                                         */
43 /* max_i: max neighbor degree in local
      view                                                  */
44 HubConnection ()
45     if max_i ⩽ 2 × min_i then
46         foreach s_j ∈ Λ_i do
47             s_j.forwarder = true
```

Algorithm 7: Hub-Based Gossip (at s_i)

```
48 /* min^ = min_i: the updated min degree
      in the network                                        */
49 /* ⟨msg⟩.min = min_i: estimated min
      piggybacked in message                                */
50 GossipHB (⟨msg⟩,-)
51     if s_i.forwarder or V_i > 2 × Approxm(⟨msg⟩) then
52         foreach s_j ∈ Λ_i do
53             Send(⟨msg⟩, s_j)
54 Approxm (⟨msg⟩)
55     if min^ > ⟨msg⟩.min then
56         min^ = ⟨msg⟩.min
57     else
58         ⟨msg⟩.min = min^
59     return min^
```

Algorithm 6 is simultaneously executed by all sites before information dissemination. A site locally suspects that it is not connected to a hub if the degree of all its neighbors is smaller or equal to $2 \times min_i$ since in the Barabàsi-Albert model, hubs have degree greater than $2 \times m$ (see line 45). In this case, by setting its neighbors' $forwarder$ variable to true, the site forces all of them to forward every message received in the second phase of the algorithm (lines 46 and 47).

The relative number of sites that need forwarders in their neighborhood is very small, which is inferred from Theorem 2. For instance, if $m = 5$, theoretically, about 1% of the total sites in $\mathcal{S}(N, m)$ require forwarders to reach a hub, whereas if $m = 15$ fewer than 6×10^{-6} of the sites need forwarder sites.

Theorem 2: Over $\mathcal{S}(N, m)$, the fraction of sites that need forwarders in their neighborhood is

$$P_{add} \leqslant \sum_{k=m}^{N-1} 0.5^k P(k)$$

, where $P(k) = \frac{2m(m+1)}{k(k+1)(k+2)}$.

Proof: Since the probability for a site s_i with degree k whose $\Lambda_i \subseteq \Pi_p$ is smaller than or equal to $(P_{connect}(per \mid s_i))^k$, then the probability that any site in Π requires a forwarder is

$$P_{add} \leqslant \sum_{k=m}^{N-1} (P_{connect}(per \mid s_i))^k \, P(k)$$

As deduced in Lemma 1 that $P_{connect}(per \mid s_i) = 0.5$, the result is obtained. ∎

After the first phase, the second phase of the algorithm, denoted *GossipHB*, starts up message dissemination. If the site is a either a $forwarder$ or a hub, it should retransmit the messages it receives to all its neighbors (see line 51). For a site to conclude that it is a hub, it must deduce m of $\mathcal{S}(N, m)$. To this end, it calls the function `Approxm`($\langle$*msg*$\rangle$). Every message msg piggybacks $msg.min$, i.e., the minimum degree of the graph known by the sender of the message. If the value in the message is smaller than the minimum degree value kept by the receiver of the message in the its local approximation $\hat{min}$ variable, the receiver updates this variable (lines 55 and 56).

Thanks to this approximate estimation, a site eventually deduces the mean degree of $\mathcal{S}(N, m)$ (i.e., $2 \times \hat{min}$) which distinguishes *hubs* from *peripheries* (see Section II).

Theorem 3 shows the message complexity induced by our algorithm, without considering the number of messages sent by the sites for hub-neighbor connection requirement of the first phase. *GossipHB* saves half of the message complexity (i.e., $2 \times m$) compared to pure flooding algorithm where all sites relay the message to all their neighbors.

Theorem 3: Over $\mathcal{S}(N, m)$, message complexity of *GossipHB* is m.

Proof: Since $\hat{min}$ converges to m, according to Algorithm 7, then $P(k) = \frac{2m(m+1)}{k(k+1)(k+2)}$. According to the theory in [17], we calculate the message complexity as

$$\begin{aligned} M &= \sum_{k=2m+1}^{N-1} P(k)k \\ &= 2m(m+1) \sum_{k=2m+1}^{N-1} \left(\frac{1}{k+1} - \frac{1}{k+2} \right) \\ &= m \end{aligned}$$

∎

VI. Performance Evaluation

In this section, we present and discuss some evaluation performance results concerning the five algorithms described in both Sections IV and V: *GossipFF*, *GossipPE*, *GossipPB*, *GossipDT*, and our algorithm.

As the first phase of our proposed algorithm is executed only once before message disseminations, we denote our algorithm *GossipHB* in the sequel. For *GossipDD*, we have fixed p_{high} and p_{low} to **1** and **0** respectively in order to ensure that only sites with higher degree retransmit the message. This version of *GossipDD* is named Degree Threshold Gossip, and denoted *GossipDT* hereafter.

Experiments have been conducted on top of the simulator OMNET++ [1]. We have considered two $\mathcal{S}(N, m)$ topologies with 1000 and 10000 sites respectively. For each value of m between 2 and 15 and $m_0 = m + 2$ for the initial clique, we generated 50 graphs with different seeds and then chose 200 different random sources in each graph. All results represent the average of these experiments.

Figures 3 and 4 aim at respectively studying the reliability and latency of gossip algorithms that require a pre-specified parameter, i.e., *GossipHB* was not included in these studies. We fixed the parameter values to reach a given message complexity, and then we evaluated reliability and latency metrics of the four algorithms.

A. Reliability of pre-specified parameter algorithms

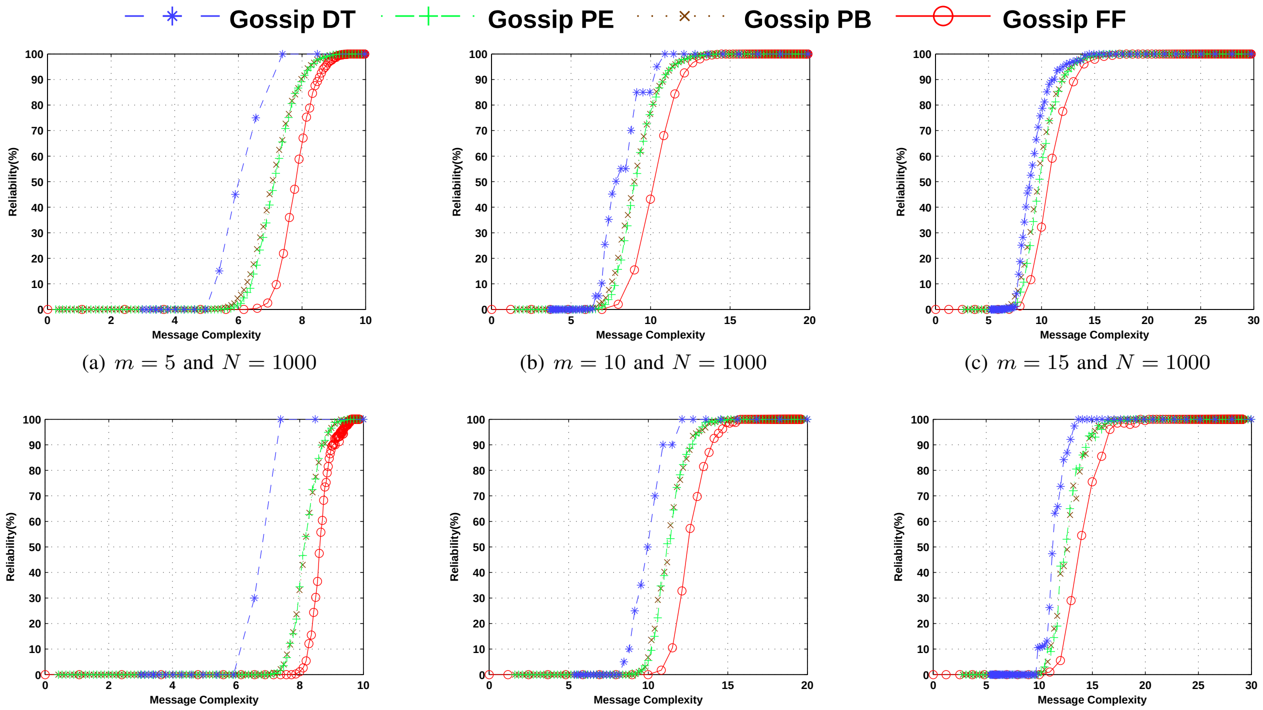
Figure 3 shows, for different topologies, the **reliability** R in regard to **message complexity** M. We can observe that, for reaching high reliability, beyond a given message complexity value, *GossipFF*, *GossipPE*, *GossipPB*, and *GossipDT* present a threshold behavior which is in accordance with the percolation theory [12].

Another interesting comparison result is that, in order to reach the same reliability, *GossipPE* and *GossipPB* present the same message complexity, while *GossipFF* and *GossipDT* induce the most and the least message complexity respectively. The difference in performance can be explained since in *GossipDT* only sites with higher degree, for example the hubs, which are a minority in the network, are responsible for message retransmission while in the other probabilistic algorithms, all peripheries relay messages as well.

B. Latency of pre-specified parameter algorithms

Figure 4 presents the **latency** L in relation to **message complexity** M. We only present the performance when the reliability reaches at least 85%.

After a given message complexity, latency does not decrease anymore, but converges towards pure flooding approach (i.e., the shortest routes between the source and the other sites). *GossipDT* converges to the minimum latency with the lowest message complexity when reliability is over 99.9%, whereas *GossipFF* entails quite substantial message complexity for converging. The reason for *GossipDT* better performance is that when atomicity is reached, the sites that are responsible for retransmission form a connected subgraph of hubs whose diameter is smaller than $\mathcal{S}(N, m)$.



(a) $m = 5$ and $N = 1000$ (b) $m = 10$ and $N = 1000$ (c) $m = 15$ and $N = 1000$

(d) $m = 5$ and $N = 10000$ (e) $m = 10$ and $N = 10000$ (f) $m = 15$ and $N = 10000$

Figure 3. Reliability comparison over $\mathcal{S}(N, m)$.

A first conclusion from both studies is that *GossipDT* is the best choice for $\mathcal{S}(N, m)$ in terms of message complexity and latency. Nevertheless, for reaching such a performance, the threshold d parameter must be set to an optimum value and the latter should be known by all sites. Since *GossipHB* overcomes this limitation by estimating m (and therefore the mean degree of the network) at runtime, it turns out quite interesting to compare the results of our algorithm with the other algorithms. For comparison reasons, we have also included in our study a flooding algorithm since the complexity of this algorithm increases linearly with m and proves, if necessary, the interest of gossip algorithms.

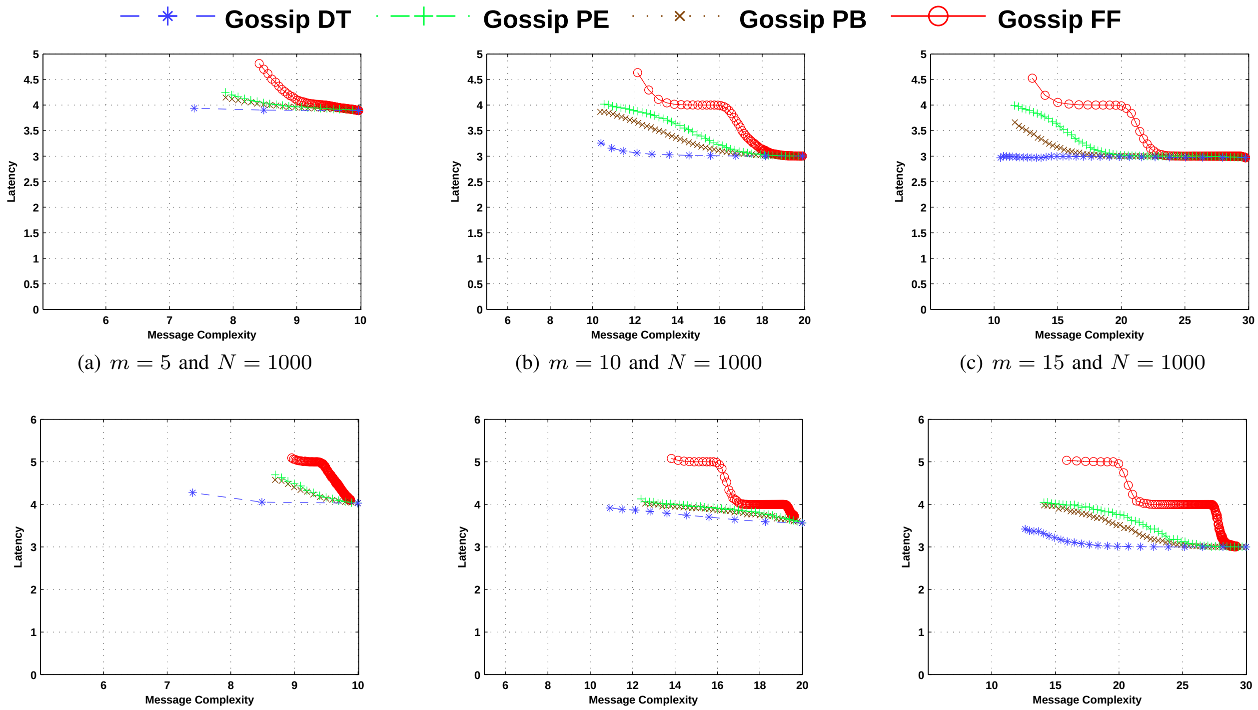
C. Comparison of the best algorithms' performance

Figure 5 presents, for each algorithm, the minimum message complexity to obtain $R > 99.9\%$ for m within 2 to 15. The minimum for each gossip algorithm has been empirically deduced by varying the values of its corresponding parameters till reaching such reliability.

We can observe that the greater the value of m, the greater the number of redundant messages received by each site regardless network size. Intuitively, when a site degree increases, there will be more message transmission paths towards the same site from other sites.

On one hand, *GossipFF* has the highest message complexity, while *GossipPB* and *GossipPE* have almost the same message complexity. On the other hand, *GossipDT* and our algorithm outperform them on all network topologies. When $m \geqslant 5$, which implies that the number of sites that require forwarders in our algorithm's first phase is very small (see Theorem 2), *GossipHB* presents a linear relation between message complexity and m, which confirms Theorem 3. This linear behavior is also responsible for reducing half of the messages in comparison to the pure flooding algorithm that presents message complexity $2 \times m$ (i.e., the mean degree of graphs). Furthermore, when $m < 5$, though not having such a linear message gain, our algorithm still presents the lowest message complexity, whereas the other algorithms perform closely to the flooding one.

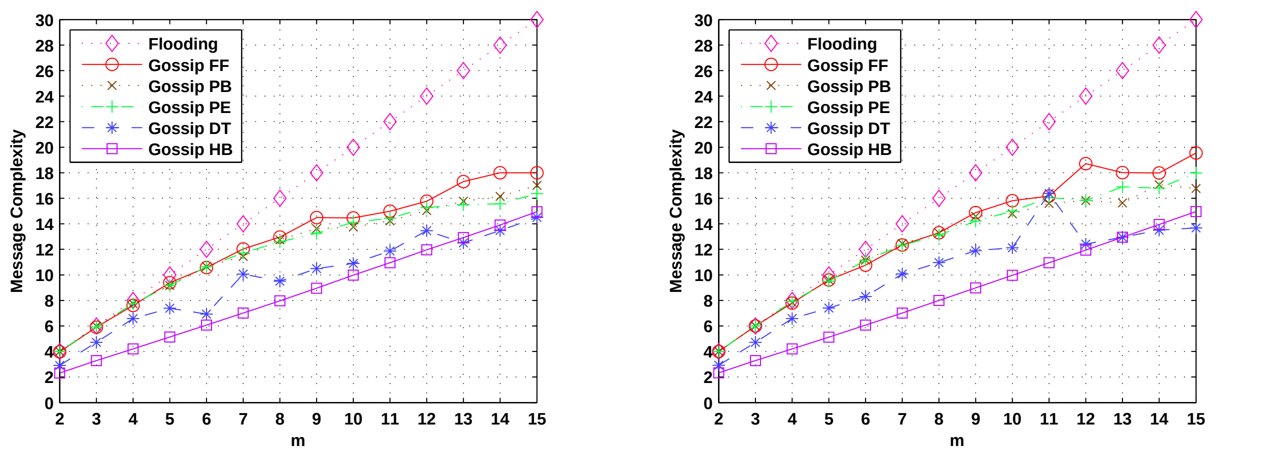
We should also point out that with regard to *GossipFF*, *GossipPB*, and *GossipPE*, our algorithm's message complexity gain is considerable. For *GossipDT* we observe that the growth of message complexity slows down with m. In fact, the greater the value of m, the higher the hub dissemination power and the smaller the number of hubs needed to ensure reliability. Thanks to the first phase, *GossipHB* algorithm ensures that hubs are connected by paths composed by very few periphery sites, (i.e., the forwarders), whereas



(a) $m = 5$ and $N = 1000$ (b) $m = 10$ and $N = 1000$ (c) $m = 15$ and $N = 1000$

(d) $m = 5$ and $N = 10000$ (e) $m = 10$ and $N = 10000$ (f) $m = 15$ and $N = 10000$

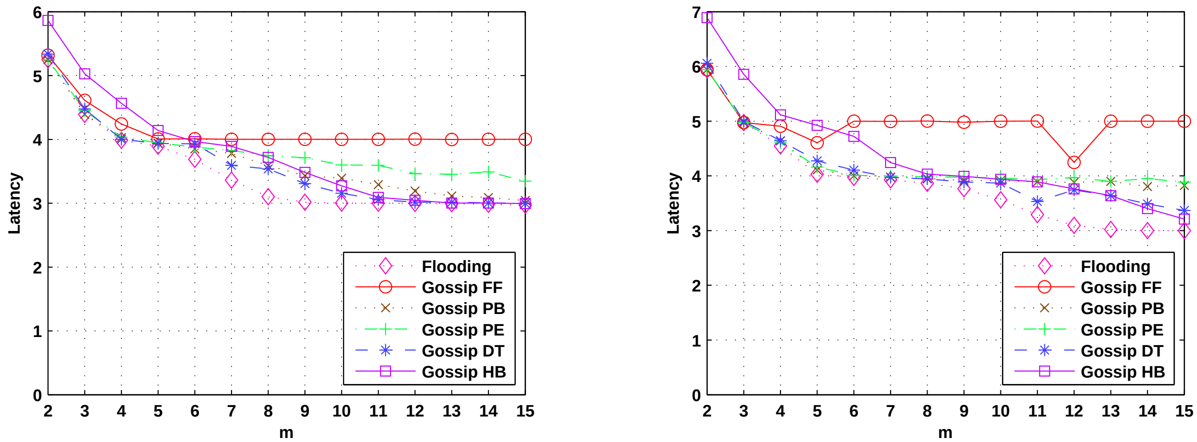
Figure 4. Latency comparison over $\mathcal{S}\,(N, m)$.



(a) $\mathcal{S}\,(N, m)$ with $N = 1000$ (b) $\mathcal{S}\,(N, m)$ with $N = 10000$

Figure 5. message complexity comparison with reliability over 99.9%.

GossipDT does not for topologies whose mean degree is small. This difference explains why below a given threshold value of m, *GossipHB* presents better message complexity than *GossipDT* since in the former there is fewer message retransmission by periphery sites than in the latter. However, beyond such a threshold value, where the connectivity of hubs is ensured without periphery paths (i.e., forwarders in the case of *GossipHB*), their complexity message performance is inverted: in *GossipHB*, all hubs relay messages while in *GossipDT* only a subset of hubs, i.e., those that



(a) $\mathcal{S}(N,m)$ with $N = 1000$

(b) $\mathcal{S}(N,m)$ with $N = 10000$

Figure 6. Latency comparison with reliability over 99.9%.

have degree greater than d. Notice that the value of this m threshold increases when N increases ($m = 13$ and $m = 14$ in Figures 5(a) and 5(b) respectively).

In Figure 6, for m within 2 to 15, we show some results related to latency, aiming $R > 99.9\%$. Latency decreases in function of m, since the greater the mean degree is, the larger the number of short-cuts in the graphs. In addition, compared to the system with 1000 sites, the one with 10000 sites has longer diameter, which results in higher latency. In the two systems, the latency of our algorithm is close to *GossipDT*'s. If $m < 5$, both latencies are higher than *GossipPB* and *GossipPE*, requiring many more transmissions per site (see Figure 5), thus creating many shorter paths in the graphs. Otherwise, their latencies are equal or lower than the other two algorithms, since the greater is m, the greater the degree of hubs, which induces shorter paths from the source. In particular, when the mean degree of graphs is very small, for example, 4, all algorithms present performance close to flooding. As expected, the flooding algorithm presents the shortest latency, but at the expense of the largest number of redundant messages, which expresses the tradeoff between message complexity and latency. Moreover, *GossipFF* has the worst message complexity performance which can be explained by its poor exploitation of hubs [17].

VII. Related Work

Many existing works propose to reduce message complexity of gossip algorithms provided that both high reliability and low latency are satisfied.

Some overlay broadcast systems apply push-pull algorithms [6] to ensure high reliability. However, as the pull phase is difficult to implement [19], we have restrained this related work to pure push algorithms. The latter can be grouped into localized and globalized methods. Globalized methods require global topology information and attempt to work out optimums, in terms, for instance, of message complexity, broadcast tree, etc. Their computing complexity is usually NP-Hard [22]. Localized methods exploit merely information of one or two-hop neighborhood or some other intrinsic information.

As *GossipHB* exploits only localized information, we discuss in the following some works that also use some localized method.

In [23], a piggybacked forwarding list helps in the decision of which sites within two hops to retransmit the message. Nonetheless, finding such list is NP-complete problem. Multipoint Relaying Pruning (MPR) [28] uses two-hop neighborhood knowledge to reduce message redundancy. In [25], the network topology is divided into several disjoint overlapping clusters whose size is bounded by two values a priori. Each cluster elects one site as the cluster-head. The cluster-head of each cluster is responsible for the retransmission. Another type of site, the gateway, has two or more cluster-heads as its neighbors and also relays messages. However, cluster-head election requires every site to have a unique identity in the system, and, contrarily to *GossipHB*'s first phase, the identity of the elected cluster-based sites cannot be simultaneously deduced by all sites in just one-hop information exchange.

In [33], the authors describe a distributed algorithm for calculating Connected Dominating Set (CDS) in ad hoc wireless networks. Message retransmission is restricted to sites of the CDS since any site in the network either belongs to the set or is a direct neighbor of some site of it. Each site requests information about its two-hop neighborhood.

Unfortunately, the problem of finding such a minimum CDS has been shown to be NP-complete [13]. Several phases are required for meta-information exchanges amongst two-hop neighbors to determine the forwarding list, which may consume much bandwidth and result in high latency, contrarily to our algorithm, where, for ensuring atomicity, sites exchanges only once the degree knowledge.

Another family of algorithms that exploit localized methods is composed by probabilistic gossip algorithms, as those described in Section IV. Compared with deterministic approaches, they highlight their simplicity and scalability [9]. Either applied to overlay networks [19], or exploited in wireless ad hoc [14] and sensor networks [31], they reduce message redundancy and well satisfy latency application constraints.

The reliability and the latency of *GossipFF*, *GossipPB*, and *GossipPE* have been fairly compared over scale-free topology in [17]. Dissemination latency of a modified version of *GossipPB*, where every site sends message to one neighbor with certain probability several times over $\mathcal{S}(N,m)$, has been theoretically studied in [10] by using SIS (Susceptible-Infective-Susceptible) model. Nevertheless, reliability is not taken into account in the authors' study. Moreover, [4] and [7] also give insights of the latency in the scale-free topology.

Similarly to *GossipHB*, the adaptive gossip algorithm for sensor networks proposed by [20] precludes the need of pre-configured parameter values. It assigns different gossip probabilities to different sites, based on the network topology. The probabilities are revised periodically based on potential changes in topology. Moreover, in [34], the authors have proposed a probabilistic approach which combines both probabilistic and CDS based methods. They classify all sites into four groups according to their connectivity characteristics in 2-hop neighborhood information, and assign the sites in each group with a different probability heuristically fixed a priori. The authors in [2] simply adapt the probability in each level proportionally inverse to the number of the level for sensor network, which ensures sites in the sparse density area to forward with higher probability. Exploiting this approach, authors in [18] propose an adaptive method that depends on the direction of message flow from the source to adjust probability for every site in each of the four group levels. On contrary, our algorithm does not need any pre-configured input parameter.

The impact of scale-free topologies on some of the probabilistic gossip algorithms of Section IV has been evaluated by simulation in both [5] and [11] in the context of ad hoc networks and heterogeneous large-scale networks respectively. Yet, neither of these works proposes a new algorithm tailored for scale-free topologies whose execution does not depend on pre-specified parameters.

VIII. Conclusion

Based on the property that hubs are highly connected in scale-free networks, we have presented a new gossip algorithm, *GossipHB*, where periphery sites directly connected to hubs do not retransmit messages. As the number of the former is much larger than the latter in scale-free networks, the message complexity of the algorithm, when compared to flooding one, is considerably reduced. In our algorithm, the average degree of the network is not a pre-defined parameter of the algorithm, but deduced during the execution of the algorithm. In order to ensure high atomicity, *GossipHB* has a first phase algorithm, where each site easily deduces if it has a direct connection to a hub or not. Thus, without a global view, this phase establishes short paths that connect hubs, ensuring then connectivity of hubs, necessary for information dissemination over the network.

Theoretical analysis and evaluation performance results confirm the correctness and effectiveness of our algorithm. Compared to other well-known probabilistic gossip algorithms, simulation results over $\mathcal{S}(N,m)$, show that *GossipHB* reduces message complexity, while the minimum latency is held and high reliability ensured.

We have observed that only if connectivity of hubs is ensured, *GossipDT* outperforms *GossipHB* in terms of message complexity. However, for such gain it is necessary to set its input parameter to an optimum value at initialization phase. We claim that global choice of parameter values is not suitable for gossip algorithms in networks where the structure of the network is not known. Therefore, *GossipHB* turns out to be the best choice, since the mean degree of the network is deduced at runtime and it exploits at the maximum the dissemination power of hubs of scale-free networks.

As a near future work, we conduct new performance experiments using social networks traces, such as Facebook [30].

References

[1] Omnet++: Discrete event simulation system *http://www.omnetpp.org/*.

[2] J.-D. Abdulai, A. Mohammed, K. S. Nokoe, and E. Oyetunji. Route discovery in wireless mobile ad hoc networks with adjusted probabilistic flooding. adaptive science technology. *ICAST*, 2009.:99–109, 2009.

[3] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, Jan. 2002.

[4] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. *Theor. Comput. Sci.*, 412(24):2602–2610, 2011.

[5] G. D'Angelo and S. Ferretti. Simulation of scale-free networks. *International Conference on Simulation Tools And Techniques*, pages 1–10, 2009.

[6] B. Doerr, M. Fouz, and T. Friedrich. Asynchronous rumor spreading in preferential attachment graphs. *SWAT*, pages 307–315, 2012.

[7] B. Doerr, M. Fouz, and T. Friedrich. Why rumors spread so quickly in social networks. *Commun. ACM*, 55(6):70–75, 2012.

[8] S. Ferretti and G. D'Angelo. Multiplayer online games over scale-free networks: a viable solution? *ICST*, 2010.

[9] R. Friedman and A. C. Viana. Gossiping on manets: The beauty and the beast. *Operating Systems Review*, 41(5):67–74, Oct. 2007.

[10] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. *INFOCOM*, pages 1455–1466, 2005.

[11] B. Garbinato, D. Rochat, and M. Tomassini. Impact of scale-free topologies on gossiping in ad hoc networks. *NCA*, pages 269–272, 2007.

[12] G. Grimmett. *Percolation*. Springer, 1989.

[13] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, Apr. 1998.

[14] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. *INFOCOM*, 3:1707–1716, 2002.

[15] C. Herrera and P. Zufiria. Generating scale-free networks with adjustable clustering coefficient via random walks. *Network Science Workshop*, pages 167–172, 2011.

[16] P. Holme and B. Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, 2002.

[17] R. Hu, J. Sopena, L. Arantes, P. Sens, and I. Demeure. Fair comparison of gossip algorithms over large-scale random topologies. *31th IEEE International Symposium on Reliable Distributed Systems*, Oct. 2012.

[18] T.-C. Huang, Y.-P. Lin, and L. Tang. Neighbor-aware gossip-based broadcasting scheme for wireless sensor networks. *CMC*, pages 293–297, 2010.

[19] A.-M. Kermarrec, L. Massoulié, and A. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE TPDS*, 3:248–258, Mar. 2003.

[20] P. Kyasanur, R. R. Choudhury, and I. Gupta. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. *In Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 91–100, Oct. 2006.

[21] J. Leitao, J. Pereira, and L. Rodrigues. Epidemic broadcast trees. *Int. SRDS*, pages 301–310, 2007.

[22] F. Li and I. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. *In Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, pages 193–202, 2001.

[23] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. *Transactions on Parallel and Distributed Systems*, pages 61–68, 2000.

[24] X. Liu, X. Jia, H. Liu, and L. Feng. A location aided flooding protocol for wireless ad hoc networks. *Mobile Sensor Networks*, pages 302–313, 2007.

[25] R. Oda, T. Ohta, and Y. Kakuda. An efficient on-demand hierarchical routing protocol based on autonomous clustering for mobile ad hoc networks. *ISADS*, pages 180–187, 2007.

[26] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of Internet: A Statistical Physics Approach*. Cambridge University Press, 2004.

[27] A. Polynikis. Random walks and scale-free networks. Master's thesis, University of York, UK, 2006.

[28] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks. Technical Report RR-3898, INRIA, 2000.

[29] C.-C. Shen, Z. Huang, and C. Jaikaeo. Directional broadcast for mobile ad hoc networks with percolation theory. *IEEE Transactions on Mobile Computing*, 5(4):317–332, Apr. 2006.

[30] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. 2011. arXiv:1102.2166.

[31] G. Vakulya. Energy efficient percolation-driven flood routing for large-scale sensor network. *IMCSIT*, 3:877–883, 2008.

[32] Y. Wang, G. Xiao, J. Hu, T. H. Cheng, and L. Wang. Imperfect targeted immunization in scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 388(12):2535–2546, 2009.

[33] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems Journal*, 3:63–84, 2001.

[34] Q. Zhang and D. P. Agrawal. Performance evaluation of leveled probabilistic broadcasting in manets and wireless sensor networks. *Simulation*, 81(8):533–546, 2005.