Epidemic dissemination algorithms in large-scale networks: comparison and adaption to topologies

Ruijing Hu

ED130
École doctorale Informatique, Télécommunications et Électronique (Paris)

T H È S E

présenté a
L'UNIVERSITÉ PIERRE ET MARIE CURIE
SPÉCIALITÉ : INFORMATIQUE

par Ruijing HU

pour obtenir le grade de **DOCTEUR**

Algorithmes de dissémination épidémiques dans les réseaux à grande échelle : comparaison et adaptation aux topologies

Directeur de thèse : Pierre SENS et Isabelle DEMEURE
Encadrant : Julien SOPENA et Luciana ARANTES

Soutenue le 2 décembre, 2013

Devant la commission d'examen formée de :

Rapporteurs :	Xavier DEFAGO	-	JAIST, Japan
	Aline VIANA	-	Inria, Saclay
Examinateurs :	Julien SOPENA	-	LIP6, UPMC
	Pierre SENS	-	LIP6, UPMC
	Isabelle DEMEURE	-	Telecom ParisTech (Invitée)
	Matthieu ROY	-	LAAS, CNRS
	Bertil FOLLIOT	-	LIP6, UPMC

UNIVERSITÉ PIERRE ET MARIE CURIE

DOCTORAL SCHOOL EDITE

Ecole doctorale Informatique, Télécommunications et Électronique (Paris)

P H D T H E S I S

to obtain the title of

PhD of the University of Paris 6

Specialty : Computer Science

Defended by

Ruijing Hu

Algorithmes de dissémination épidémiques dans les réseaux à grande échelle : comparaison et adaptation aux topologies

Thesis Advisors: Julien Sopena, Luciana Arantes, Pierre Sens, and Isabelle Demeure

prepared at LIP6, REGAL Team

defended on December 2, 2013

Jury :

Reviewers:	Xavier Defago	-	JAIST, Japan
	Aline Viana	-	Inria, Saclay
Advisors:	Julien Sopena	-	LIP6, UPMC
	Pierre Sens	-	LIP6, UPMC
	Isabelle Demeure	-	Telecom ParisTech (invited)
Examinators:	Matthieu Roy	-	LAAS, CNRS
	Bertil Folliot	-	LIP6, UPMC

Acknowledgments

In the past three years of thesis, I was very glad that I met many people who influenced and taught me in my work and my life. I would like to show my appreciation to these people for their invaluable help and support.

First I would like to express my deepest gratitude to my supervisors Julien Sopena, Luciana Arantes, Pierre Sens and Isabelle Demeure for offering me such an opportunity to accomplish my thesis. I also give my greatest thanks to my research colleague Leander Nokolaus Jehl in Norway. Their continuous guidance and advice have helped me throughout my entire PhD study. Their passion and attitude to research changed my previous view of academy. Their diligence and patience motivated me to improve myself. And I also want to thank Yann Thierry-Mieg, Olivier Fourmaux, Olivier Sigaud, Nocolas Labroche, Matthieu Cord, Emmanuel Chailloux, Franck Petit, Sebastien Monnet, Andrea Pinna, Olivier Marin, and Aurélie Beynier for supporting and advising me about my teaching classes.

It is an honor to have Xavier Defago and Aline Viana to review my thesis. Their time and careful reading are highly appreciated, and their detailed and professional comments on the thesis have led to great improvement on it. And I also want to express my warm thanks to Matthieu Roy and Bertil Folliot for attending the thesis defense as jury members.

I would like to give my thanks to members of Regal team: Thomas Preud'homme, Jonathan Lejeune, Karine Pires, Maxime Verron, Maxime Maxime Lorrillere, Anissa Lamani, Swan Dubois, Pierre Sutra, Pierpaolo Cincilla, Masoud Saeida Ardekani, Marek Zawirski, Suman Saha, Guthemberg Silvestre, Luiz Antonio Rodrigues, Lokesh Gidra, Florian David, Lisong Guo, and so on.

I also appreciate the friendship with some members of MoVe team: Yan Zhang, Laure Millet, and others.

The technical support of the laboratory gave me a great deal of help for my research. I would like to thank these people, even though I do not know their names yet.

I wish to thank my parents for supporting me throughout all my studies in France.

Contents

Chapter 1

Introduction

Contents

This thesis presents our work about information dissemination in large-scale networks. The probabilistic gossip algorithms are studied over random topologies, which model many typical distributed applications. We have firstly introduced a new parameter denoted *Effectual Fanout*, to uniformly evaluate the different gossip algorithms over the random graphs. Then, an efficient dissemination algorithm is proposed for scale-free topologies that are representative of some social networks.

In this chapter, the motivation of the thesis is presented in 1.1. Section 1.2 introduces two families of gossip protocols, while the impact of topology over gossip algorithms' performance is pointed out in Section 1.3. Two main contributions of our work are stated in Section 1.4. Finally, Section 1.5 shows the road map of the following chapters.

1.1 Motivation

In this thesis, we focus on one of the basic problems in many distributed systems and applications: information dissemination in large-scale networks. More precisely, a message generated by a source site in a considerably wide system should be broadcasted to all other sites throughout the network by successive retransmissions. For example, in an RSS delivery system, to achieve a desirable quality of service, an updated stream from any publisher needs to notify every site when the dissemination ends.

Information dissemination is an important subject, as well as a difficult issue for the large-scale networks since the past decades. A large number of gossip algorithms have been proposed to deal with it. However, as far as we know, there was not a general method to compare the gossip algorithms. Therefore, in the first step of our

research, we gave a method to conduct a fair comparison of some principle gossip algorithms. We have observed different dissemination power that widely used gossip algorithms have over various random graphs, which typically model peer-to-peer system, sensor network, and social network. We thus introduced a new parameter denoted *Effectual Fanout* to uniformly quantify such a power. Then, the impact of the topology on the performance is fairly compared for the gossip algorithms with different natures of pre-configured input parameters. This gives us insight how to combine the gossip algorithm and the topology to have the best gain in terms of reliability.

Since the global choice of pre-specified input parameter for gossip algorithms is not always feasible for gossip algorithms, in the second step, we proposed an efficient distributed gossip algorithm that detects and exploits property of the underlying network topology, which is a scale-free network, to take retransmission decision. It outperforms some classic gossip algorithms.

1.2 Gossip Protocols

The most straightforward distributed solution for information dissemination is the *pure flooding* protocol [82]. Basically, every site forwards the message that has been received for the first time to all its neighbors. In principle, when the message dissemination finishes, 100% of the sites have delivered a copy of the message. Nevertheless, actually, broadcast storm [111] may occur. It is a phenomenon that extreme amounts of broadcast messages are generated in the network, which can turn to a serious bottleneck, entailing for instance network congestion and message loss, hindering normal retransmissions within the large-scale broadcast.

In order to minimize the above drawbacks, many optimized *gossip protocols* have been provided, such that a study field of gossiping or epidemics algorithms emerges. For instance, upon a disease contagion, every contaminated site randomly infects some of its neighbors. After some time, the epidemics will sweep through the whole system, whereas global information about every site infection is never required. The effectiveness of gossip algorithms can be evaluated by some metrics, such as reliability, message complexity, and latency [65, 86, 92]. When the information dissemination finishes, the *reliability* measures the percentage of messages received by all the sites in the system; the *message complexity* expresses the average message redundancy in every site; and *latency* defines the time required to send a message from the source site to the last site that delivers the message.

There are three basic approaches for dissemination algorithms that are presented in [35]. They are respectively push, pull, and push-pull algorithms. Push algorithms allow every site to push a message to its neighbors, while in pull algorithms, every site asks its neighbors to send it missing messages if they have. Push-pull algorithms in turn combine the two methods correspondingly into two phases to disseminate information. As shown in the literature, the pull algorithm is difficult to implement [86], and constraints of push-pull algorithm mainly consist in its push phase [78].

We thus concentrate our work on push algorithms. We discuss in the future work how our solution can leverage the performance of push-pull algorithms in our future work.

Basically, push algorithms require some sites of the system to be responsible for relaying message to some of their neighbors. In contrast to the pure flooding, they substantially reduce duplicated messages during message dissemination. On the other hand, as there are usually many paths from the source to every site, protocols that choose the optimal ones are able to ensure high reliability with low message complexity (i.e., in an ideal case the message reaches every site at the end of the dissemination). Moreover, the latency also catches a great attention. However, as shown in [92], there are trade-offs amongst reliability, latency, and message redundancy. The avoidance of message transmission over some channels of a network can eliminate some short routes from the source site to every site, or even cut the unique path between them. In this case, both the reliability and the latency would degrade. Therefore, an efficient dissemination algorithm must provide high reliability, while minimizing both message redundancy and latency. To this end, a large sum of literature answers how to efficiently choose such forwarding sites and forwarded neighbors to satisfy application constraints.

The decision of such a forwarding scheme is either carried out in a deterministic or probabilistic way. The former can ensure high reliability, but its implementation is commonly very hard (e.g., some of them are proved to be NP-hard to reach the optimal performance). For the latter, it lightly takes advantage of local information of the sites in one-hop or two-hop neighborhood, while the reliability depends on the choice of their pre-configured input parameters.

1.2.1 Deterministic Gossip Protocols

Generally, there are two principal approaches behind the deterministic gossip algorithms: *neighbor coverage based algorithms* [96, 143] and *dominating sets based algorithms* [152, 153]. In the former, a site forwards the message if some of its neighbors cannot receive the message from other sites. Thus, it ensures that every site can receive the message before the end of dissemination. On the other hand, it reduces message complexity compared to the pure flooding. The second principle exploits the fact that every site is either in a forwarding list or at least one of its neighbors is in the list. Thereby, if the sites in such a list are connected, all of them are thus connected, which ensures reliable information dissemination. Yet, both principles are proved to be NP-complete problems.

There are some other heuristical techniques to implement the gossip algorithm. *Resource aware algorithms* [97, 139] take the site resource availability (e.g., battery energy) into consideration, when deciding whether to relay the message. Cross-layer designs are thus required, which is not trivial at all.

Radius based algorithms [23, 141] help every site fine-tune its radius to minimize overlap of neighboring sites' transmission region. Latency may increase, since the message that could have been transmitted from one site to another in merely one hop

must cross several hops whose transmission ranges are shorter than before. Finally, *counter based algorithms* [87] use the number of duplicated copies overheard in every site to determine retransmission policies. It is simple, though their reliability is not ensured on arbitrary topology.

1.2.2 Probabilistic Gossip Protocols

Probabilistic algorithms are widely applied either to overlay networks [47, 52, 86], as well as wireless ad-hoc and sensor networks [15, 61, 65, 132, 145]. Simply with the information of the one-hop neighbors, probabilistic gossip algorithms mitigate the undesirable *broadcast storm* phenomenon [54] by reducing at random the number of edges over which messages are transmitted [32, 51, 61, 86] or by forwarding messages with some probability [65]. Despite such advantages, probabilistic gossip protocols do not always ensure 100% of reliability (i.e., all sites receive the message by the end of the dissemination). Moreover, neither the probability nor the size of randomly chosen subset of edges to forward the message can be easily pre-fixed. Consequently, the probabilistic algorithms found in the literature mainly focus on how to finely tune the input parameters of different natures in order to find a good trade-off between the reliability and message complexity. A percolation phenomenon has been observed: above some threshold of message complexity, a zero reliability immediately goes close to 100%, which is in accordance with the percolation theory [63].

There are three principal probabilistic gossip families, which we denote (1) Fixed Fanout Gossip (*GossipFF*) [86], (2) Probabilistic Edge Gossip (*GossipPE*) [132], and (3) Probabilistic Broadcast Gossip (*GossipPB*) [65]. Many other probabilistic broadcast protocols are based on these gossip algorithms (see Section 2.4.3). *GossipFF*, has as input, the $fanout$ which is the number of randomly selected neighbors that a site should send a message. In *GossipPE*, based on an input probability parameter, a site randomly chooses those edges over which received message should be retransmitted. In *GossipPB*, the input parameter defines the probability with which a site broadcasts the message to all its neighbors. In addition, in this work, Degree Dependent Gossip (*GossipDD*) [32, 51, 61] is an improved version of *GossipPB*. It pre-specifies a degree threshold value. If a site degree is greater than this value, its retransmission probability is different from the case where the site degree is smaller than the threshold value.

Unlike deterministic gossip algorithms, probabilistic algorithms highlight their simplicity, scalability, and high reliability [54, 144], which is required by many applications in large-scale systems.

1.3 Impact of the Topology

It should be pointed out that an efficient gossip algorithm in one network may be inefficient in another, due to the different properties of the underlying graph.

The *edge dependency or clustering coefficient*, for instance, is an important graph property, which has a significant impact over gossip algorithms. It expresses which sites in a graph tend to cluster together (see Section 2.3). In wireless ad-hoc networks, it is large, while social networks have a very low edge dependency.

In [53], the authors exploit this property to propose an algorithm for wireless ad-hoc network, where the fewer the number of neighbors of a site, the higher the probability that such a site forwards its first time received message. It outperforms some classic gossip algorithms, while taking advantage of the graph's property that, in such a network, low degree sites are clustered together, as well as high degree sites. Therefore, if a low site does not retransmit the message, it is highly probable that some of its neighbors will never get it later. On the other hand, a high degree site may waste retransmission copies, if its forwarding probability is large, since its neighbors may have many opportunities to receive the message from other sites.

Contrarily, this approach cannot be applied for a social network with low edge dependency, where low degree sites are commonly connected to sites with high degree. Every new participant of social network usually becomes friend with popular people, who becomes, therefore, more popular. Thereby, the popular participants compose the heart of the network. Moreover, if the latter do not gossip the message at all, the communication system might be partitioned, since some of their friends that are solely acquainted with them will never receive it. Thereby, sites with higher number of neighbors should retransmit messages with greater probability in this topology.

1.4 Contributions

In this thesis, we are interested in the impact of topologies over the performance of gossip algorithms. Three distinct random topologies have been studied, as they model typical networks for real applications. Bernoulli (or Erdős-Rényi) graph [45] is applied to model the overlay of peer to peer system in [86], since every site independently connects to each other with certain probability. Random geometric graph [7, 119] is extensively used to simulate wireless ad-hoc environment [65]. All sites are uniformly distributed at random in an area at first. Each site has, therefore, limited signal transmission region, which is generally considered as a disc with certain radius. Finally, a scale-free topology generated by Barabási-Albert model represents social networks [6]. It is based on *preferential attachment* social behavior as in Facebook and Twitter [39]. In such networks, the new users tends to connect to popular participants who have a great number of neighbors rather than a person with very few friends. Having understood their specific characteristics for each graph, we are able to evaluate the impact of their properties over gossip algorithms.

In the context of information dissemination over large-scale networks, this thesis has two main contributions.

Contribution 1: Both the configuration parameters of a probabilistic gossip algorithm and the properties of the underlying random topology on top which it

executes (e.g., edge dependency) have an impact on the performance of the algorithms. For example, the number of messages retransmitted by a probabilistic gossip algorithm depends not only on its respective input parameters (e.g., the number of target neighbors for reception, or the probability of forwarding), but also on the graph's degree distribution (i.e., the fraction of sites having a specific number of neighbors). Nonetheless, the nature of such parameters is very different, as well as the properties of the graph. Aiming at conducting a fair uniform comparison of gossip algorithms over the random topologies, it is necessary to be enabled to evaluate the interaction between the gossip algorithms and random topologies for the sake of performance. We have, therefore, introduced a new parameter, denoted **Effectual Fanout**. For a fixed topology and gossip algorithm, the effectual fanout characterizes the mean dissemination power of the sites that have received the message from the source site. It is used to theoretically analyze the influence of topology over the performance of gossip algorithms. Performance evaluation metrics are tightly related to its value. Thanks to the effectual fanout concept, it is possible to define the reliability of gossip algorithms as a function of the algorithm's input parameters.

Contribution 2: Notwithstanding the simplicity exhibited in the probabilistic gossip algorithms, we argue that the choice of optimum values, for the input parameters of a gossip algorithm at initialization phase is not feasible for gossip algorithms in networks whose structure is unknown. Therefore, after having studied the impact of random topologies on the performance of gossip algorithms, we focus our work on scale-free graphs that model social networks, web, and complex networks, where some sites named *hubs* have much more dissemination power than the others. By exploiting the information given by neighbors in every site, we proposed an algorithm that, without any pre-fixed input argument, can automatically detect hubs in a distributed way. By a simple hub connection phase, the algorithm eventually guarantees that all hubs are well connected and every site has at least one hub or a forwarder in its one-hop neighborhood. Then, solely the hubs and forwarders take charge of message retransmissions. The reception of message by all sites is ensured in the end of information dissemination phase. The number of redundant messages is significantly reduced, which is even more outstanding than the probabilistic gossip algorithms. Latency is also reduced, when compared to other probabilistic algorithms, since hubs create many short-cuts.

1.5 Road Map

The road map of this thesis is organized as follows.

Chapter 2 gives the state of the art about gossip algorithms and dissemination of information where performance evaluation metrics, random networks, and existing information dissemination algorithms are addressed.

In Chapter 3, we present a thorough performance comparison of three widely used probabilistic gossip algorithms over well-known random graphs. These graphs represent some large-scale network topologies: Bernoulli (or Erdős-Rényi) graph,

random geometric graph, and scale-free graph. In order to conduct such a fair comparison, particularly in terms of reliability, we propose a new parameter, called *effectual fanout*. It enables to make an accurate analysis of the behavior of a gossip algorithm over a topology. Furthermore, it simplifies the theoretical comparison of different gossip algorithms on the topology. Based on extensive experiments on top of OMNET++ simulator, which make use of the effectual fanout, we discuss the impact of topologies and gossip algorithms on performance, and how to combine them to have the best gain in terms of reliability.

Chapter 4 presents a new 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. 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 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.

Finally, Chapter 5 concludes our work and gives some perspectives for future work.

Chapter 2

State of the Art

Contents

2.1 Introduction

Information dissemination is essential for many distributed systems and applications, including large-scale ones. Basically, if sites have not direct connection to all others, a source site attempts to broadcast messages to all the other sites across the network by successive retransmissions. For example, the Usenet newsgroup servers spread post. Many routing protocols [15, 102] also rely on communications amongst routers to exchange the up-date traffic information, thereby improving routing tables. Another important application comes from the area of unstructured

peer-to-peer computing. In order to successfully locate an object, which has been replicated throughout the network, a typical solution is to apply to a kind of query broadcast [103].

The principle behind information dissemination mimics the spread of epidemics, which in [49], is denoted epidemic information dissemination. It is similar to a contagious disease which is transmitted one by one from a sick person to a large population. During transmission, even though many infected people either die before infecting others, or are immunized, such epidemics are still able to be propagated across populations. This phenomenon is analyzed by Susceptible-Infection-Recovery (SIR) model in [78]. More precisely, differential equations are established to follow the rate of site contamination and immunization in a large-scale system.

Another close analogy for such an information dissemination is the rumor gossiping in social network [78]. A person is aware of some news, and tells it to some others, who in turn, continue to gossip it around, and so forth. In the end, the news is diffused to everyone in the society.

Due to these analogies, in the sequel, epidemiological and gossip terminology is interchangeable in the thesis. Sites that have received at least once the message are denoted **infected sites**, while those that received no message are denoted **isolated sites** hereafter.

Other than controlling [93] or resisting [135] such a propagation of disease or rumor, the distributed applications in which we are interested, require a dissemination protocol that efficiently broadcasts messages as far and as soon as possible. In other words, it provides high reliability, which expresses the percentage of broadcast messages that are received by all sites of the system, with both low latency and message complexity.

A straightforward but inefficient method to disseminate information throughout the network is using *pure flooding* protocol [82] which does not require global view of the system. In this protocol, upon the first reception of a message, every site of the network relays it once to its neighbors. The inefficiency of such an approach is due to the fact that a very large number of messages may be generated, which entails broadcast storm problems [111].

Different types of dissemination protocols aim at resolving such a storm problem in a more scalable and reliable way. More precisely, only some sites in the system forward the message to some of their neighbors, which can still ensure that every site delivers the message by the end of the dissemination. To evaluate such improvement in contrast to the *pure flooding*, many metrics have been employed, which will be addressed in Section 2.2. For example, message redundancy is thus reduced. In Section 2.3, three families of random topologies are introduced. They are Bernoulli graph, random geometric graph, and scale-free graphs, whose properties are exploited in our study. Then, in Section 2.4, we classify the gossip protocols. However, the underlying topologies restrict the choice of the forwarding sites and their forwarded neighbors, having thus an important impact on the performance of gossip algorithms. For example, some site with only one neighbor must require such a neighbor to forward a message. Otherwise, it will become an isolated site. In

Section 2.5, we focus on four probabilistic gossip protocols used in our work.

2.2 General Metrics

In the context of information dissemination, many metrics have been used for performance evaluation. There are principally three basic metrics: reliability, message complexity, and latency. Achievement of a good reliability is a direct criterion of a reliable information dissemination, which is presented in 2.2.1. As defined in Section 2.2.2, message complexity mainly mesures redundant message copies generated by gossip algorithms. Latency introduced in Section 2.2.3 is related to information dissemination rate. Furthermore, some other metrics that are also used in some specific contexts are shown in Section 2.2.4.

2.2.1 Reliability

Definition 1: Reliability in [92] is defined as the percentage of correct sites in a system that delivers a given broadcast message when dissemination ends. As expected, ideal gossip protocols must be able to obtain a reliability of 100% despite network omissions or site failures.

This definition also corresponds to **Fraction of Sites** that have received the message in [65]. Then, in [147], without calculating the proportion, **Number of Infected Sites** is taken as a straightforward performance metric. On the other hand, in [35], a complementary metric, **Residue**, represents the number of non-infected sites (i.e., the sites did not receive the message) when epidemic finishes. The main goal of gossip algorithm is to have such a residue as small as possible.

Besides considering the infected sites, some literature mesures the percentage of experimentations that can have a given number of infected sites. For example, in [16], **Fraction of Executions** expresses the percentage of experiment executions to reach a certain *Fraction of Sites* that have delivered the message.

Definition 2: In [48, 102], they let Δ represent any pair of real numbers (ψ, ρ) $(\psi, \rho \in [0, 1])$. A broadcast protocol is Δ-reliable, iff the three properties below are simultaneously satisfied with probability Ψ:

- *Integrity*: for any message m, every correct site delivers m at most once, and only if m was previously broadcast by the source.
- *Validity*: if a correct site broadcasts a message, then it eventually delivers such a message.
- Δ-*Agreement*: if a correct site delivers a message, then eventually at least a fraction ρ of correct sites deliver such a message.

In this way, ρ is the same as the *reliability* that has been addressed at first, while ψ indicates the probability that once a message is diffused by a correct site, the fraction ρ of correct sites eventually deliver it.

In [86], **Atomic Broadcast**[1]is defined as a broadcast where all sites in the system deliver the message in end of its dissemination. Both the percentage of such a broadcast and average fraction of infected sites in non-atomic broadcast are evaluated in their work.

2.2.2 Message Complexity

Message Complexity generally indicates the number of redundant messages copies received by every site. The higher the message complexity, the worse the efficiency of the gossip algorithm. In [92], they measured **Relative Message Redundancy (RMR)**. It captures the message overhead in a gossip broadcast protocol, which is defined as:

$$RMR = \frac{\Omega}{N-1} - 1, \tag{2.1}$$

where Ω is the overhead (i.e., the total number of messages exchanged during the dissemination), and N is system size (i.e., the total number of sites in the system) which is greater than 2 due to the fact that the denominator should not be zero. In [3, 7], the total overhead Ω is straightly used without this constraint. The optimal value for RMR is 0, which means that no redundant copy of the message is exchanged for each site. Contrarily, the higher its value, the poorer usage of network resources. Due to the trade-off between reliability and RMR, this metric is solely comparable for protocols that exhibit similar reliability. Beyond it, in [87], **Number of Rebroadcasts** is used to evaluate the number of sites that forward message.

Control messages are ignored in this metric. The reason comes from the fact that their size is typically smaller with regard to payload messages; they can usually be sent by delay and piggyback strategies, thereby saving the available network resources. However, in [112], the **Number of Control Packets** can be also considered as metric. A similar evaluation on *control overhead* is also presented in [90], since the authors believe that control communications can cause collision and channel contention as well.

2.2.3 Latency

Latency is the time between the first and last instant that the broadcast message is received (see [87]). In [92], they evaluate **Last Delivery Hop (LDH)**, which 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. Intuitively, this metric is related with the *diameter* (see Section 2.3) of the network's topology. Its value should be minimized in a dissemination protocol. Moreover, LDH can provide some comparative

[1]This notion is not the classic total order broadcast as introduced in many distributed systems [34].

measure relatively to the *latency* of information dissemination. For example, if all retransmissions present the same latency, the total latency of a gossip broadcast is the *LDH* multiplied by the *per hop latency* that is defined in [90]. As shown in [86], like the trade-off between reliability and *RMR*, the latency has a trade-off with redundant message.

2.2.4 Other Metrics

For some specific contexts, some other metrics should also be evaluated. For instance, in [10], their analysis aims at obtaining fast data replication. Then, *data accuracy* and *exchange buffer size* are taken into account. A performance metric as *function of time* is optimized in [109] for delay tolerant networks.

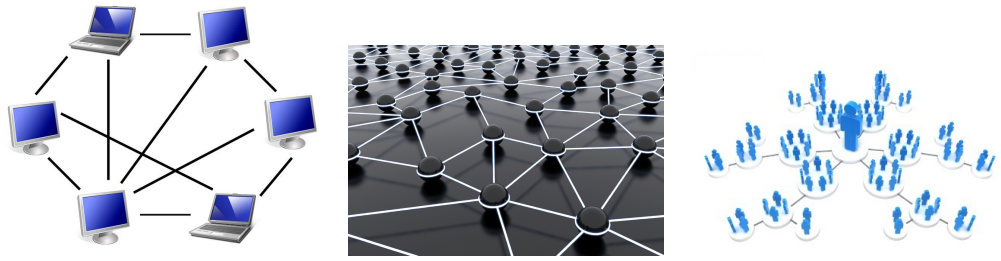
Furthermore, the reliability defined by the percentage of infected sites is measured in different ways. In [81], **Packet Delivery Ratio** is considered, while in [75], **Reception Percentage** is taken into account. In [7], they observed **Giant Component Size (GCS)**, which indicates the connectivity of a network. Thus, when *GCS* is equal to 1, the entire network is connected, which implies that every site has a message transmission path from the source site. This term is also defined as **Coverage** in [90], or **Reachability** in [87].

Instead of counting the message copies in the network such as *RMR* to compare the message complexity, some routing protocol comparisons in ad hoc networks [108, 125, 154], directly study the average energy consumption at each site for every message dissemination. **Energy Efficiency** defined in [90] evaluates the ratio of the number of sites that received the message over the number of transmissions per time. In [3], they compared **Average Collision Rate** that is the total number of packets dropped resulting from the collisions at the MAC layer. Another metric, **Throughput** was also used. It is defined as the total number of data packets received (bytes) at destinations per second. In [2], the **Normalized Throughput** is introduced. It normalizes the *throughput* by theoretical throughput. The authors in addition, presented the **Connectivity Success Ratio**, that is the ratio of the number of route reply packets received over the number of route request packets transmitted at the source site. This metric indicates the success rate of establishing paths.

In terms of latency, the **End to End Delay** is used in [88] to describe the average delay that a data packet takes to travel from source to destination. This metric includes all possible delays caused, such as buffering during route discovery delay and queuing at the interface.

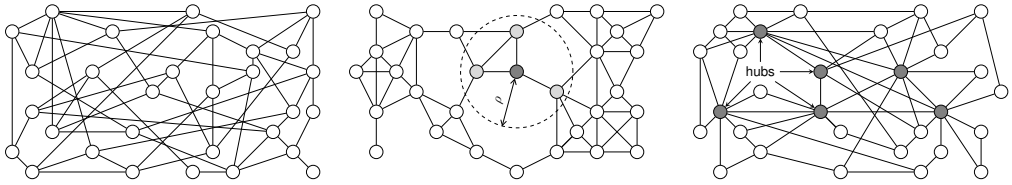
2.3 Large-scale Network Topologies

Three widely used networks have caught our attention for information dissemination: peer-to-peer system overlay, wireless sensor network, and social network. See Figures 2.1(a), 2.1(b), and 2.1(c).



(a) Peer-to-Peer System (b) Sensor Network (c) Social Network

Figure 2.1: Three real application networks



(a) Bernoulli Graph (b) Random Geometric Graph (c) Scale-Free Graph

Figure 2.2: Examples of the three random topologies with 30 sites and mean degree=4

Their properties (e.g., degree distribution, edge dependency, etc.) are analyzed by random graphs. Figures 2.2(a), 2.2(b), and 2.2(c) show Bernoulli (or Erdős-Rényi) graph $\mathcal{B}(N, p_N)$ [45], random geometric graph $\mathcal{G}(N, \rho)$ [119], and scale-free graph $\mathcal{S}(N, m)$ [6] respectively. These random topologies model the peer-to-peer system in [86], the wireless sensor network in [65], and the social network in [26] respectively.

In the sequel, $\mid l \mid$ denotes the size of set l. $P_{connect}(\centerdot)$ denotes the probability that concerned sites are connected to each other.

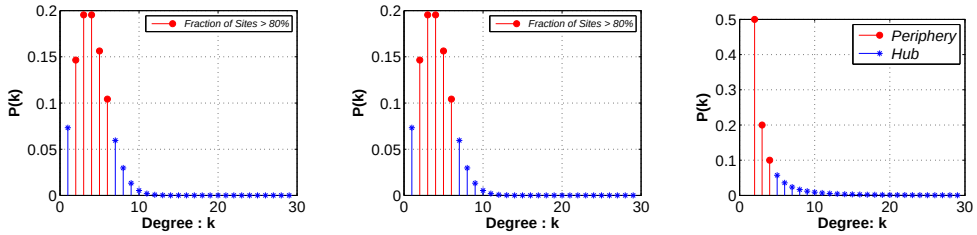
A network underlying a large-scale dissemination system Π can be viewed as a bidirectional or undirected graph. It is comprised of N sites (or vertices) $\{s_1, s_2, \cdots, s_N\}$. The set of all s_i's neighbors that have an edge with s_i (i.e., $s_i \sim s_j$), is denoted Λ_i and $V_i = \mid \Lambda_i \mid$ denotes the degree of s_i. Of a given topology, we define three important graph's properties, which are the degree distribution, edge dependency, and the diameter of a graph.

Degree Distribution, denoted $\mathbf{P(k)}$ is the fraction of sites with degree k in the graph.

The corresponding degree distributions for the three random graphs are illustrated in Figure 2.3.

Then, we denote the mean degree as $\bar{V}$. Therefore, $\bar{V} = \sum_{k=0}^{N-1} P(k) \cdot k$.

The degree distribution expresses the distribution of dissemination power in a given network.



(a) $\mathcal{B}(N, p_N)$ (b) $\mathcal{G}(N, \rho)$ (c) $\mathcal{S}(N, m)$

Figure 2.3: The degree distributions of the three random graphs

Edge Dependency (or Clustering Coefficient), denoted C of a given random graph, for distinct sites s_i,s_j,s_k, is defined as the conditional probability that, given the existence of edges $s_i \sim s_k$ and $s_j \sim s_k$, an edge $s_i \sim s_j$ also exists (i.e., $P_{connect}(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k)$).

In [13, 110], the authors proposed an alternative definition that is widely used to measure such a conditional probability. The local edge dependency at site s_i is firstly quantified as

$$C_i = \frac{number\ of\ triangles\ connected\ to\ s_i}{number\ of\ triples\ centered\ on\ s_i},$$

where a *triple centered on* s_i means an unordered pair of neighbors of s_i in an undirected graph. Thereby, the denominator is $\frac{V_i \cdot (V_i - 1)}{2}$. Figure 2.4 illustrates an example. The edge dependency of a colored site s_i is derived in three different cases, where solely solid lines represent connected edges. Then, the edge dependency for the whole graph is the average across all sites. Thus,

$$C = \frac{1}{N} \sum_{i=1}^{N} C_i$$

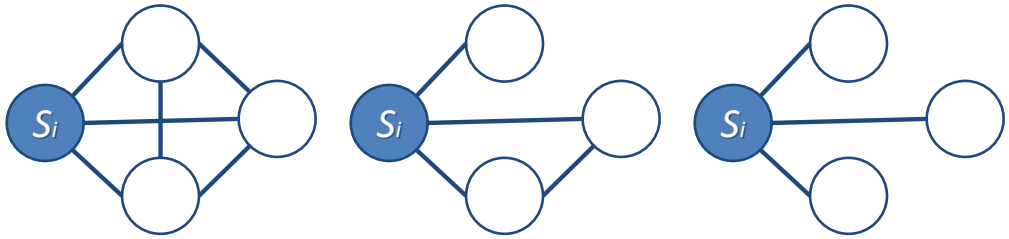
.

This property has a high impact on the number of redundant copies received by sites for *pure flooding* broadcast, where the greater the edge dependency, the higher the redundancy.

The **Diameter** of a graph is the longest length of shortest path amongst all shortest path between any two sites in the graph.

In other words, a graph's diameter is the largest number of sites which must be traversed in order to travel from any site to another one. Hence, as described in Section 2.2.3, this property is closely related to the latency.

The differences in properties of the random graphs will be addressed in the next sections, and they are also resumed in Table 2.1.



(a) $C_i = 1$ (b) $C_i = 1/3$ (c) $C_i = 0$

Figure 2.4: The edge dependency of site s_i

Graph	Notation	Degree Dist. $P(k)$	Degree Variance	Edge Dependency	Diameter	P.A.[2]
Bernoulli	$\mathcal{B}(N, p_N)$	$\exp\left(-\bar{V}\right)\frac{\bar{V}^k}{k!}$	$\bar{V}$	p_N	$\frac{\log(N)}{p_N \cdot N}$	No
Random Geometric	$\mathcal{G}(N, \rho)$	$\exp\left(-\bar{V}\right)\frac{\bar{V}^k}{k!}$	$\bar{V}$	0.5865	$\frac{\sqrt{a^2+b^2}}{\rho}$	No
Scale-free	$\mathcal{S}(N, m)$	$\frac{2m(m+1)}{k(k+1)(k+2)}$	∞	$\frac{m_0-1}{8}\frac{(\log N)^2}{N}$	$\frac{\log N}{\log\log N}$	Yes
	Deg. Seq. Based	$\frac{k^{-\tau}}{\zeta(\tau)}$	∞	–	–	No

Table 2.1: Properties of Graphs

Beyond the properties that have been presented so far, there exist some others, such as the **Connectivity** and the **Accuracy** discussed in [92]. In order to evaluate the performance of dissemination algorithms, the former should be satisfied in order to ensure that all sites are connected in the network (i.e., the *connectivity* is 1). Otherwise, isolated sites will inevitably appear, and the reliability never reaches 100%, even with execution of *pure flooding* protocol. The *accuracy* of site is defined as the number of its neighbors that have not failed till the end of broadcast divided by its degree. The fault-tolerance is not taken into account in our research yet, but will be tackled in our future work. Furthermore, **Expansion** properties (e.g., *edge expansion*) is also exploited in some existing work in order to study some limit conditions for spread of epidemics in [58]. Informally, a graph can efficiently and quickly diffuse information, if it has a low mean degree and high expansion parameters. Besides, in [107], *conductance* property is used to show connectivity properties of Internet topology. It can be used to explain the information dissemination rate. Some further properties, such as *network resilience*, *degree correlations*, *mixing patterns*, *community structure*, etc. are discussed in [110] for complex networks.

[2]Preferential Attachment: a new site preferring connecting to an existing site with high degree in the network.

2.3.1 Bernoulli Graph $\mathcal{B}(N, p_N)$

Unstructured overlays of peer-to-peer system can be modeled by Bernoulli Graph in [86]. Bernoulli (or Erdős-Rényi) graph $\mathcal{B}(N, p_N)$ is a random bidirectional graph constructed by connecting N sites randomly with probability p_N, independently of other edges. Based on [44], $p_N > \frac{(1+\varepsilon)\cdot\ln(N)}{N}$, with a positive constant ε, aiming at having a *giant component* which would have N sites.

Degree Distribution: In $\mathcal{B}(N, p_N)$, every pair of sites (s_i, s_j) is connected independently with probability p_N. Thereby, $P_{connect}(s_i \sim s_j) = p_N$. Then the degree distribution is given by $P(k) = \binom{N-1}{k} p_N^k (1-p_N)^{(N-1-k)}$. If N is large, we can approximate that by Poisson-law distribution $P(k) = \exp\left(-\bar{V}\right) \frac{\bar{V}^k}{k!}$, where $\bar{V} = p_N \cdot N$ (See Figure 2.3(a)).

Edge Dependency: In [8], it has been proved that $\mathcal{B}(N, p_N)$ has little edge dependency, i.e., the existence of an edge over $\mathcal{B}(N, p_N)$ does not depend on the others. Therefore, $C = P_{connect}(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k) = P_{connect}(s_i \sim s_j) = p_N$.

Diameter: In [27], the authors showed that, if $p_N \cdot N \geqslant 1$ then, asymptotically, the diameter tends to $\frac{\log(N)}{p_N \cdot N}$, as $N \to \infty$.

2.3.2 Random Geometric Graph $\mathcal{G}(N, \rho)$

The random geometric graph $\mathcal{G}(N, \rho)$ is a graph whose sites are positioned uniformly at random in a bounded region. In our research, such a region is a rectangular plane with length a and width b, as introduced in [65]. Furthermore, two sites are connected, whenever the distance between them is at most ρ. Based on [120], we can fine-tune $\rho > \sqrt{\frac{(1+\varepsilon)\cdot\ln(N)\cdot a\cdot b}{N\cdot\pi}}$ with a positive constant ε in order to ensure that the graph is connected [119].

There are variants in the descriptions on $\mathcal{G}(N, \rho)$, either considering it as a finite graph [7], or an infinite graph [119] without border effect, when the topology of a wireless sensor network is analyzed. The **Border Effect** is a phenomenon that the sites on the borders of a graph have fewer neighbors than those inside the graph. Thereby, the border effect can make message transmission at border sites difficult and then, average dissemination power decreases. The reliability is thus reduced. The influence of *border effect* on the finite graph will be elaborated in Section 3.5.

Degree Distribution: The probability of any pair of sites s_i and s_j being connected is equal to the probability that s_i is located in the circle of radius ρ around s_j. Then, $P_{connect}(s_i \sim s_j) = \frac{\pi\rho^2}{a\cdot b}$. Due to the independent placements of each site, the degree distribution is given by $P(k) = \binom{N-1}{k} \left(\frac{\pi\rho^2}{a\cdot b}\right)^k \left(1 - \frac{\pi\rho^2}{a\cdot b}\right)^{(N-1-k)}$. Similar to random process of $\mathcal{B}(N, p_N)$, $\mathcal{G}(N, \rho)$ follows the Poisson-law degree distribution

$P(k) = \exp\left(-\bar{V}\right)\frac{\bar{V}^k}{k!}$, where $\bar{V} = \frac{N\cdot\pi\rho^2}{a\cdot b}$ when ignoring the border effect of the region (See Figure 2.3(b)).

Edge Dependency: In [8], $\mathcal{G}(N,\rho)$ presents a high edge dependency and the existence of edges is correlated. More precisely, when border effect is neglected $C = P_{connect}\left(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k\right) = 0.5865$, a value typically greater than the probability p_N in $\mathcal{B}(N,p_N)$ that models the peer-to-peer system overlay.

Diameter: According to [20], for our rectangular $\mathcal{G}(N,\rho)$, the diameter can be bounded as $\frac{\sqrt{a^2+b^2}}{\rho}$ if N is large. In regard with $\mathcal{B}(N,p_N)$ whose diameter is small with rare cliques, over $\mathcal{G}(N,\rho)$, its diameter tends to be large, and many small cliques turn out.

2.3.3 Scale-Free Graph $\mathcal{S}(N,m)$

Social networks like Twitter or Facebook can be modeled as scale-free graph, which characterizes their properties, as, for example, power-law degree distribution. Consequently, unlike $\mathcal{B}(N,p_N)$ and $\mathcal{G}(N,\rho)$, scale-free graph's degree variance is quite high. There are mainly two ways to generate a graph satisfying such a condition. One proposed in [136] is based on degree sequence; the other applies Barabási-Albert model [6]. In regard with the former, the latter introduces the so-called *preferential attachment*, which conforms to some social behavior [110].

Graph Generation 1: A feasible generation given in [136] is achieved in two phases. First, a degree sequence $\{V_1, V_2, \cdots, V_N\}$ is obtained by sampling the N values from a power-law distribution $P(k) = \frac{k^{-\tau}}{\zeta(\tau)}$, where τ is an exponent parameter, and $\zeta(x)$ is the Riemann zeta function [155]. Then, $\sum_{i=1}^{N} V_i$ (an even value) labeled balls are put inside an imaginary urn, where label i between 1 and N is given to exactly V_i balls. In the second phase, a pair of balls labeled by i and j is randomly drawn from the urn and an edge $s_i \sim s_j$ is added to the graph. This process is repeated until the urn becomes empty.

Obviously, the degree distribution follows the power-law, by which the degree sequence is produced. However, the theoretical study on its other properties has not been found so far. Even though, we can have some intuitions. Due to the independent ball picking for edge connection, the edge dependency for this graph can be bounded in order of $1/N$, while its diameter must be in the same order of $\mathcal{B}(N,p_N)$.

Despite this straightforward manner, our attention focus on the second method i.e., Barabási-Albert model, to carry out a scale-free graph $\mathcal{S}(N,m)$, which is widely used to explain some inherent social actions.

Graph Generation 2: Scale-free graph $\mathcal{S}(N,m)$ is a random bidirectional graph generated by Barabási-Albert model [6]. Starting from a small *clique* of m_0 sites,

at every time step a new site is added such that its m ($\leqslant m_0 \ll N$) edges connect it to m different sites already present in the graph. The probability p that a new site will be connected to an existing site is proportional to the degree of the latter. This is called **Preferential Attachment**. This behavior will be studied in Chapter 4. Figure 2.5 shows that due to the preferential attachment, a new site s_6 is more likely to connect to sites with high degree. Therefore, much more probably, s_6 will connect to sites s_2 and s_3 that have highest degree in the graph at the moment.

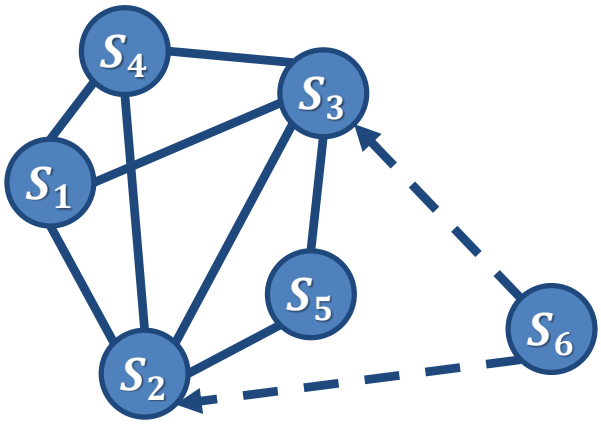


Figure 2.5: Preferential attachment in Barabási-Albert model with $m_0 = 4$ and $m = 2$

Degree Distribution: This process ensures that the graph $\mathcal{S}(N, m)$ is connected with power-law degree distribution approximately equal to $P(k) = \frac{2m(m+1)}{k(k+1)(k+2)}$ where $k = m, m+1, \cdots, N-1$ and $\bar{V} = 2m$ which does not depend on N [123] (See Figure 2.3(c)). In this network, there are **hub** and **periphery** sites which have degree greater than $2m$ and between m and $2m$ respectively. Hence, the system Π is composed of the set of hubs denoted Π_h and the set of peripheries denoted Π_p. Some theoretical results on their characteristics will be shown in theorems of our research later. For instance, we can deduce that $\mid \Pi_p \mid > 3 \mid \Pi_h \mid$.

Edge Dependency: In [55], Fronczak et al. deduced edge dependency in graph $\mathcal{S}(N, m)$. The authors obtained $C = \frac{m_0 - 1}{8} \frac{(\log N)^2}{N}$, if both N and m_0 are large. Notice that the edge dependency is very low in $\mathcal{S}(N, m)$, i.e., almost in the same order of $\mathcal{B}(N, p_N)$.

Diameter: The diameter of the network $\mathcal{S}(N, m)$ increases logarithmically with network size. If $\gamma = 0.5772$ denotes the Euler constant, then in [56], it has been demonstrated that the mean average shortest path length of $\mathcal{S}(N, m)$ is $\frac{\log N - \log(m_0/2) - 1 - \gamma}{\log\log N + \log(m_0/2)} + \frac{3}{2}$. From another perspective, compared to the two other

graphs, $\mathcal{S}(N, m)$ has the smallest diameter due to the *hubs* that create short-cut paths [17].

2.4 Taxonomy of Algorithms

Implementation of a gossip algorithm may be carried out in a probabilistic or deterministic way, based on whether or not a random number selection was used to make decisions. Section 2.4.1 presents three basic approaches to disseminate information. Deterministic gossip protocols are introduced in Section 2.4.2, while Section 2.4.3 shows three families of probabilistic gossip algorithms and their percolation models. Finally, in section 2.5 we compare four gossip protocols that have been analyzed in our study.

2.4.1 Dissemination Algorithms

According to [35], there are essentially three basic approaches as follows to retransmit message at every site for information dissemination.

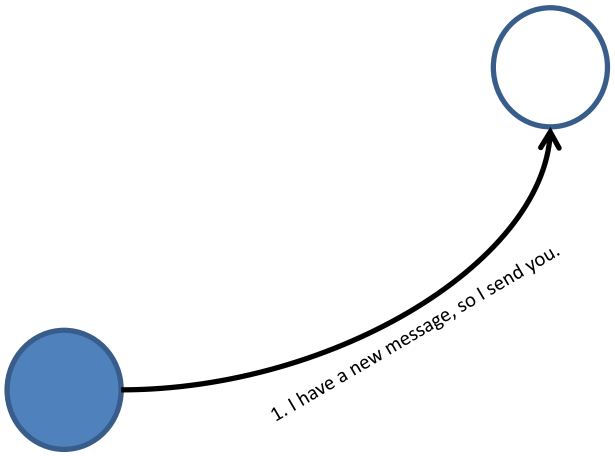


Figure 2.6: Push Algorithm

- **Push Algorithm**: upon the first reception of the message generated by a source site, every site locally decides how to forward the message to its neighbors. An example is illustrated in Figure 2.6.
- **Pull Algorithm**: periodically, every site queries its neighbors for information about their recently received or available messages. Once being aware of any message that has not been received yet from them, the site explicitly requests the concerned neighbors to forward the missing message. Figure 2.7 shows an example. This is a strategy that works best as a complement to a best-effort broadcast mechanism (i.e., IP Multicast [33]). Trivially, though enabling

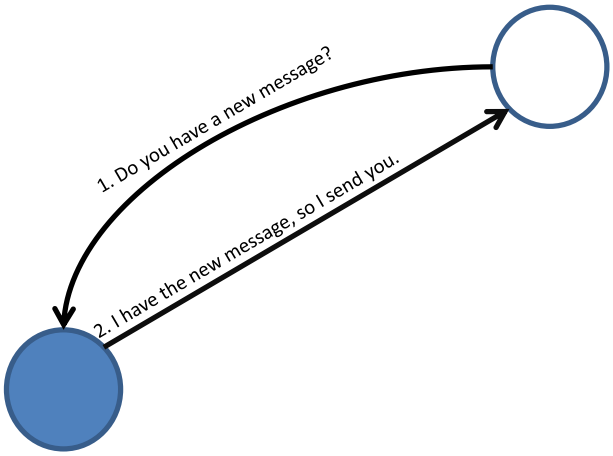


Figure 2.7: Pull Algorithm

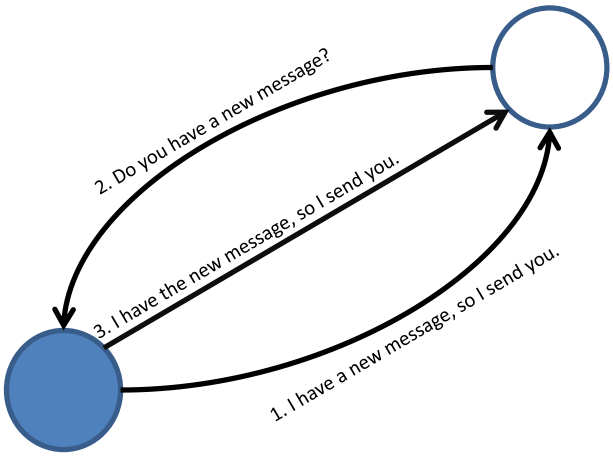


Figure 2.8: Push-Pull Algorithm

higher reliability to be reached, it requires both periodic meta-information exchanges. As shown in [78], it has lower information dissemination rate than *Push Algorithms* at the beginning when very fewer sites have the message, while it can spread information significantly faster than *Push Algorithms*, whenever less than half of sites have not been infected.

- **Push-Pull Algorithm**: as the hybrid name implies, two distinct phases are involved. We can observe it from Figure 2.8. The first phase applies *Push Algorithm* to disseminate a message in a best-effort manner. Within the second phase, *Pull Algorithm* is used to recover message loss due to some isolated sites. Owing to the second phase, the forwarding decision in the first phase can be

somewhat conservative to entail fewer retransmissions. The idea is to ensure a high reliability with tolerable extra information exchanges.

The study of these three algorithms can give a sound knowledge to propose new efficient reliable gossip algorithms.

In [121], the authors discussed the latency of Push Algorithm, while in [84], *Phone-Call* model is exploited to study message complexity and latency resulted from Push-Pull algorithm. Moreover, the bounds of both message complexity and latency are conjectured on top of simulations in [35] for the three algorithms. The theoretical results obtained in [84] are outlined in Table 2.2.

Algorithm	Message Complexity[3]	Latency (cycles)
Push	$\ln(N)$	$\log_2(N) + \ln(N)$
Pull	$\ln\ln(N)$	$\ln(N) + \ln\ln(N)$
Push-Pull	$\ln\ln(N)$	$\log_3(N) + \ln\ln(N)$

Table 2.2: Performance of three dissemination approaches

We observe that Push-Pull algorithm takes advantages of both Push Algorithm and Pull Algorithm in terms of message complexity and latency. In fact, in order to ensure high reliability it is necessary to detect and retrieve lost messages for some protocols after push algorithm has been executed [114].

In practice, Push-Pull Algorithm is widely implemented, for instance, in anonymous gossip [25] and in Publish-Subscribe system [30]. Some overlay broadcast systems apply Push-Pull algorithms [38] to ensure high reliability. Setting start-up time for the pull phase has an impact on the performance of Push-Pull Algorithm. Furthermore, as concluded in [86], the pull phase is difficult to implement, and then constraints of push-pull algorithm mainly consist in its push phase [78]. Therefore, we restrain our following stated works to pure push algorithms.

Algorithm and Variants: Aiming at better performance, some authors propose other hybrid algorithms.

Smart Gossip protocol [90] *introduces a push-pull-push* algorithms based on heuristics. More precisely, after a time interval for the first push phase, whenever a site has not obtained as many local information updates from its neighbors as a pre-configured threshold, it will send a request to the last neighbor that updated its information. The latter directly transmits its message to the former. After a random period, holding such a message, the site pushes it to its neighbors with probability as a *sigmoid* function of the number of infected neighbors. Moreover, the threshold and transmission probability can be updated for message disseminations.

In [92], a *lazy push algorithm* is considered as a similar strategy to pull algorithm in terms of latency. In this approach, once receiving a new message, a site

[3]The authors only considered the messages that are disseminated from the source.

gossips only the message identifier (e.g., a hash of the message) to its neighbors. If the neighbors receive an identifier for a message they have not yet received, they explicitly request the message from the sender.

In [24], the authors replace the pull phase by the lazy push in their push-pull algorithm. It is worth pointing out that in terms of memory usage perspective, similarly to the pull approach, lazy push gossip strategy requires a site to maintain message copies for later retransmission upon request.

2.4.2 Deterministic Gossip Algorithms

In this section, we give an overview of deterministic push algorithms, which are concerned without any probabilistic parameter. We can group them into localized and globalized methods. Globalized methods require global topology information and attempt to work out optimum executions, in terms, for instance, of message complexity, broadcast tree, etc. Their computing complexity is usually NP-Hard [95]. Localized methods exploit merely information of one or more hop neighborhood or some other intrinsic information. As our work focuses on Push Algorithms without global knowledge of network topology, we are going to discuss some localized deterministic gossip algorithms proposed in the literature.

2.4.2.1 Neighbor Coverage Based Algorithms

Generally, if only the local view about one-hop neighborhood is available, it is not efficient to implement deterministic gossip algorithms. Neighbor Coverage Based Algorithms exploit neighborhood information within at least 2 hops to compose a forwarding list with the minimum neighboring sites in order to connect all 2-hop neighborhood sites.

In [96, 143], a piggybacked forwarding list helps in the decision of which sites within two hops from a site are eligible to retransmit the message of this site. The authors' approach is called *Dominant Pruning* (DP). For example, a site will not retransmit a message, whenever detecting that its neighbors will be infected by others. Nonetheless, finding such list is a NP-complete problem. In [100], the authors offer two pruning algorithms: Total Dominant Pruning and Partial Dominant Pruning. The former benefits from the piggybacked information of three-hop neighbors in order to take a decision about retransmission, while the latter is similar to DP.

Multipoint Relaying Pruning (MPR) in [124] uses two-hop neighborhood knowledge to reduce message redundancy, and it is implemented in OLSR protocol [77]. The principle of the algorithm is to create a *cover set*. This set contains two kinds of neighbors of a site. Firstly, the neighbors connect to the sites that will be isolated, if such neighbors do not retransmit the message. Secondly, the neighbors have very high degree, thus mighty dissemination power to infect sites within 2-hop neighborhood. In [4], they construct this cover set by ordering the identifiers of the two hop neighbors. MPR is a source aware pruning, where the forwarding list depends on the preceding broadcasting site.

The neighborhood information can also be exploited to decide forwarding priority. A scalable broadcast algorithm proposed in [118] introduces a *broadcast delay* for every site, aiming at defining a retransmission priorities. The delay is the ratio of the max degree amongst the neighbors of a site over the degree of the site. The authors in [140] designed a Lightweight and Efficient Network-Wide Broadcast algorithm in an asynchronous and distributed manner. It gives the retransmission priority to higher degree sites.

2.4.2.2 Dominating Sets Based Algorithms

In [152, 153], the authors describe a distributed algorithm for calculating *Connected Dominating Set* (CDS) in ad hoc wireless networks, while site identifiers should be totally ordered. In [138] instead of the identifiers, site coordinates are used to obtain a CDS. Such algorithms require two-hop neighborhood information available for each site.

More precisely, CDS is such a connected set, that any site in the network either belongs to the set or is a direct neighbor of some site in it. Message retransmission is restricted to sites in the CDS, which therefore, infect all sites. Unfortunately, the problem of finding such a minimum CDS has been shown to be NP-complete [64]. 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.

There are a great number of variants using CDS to compose forwarding list.

The hexagonal dominating set flooding algorithms of [116] proposes to reduce message redundancy. Every site chooses six closer neighbors that better form an approximate regular hexagon to forward messages. However, as observed in [89], there is a consensus problem on the choice of ideal neighbors at every site. Thus, an additional stopping condition is proposed to resolve it. Evidently, both of them cannot ensure that all sites are infected when dissemination finishes.

Compared to MPR, though independent of the source, the abovementioned CDS is not optimal due to many redundant sites that compose the CDS itself. Then, a new algorithm comes out by combination of the two principles. In [4], MPR-Dominating set is determined by two conditions: if a site has the smallest identifier in its neighborhood, and if the site is a forwarding neighbor of the neighbor with the smallest identifier, then the site belongs to such a set. To further reduce the size of relaying set, [151] enhanced the first condition as follows: the site has the smallest identifier in its neighborhood and also two unconnected neighbors. Moreover, both approaches compute the set by heuristic greedy set cover algorithm proposed in [124] for MPR.

There are also alternative ways to find out a CDS, which adapt to some specific application context.

In ad hoc networks, the available energy resource can also be taken into consideration to improve performance. Cross-layer designs emerge as such awareness is involved. In [139], a scheduling scheme is presented to maximize lifetime of every

site and networks. A site is either in an active or a passive state. All sites are dynamically and fairly activated to create a CDS to disseminate information. Then, sites in passive mode can save their energy when *sleeping*.

In [112], 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 message 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 the identity of the elected cluster-based sites cannot be simultaneously deduced by all sites in just one-hop information exchange. To some extent, the cluster-based broadcasting can be seen as a distributed way to get a CDS.

2.4.2.3 Other Approaches

Besides the two main methods discussed above, there are also some other distributed approaches to ensure the connectivity of networks.

In some wireless networks, some works in the literature optimize the performance by varying the transmission radius. Namely, underlying topology is fine-tuned to make dissemination redundancy as little as possible. In [141], they studied a *Relative Neighborhood Graph* (RNG), which minimizes the mean degree, and on the other hand, it holds graph's connectivity. In [22], the authors proposed a distributed algorithm to find out a RNG relay subset. Solely the sites in such a subset forward messages.

Localize Minimum Spanning Tree (LMST) described in [23] requires one-hop neighbor information. However, it is not a tree anymore but with loops inside. Furthermore, they proved that LMST is a subset of RNG. Consequently, LMST always performs better than RNG in terms of message redundancy.

TreeCast, proposed in [83], is a distributed and asynchronous algorithm which detects and repairs the broken links in the tree. Since the radius depends on site power as well, in [98] every site decides a minimal Power Broadcasting Tree (PBT) (i.e., a substitution of a high power broadcast at a site with two or more successive lower power diffusions), when being aware of the power of its neighbors. Trivially, the latency is increased. In [76], a localized Broadcast Incremental Power algorithm intends to find out such a tree within k-hop neighborhood. Thereby, redundant copies are reduced to the detriment of rise of controlling messages.

Some deterministic protocols make use of counters. In [87], a color-based scheme has been proposed. In the authors' approach, each node forwards a message if it can assign it a color from a given pool, which it has not yet overheard after a time. Using geometric analysis, they have shown that the size of the rebroadcasting group is bounded by a small constant factor times the minimum CDS size. The color-based scheme is actually a kind of a counter-based scheme, whereas it does not guarantee high reliability on arbitrary topologies.

Furthermore, in some algorithms, message acknowledgement (ACK) is also in-

troduced [101, 115, 133] to ensure reliable broadcast [34].

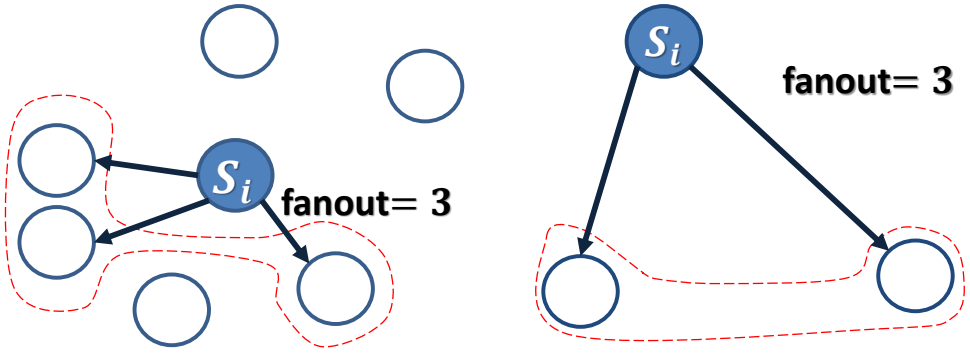
All the localized deterministic methods are summarized in Table 2.3.

Principle	Protocol	Source Aware	Neighborhood (hop)	Radius-Based	Degree-Based	Location	Pre-configured Input	ID	Reliable	Complexity ($\mathcal{O}$)
DP	[96, 143]	Yes	2						Yes	NP-complete
	[118]	Yes	2		Yes		Broadcast Delay		Yes	NP-complete
	[140]	Yes	2		Yes				Yes	NP-complete
	Total DP [100]	Yes	3						Yes	NP-complete
	Partial DP [100]	Yes	2						Yes	NP-complete
MPR	[124]	Yes	2						Yes	NP-complete
	[4]	Yes	2					Yes	Yes	NP-complete
CDS	[152, 153]		2						Yes	NP-complete
	[138]		2			Yes			Yes	NP-complete
Hexagonal DS	[89, 116]		1			Yes				
MPR-DS	[4]		2					Yes	Yes	NP-complete
Clusters	[112]		2				Cluster Size	Yes	Yes	
Energy + CDS	[139]		2						Yes	
Counter-Based	[87]		2				Counter Threshold			
RNG	[22, 141]		2	Yes					Yes	
LMST	[23]		2	Yes					Yes	
PBT	[76, 98]		k	Yes					Yes	

Table 2.3: Deterministic Algortithms

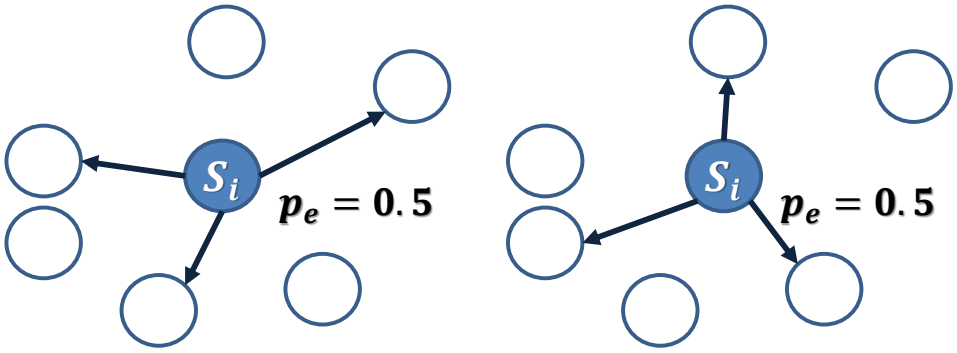
2.4.3 Probabilistic Gossip Algorithms

In contrast to deterministic approaches, probabilistic gossip algorithms highlight their simplicity, high reliability, and scalability [54, 144]. Either applied to overlay networks [47, 52, 86], or exploited in wireless ad hoc and sensor networks [15, 61, 65, 132, 145], they reduce the number of duplicated messages and well satisfy application constraints. Nevertheless, probabilistic gossip protocols do not always ensure that 100% sites are infected. As a result, aiming at a trade-off between reliability and message complexity, the algorithms found in the literature fine-tune their input parameters of different natures. There are three main probabilistic gossip families, namely (1) Fixed Fanout Gossip (*GossipFF*) [86], (2) Probabilistic Edge Gossip (*GossipPE*) [132], and (3) Probabilistic Broadcast Gossip (*GossipPB*) [65]. Many other probabilistic broadcast protocols are based on these gossip algorithms.



(a) *Degree* $\geqslant$ *fanout* (b) *Degree* $<$ *fanout*

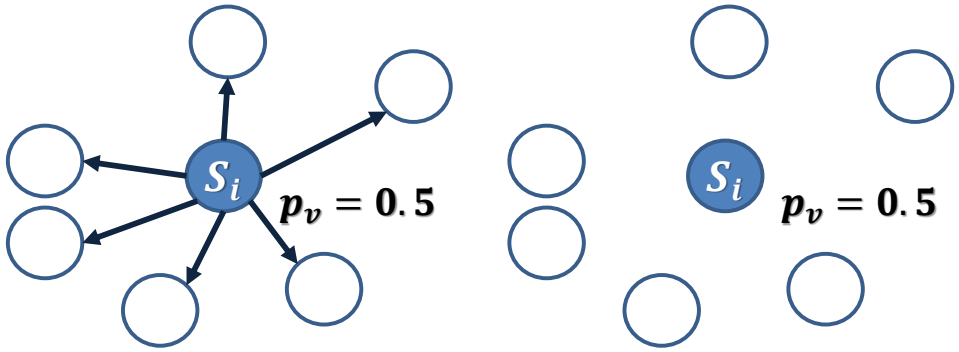
Figure 2.9: *GossipFF* with $fanout = 3$



(a) Edge Choice with $p_e = 0.5$ (b) Edge Choice with $p_e = 0.5$

Figure 2.10: *GossipPE* with $p_e = 0.5$

- *GossipFF* applies as input the *fanout* parameter, which defines the number of



(a) Retransmission (b) No Retransmission

Figure 2.11: *GossipPB* with $p_v = 0.5$

neighbors that every site selects at random to forward the received message. Figures 2.9(a) and 2.9(b) show that in two different topologies, site s_i needs to forward a message using *GossipFF* with $fanout = 3$. In the first scenario, the number of s_i's neighbors is greater than the value of the fanout, whereas in the second scenario (Figure 2.9(b)), the value of the fanout is larger than s_i's degree.

- *GossipPE* is based on an input probability parameter p_e; each site randomly chooses one by one, the edges over which received message will be retransmitted. Figures 2.10(a) and 2.10(b) show two scenarios after execution of *GossipPE* with $p_e = 0.5$ in a given topology.

- *GossipPB* has as input parameter probability p_v, with which a site broadcasts the message to all its neighbors. Similarly to *GossipPE*, in a given topology, two different execution scenarios obtained by *GossipPB* with $p_v = 0.5$ are shown in Figures 2.11(a) and 2.11(b). In the first scenario, s_i transmit the message to all its neighbors, contrary to the second one.

2.4.4 Percolation

The typical random graphs are already described in 2.3. As a matter of fact, the performance of gossip algorithms over the random graphs is always studied by the percolation theory, since many mathematical results are ready to be used for reference. Thereby, a brief presentation about it is given in this section.

2.4.4.1 Percolation Theory

Percolation theory is a research field in the domain of spatial random processes [105]. It is widely applied in electro-engineering [150], physics [122, 131], biology [128], and many other fields. The term "percolation" comes from a physical phenomenon, when

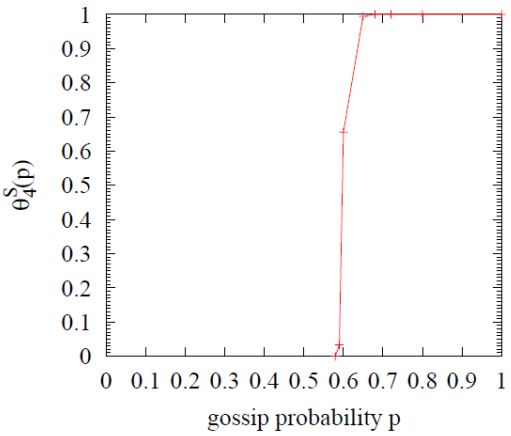


Figure 2.12: Percolation probability $\theta_4^S(p)$ as a function of the probability p. $p_c = 0.65$ is the critical value when the system changes its behavior so that $(\theta_4^S(p) = 1)$, e.g., an infinite component turns out or, in the context of wireless networks, almost all the sites receive the packets. [65]

a layer of some porous, for instance, Styrofoam or pumice stone is considered. If the cavities in it are distributed by some random process, how likely is it that some water can percolate through such a layer? Some other examples are the spreading of diseases in a population that is deployed at random over an area, or the forming of wet areas (pools of water) when rain randomly falls.

A system is said to percolate at a specific probability p_c (critical point), when a phase transition is observed. In the example with the porous material, where the probability p decides whether there is a cavity at each position: if $p \geqslant p_c$ (supercritical) there will be a path from the top to the bottom; otherwise there is none (subcritical). For a probabilistic gossip broadcast, the percentage of messages that are delivered to all sites can be as function of the probability of message retransmission in every site [65] (see Figure 2.12). In the subcritical phase there is very limited opportunity that the system percolates, whereas in the supercritical phase the percentage is almost 100%. It should be pointed out that the phase transition can exhibit different shapes [5], which are either quicker or slower. Thus, p_c becomes an interval, beyond whose bounds there are the two distinct phases, though usually in percolation theoretical problems the transition is a rapid process. Further on, the infinite systems [67] are taken into account as it makes the mathematical study much more simple, e.g., the *border effect* (see Section 2.3.2) can be ignored. Although classic percolation models are static, they can be dynamic or move according to some stochastic process [146].

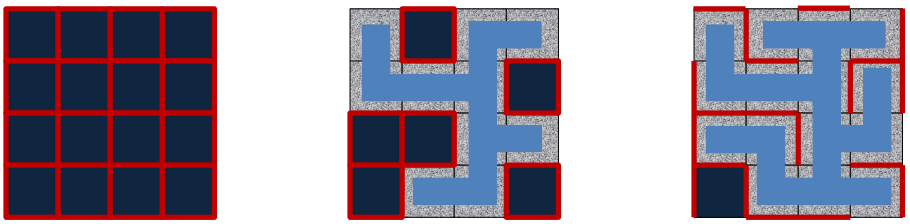
The analysis of the models of percolation theory are always related to the research on *random graphs* [44]. These graphs are generated by some random processes (see Section 2.3). A common question for the models is to determine the p_c so that all sites in the network are connected to each other with high chance.

2.4.4.2 Percolation Models

There are two types of percolation in the percolation theory: **Site Percolation** and **Bond Percolation**, which model some probabilistic gossip algorithms. Without loss of generality, we take Figures 2.13(b) and 2.13(c) to present the *site percolation* model and *bond percolation* model respectively. As shown in Figure 2.13(a), the network is an originally empty square lattice that is divided into 4×4 squares.

For *site percolation* model, the small squares (sites) in the lattice is either occupied with probability p or empty with probability $1 - p$ (see Figure 2.13(b)). The occupation or emptiness of a square is totally independent of the state of its neighboring squares. In Figure 2.13(b), all occupied squares next to each other are connected and form a *component*. With increasing value of p, more squares are occupied and larger component turns out. Evidently, if a square is around by four other occupied squares becomes occupied, all of the five squares are connected. The site percolation model well represents the porous material in the above examples.

The bond percolation model depends on the sides of the squares in a square lattice. With probability p, a side is open and with probability $1 - p$ it is closed (see Figure 2.13(c)). Neighboring squares form a *component* only when the sides in between are open.



(a) Orignal Square Lattice. All squares are empty and their sides are closed.

(b) Site Percolation. Horizontally or vertically neighboring occupied squares are connected to each other.

(c) Bond Percolation. The edges connect neighboring squares whose sides in between are open.

Figure 2.13: Percolation models over a 4×4 square lattice

The square lattice in 2-dimension as we have explained above can be replaced by many other finite or infinite structures: higher dimensional lattices, honeycombs, hypercube, and so on. The critical value p_c depends on the network type and the percolation model. Some values are selected in Table 2.4. We notice that only for some of these combinations, analytical results are available, whereas for the others, such a p_c is based on the empirical studies (i.e., simulations) [137]. Overall it is observed that networks with higher dimensions have lower critical values. Interestingly, as demonstrated in [63], in any topology, the critical value p_c for the

Lattice Type	Site Degree	Site Percolation	Bond Percolation
Honeycomb	6	0.6962	0.65271
Square	4	0.592746	0.5
Triangular	3	0.5	0.34729
Diamond	4	0.43	0.388
Simple Cubic	6	0.3116	0.2488
Body-centered Cubic		0.246	0.1803
Face-centered Cubic		0.198	0.119
Hypercubic (4d)		0.197	0.1601
Hypercubic (5d)		0.141	0.1182
Hypercubic (6d)		0.107	0.0942
Hypercubic (7d)		0.089	0.0787

Table 2.4: Critical values for the site and bond percolation models [137]

site percolation model is never smaller than for the corresponding bond percolation problem.

Percolation is also usable in other random graphs to study other problems. In [21], the authors argue that some power-law graphs such as Internet [50] are tolerant to a single site failures (i.e., the graphs are still well connected). However, the latter is true only for the random sites failures. If a particular fraction of sites with very high degree fail, the *component* in the graph may be disconnected.

2.5 Probabilistic Gossip Protocols

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 probabilistic gossip protocols that we are interested in to carry out the retransmission Gossip() procedure, namely (1) Fixed Fanout gossip (*GossipFF*)

[86], (2) Probabilistic Edge gossip (*GossipPE*) [132], (3) Probabilistic Broadcast gossip (*GossipPB*) [65], and (4) Degree Dependent gossip (*GossipDD*) [61]. They comprise the three families addressed in Section 2.4.3. Besides the received message, all these algorithms receive one or more parameters whose value is the same for all sites.

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

2.5.1 Fixed Fanout Gossip (*GossipFF*)

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.

2.5.1.1 Theoretical Analysis

In [86], the reliability of *GossipFF* in peer-to-peer system is firstly studied by both simulation and random graph theory [44]. The article mainly concludes that in a system with N sites to have the reliability equal to R, it requires to fix $fanout = -\ln\left(\frac{-\ln(R)}{N}\right)$. It is essential to know the system size in advance. Distributed algorithms in [80, 106] can estimate the system size.

Besides, in [47], a Markov Chain model is introduced to explain its infection process.

2.5.1.2 Algorithm and Variants

When *GossipFF* is carried out in real applications without theoretical assumptions, some problems in [49] are issued, such as membership maintenance, network aware-

ness, buffer management, and message filtering. Therefore, many solutions have been proposed to make it adaptable to the applications.

To optimize the choice of the input parameters for some real applications, local estimations are introduced during message dissemination. Based on the relation amongst reliability, system size, and *fanout* in *GossipFF* over Bernoulli (or Erdős-Rényi) graph, the authors in [52] make use of TCP connections to implicitly figure out available bandwidth, thus dynamically varying *fanout* according to the bandwidth.

In [129], the buffer resources at every site is managed in a realistic way. It exploits message history and discards a message on the basis of its *age* and site buffer at hand. The *age* is the number of times that a message has been forwarded. Furthermore, in [41], an overlay called *irrigation graph* is established in sensor network by *GossipFF*, which outperforms *GossipPE* and *GossipPB* in terms of reliability.

The temporally adaptive *fanout* calculated in [147] requires the information of infected and non-infected sites for every hop, which seems unrealistic.

Random Walks in [12] can be generalized as a *GossipFF* with $fanout = 1$. More precisely, every site relays a rumor message, called *token* to one of its neighbor, which is chosen with probability as a function of its degree. Moreover, the length of such random walks is heuristically pre-specified to avoid infinite information dissemination in the overlay.

2.5.1.3 Applications

GossipFF is a basic algorithm that a large number of protocols have employed so far to achieve some real requests.

The first introduction of *GossipFF* into traditional information dissemination protocols dates from [14] in its anti-entropy phase.

In [134], dynamic wired network system exploits it to meet requirements in mobile wireless applications. This random peer selection algorithm is widely utilized in publish-subscribe systems [9, 46] and the data aggregation [19, 37].

Particularly, in order to uniformly maintain a partial view (i.e., the recognition of neighbors) for a site of an overlay, *GossipFF* is the simplest way to sample and diffuse network information [79]. To this end, this algorithm is either implemented in a proactive protocol or a reactive one.

The typical proactive protocol is Cyclon [148]. In it, a partial view is periodically updated by exchange of randomly selected one or more neighbors. Thus, a partial view can be changed, even if the global system membership is stable.

The reactive protocol, Scamp [59, 60], makes the site's partial view evolved in response to some detected changes (e.g., churns) of system. Unless the membership is instable, the partial view remains unaltered.

Furthermore, in [36], a gossip protocol for rapid dissemination (i.e., with low latency), CREW applies random walks to maintain the partial view.

Geographic Gossip in [37] is also based on random walks. It improves the estimation accuracy for data aggregation, on account of exploitation of topology infor-

mation.

Despite the random walks, Astrolabe in [127] and Spatial Gossip in [85] spread information with $fanout = 1$, to fulfill network resource location service.

2.5.2 Probabilistic Edge Gossip (*GossipPE*)

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.

2.5.2.1 Theoretical Analysis

The theoretical study on *GossipPE* always goes along with *GossipPB*, since they are referred to the results in **Percolation Theory**.

As mention in 2.4.4.2, there exist two types of percolation in the percolation theory: **Site Percolation** and **Bond Percolation**. In the former, a site becomes inactive (respectively, active) with probability $1-p_v$ (respectively, p_v), blocking (respectively, forwarding) the disseminating information from itself to all its neighbors; for the latter, the edge of a site is removed (respectively, kept) with probability $1-p_e$ (respectively, p_e), blocking (respectively, forwarding) the disseminating information from the site to the neighbor linked through it.

Thereby, *GossipPE* is modeled by bond percolation in [132], while *GossipPB* matches site percolation in [108].

Both percolations over Bernoulli graph and scale-free graph are theoretically analyzed in [11]. Moreover, the percolation threshold of site percolation is always greater than that of bond percolation in any topology [63]. That is the reason that *GossipPE* entails less message redundancy than *GossipPB* to reach the same reliability. The related protocols will be stated in the following.

2.5.2.2 Applications

GossipPE is implemented for the directional antenna broadcast in wireless ad hoc networks in [132]. It presents better performance than *GossipPB* over random geometric graph $\mathcal{G}(N, \rho)$, which is studied as a function of the system size (or the site degree).

In [126], it is pointed out that the choice between *GossipPE* and *GossipPB* depends on the different application constraints over $\mathcal{G}(N, \rho)$.

2.5.3 Probabilistic Broadcast Gossip (*GossipPB*)

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.

2.5.3.1 Theoretical Analysis

Besides the analysis by the percolation theory, *GossipPB* is also modeled in some other ways.

A relation between the reliability and p_v is revealed by a recurrence model in [102] for a route driven gossip protocol.

In [113], not only the reliability is discussed by percolation property over Bernoulli graph $\mathcal{B}(N, p_N)$, but also the asymptotic expressions with respect to the average number of messages and the average time required to complete network coverage are derived, showing the benefits of a proper choice of p_v.

Dissemination latency of a modified version of *GossipPB*, where every site sends message to one neighbor with certain probability several times over a scale-free graph $\mathcal{S}(N, m)$, has been theoretically studied in [58] by using SIS (Susceptible-Infective-Susceptible) model. Nevertheless, reliability is not taken into account in the authors' study.

2.5.3.2 Algorithm and Variants

GossipPB is a simple approach addressed in [65] for ad hoc network broadcast. It turns out a *bimodal behavior*: for a majority of broadcasts, either a large or a small proportion of the sites can receive message when the probabilistic broadcast ends.

There are many other variant probabilistic algorithms from *GossipPB* that are classified as follows.

Algorithm and Variants with Globally Chosen Input Parameters: Some algorithms globally pre-specify the same input parameters for every site in the networks or sub-networks before information dissemination.

In [65, 15], the authors proposed and studied the following variants of *GossipPB*.

- *GossipPB*0(p_v): each site, after receiving a message for the first time, relays it with some probability $p_v < 1$.

- *GossipPB*1(p_v, k): it extends Gossip0(p_v) by setting $p_v = 1$ in the first k hops of a broadcast. In the remaining hops, sites relay with probability $p_v < 1$.
- *GossipPB*2(p_{v1}, k, p_{v2}, d): p_{v1} and k retain the same meaning described for *Gossip*1(p_v, k). However, $p_{v2}(> p_{v1})$ specifies the probability of relay for sites receiving the message from a site with less than d neighbors.
- *GossipPB*3(p_v, k, z): like *Gossip*1(p_v, k), when a site receives a message for the first time, it will relay it with probability p_v, with $p_v = 1$, if the source is less than k hops away. Then, each site decides not to relay, if a sufficient number (z) of received redundant copies is reached. In this way, the broadcast does not die during the gossip. The site will relay if it does not hear z duplicates within a short period of time. This may be advantageous for wireless senor networks, since it reduces the number of collisions in multiple access MAC protocols as [28] without additional cost. However, the reception of duplicates would be problematic, as it requires substantial energy due to amplification of signal.
- *GossipPB*4(p_v, k, k'): it is similar to *Gossip*1(p_v, k), while it limits its focus on routing a unicast message by dividing the systems into several zones with diameter of k' hops. When a site receives a message for the first time in the source zone, it will relay it with probability $p \leqslant 1$, while setting $p_v = 1$ in the first k hops. When the message goes outside the source zone, it will be directly forwarded to the destination in the zone.
- *GossipPB*5$(p_v, k, z = V_i)$: it is a *Gossip*3(p, k, z) where z equals to the degree of site s_i.

Moreover, in [156], the authors have proposed a probabilistic approach which combines both *GossipPB* and CDS based methods. They classify all sites into four groups according to their connectivity characteristics in two-hop neighborhood information, and assign the sites in each group with a different probability heuristically fixed a priori.

Algorithm and Variants with Locally Chosen Input Parameters: In some algorithms, every site fine-tunes the forwarding parameters according to local information from neighbors.

Enhanced RAPID in [40] gives a better infection performance with lower overhead than *GossipPB*3. After receiving a message for the first time, the algorithm waits for a small random delay, while monitoring the network. The message relay is canceled, either by none of its copy that has been heard from any other site, or by a locally chosen probability $1 - \min\left\{1, \frac{c_v}{V_i}\right\}$, where V_i is the degree of site s_i and a reliability factor c_v is related with the number of sites that take responsibility to forward message in one hop neighborhood. Then, a site that decides not to relay continues to monitor the network for an additional random time. This second monitoring period has a larger interval. The site will relay with probability 1, if it does

not hear any copy during this period. On top of this, an up-graded version in [53] performs against malicious attacks.

In [3], every site tailors the probability as a function of the number of neighbors that probably have not received the message yet, which is estimated by the former message disseminations. Notwithstanding, their optimal forwarding schemes require converge time and need a buffer to store the message reception history.

Similarly to [156], the authors distinguish all sites by four levels in [2]. Then, they simply adapt the probability in each level proportionally reverse to the number of the level for sensor network, which ensures that sites in the sparse density area forward with higher probability. Exploiting this approach, authors in [75] propose an adaptive source-dependent method that reckons on the direction of message flow from the source to adjust probability for every site in each of the four group levels.

Smart Gossip protocol proposed in [91] associates deterministic principles (see Section 2.4.2) with *GossipPB*, which shows an improvement compared to *GossipPB*2 and aforementioned protocol in [94]. It automatically and dynamically adapts forwarding probability at each site to network topology. Knowing two-hop neighborhood information, every site constructs a local relationship tree, which is composed of its parents, children and siblings. Every child in turn calculates the probability p_v for its parent to optimize message redundancy. Such a probability choice satisfies both reliability and local topology constraints. Then, the parent selects the maximum p_v from his children. Trivially, like many deterministic broadcast protocols, it requires two-hop neighborhood maintenance.

Some works, such as Gossip-based Sleep Protocols [69, 108] and NAPS [62], randomly decide whether a site can go to sleep. A message is eligible to be retransmitted by a site, whenever the site is awake once receiving it for the first time. The probability in NAPS [62] is set such that site's sleeping time is proportional to its degree. Article [42] reviewed some protocols with probabilistic sleeping mode. Not only the probability can be as a function of site degree, but also it can depend on the traffic that satisfies delay constraints, or available device battery. Yet, their goal is to save energy while still maintaining connectivity in a network, whereas our purpose is to disseminate data to all sites efficiently.

In delay tolerant networks (DTN), the forwarding scheme can be modeled as an optimization problem in [109]. Every site finally chooses a message relay probability, which is a consensus value. Such a result aims at minimizing the cost as a function of time to meet application requirements.

2.5.3.3 Applications

GossipPB and variants are initially applied and compared in traditional AODV routing protocol over regular and random geometric topologies [65]. Compared to pure flooding in mobile ad hoc networks, the most significant performance improvement is obtained by *GossipPB*3, while *GossipPB*0 performs the worse, since its execution terminates at sites very few hops away from the source. Unlike our problem, *GossipPB*4 is integrated in unicast routing protocols. However, to reach desirable

reliability, the input parameters of the algorithms are heuristic on top of simulation.

A realistic implementation without faulty sites in [15] on DES-Testbed [16] tests the abovementioned *GossipPB*1, *GossipPB*2, *GossipPB*3, and *GossipPB*5. Some explications are given for their experimental results in the end.

The percolation driven flood routing algorithm proposed in [145] can be seen as an advanced version of *GossipPB*1 in large-scale sensor networks. It gives an answer how to choose p_v to ensure high reliability. In [104], the protocol improves *GossipPB*1 by tailoring its forwarding probability as a function of site degree. Its performance is explained by a theoretical model.

In regard to *GossipPB*3, variant protocols arm a time counter as shown in [111] for copy accumulations. Hop Counter Aided Broadcast (HCAB) protocol in [74] makes the sites, upon the first reception of a message, start a random timer and record the value of hop counter. The message will be relayed by the site if, when the timer expires, no message with a hop counter higher than the first was received. Implicitly, every site in HCAB attempts to know whether each relay is done by some other sites, hopefully covering (i.e., infecting) additional regions of the network.

Another distance-based protocol is also proposed in [74], which is called Self-Adaptive Probability Broadcasting (SAPB) protocol. It stipulates that the probability of a site to relay a message is given by an aggregation of three metrics: $p_v = K \cdot f(z) \cdot g(S_x)$, where K is a constant to make $p_v \leqslant 1$, z is the number of the copies received by the site during a time interval, and S_x is the maximum of *Received Signal Strength Indicator* (RSSI) of all copies of the message received during the time interval, which implies coverage (i.e., infection) contribution brought by this site. And Functions $f(x)$ and $g(x)$ should be monotonously decreasing, so that the probability of relay decreases with the number of relay messages heard and with their RSSI.

In [130], the authors exploit *GossipPB*3 to make a robust Geocast against DoS attacks by fixing an adaptive threshold of duplicate messages during information dissemination. Unless the number of copies is under such a threshold, the gossip message is not forwarded anymore.

The protocol in [94] is based on *GossipPB*2 and *GossipPB*3, which has forwarding probability p_v for a message with sequence number n_{seq}, in reverse proportion to the number of duplicate messages overheard for message $n_{seq} - 1$.

Furthermore, [18] shows that gossip-based protocols can benefit from Kleinberg-like small-world overly topologies to reduce latency with very little cost. The impact of scale-free topologies on some of the probabilistic gossip algorithms has been evaluated by simulation in both [32] and [61] in the context of ad hoc networks and heterogeneous large-scale networks respectively. Yet, their analysis remain at experiment level to give intuitions.

Beyond this, in [26] and [39], theoretical insights of the latency performed by gossip algorithms are offered for information dissemination over a scale-free topology $\mathcal{S}(N, m)$. In [57], an optimal latency bound is derived for mobile ad hoc networks that are modeled by random geometric graph $\mathcal{G}(N, \rho)$.

2.5.4 Degree Dependent Gossip (*GossipDD*)

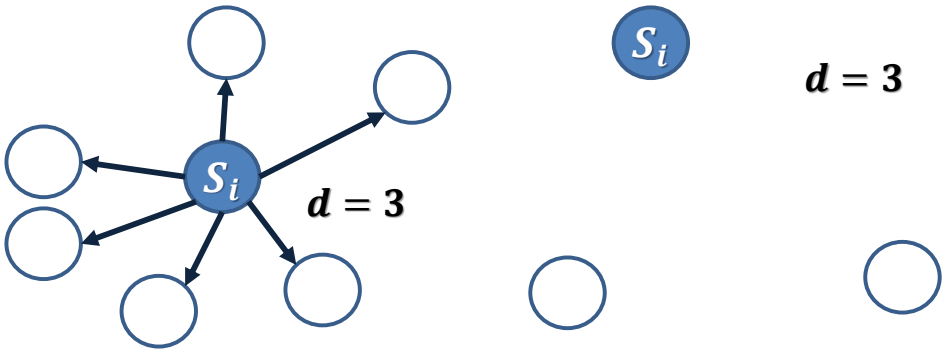
Algorithm 5: Degree Dependent Gossip (at s_i)

```
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).

2.5.4.1 Algorithm and Variants



(a) *Degree* $> d$ (b) *Degree* $\leqslant d$

Figure 2.14: *GossipDT* with $d = 3$

If we fix $p_{high} = 1$ and $p_{low} = 0$, then only sites with higher degree retransmit the message, which is named Degree Threshold Gossip, and denoted (*GossipDT*).

Our study will address this gossip protocol in Chapter 4.

Figures 2.14(a) and 2.14(b) show two scenarios that respectively represent the site degree is higher and lower to the threshold value d. Hence, the retransmission only happens in the first scenario, where site s_i's degree is greater than d.

2.5.4.2 Applications

GossipDD marries *GossipPB* and a degree-based deterministic algorithm, which has been evaluated in scale-free networks [32, 51], or compared in several topologies [61]. In scale-free networks, *GossipDD* shows its better performance than some other probabilistic gossip algorithm, since the sites with high degree are better exploited for message forwarding. However, in wireless ad hoc network the high degree sites should have lower relay probability in order to reduce collisions.

The comparison of all the probabilistic gossip protocols is summarized in Table 2.5.

Family	Protocol & Reference		Neighborhood (hop)	System Size	Degree-Based	Location	Pre-configured Input	ID	Counter	Networks	Theory
GossipFF	[86]		1	Yes			$fanout$			$\mathcal{B}(N, p_N)$	Random Graph
	[47]		1	Yes			$fanout$			$\mathcal{B}(N, p_N)$	Markov Chain
	[14]		1				$fanout$			Complete Graph	
	[12, 36]		1		Yes		Stop Time			$\mathcal{G}(N, \rho)$	Bins and Balls
	[147]		all	Yes			$fanout$			$\mathcal{B}(N, p_N)$	Recurrence
	[52]		1	Yes			$fanout$			on TCP	
	[129]		1	Yes			$fanout$		Yes	$\mathcal{B}(N, p_N)$	
	[41]		1				$fanout$			$\mathcal{G}(N, \rho)$	Random Graph
	[32, 61]		1				$fanout$			$\mathcal{B}(N, p_N)$, $\mathcal{S}(N, m)$	
	[51]		1		Yes		$fanout$			$\mathcal{S}(N, m)$	
GossipPE	[132]		1			Yes	p_e			$\mathcal{G}(N, \rho)$	Percolation
	[126]		1		Yes		p_e		Yes	$\mathcal{G}(N, \rho)$	Percolation
	[32, 61]		1				p_e			$\mathcal{B}(N, p_N)$, $\mathcal{S}(N, m)$	
	[51]		1		Yes		p_e			$\mathcal{S}(N, m)$	
GossipPB	*GossipPB1*	[15, 65]	1				(p_v, k)		Yes	$\mathcal{G}(N, \rho)$	Percolation
		[145]	1		Yes		p_v			$\mathcal{G}(N, \rho)$	Percolation
		[109]	1				Delay, Cost		Yes	DTN	Opt. Function
		[51]	1		Yes		p_v			$\mathcal{S}(N, m)$	
	GossipPB2	[15, 65]	1		Yes		(p_{v1}, k, p_{v2}, d)		Yes	$\mathcal{G}(N, \rho)$	Percolation
		[40, 53]	1		Yes		p_v		Yes	$\mathcal{G}(N, \rho)$	Random Graph
	GossipPB3	[94]	1				p_v	Yes	Yes	$\mathcal{G}(N, \rho)$	
		[15, 65]	1				(p_v, k, z)		Yes	$\mathcal{G}(N, \rho)$	Percolation
		HCAB [74]	1		Yes		(p_v, k)		Yes	$\mathcal{G}(N, \rho)$	
		SAPB [74]	1				(p_v, z)		Yes	$\mathcal{G}(N, \rho)$	
GossipDD	[32, 61]		1		Yes		(p_{v1}, p_{v2}, d)			$\mathcal{B}(N, p_N)$, $\mathcal{S}(N, m)$	
	[51]		1		Yes		(p_{v1}, p_{v2}, d)			$\mathcal{S}(N, m)$	

Table 2.5: Probabilistic Algorithms

2.6 Conclusion

In this chapter, we have discussed some concept and existing works related to information dissemination over large-scale networks. To avoid the broadcast storm problem, many gossip algorithms protocols have been proposed in a deterministic or a probabilistic way. Though deterministic algorithms present 100% of reliability and substantial message redundancy reduction, most of them are proved to be NP-complete to achieve optimal performance. Therefore, we have focused our work on probabilistic gossip algorithms and several described existing works. We have also presented a discussion about the percolation theory, which explains dissemination power of some probabilistic gossip algorithms. Since the network topology over which the gossip algorithm runs has an impact on its performance we have analyzed some typical properties of three random topologies that model some real networks. Such a study has given us some intuition on how to improve the performance of gossip algorithms by taking such properties into account.

On the other hand, input parameters of the gossip protocols are not always the same, as for instance, probability in *GossipPB*, or $fanout$ in *GossipFF*. Furthermore, topology properties like *degree distribution* or *edge dependency* may differ depending on the random graph.

Chapter 3

Fair Comparison of Gossip Algorithms over Large-Scale Random Topologies

Contents

3.1 Introduction

As discussed in Section 2.4.3, there are essentially three families of probabilistic gossip algorithms. Although they do not always ensure 100% of reliability, they are widely used to reduce the number of messages and satisfy application constraints, thanks to their simplicity and scalability [54, 144]. The algorithms usually have input parameters of different natures. *GossipFF* applies as input the $fanout$, which is the number of neighbors that a site will send the message; in *GossipPE*, based on an input probability parameter, a site randomly chooses those edges over which received message should be retransmitted; in *GossipPB*, the input parameter defines the probability with which a site broadcasts the message to all its neighbors. Besides the configuration parameters, the properties of underlying topology (e.g., degree distribution, edge dependency, etc.) have also an impact on the performance of gossip algorithms, such as message complexity, fraction of infected sites, reliability, and latency.

Considering the above discussed differences, our aim is to compare the three families of gossip algorithms: *GossipFF*, *GossipPE*, and *GossipPB* (see Section

2.5), evaluating them over three random graphs (see Section 2.3): Bernoulli (or Erdős-Rényi) graph $\mathcal{B}(N, p_N)$ [45], random geometric graph $\mathcal{G}(N, \rho)$ [119], and scale-free graph $\mathcal{S}(N, m)$ [6], which respectively model peer-to-peer system [86], wireless sensor network [65], and the Internet [110]. We should point out that our study can be extended for other gossip algorithms and random topologies.

In order to carry out a fair comparison, we have introduced a new parameter, called *effectual fanout* which expresses the average number of messages per retransmission. It characterizes, therefore, the mean dissemination power of infected sites, i.e., those that received the message at least once. In large scale systems, the effectual fanout has thus a strong linear correlation with message complexity metric as we show and prove in Section 3.3. For a given value, the *effectual fanout* can be analytically calculated as function of the input parameter of the corresponding gossip algorithm (e.g., fanout, probability, etc.) and thus, it simplifies the theoretical comparison of different gossip algorithms on a fixed topology. The advantage of using the effectual fanout compared to message complexity metric is that, the former can be easily calculated analytically, while the latter requires knowing the total number of messages generated by each algorithm as function of the topology.

Exploiting the effectual fanout, we will also present in this chapter results of an extensive performance evaluation, conducted on top of OMNET++ [1]. We have compared *GossipFF*, *GossipPE*, and *GossipPB* over the above mentioned three topologies. In order to have a fair comparison, the value of the respective input parameter of each algorithm has been varied and, the effectual fanouts for the different values have been applied (see Section 3.3). The performance evaluation results are then presented as a function of the effectual fanout: for different values of the effectual fanout, we have evaluated the fraction of infected sites, reliability, and latency of the three gossip algorithms over the three topologies.

The remainder of this chapter is organized as follows. Section 3.2 introduces some performance metrics. The effectual fanout is presented in Section 3.3. Section 3.4, Section 3.5, and Section 3.6 respectively show simulation results of the gossip algorithms over Bernoulli (or Erdős-Rényi) graph, random geometric graph, and scale-free graph on top of OMNET++, while Section 3.7 shows how to combine them to have the best gain in terms of reliability. Section 3.8 discusses some previous work. Finally, Section 3.9 concludes this work.

3.2 Performance Metrics

In the context of information dissemination, there are many existing metrics to evaluate performance of gossip algorithms (See Section 2.2). We use the common metrics introduced in the literature [65, 86, 92] as follows:

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

$$M = \frac{\Omega}{N-1}, \tag{3.1}$$

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

Fraction of Infected Sites, denoted α: is defined as the percentage of all sites in the system that delivered a message generated by a source in the end of 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., $\alpha = 100\%$ for any given message) ensuring thus *atomicity* similarly to pure flooding algorithms [86].

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 both large fraction of total infected sites and high reliability, while minimizing both message complexity and latency.

3.3 Effectual Fanout

The number of retransmitted messages of the three gossip algorithms, and therefore their message complexity, depend on their respective input parameters (p_v, p_e or $fanout$), which are, in fact, quite different. Hence, aiming at conducting a fair uniform comparison of these algorithms over the topologies described in Section 2.3, we have introduced a new parameter: ***effectual fanout*** denoted F_{eff}. The latter enables to make an accurate analysis of the behavior of a gossip algorithm over a topology. Furthermore, it simplifies the theoretical comparison of different gossip algorithms on the topology. For a fixed topology and gossip algorithm, the effectual fanout characterizes the mean dissemination power of infected sites. Thereby, when the number of sites of the system is very large, the effectual fanout has a strong linear correlation with message complexity, as shown in Theorem 7 of the current section. Notice that as function of both an algorithm and a topology, it is possible, for a given effectual fanout to deduce the value of the mentioned input parameter of the gossip algorithm in question, as shown in the following.

Based on *GossipPE*, *GossipPB*, and *GossipFF* algorithms, we define respectively that:

$$F_{eff}^{GossipPE} = p_e \cdot \bar{V} \tag{3.2}$$

$$F_{eff}^{GossipPB} = p_v \cdot \bar{V} \tag{3.3}$$

$$F_{eff}^{GossipFF} = \sum_{k=1}^{fanout-1} P(k) \cdot k + \sum_{k=fanout}^{N-1} P(k) \cdot fanout \tag{3.4}$$

due to the two conditions in Algorithm 9 described in Section 2.5.1 (lines 11 and 13).

We respectively denote $\mathbf{U_h}$ and $\mathbf{I_h}$ the expected number of sites that have not been infected before the end of hop h and the expected number of newly infected sites within hop h, for $1 \leqslant h \leqslant L$ where L is the latency. Observe that U_0 equals to $N-1$, I_0 equals to 1, and $U_L = (1-\alpha)N$.

For hop h, U_h and I_h are related as follows:

$$I_h = U_{h-1} - U_h,\ 1 \leqslant h \leqslant L \tag{3.5}$$

Theorem 1. *For the three probabilistic gossip algorithms over large scale random topologies ($N \gg 1$), message complexity is $M \approx \alpha F_{eff}$.*

Proof. Since there is no loss of messages, the total number of messages received by each site is equal to the number of transmitted messages. In every hop h, a site will relay F_{eff} messages to its neighbors, while the expected number of newly infected sites in hop h is I_h. Thus, the expected number of transmitted messages in hop h is $F_{eff} \cdot I_h$.

Considering all hops and Equation (3.5), we obtain Ω, the total number of received messages:

$$\Omega = \sum_{h=1}^{L} F_{eff} \cdot I_h = F_{eff} \cdot \sum_{h=1}^{L} I_h = F_{eff} \cdot (N - 1 - U_L)$$

By Equation (4.7), $M = \frac{(N-1-U_L)}{N-1} \cdot F_{eff} = \frac{(\alpha N-1)}{N-1} \cdot F_{eff}$ and, since N is very large we then get $M = \alpha F_{eff}$.

□

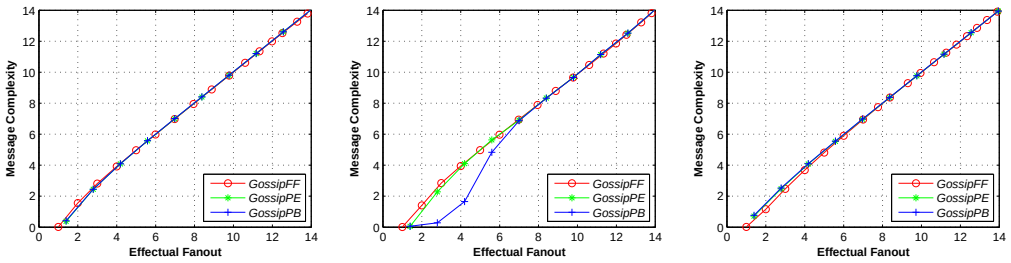
In fact, using the number of redundant transmissions resulted in delivering a given message to an infected site, which is measured in [29], we may obtain the same result.

Corollary 2. *In order to have high reliability for the three probabilistic gossip algorithms over large-scale random topologies, message complexity is $M \approx F_{eff}$.*

Proof. When the high reliability is reached (e.g., heuristically, over 95% of *total* sites are infected on average at the end of a message dissemination), α is very close to 100% and, according to Theorem 7, the result is obtained. □

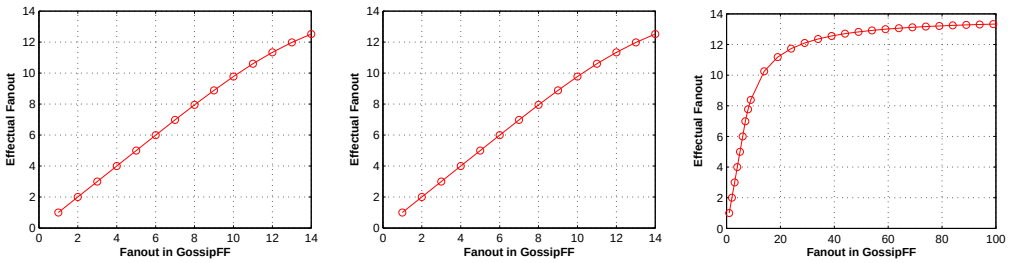
From now on, we will present a series of experimentations implemented over OMNET++. From that, we can understand the importance of *effectual fanout* on the fair performance evaluations.

We consider that the network is composed of $N = 1000$ sites and, in order to ensure connectivity, $\varepsilon = 1$ for $\mathcal{B}(N, p_N)$ and $\mathcal{G}(N, \rho)$. Since we aim at having almost the same mean degree for all topologies ($\bar{V} \approx 14.0$), the topology parameters were chosen as shown in Table 3.1:



(a) On $\mathcal{B}(N, p_N)$ (b) On $\mathcal{G}(N, \rho)$ (c) On $\mathcal{S}(N, m)$

Figure 3.1: Relation between Message Complexity and Effectual Fanout



(a) On $\mathcal{B}(N, p_N)$ (b) On $\mathcal{G}(N, \rho)$ (c) On $\mathcal{S}(N, m)$

Figure 3.2: Difference between Fanout in *GossipFF* and Effectual Fanout.

For each gossip algorithm, 200 different messages are generated by 200 different sources that are chosen uniformly amongst 1000 sites over 20 different graphs related to each of the topologies. Then, the results for each effectual fanout are averaged by the $200 \times 20 = 4000$ message disseminations.

As explained, our aim is to fairly compare the performance of the three gossip algorithms, especially the reliability, using the effectual fanout. Such fairness requires the equivalence in terms of message complexity of the three algorithms over a given topology. Therefore, we would like to verify the linear relation between the effectual fanout and message complexity, which is proved in Corollary 8. Figures 3.1(a),

[1]Since the graph is a rectangular area with border effect, its degree distribution should be adjusted on basis of the results in Section 2.3.2 for infinite $\mathcal{G}(N, \rho)$. The choice of ρ will be pinpointed in Section 3.5.

TOPOLOGY	PARAMETERS
$\mathcal{B}(N, p_N)$	$p_N = 0.014$
$\mathcal{G}(N, \rho)$	$a = 7500$, $b = 3000$, $\rho = 330$[1]
$\mathcal{S}(N, m)$	$m_0 = 9$ $(m_0 - clique)$, $m = 7$

Table 3.1: Topology Parameters

3.1(b), and 3.1(c) show this relation. They confirm that the linearity $F_{eff} = M$ holds whenever F_{eff} value is great enough ($F_{eff} > 3$). On the other hand, for smaller F_{eff} values, the fraction of infected sites (α) is too small to be neglected in the equation of Theorem 7. The only exception is for *GossipPB* over $\mathcal{G}(N, \rho)$ (Figure 3.1(b)) since, in this case, a clustering effect (see Section 3.5) prevents this algorithm to benefit from the growth of the dissemination power. Notice that such value is much smaller than the dissemination power value which provides reliability. Hence, the fairness of the algorithm comparison is ensured in this case.

Fanout vs. Effectual Fanout: Contrarily to the *fanout* in *GossipFF*, the effectual fanout takes into account the degree distribution of the topology, which makes measures and performance evaluation fairly comparable for all the three topologies. In order to well understand the difference between the $fanout$ in *GossipFF* and effectual fanout, Figures 3.2(a), 3.2(b), and 3.2(c) present the value of effectual fanout for each random topology as a function of $fanout$ in *GossipFF*. We observe that for small values there is an equality between them, while their values diverge for large $fanout$ values (i.e., the effectual fanout is proportionally smaller). In fact, those sites whose number of neighbors is inferior to $fanout$ always retransmit less than fanout messages. In $\mathcal{S}(N, m)$, where the degree variance is very high and the number of sites, such as peripheries, with small number of neighbors is very large, this phenomenon is much more remarkable (see Figure 3.2(c)). Such a phenomenon also explains why in a great number of theoretical studies, the $fanout$ of *GossipFF* is always considered as inferior or equal to the minimum degree of the graph.

3.4 Algorithms Comparison over $\mathcal{B}(N, p_N)$

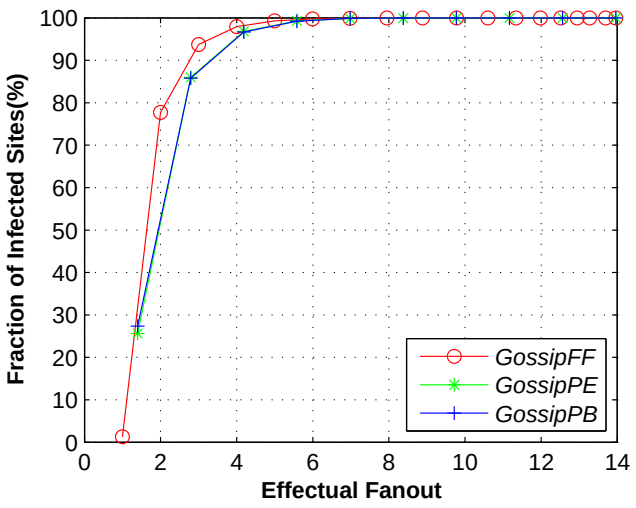


Figure 3.3: Infected Sites over $\mathcal{B}(N, p_N)$

We discuss now the performance evaluation for the gossip algorithms over

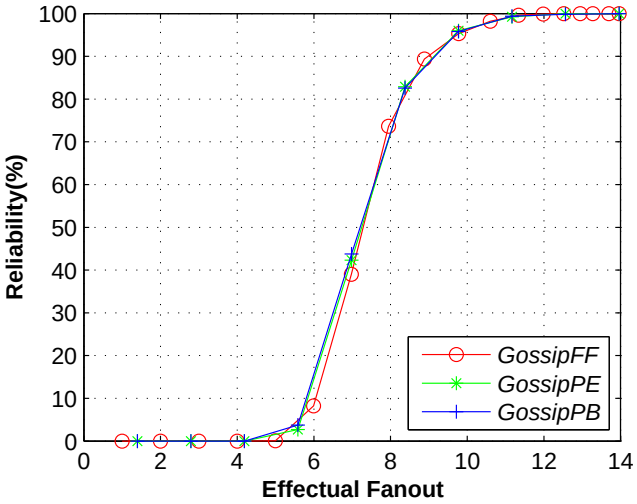


Figure 3.4: Reliability over $\mathcal{B}(N, p_N)$

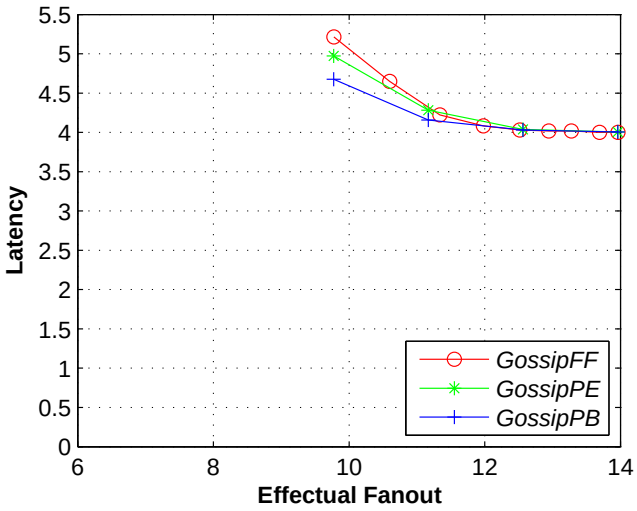


Figure 3.5: Latency (hops) over $\mathcal{B}(N, p_N)$

$\mathcal{B}(N, p_N)$. On the one hand, both the fraction of infected sites (α) and the reliability (R) in Figures 3.3 and 3.4 present a threshold effect as a function of effectual fanout. In other words, the fraction of infected sites or the reliability equals to 0 for some small effectual fanout values, but it quickly comes to 100% for a threshold value (effectual fanout = 4). On the other hand, we observe that the performance for all gossip algorithms is the same for the same effectual fanout.

However, if we compare the thresholds for the fraction of infected sites and reliability, they are different. The fraction of infected sites percolates with smaller effectual fanout than reliability (respectively, $F_{eff} = 4$ in Figure 3.3 and $F_{eff} = 13$

in Figure 3.4). As a matter of fact, when the effectual fanout is great enough such that almost all sites receive each message (i.e., $\alpha \approx 100\%$), none of the messages is received by all sites (i.e., $R = 0$). Only when the effectual fanout equals to 13 that almost all messages are surely received by every site (i.e., $R \approx 100\%$). We thus observe a great gap in terms of effectual fanout value between the dissemination power necessary for infecting almost every site and high reliability.

Since the algorithms have the same behavior over $\mathcal{B}(N, p_N)$, then we can use the theoretical result of *GossipFF* [86] to determine the corresponding thresholds for *GossipPE* and *GossipPB*: $fanout = -\ln\left(\frac{-\ln(R)}{N}\right)$. For instance, for $R = 99.4\%$, $fanout = -\ln\left(\frac{-\ln(.994)}{1000}\right) \approx 12$. By Equation (6.3) in Section 3.3 we obtain $F_{eff} = 11.3$. Thereby, $p_e = p_v = F_{eff}/\bar{V} = 11.3/14 = 0.81$. Hence, it becomes possible to dimension the input probabilities of *GossipPE* and *GossipPB* to obtain a desired reliability.

In Figure 3.5, after a given effectual fanout, latency does not decrease anymore, but converges towards pure flooding approach (i.e., the shortest routes between the source and the other sites), and therefore, towards the minimum latency.

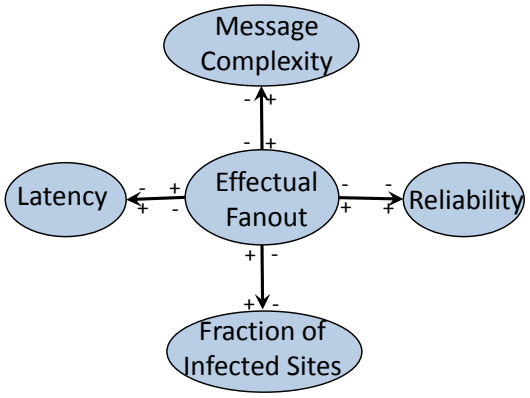


Figure 3.6: Trade-off amongst the metrics related to *Effectual Fanout*

By the means of *effectual fanout*, we can summarize the trade-off amongst the four metrics in Figure 3.6. The positive (resp., negative) sign in the figure represents a rise (resp., fall) of the corresponding value. Such relations are confirmed not only over $\mathcal{B}(N, p_N)$, but over other random graphs as well. More precisely, on the one hand, the increase (resp., decrease) of effectual fanout results in the rise (resp., fall) of message complexity, reliability, and fraction of infected sites. On the other hand, it reduces (resp., raises) the latency for the gossip algorithms.

3.5 Algorithms Comparison over $\mathcal{G}(N, \rho)$

We now present simulation results related to the performance of the gossip algorithms on $\mathcal{G}(N, \rho)$ (Figures 3.8, 3.9, 3.10, and 3.11). If the performance of the

gossip algorithms is identical on $\mathcal{B}(N,p_N)$, it is not always the case for other random topologies.

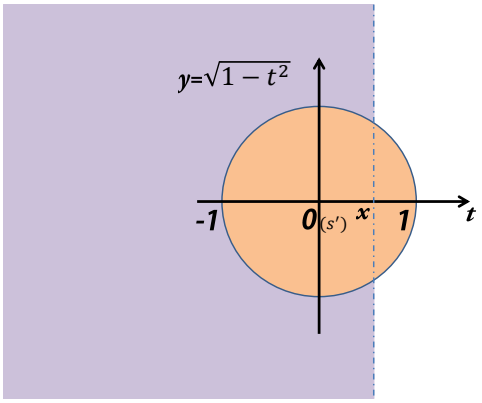


Figure 3.7: Percentage of a circle covered by s'

Since $\mathcal{G}(N,\rho)$ in our simulation is a finite graph, we firstly adjust its degree distribution that is shown in Section 2.3.2, where the *border effect* is not taken into consideration. To this end, we consider a site s' located near the border. Then, we estimate the percentage of a circle covered by s' that lies inside the rectangle (see Figure 3.7). Since our interest is the percentage, we can assume that $\rho = 1$. Clearly, if the distance of s' from the edge of the field is $0 \leqslant x \leqslant 1$, the area of the disc lying inside the field is given as

$$\int_{-1}^{x} 2\sqrt{1-t^2}\mathrm{d}t = \left(x\sqrt{1-x^2} - \arccos(x)\right) + \pi$$

Thereby, the fraction is given by:

$$F(x) := \frac{1}{\pi}\left(x\sqrt{1-x^2} - \arccos(x)\right) + 1$$

We can now compute the degree distribution as: $P(k,x) = \exp\left(-\bar{V}(x)\right)\frac{\bar{V}(x)^k}{k!}$, where $\bar{V}(x) = \bar{V}\cdot F(x)$. Finally using $P_s(k) = \exp\left(-\bar{V}\right)\frac{\bar{V}^k}{k!}$, we can write

$$P(k) = P_s(k)\left(\frac{(a-2\rho)(b-2\rho)}{ab} + \frac{2a\rho + (b-2\rho)\rho}{ab}\varphi(k)\right),$$

where $\frac{(a-2\rho)(b-2\rho)}{ab}$ is the probability that the round area covered by a site is totally inside the rectangle, $\frac{2a\rho+(b-2\rho)\rho}{ab}$ is the complementary probability to the previous one, and $\varphi(k)$ represents the average impact of the border effect, calculated as:

$$\varphi(k) = \int_0^1 \exp\left(-\bar{V}\left(F(x)-1\right)\right)\cdot F(x)^k\mathrm{d}x$$

It should point out that the above formula ignores the corners, where the expected degree of a site is even smaller. However, when $\rho \ll a$ and $\rho \ll b$, such an effect is negligible.

Therefore, its mean degree can be calculated as $\bar{V} = \sum_{k=1}^{N-1} P(k) \cdot k$. This is reason that we have chosen $\rho = 330$ to ensure $\bar{V} = 14$ in our simulation.

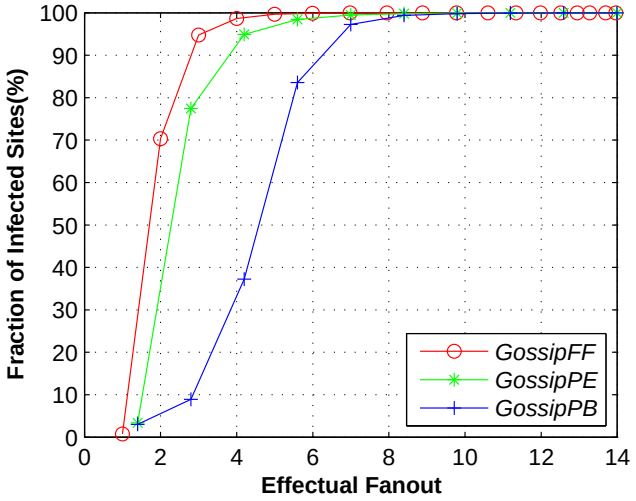


Figure 3.8: Infected Sites over $\mathcal{G}(N, \rho)$

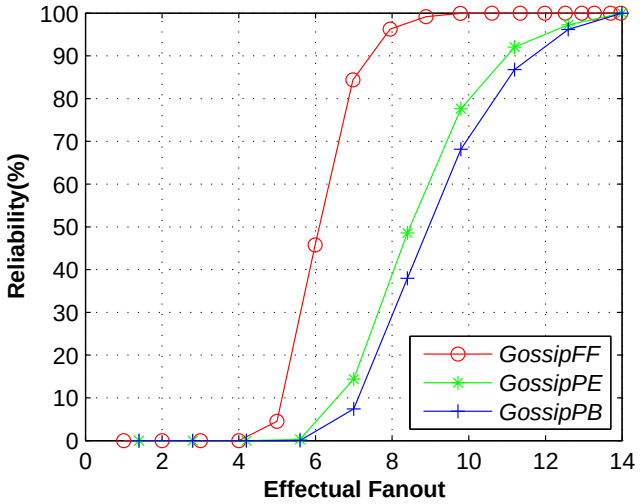


Figure 3.9: Reliability over $\mathcal{G}(N, \rho)$

Now, we can compare the performance for three gossip algorithms in $\mathcal{G}(N, \rho)$ with border effect. If we consider the reliability (see Figure 3.9), we notice that *GossipFF* is much more efficient (i.e., with merely $F_{eff} = 8.5$, $R = 99\%$ is reached) than the two other algorithms that require $F_{eff} = 14$ to reach $R = 99\%$. Furthermore, we can observe in the same figure that the threshold effect for both *GossipPE* and *GossipPB* is much smoother (i.e., from 5.5 to 14) than for *GossipFF* and, that

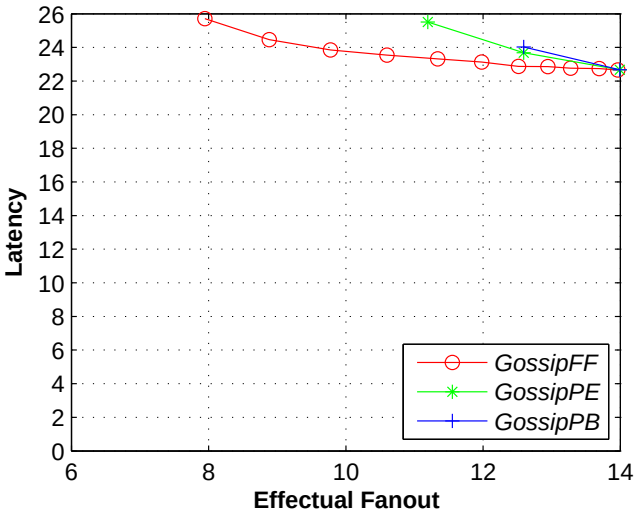


Figure 3.10: Latency (hops) over $\mathcal{G}(N, \rho)$

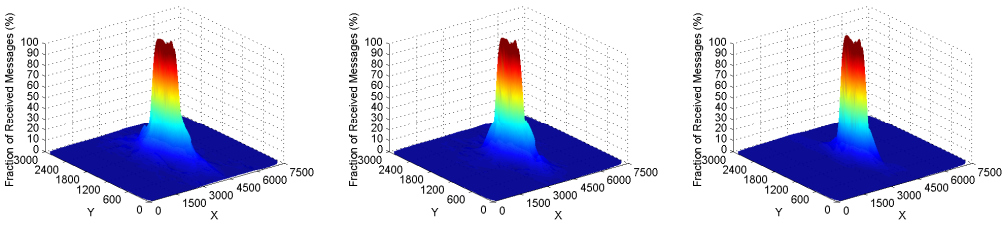
GossipPE presents a slightly better performance than *GossipPB*. However, if we look at the performance in terms of the fraction of infected sites in Figure 3.8, the comparison results are quite different. *GossipPE* has very similar performance to *GossipFF* which is the most efficient. *GossipPB* shows the worst performance: it requires about $F_{eff} = 8$ in order to infect almost every sites.

The behavior of the latency curves of Figure 3.10 for $\mathcal{G}(N, \rho)$ is similar to that of $\mathcal{B}(N, p_N)$, except that the minimum latency value is around 22 hops since the diameter of the former is greater than the latter.

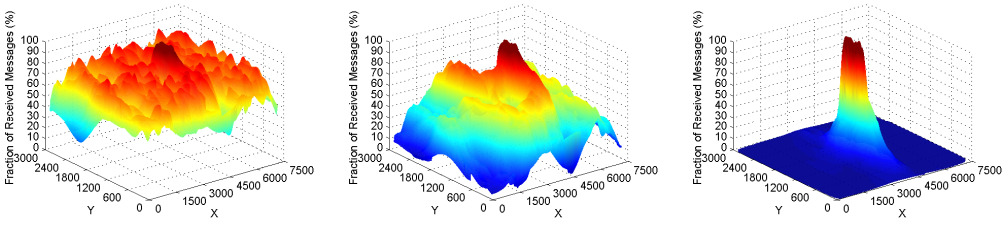
In order to thoroughly analyze the results, we conducted a series of experimentations by placing the source in the center of a rectangular plane of dimension 3000×7500 with 1000 sites uniformly distributed at random. The mean degree corresponding to the radius $\rho = 330$ is about 14. The results are presented in Figure 3.11. Several values of F_{eff} are chosen for the three gossip algorithms. The x-axis and y-axis represent the geographic position of the site in the graph, whereas the z-axis characterizes the percentage of messages received by every site. The greater the value towards the z-axis for a site, the greater the number of messages received by the site. The plane *z-axis=0* indicates the sites that never received any message.

The performance of *GossipFF* is shown in the first column of Figure 3.11. We can verify that this algorithm is the most efficient for infecting all sites (i.e., with F_{eff} merely equal to 6.97) contrarily to the other two algorithms (see Columns 2 and 3) that cannot broadcast every message from the source to the whole system Π until $F_{eff} = 9.77$. Even though these two algorithms complete the broadcast with almost the same performance, the evolution of their infections is quite different.

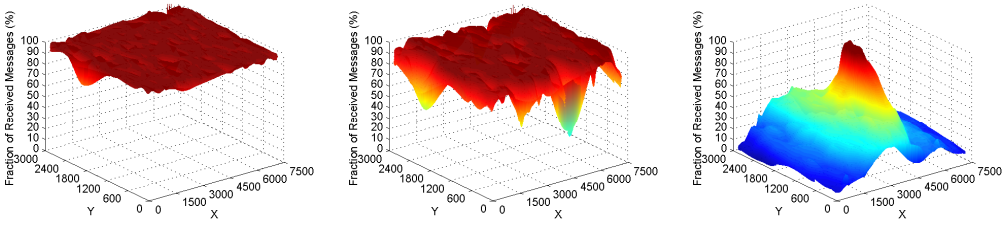
On the last column, we notice that *GossipPB* presents a peek for several values of F_{eff} in the graphs. Such a behavior implies that the infected sites are located around the source and message dissemination stops quickly. It can be explained



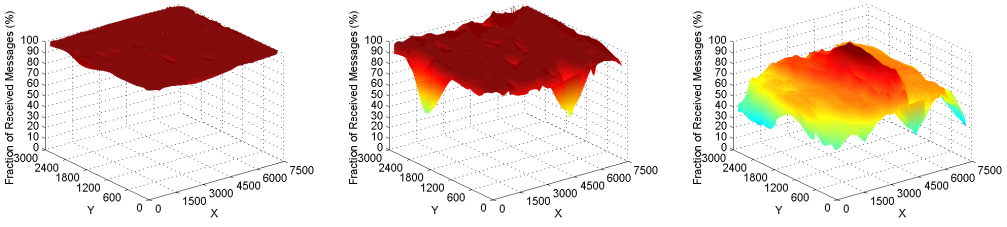
(a) *GossipFF* with F_{eff} =1 (b) *GossipPE* with F_{eff} =1 (c) *GossipPB* with F_{eff} =1



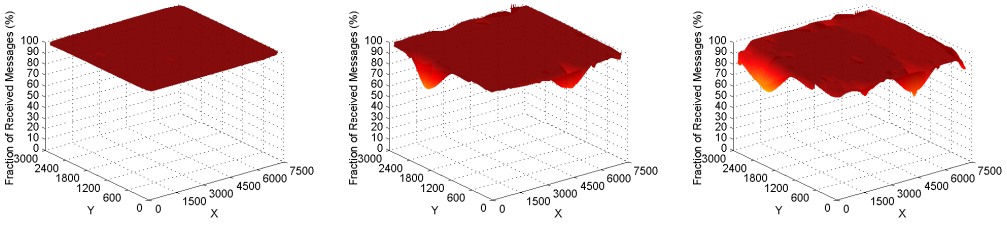
(d) *GossipFF* with F_{eff} =2 (e) *GossipPE* with F_{eff} =2 (f) *GossipPB* with F_{eff} =2



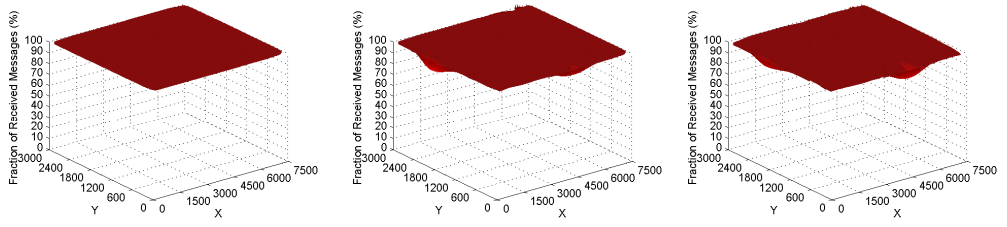
(g) *GossipFF* with F_{eff} =3.99 (h) *GossipPE* with F_{eff} =3.99 (i) *GossipPB* with F_{eff} =3.99



(j) *GossipFF* with F_{eff} =4.99 (k) *GossipPE* with F_{eff} =4.99 (l) *GossipPB* with F_{eff} =4.99



(m) *GossipFF* with F_{eff} =6.97 (n) *GossipPE* with F_{eff} =6.97 (o) *GossipPB* with F_{eff} =6.97



(p) *GossipFF* with F_{eff} =9.77 (q) *GossipPE* with F_{eff} =9.77 (r) *GossipPB* with F_{eff} =9.77

Figure 3.11: The message reception of every site over $\mathcal{G}(N, \rho)$

by the clustering effect entailed by the broadcast probability p_v: in this algorithm, sites stop retransmitting the message with probability $1 - p_v$. If this probability is high, the sites that do not relay the message give rise to a confinement around the source (i.e., the border of the peak). On the other hand, by increasing p_v, the clustering effect is reduced and the message can be received by every site. The study of such phenomenon is particularly important since, as explained in Section 2.3.2, $\mathcal{G}(N, \rho)$ has very high edge dependency which induces a higher number of possible peak borders which increase the risk of dissemination stopping.

Inversely, for *GossipPE* (see the second column), by slightly increasing the value F_{eff}, almost all sites receive every message from the source. Nevertheless, contrarily to *GossipFF*, there are always some sites which receive only a few messages. Such sites are located either on the border of the rectangular plane or in areas with small site density of the graph. As a matter of fact, *GossipPE* imposes random choice for each edge of every site no matter its degree. Therefore, sites having very few neighbors, with high probability, do not receive all messages. This explains *GossipPE*'s bad reliability (see Figure 3.9) even when almost all sites are infected (see Figure 3.8). For instance, when $F_{eff} = 5$, $\alpha \approx 100\%$ whereas there is zero reliability.

This study thus shows why *GossipFF* is particularly efficient over $\mathcal{G}(N, \rho)$. By forcing each site to retransmit some messages, it reduces the clustering effect more efficiently than the other algorithms. Furthermore, by obliging the sites with small degree (i.e., smaller than $fanout$) to broadcast the message to all its neighbors, it prevents the risk of dissemination stopping of small density areas.

3.6 Algorithms Comparison over $\mathcal{S}(N, m)$

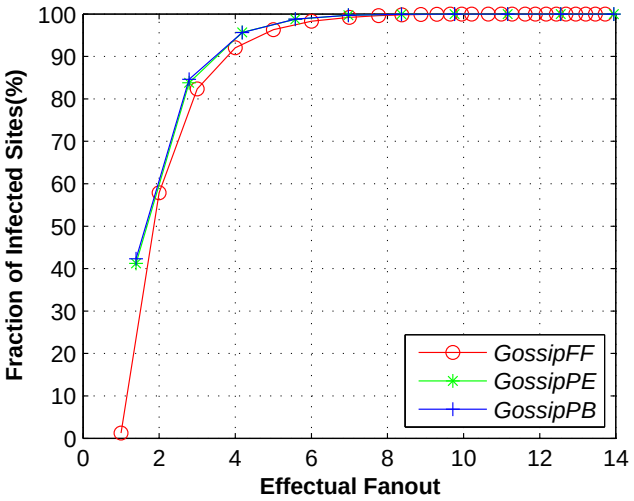


Figure 3.12: Infected Sites over $\mathcal{S}(N, m)$

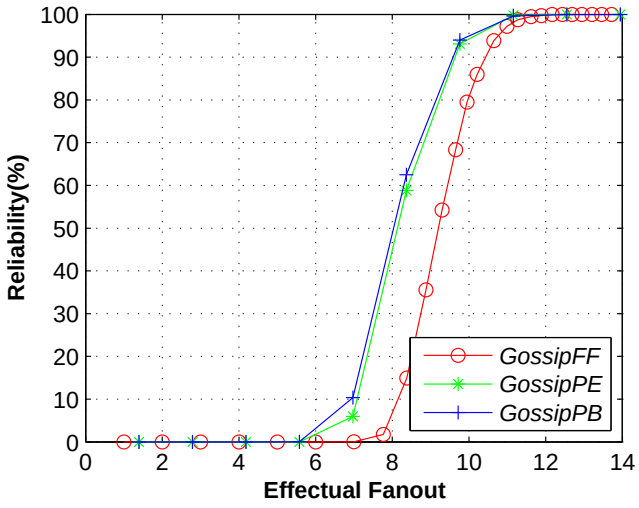


Figure 3.13: Reliability over $\mathcal{S}(N, m)$

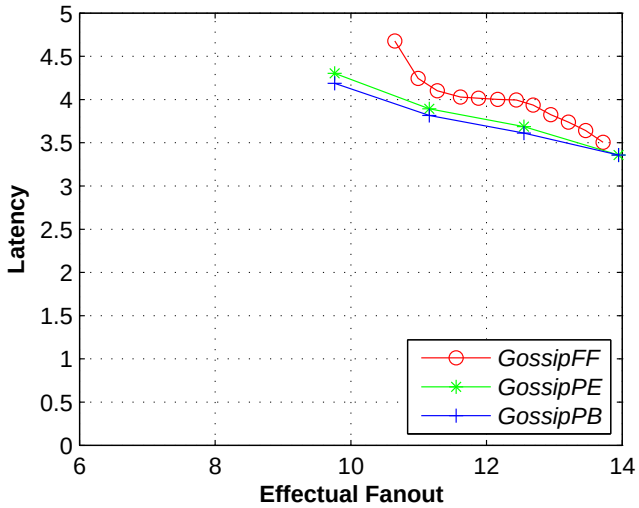


Figure 3.14: Latency (hops) over $\mathcal{S}(N, m)$

We discuss now the performance of the different gossip algorithms over $\mathcal{S}(N,m)$. Results are completely different from the other two random topologies. Similarly to $\mathcal{B}(N,p_N)$, the three algorithms present the same performance behavior in terms of the fraction of infected sites as shown in Figure 3.6. Nevertheless, contrarily to the reliability over $\mathcal{G}(N,\rho)$, *GossipFF* turns to be the worst choice (see Figure 3.6).

Such a performance behavior is a consequence of the degree distribution of the graph which has sites with higher degrees, the **hubs** (see Section 2.3.3). Such a group of sites only consists of less than 1/4 of total sites in the system, which is proved in Theorem 3. In addition, the **peripheries** whose degrees are much lower, connect to the *hubs* with high probability due to the **preferential attachment** in $\mathcal{S}(N,m)$. Thus, intuitively, the *hubs* can be infected before the *peripheries*, and to some extent, the former determine the final broadcast performance.

Theorem 3. *In $\mathcal{S}(N,m)$, we have $\mid \Pi_p \mid > 3 \mid \Pi_h \mid$. where $\mid \Pi_p \mid$ is the total number of periphery sites, and $\mid \Pi_h \mid$ is the total number of hubs.*

Proof. First of all, we compute the percentage of the peripheries in the system as follows.

$$
\begin{aligned}
\frac{\mid \Pi_p \mid}{N} &= \sum_{k=m}^{2m} \frac{2m(m+1)}{k(k+1)(k+2)} \\
&= 2m(m+1) \sum_{k=m}^{2m} \frac{1}{k} \left(\frac{1}{k+1} - \frac{1}{k+2} \right) \\
&= 2m(m+1) \sum_{k=m}^{2m} \left[\left(\frac{1}{k} - \frac{1}{k+1} \right) - \frac{1}{2} \left(\frac{1}{k} - \frac{1}{k+2} \right) \right] \\
&= 2m(m+1) \sum_{k=m}^{2m} \left(\frac{1}{k} - \frac{1}{k+1} \right) - m(m+1) \sum_{k=m}^{2m} \left(\frac{1}{k} - \frac{1}{k+2} \right) \\
&= 2m(m+1) \left(\frac{1}{m} - \frac{1}{2m+1} \right) - m(m+1) \left(\frac{1}{m} + \frac{1}{m+1} - \frac{1}{2m+1} - \frac{1}{2m+2} \right) \\
&= 2(m+1) - \frac{2m(m+1)}{2m+1} - (m+1) - m + \frac{m(m+1)}{2m+1} + \frac{m}{2} \\
&= 1 - \frac{m(m+1)}{2m+1} + \frac{m}{2} \\
&= 1 - \frac{m}{2(2m+1)} \\
&= 1 - \frac{1}{2} \frac{1}{2 + \frac{1}{m}} \\
&> 0.75.
\end{aligned}
$$

Since $\mid \Pi_h \mid + \mid \Pi_p \mid = N$, then

$$\frac{\mid \Pi_h \mid}{N} < 0.25.$$

Thereby, we obtain

$$| \Pi_p | > 3 | \Pi_h | .$$

□

In order to understand the dissemination power of the hubs, we have measured, for different F_{eff} values, the reliability of the hubs (i.e., the proportion of messages generated by the source that are received by all hubs in the system Π). The results are presented in Figures 3.6, 3.6, and 3.6. For each algorithm, the latter are compared with the reliability when we consider all sites of Π (denoted global reliability). This comparison shows that hubs are infected on priority no matter which algorithm is applied. Thereby, with a F_{eff} equal to 6, the reliability of the hubs is 100% for all the three algorithms whereas almost none of the messages is received by all sites (i.e., the global reliability is still zero).

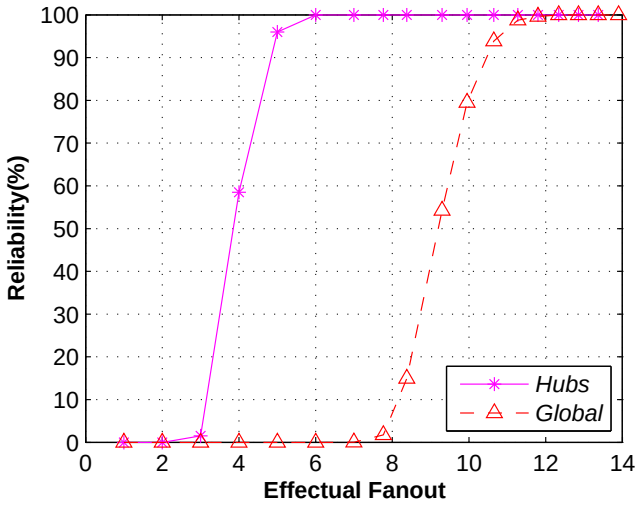


Figure 3.15: The reliability of all hubs and all sites by *GossipFF* over $\mathcal{S}(N, m)$

GossipFF presents the worst performance which can be explained by its poor exploitation of hubs. In fact, even if hubs degree is quite high, the algorithm limits their dissemination power to the value fixed by the $fanout$. On the other hand, it should be understood that a transmission of 10 messages by one site is more powerful than a transmission of 1 message by 10 sites. In the first case, all receivers are different, which ensures a better message dissemination with less message redundancy.

The fact that the dissemination potential of hubs is not fully exploited also explains the latency of Figure 3.6. As we can observe, *GossipPB* and *GossipPE* present the same latency but not *GossipFF* which has a higher latency when its reliability is near to 100%. As a matter of fact, in $\mathcal{S}(N, m)$, the hubs are the heart of the network: the peripheries have at least one hub in its neighborhood with high

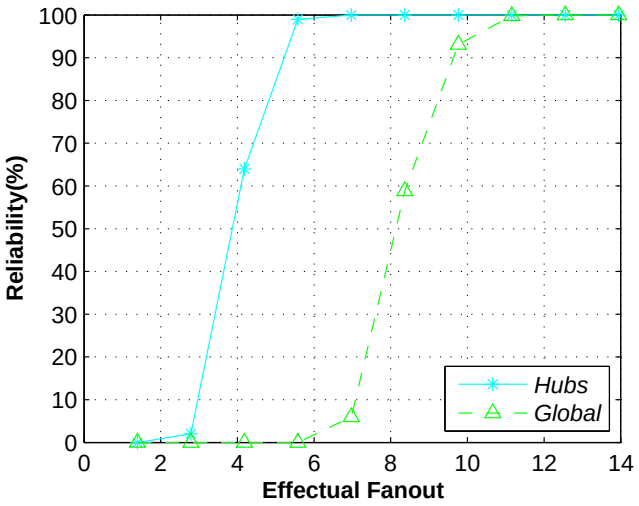


Figure 3.16: The reliability of all hubs and all sites by *GossipPE* over $\mathcal{S}(N, m)$

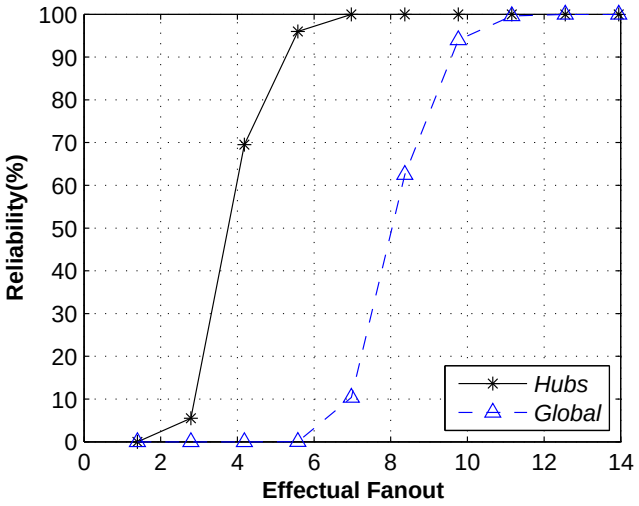


Figure 3.17: The reliability of all hubs and all sites by *GossipPB* over $\mathcal{S}(N, m)$

probability. By limiting the dissemination power of the hubs, *GossipFF* discards numerous short-cut paths.

3.7 Impact of the Topology on the Algorithms

In the previous sections, by applying the *effectual fanout* parameter, we have finely disclosed the reliability, fraction of infected sites, and latency for the three families of gossip algorithms in each of the three typical random topologies. On the other hand, the *effectual fanout* is also useful to uniform comparison of all reliability obtained from gossip algorithms over random networks with different degree distributions. In this way, we can discover the impact of the topology on the algorithms.

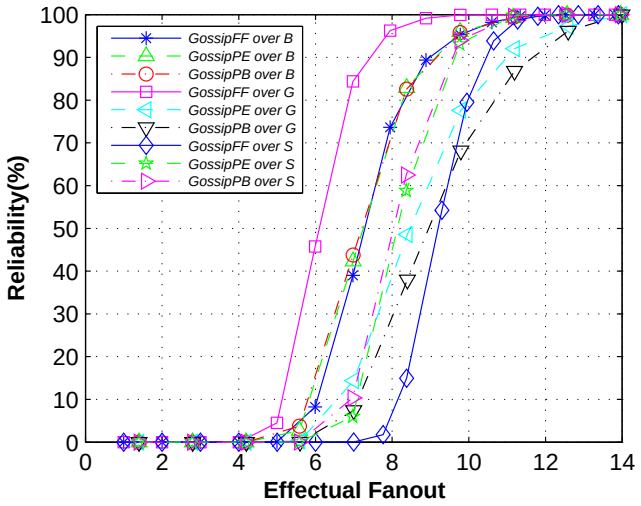


Figure 3.18: Topologies impact on algorithms

Since in our simulations we have considered topologies with the same mean degree ($\bar{V} \approx 14.0$), we can compare the reliability of the algorithms over the different topologies, which is shown in Figure 3.18 and summarized in Table 3.2. When the graph has low edge dependency and low degree variance as in ($\mathcal{B}(N, p_N)$), the three algorithms present the same behavior. When edge dependency (resp., degree variance) is introduced in the graph, but the degree variance (resp., dependency) does not change as in $\mathcal{G}(N, \rho)$ (resp., $\mathcal{S}(N, m)$), the performance of *GossipPB* and *GossipPE* (resp., *GossipFF*) decrease.

Such results confirm that the best algorithm choice for the reliability with the same message complexity depends on the properties of network topology. It should be pointed out that the performance of *GossipPE* is never worse than *GossipPB*, since in the percolation theory [63], the former can be modeled by bond percolation while the latter matches the site percolation. The percolation threshold of site percolation is always greater than that of bond percolation in any topology.

We have also measured the relative effectual fanout gain of the gossip algorithms

	Low Degree Variance	**High Degree Variance**
Low Edge Dependency	$\mathcal{B}(N, p_N)$: ***GossipFF***, ***GossipPE***, ***GossipPB***	$\mathcal{S}(N, m)$: ***GossipPE***, ***GossipPB***
High Edge Dependency	$\mathcal{G}(N, \rho)$: ***GossipFF***	$--$

Table 3.2: Algorithm choice

in comparison to the effectual fanout needed by the flooding algorithm (i.e., the maximum message complexity) when reliability reaches 80% and 99% and the fraction of infected sites is high (i.e., α is approximate to 100%). The results over the three random topologies are shown in Tables 3.3 and 3.4 respectively. We observe that over $\mathcal{G}(N, \rho)$ to reach $R = 99\%$ the gain of *GossipPB* and *GossipPE* is zero. Hence, they need almost the same message complexity as the pure flooding. On the other hand, to reach $R = 80\%$, $\mathcal{B}(N, p_N)$ exhibits the best gain for *GossipPB* and *GossipPE* amongst all random topologies. Furthermore, *GossipFF* over $\mathcal{G}(N, \rho)$ is the best combination for achieving the highest gain.

	$\mathcal{B}(N, p_N)$	$\mathcal{S}(N, m)$	$\mathcal{G}(N, \rho)$
GossipFF	14%	14%	**43%**
GossipPB	14%	**21%**	0%
GossipPE	14%	**21%**	0%

Table 3.3: Gain in terms of effectual fanout to reach $R = 99\%$

	$\mathcal{B}(N, p_N)$	$\mathcal{S}(N, m)$	$\mathcal{G}(N, \rho)$
GossipFF	40%	23%	**52%**
GossipPB	40%	**34%**	27%
GossipPE	40%	**34%**	31%

Table 3.4: Gain in terms of effectual fanout to reach $R = 80\%$

In conclusion, in order to reduce message complexity of gossip algorithms in contrast to the flooding algorithm, it is necessary to consider both the gossip algorithm and the topology.

3.8 Previous Work

Many works have already established some similar implementations and analysis, but solely for a given algorithm over one or two graphs in terms of some restricted

metrics. To the best of our knowledge, our study is the first time to fulfill a complete comparison.

In the following, we provide a brief presentation of these works which are summarized in Table 3.5.

	$\mathcal{B}(N, p_N)$	$\mathcal{S}(N, m)$	$\mathcal{G}(N, \rho)$
GossipFF	[11] [13] [20]	[16]	[12]
GossipPE			[29] [30]
GossipPB	[7] [25]	[15]	[4] [18]

Table 3.5: Previous studies of random gossip algorithms

The reliability of the information dissemination is studied in [86] by applying *GossipFF* over $\mathcal{B}(N, p_N)$. The authors assumed that the *fanout* of every site is always smaller than the number of its neighbors. The article mainly concludes that in a system with N sites to have the reliability equal to R, it requires to fix $fanout = -\ln\left(\frac{-\ln(R)}{N}\right)$. Results for *GossipFF* over $\mathcal{B}(N, p_N)$ which are based on simulations are also discussed in [47, 52]. However, the other gossip algorithms are neither studied nor compared in the articles. It is worth pointing out that since the $fanout$ is not linear to message complexity and the other two gossip algorithms take the probability as input parameter, the comparisons amongst them as a function of their probabilistic input become difficult due to the lack of one generic parameter like the effectual fanout.

The performance of *GossipPB* over $\mathcal{G}(N, \rho)$ is discussed and implemented in [15, 65] and it is theoretically analyzed over $\mathcal{B}(N, p_N)$ in [31]. The former is also studied in [145], concentrating on answering how to choose p_v in order to reach high reliability. Besides the discussion about the reliability by percolation property over $\mathcal{B}(N, p_N)$, the asymptotic expressions in [113] with respect to the average number of messages and the average time required to complete network coverage are derived as well, showing the benefits of the properly choice of p_v. However, compared to our work, their efforts are focused on the performance of one gossip algorithm over a certain random topology, which can be considered as one aspect of our discussion. Over the two random topologies $\mathcal{S}(N, m)$ and $\mathcal{B}(N, p_N)$, the latency of a modified version of *GossipPB* algorithm which lets every site send message to one neighbor with certain probability several times is theoretically studied by SIS (Susceptible-Infective-Susceptible) model in [58].

According to the heuristic results firstly shown in [61], the performance of the

three probabilistic gossip algorithms over $\mathcal{S}(N, m)$ is better than $\mathcal{B}(N, p_N)$. However, without effectual fanout, they cannot obtain the quantitative gains for all gossip algorithms in terms of message complexity to reach the same reliability.

GossipPE presents better performance than *GossipPB* over $\mathcal{G}(N, \rho)$, which is studied as a function of the system size (or the site degree) in [132]. Moreover, the three algorithms over $\mathcal{S}(N, m)$ are compared in the same way in [51]. Article [126] points out that the choice between *GossipPE* and *GossipPB* depends on the different application constraints over $\mathcal{G}(N, \rho)$. Compared to their work, we exploit the effectual fanout, which is linear to message complexity, and thus, the difference of all metrics can be fairly compared.

3.9 Conclusion

Contrarily to the *fanout* in *GossipFF*, the effectual fanout takes into account the degree distribution of the topology, which makes measures and performance evaluation fairly comparable for all the three topologies. Therefore, effectual fanout is a useful parameter to uniform comparison of gossip algorithms over random networks with different degree distributions. It characterizes the mean dissemination power of infected sites.

By exploiting the *effectual fanout* parameter, we have fulfilled the comparison of the reliability, fraction of infected sites, message complexity, and latency for the three families of gossip algorithms over three typical random topologies. We have shown that in terms of reliability, *GossipFF* is the best algorithm on $\mathcal{G}(N, \rho)$ but the worst on $\mathcal{S}(N, m)$ and that the three algorithms have the same performance on $\mathcal{B}(N, p_N)$.

The results obtained in this study help in the decision of choosing the most suitable combination between a gossip algorithm and a random topology that satisfies the requirements of an application. Furthermore, since the effectual fanout simplifies the theoretical comparison of different gossip algorithms on a topology, it can thus be used to theoretically analyze some of the algorithms' behaviors as, for instance, the clustering effect of *GossipPB* on $\mathcal{G}(N, \rho)$ or the fact that hubs are infected before peripheries on $\mathcal{S}(N, m)$, regardless of the gossip algorithm.

Chapter 4

Efficient Dissemination Algorithm for Scale-Free Topologies

Contents

4.1 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.

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 chapter a dissemination algorithm suitable for scale-free topologies generated by Barabási-Albert model [6]. This model makes use of

preferential attachment which is a basic idea used by many other models [66, 68] 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 this chapter 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 that entails the broadcast storm problem [111], as confirmed by some performance evaluation results on top of OMNET++ [1].

The road map of this chapter is organized as follows. Section 4.2 gives an overview of scale-free random networks. Section 4.3 introduces our algorithm and presents an example of its execution, while simulation results on OMNET++ are shown in Section 4.4. Finally, Section 4.5 concludes this work.

4.2 Scale-Free Random Topology

In scale-free networks, 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**. Moreover, in social networks, new participants are more likely to make friends with people who have a great number of neighbors than a person with very few friends. We call it *preferential attachment* behavior, which can be imitated by the network $\mathcal{S}(N, m)$ generated by Barabási-Albert model [6] (see Section 2.3.3). In $\mathcal{S}(N, m)$, **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 have deduced

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

in Theorem 3 (see Section 3.6 in Page 57).

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

Lemma 4. *Over* $\mathcal{S}\left(N, m\right)$, $P_{connect}\left(hub \mid s_i\right) = P_{connect}\left(per \mid s_i\right) = 0.5$

Proof. $\mathcal{S}\left(N, m\right)$ generated by Barabási-Albert model is an uncorrelated network described in [149]. In such a network [117], the probability that a site s_i connects to another site s_j can be written as:

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

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

$$P_{connect}\left(hub \mid s_i\right) = \sum_{k=2m+1}^{N-1} \frac{P\left(k\right) k}{2m} \tag{4.1}$$

$$= \sum_{k=2m+1}^{N-1} \frac{k}{2m} \cdot \frac{2m(m+1)}{k(k+1)(k+2)} \tag{4.2}$$

$$= \sum_{k=2m+1}^{N-1} \frac{(m+1)}{(k+1)(k+2)} \tag{4.3}$$

$$=(m+1) \cdot \sum_{k=2m+1}^{N-1} \left(\frac{1}{k+1} - \frac{(1)}{k+2}\right) \tag{4.4}$$

$$=0.5, \tag{4.5}$$

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

$$P_{connect}\left(per \mid s_i\right) = \sum_{k=m}^{2m} \frac{P\left(k\right) k}{2m} = 0.5. \tag{4.6}$$

□

4.3 Our Algorithm

The main idea of our gossip algorithm is that only the sites in Π_h (i.e., hubs whose degree is greater than 2 times the minimum degree of $\mathcal{S}\left(N, m\right)$), 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 3.6), 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 it 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. An example will explain these phases just after the study of our gossip algorithm.

4.3.1 Efficient Gossip Algorithm

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 /* m̂in = 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 m̂in > ⟨msg⟩.min then
56         m̂in = ⟨msg⟩.min
57     else
58         ⟨msg⟩.min = m̂in
59     return m̂in
```

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 (i.e., the sites in Λ_i) 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 51 and 52).

The relative number of sites that need forwarders in their neighborhood is very small, which is inferred from Theorem 5. 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 5. *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

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

As deduced in Lemma 4 that $P_{connect}(per \mid s_i) = 0.5$, the result is obtained. □

After the first phase, the second phase of the algorithm, Hub-Based Gossip denoted *GossipHB*, starts up message dissemination. If the site is either a *forwarder* or a hub, it should retransmit the messages it receives to all its neighbors (see line 15). 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 its local approximation $\hat{min}$ variable, the receiver updates this variable (lines 15 and 15).

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 2.3.3).

Theorem 6 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 6. *Over $\mathcal{S}(N, m)$, message complexity of GossipHB is m.*

Proof. Since $\hat{min}$ converges to m, according to Algorithm 7, then $P(k) = \frac{2m(m+1)}{k(k+1)(k+2)}$. Besides, Theorem 5 shows that the fraction of *forwarders* is negligible. Applying the theory in [72], we thus calculate the message complexity as

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

□

4.3.2 Example

In this section, we describe an execution of our algorithm. We consider a scale-free graph with 17 sites generated by Barabási-Albert model, where $m = 2$ and $m_0 = 3$ (see Figure 4.1(a)). Thus, the mean degree is about 4. Sites s_1, s_2, and s_3 are *hubs* whose degree is strictly higher than 4, while the others are *peripheries* (see Section 2.3.3). However, s_1 is not directly connected to the two other hubs. After the execution of the first phase in the algorithm, s_4 will become a *forwarder* to relay messages, and therefore, all hubs will be connected. These four sites compose the heart of the network.

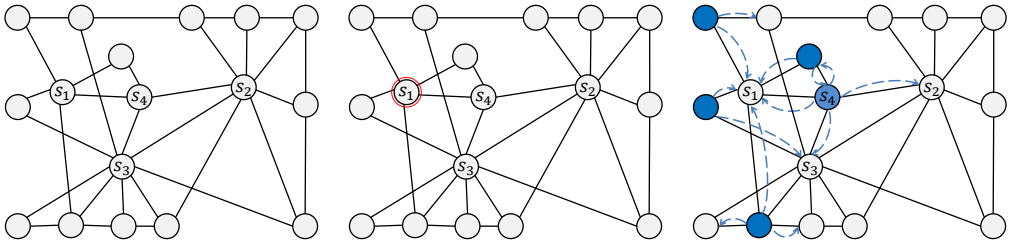


Figure 4.1: Hub Connection

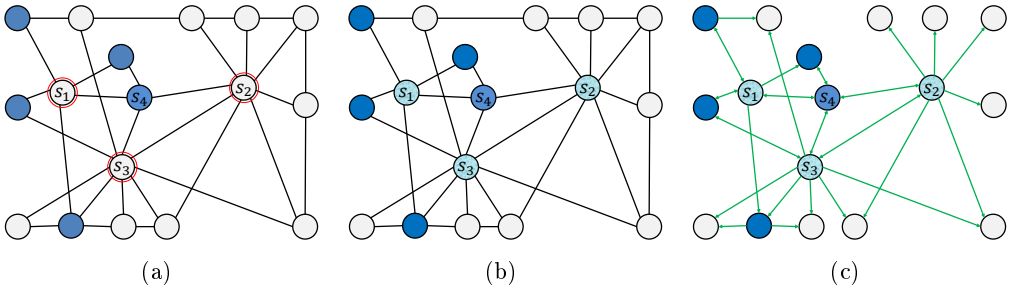


Figure 4.2: Hub Detection and Message Dissemination

Figure 4.1 shows the first phase of our algorithm, where hub connection algorithm is executed before information dissemination. It ensures, therefore, that all hubs are connected, and every periphery has at least one hub in its one-hop neighborhood. Knowing that $min_1 = 2$ and $max_1 = 4$, s_1 suspects to be isolated from hubs, due to the algorithm's condition (see line 45). Then, all its neighbors including s_4 will set their $forwarder$ variable to true. In other words, they will take charge of message retransmission in the second phase (see Figure 4.1(c)).

When the message dissemination starts, every site runs hub-based gossip algorithm as shown in Figure 4.2. Trivially, s_1 and s_3 are detected to be hubs by themselves, since their degree is strictly higher than the twice of the estimated minimum degree of the graph (i.e., it is greater than the mean degree $2 \times m = 4$). In particular, although s_2 locally assumes that $min_2 = 3$ at the beginning, it also turns to be a hub due to its degree, which is larger than 6. Upon reception of a msg from other hubs, s_2 can accurately update its knowledge on the minimum degree of the graph, when the function `Approxm`($\langle msg \rangle$) is called with the piggybacked information (see line 55). Its character of *hub* can thus be definitely updated. Then, both hubs and forwarders retransmit their newly received message (see line 51), whereas almost half of the message complexity is reduced compared to the pure flooding. The *atomicity* ($R = 100\%$) is ensured as well (see Figure 4.2(c)). We should point out that such a gain can only be obtained, when the fraction of sites that require the hub connection in the first phase is very small, and thus very few non-hub forwarders are designated to rebroadcast messages. Furthermore, both our theoretical analysis and simulation confirm that such a requirement is negligible.

4.4 Performance Evaluation

In this section, we present and discuss some evaluation performance results concerning the five algorithms described in both Sections 2.5 and 4.3: *GossipFF*, *GossipPE*, *GossipPB*, *GossipDD*, 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. Since for $m = 1$ the graph becomes a tree, we consider $m > 1$. 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.

The following metrics are used for our performance evaluation:

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

$$M = \frac{\Omega}{N-1}, \tag{4.7}$$

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 [86].

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.

Figures 4.3 and 4.4 aim at respectively studying the reliability and latency of gossip algorithms that require a pre-specified parameter, i.e., our algorithm (*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.

4.4.1 Impact of forwarders' requirement

In Section 4.3.1, we have theoretically studied the fraction of sites that require its neighbors to become forwarders. The simulation results in Table 4.1 confirm our study and that such a fraction is negligible.

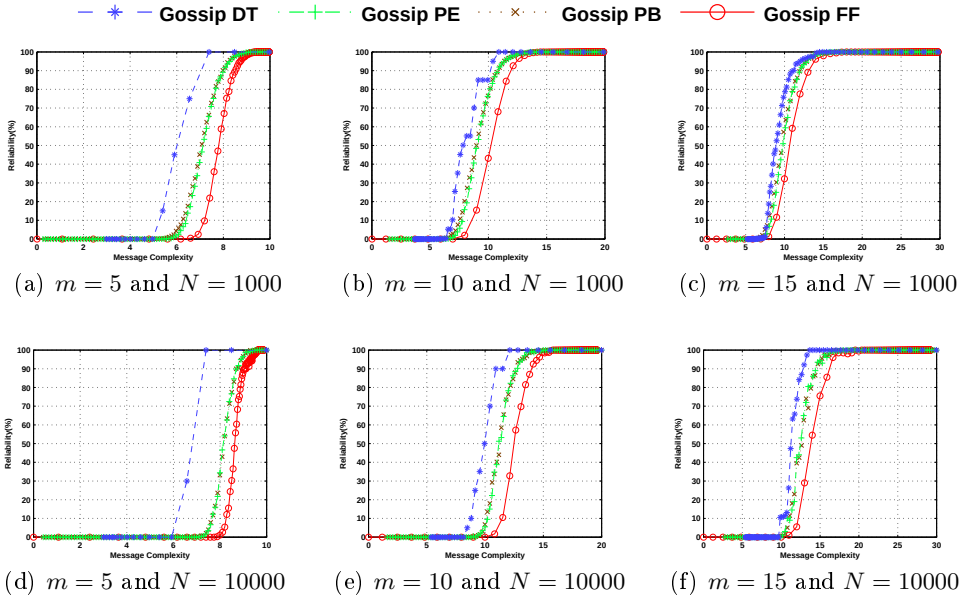
N \ m	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1000	0.0167	0.0093	0.0037	0.0017	0.0007	0.0003	0.0002	10^{-4}	10^{-4}	10^{-4}	10^{-4}	0	0	0
10000	0.0172	0.0089	0.0040	0.0020	0.0009	0.0004	0.0002	10^{-4}	4×10^{-5}	10^{-5}	10^{-5}	0	0	0

Table 4.1: Fraction of sites that require forwarders in neighborhood

4.4.2 Reliability of pre-specified parameter algorithms

Figure 4.3 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 [63].

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



(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 4.3: Reliability comparison over $\mathcal{S}(N, m)$.

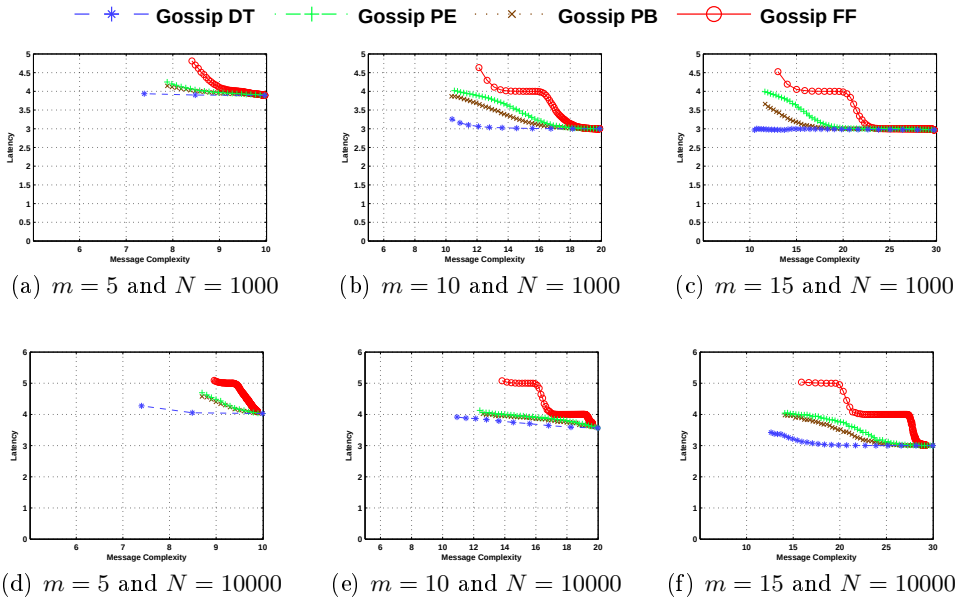
for message retransmission while in the other probabilistic algorithms, all peripheries relay messages as well.

4.4.3 Latency of pre-specified parameter algorithms

Figure 4.4 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 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 our algorithm (*GossipHB*) overcomes this restriction 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.

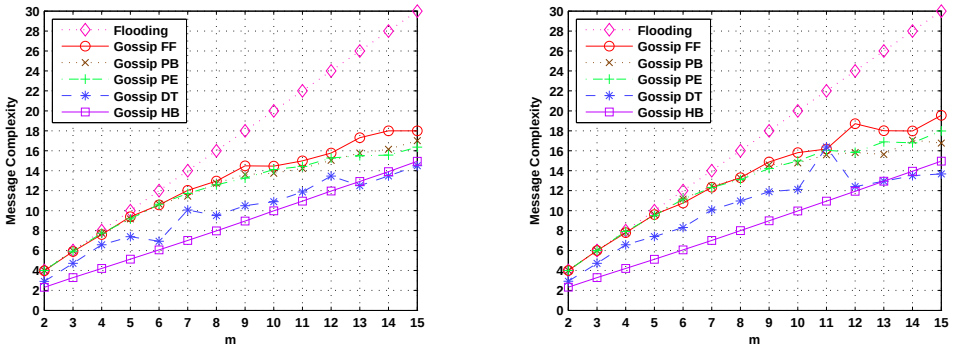


(a) $m = 5$ and $N = 1000$ (b) $m = 10$ and $N = 1000$ (c) $m = 15$ and $N = 1000$

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

Figure 4.4: Latency (hops) comparison over $\mathcal{S}(N, m)$.

4.4.4 Comparison of the best algorithms' performance

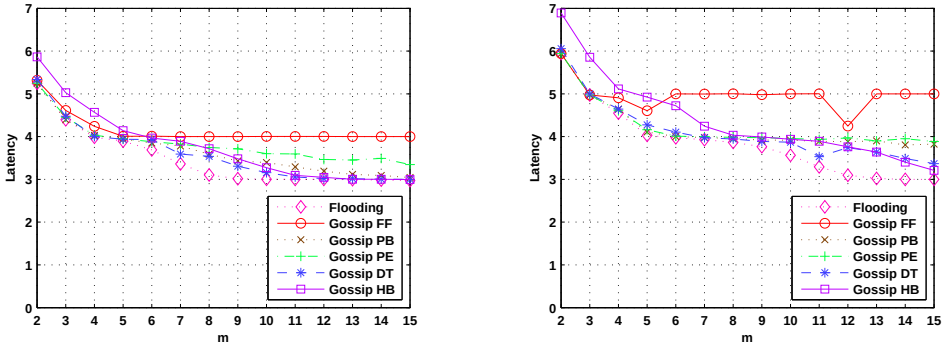


(a) $\mathcal{S}(N, m)$ with $N = 1000$ (b) $\mathcal{S}(N, m)$ with $N = 10000$

Figure 4.5: message complexity comparison with reliability over 99.9%.

Figure 4.5 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) $\mathcal{S}(N, m)$ with $N = 1000$ (b) $\mathcal{S}(N, m)$ with $N = 10000$

Figure 4.6: Latency (hops) comparison with reliability over 99.9%.

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 5), our algorithm (*GossipHB*) presents a linear relation between message complexity and m, which confirms Theorem 6. 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, as well as the latency. 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, our algorithm (*GossipHB*) algorithm ensures that hubs are connected by paths composed by very few periphery sites, (i.e., the forwarders), whereas *GossipDT* does not for topologies whose mean degree is small. This difference explains why below a given threshold value of m, our algorithm (*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 our algorithm (*GossipHB*), all hubs relay messages while in *GossipDT* only a subset of hubs, i.e., those that have degree greater than d. Notice that the

value of this m threshold increases when N increases ($m = 13$ and $m = 14$ in Figures 4.5(a) and 4.5(b) respectively).

In Figure 4.6, for m within 2 to 15, we show some results related to latency, aiming $R > 99.9\%$. Latency decreases as 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 4.5), 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 [72].

4.5 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, our algorithm (*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 our algorithm (*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 our algorithm (*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, our algorithm (*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 in [142].

Chapter 5

Conclusion

Contents

5.1 Introduction

A brief conclusion of our work is drawn in this chapter, while we also discuss some future perspectives.

5.2 Our Study and Contributions

Our work focuses on studying the behavior and performance of gossip algorithms over random graphs. To this end, we considered three families of gossip algorithms (*GossipFF*, *GossipPE*, and *GossipPB*) over three different random topologies : $\mathcal{B}(N, p_N)$, $\mathcal{G}(N, \rho)$, and $\mathcal{S}(N, m)$. After such study, we proposed a new gossip algorithm for scale-free topologies, which does not need any pre-fixed parameter to tailor the algorithm to the topology.

Contribution 1: In order to uniformly compare all probabilistic gossip algorithms over the random graphs, we introduced a new parameter, denoted **Effectual Fanout** [72]. It has two-fold exploitations. We can beforehand estimate the message complexity that is linearly correlated to it. For a fixed topology and gossip algorithm, the effectual fanout characterizes the mean dissemination power of the sites that have received the message from the source site. On the other hand, it is used to theoretically explain the influence of topology over the performance of gossip algorithms. All results become comparable. The trade-off amongst reliability, message complexity, and latency is clearly addressed by effectual fanout. The best choice of the gossip algorithms depends on the properties of their underlying topologies. Evaluation on top of OMNET++ has shown that when the graph has

low edge dependency and low degree variance as in $(\mathcal{B}(N, p_N))$, the three algorithms present the same behavior; when edge dependency (resp., degree variance) is introduced in the graph, but the degree variance (resp., dependency) does not change as in $\mathcal{G}(N, \rho)$ (resp., $\mathcal{S}(N, m)$), the performance of *GossipPB* and *GossipPE* (resp., *GossipFF*) decrease.

Contribution 2: After having gained a sound knowledge about the impact of different characteristics of random graphs on the gossip algorithms' performance, we have proposed a dissemination algorithm for scale-free topologies [73], which exploits the dissemination power of *hubs*, aiming 100% reliability. Since globally setting parameter to an optimum value at initialization phase is not suitable for gossip algorithms as we have seen in probabilistic gossip protocols, every site automatically detects whether itself is a hub merely by the knowledge from one-hop neighbors. In particular, a simple hub connection phase in our algorithm eventually guarantees that all hubs are well connected and every site has at least one hub or forwarder in its one-hop neighborhood. Messages passed by hubs and forwarders can reach every site throughout networks in the end of information dissemination. Moreover, our algorithm substantially reduces message redundancy, which is even more outstanding than the other probabilistic gossip algorithms studied in the previous contribution. Latency is reduced as well, since in contrast with other probabilistic algorithms that do not fully exploit hubs' dissemination power, many more short-cuts are created by our approach.

5.3 Perspectives

Our work opens the following perspectives.

5.3.1 Mathematical model and fair analyses of other probabilistic gossip algorithms over other random graphs

We provide a mathematical model that establishes a relation between *reliability* and *effectual fanout* for the basic probabilistic gossip algorithms over the random networks. According to such a relation, we can fine-tune input parameters in the gossip protocols to obtain a desirable reliability before information dissemination, while traditionally we resort to heuristic simulations to choose right parameters.

The effectual fanout typically requires input arguments of gossip algorithms and degree distribution of random graphs. Thus, there is no restriction for using it on other probabilistic gossip algorithms, such as the reliable broadcast protocol proposed in [53], or for other random graphs like a scale-free topology generated by [136].

Therefore, thanks to effectual fanout, we are able to evaluate probabilistic gossip algorithms over some random graphs and the impact of a given topology on the gossip algorithms. However, the best algorithm choice for a graph with both high degree variance and high edge dependency has not been analyzed yet. We could

then observe which property has a dominant effect on the performance of gossip algorithms.

Finally, we could apply the effectual fanout approach to learn about gossip algorithms' performance over mobile networks. Considering mobility model, our effectual fanout could dynamically adapt mobile context.

5.3.2 Real dissemination applications and multisource dissemination problem

Our new algorithm presented in Chapter 4 has been evaluated and compared with other probabilistic gossip algorithms in the scale-free topologies generated by Barabási-Albert model [6]. It will be executed in some other scale-free networks. The difference of their performance in real social networks like Facebook is an ongoing work, accomplished in very near future.

Even though our research is about one source dissemination problem, some results can be extended to multisource broadcast applications. In reality, the classical probabilistic gossip algorithms can also be implemented in many peer-to-peer publish-subscribe systems or sensor network code update protocols that disseminate multiple streams (e.g., RSS subscribers fetch data from multiple streams, or queries come from different sensor sites). Yet, most of gossip mechanisms handle each stream independently. Messages are sent to the same destinations from different sources when sharing common network channels. Trivially, the overhead of the network now is the sum of all the streams, as every site periodically sends gossip messages that contain constituent messages from several streams. In this context, instead of being taken into account for only one source, the reliability of dissemination will be evaluated for all streaming messages from distinct producers. One of the most straightforward questions is how every site decides whether to include a stream in its combined gossip message within the limit of channel capacity when being aware of underlying topologies. In particular, such information dissemination can take place in social networks, where dissemination power of each site is quite different one from another. Besides our evaluation metrics, fairness and maximum utilization of such a power should also be considered for multisource messages.

Furthermore, if there is message loss or wrong data due to unreliable noisy transmission channels, my previous contributions [70, 71] on source-channel conjoint coding theory can be exploited for information recovery. It will be another theme on multimedia processing in wireless network communications.

Publication

- Hu, R. and Sopena, J. and Arantes, L. and Sens, P. and Demeure, I., Efficient dissemination algorithm for scale-free topologies, 42th International Conference on Parallel Processing (ICPP'13), Lyon, France, (IEEE Computer Society) (2013)
- Hu, R. and Kieffer, M. and Duhamel, P., Protocol-Assisted Channel Decoding, 2012, IEEE Signal Processing Lettres
- Hu R. and Sopena J. and Arantes L. and Sens P. and Demeure I., Comparaisons équitables des algorithmes de gossip sur les topologies aléatoires à grande-échelle, 9ème Conférence Française sur les Systèmes d'Exploitation (CFSE'13), Chapitre français de l'ACM-SIGOPS, GDR ARP, Grenoble, France (2013)
- Hu, R. and Sopena, J. and Arantes, L. and Sens, P. and Demeure, I, Fair comparison of gossip algorithms over large-scale random topologies, 2012, IEEE Proc. Int. Symposium on Reliable Distributed Systems
- Hu, R. and Huang, X. and Kieffer, M. and Derrien, O. and Duhamel, P., Robust critical data recovery for MPEG-4 AAC encoded bitstreams, 2010, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing

Chapter 6

French Version

Contents

6.1 Introduction

La dissémination d'informations (broadcast), où un site tente de diffuser des messages à tous les autres sites dans le réseau, est essentielle pour beaucoup d'applications réparties.

Les *protocoles inondation* (*flooding*) sont une solution simple mais inefficace pour disséminer des informations. Lors de la première réception d'un nouveau message, tous les sites le retransmettent à tous leurs voisins [82, 99]. Pour réduire le nombre de messages, les algorithmes probabilistes de gossip sont apparus comme une solution efficace pour implémenter les protocoles de diffusion de façon extrêmement fiable et scalable [54, 144]. Ils sont ainsi couramment utilisés dans les réseaux couvrants (overlay) [47, 52, 86], les réseaux sans fil ad-hoc, ou encore les réseaux de capteurs [15, 61, 65, 132, 145].

Nos recherches se focalisent sur l'amélioration des algorithmes de gossip en profitant des propriétés des topologies sous-jacentes pour la dissémination d'informations (par exemple, la distribution des degrés, la dépendance d'arêtes, etc.). L'objectif est d'adapter les algorithmes de gossip aux topologies, en tenant compte des besoins des applications en termes de fiabilité.

Nous avons fait deux contributions :

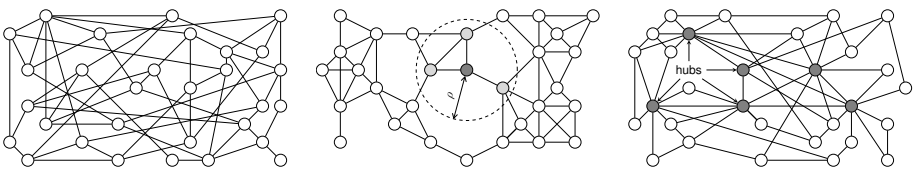
1. Nous avons premièrement proposé des études théoriques et comparé expérimentalement les trois classes d'algorithmes de gossip les plus répandus [61]

sur trois familles de graphes aléatoires représentant les topologies des réseaux à grande échelle. Plus précisément, nous avons étudié : (1) le gossip avec fanout fixé (*GossipFF*) [86], (2) le gossip avec choix d'arêtes probabiliste (*GossipPE*) [132] et (3) le gossip basé sur une diffusion probabiliste (*GossipPB*) [65]. Nous avons considéré les topologies suivantes : (1) le graphe de Bernoulli (ou Erdős-Rényi) ($\mathcal{B}(N, p_N)$), (2) le graphe géométrique aléatoire ($\mathcal{G}(N, \rho)$) et (3) le graphe scale-free ($\mathcal{S}(N, m)$). Ces trois familles des graphes modélisent respectivement un système pair à pair [86], un réseau de capteurs [65] et un réseau ad-hoc [61]. Cette étude permet de choisir le meilleur algorithme en fonction de la topologie et une fiabilité souhaitée. De plus, pour que les analyses théoriques et les comparaisons de performance soient effectuées équitablement, nous avons introduit un nouveau paramètre : le *fanout effectif*. Le fanout effectif caractérise la puissance moyenne de dissémination des sites infectés dans le système. Ce paramètre simplifie non seulement l'analyse des résultats expérimentaux, mais aussi l'étude théorique des algorithmes. Ainsi, je travaille actuellement sur une série de preuves reprenant les résultats expérimentaux.

2. Nous avons proposé un nouvel algorithme de gossip réparti : le gossip basé sur les hubs (*GossipHB*) pour un graphe scale-free $\mathcal{S}(N, m)$. *GossipHB* exploite les sites *hubs* ayant beaucoup plus de voisins que les autres. Il arrive à réduire la message complexity par rapport aux trois algorithmes de gossip avec la même fiabilité pour la diffusion des messages.

Ce chapitre est organisé comme suit. La section 6.2 décrit les trois graphes aléatoires, lorsque les trois familles des algorithmes de gossip sont présentées à la section 6.3. La section introduit les métriques que nous avons utilisées pour l'évaluation de performances. L'état de l'art est relaté dans la section . La section 6.6 introduit le fanout effectif. Finalement, la section 6.7 conclut nos recherches et dessine le travail du futur.

6.2 Topologies étudiées



(a) Le graphe de Bernoulli (b) Le graphe géométrique aléatoire (c) Le graphe scale-free

Figure 6.1: Exemples des topologies aléatoires avec 30 sites et de degré moyen=4

Dans la suite, $\mid l \mid$ dénote le cardinal de l'ensemble l.

On considère que le système de dissémination Π comprend N sites $\{s_1, s_2, \cdots, s_N\}$. L'ensemble des voisins de s_i est noté Λ_i et $V_i = \mid \Lambda_i \mid$ indique le degré de s_i. $P(k)$ représente la distribution des degrés d'un site ayant k voisins (c'est-à-dire, la fraction de sites avec un degré k) dans le graphe et $\bar{V}$ est le degré moyen du graphe. Donc, $\bar{V} = \sum_{k=0}^{N-1} P(k) \cdot k$. Aucune perte de message n'est prise en compte.

Trois topologies aléatoires sont étudiées dans nos recherches : le graphe de Bernoulli (ou Erdős-Rényi) $\mathcal{B}(N, p_N)$ [45], le graphe géométrique aléatoire $\mathcal{G}(N, \rho)$ [119] et le graphe scale-free $\mathcal{S}(N, m)$ [6] (voir respectivement les figures 6.1(a), 6.1(b) et 6.1(c)).

Le graphe de Bernoulli (ou Erdős-Rényi) $\mathcal{B}(N, p_N)$ est un graphe aléatoire bidirectionnel, construit en créant indépendamment une arête entre deux sites du système avec une probabilité p_N.

Le graphe géométrique aléatoire $\mathcal{G}(N, \rho)$ est un graphe aléatoire bidirectionnel dans une région bornée. Dans nos études c'est une région rectangulaire de longueur a et de largeur b. $\mathcal{G}(N, \rho)$ est généré en plaçant les sites uniformément, aléatoirement et indépendamment dans la région.

Le graphe scale-free est un graphe aléatoire bidirectionnel dont la distribution des degrés suit une loi de puissance. Un graphe scale-free $\mathcal{S}(N, m)$ peut être généré par le modèle Barabási-Albert [6]. La génération du réseau commence à partir d'une clique de m_0 sites ($m_0 \ll N$). Puis chaque nouveau site crée $m (\leqslant m_0)$ arêtes connectées à m différents sites déjà présents dans le graphe. La probabilité p qu'un nouveau site soit connecté à un site existant est proportionnelle au degré de ce dernier. Ceci est appelé *l'attachement préférentiel*. Ce processus assure que le graphe est connecté avec une distribution des degrés suivant la loi de puissance. Dans ce graphe, il y a **hubs** et **sites périphériques** dont les degrés respectifs sont supérieurs à $2m$ et entre m et $2m$. On peut déduire $\mid \Pi_p \mid > 3 \mid \Pi_h \mid$, où Π_h est l'ensemble de hubs et Π_p est l'ensemble de sites périphériques.

La dépendance d'arêtes (ou le coefficient de clustering) d'un graphe aléatoire donné, pour trois différents sites s_i,s_j,s_k, est définie par la probabilité conditionnelle sachant l'existence des arêtes $s_i \sim s_k$ et $s_j \sim s_k$, qu'une arête $s_i \sim s_j$ existe (c'est-à-dire, $P(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k)$).

[8] a prouvé que $\mathcal{B}(N, p_N)$ possède une très faible dépendance d'arêtes, c'est-à-dire, l'existence d'une arête dans $\mathcal{B}(N, p_N)$ ne depend pas des autres. Ainsi, $P(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k) = P(s_i \sim s_j) = p_N$. En revanche, $\mathcal{G}(N, \rho)$ présente une très forte dépendance d'arêtes et l'existence d'arêtes corrélées : $P(s_i \sim s_j | s_i \sim s_k, s_j \sim s_k) = 0.5865$, si l'on néglige l'effet de la borne. Cette valeur est plus importante que la probabilité p_N dans $\mathcal{B}(N, p_N)$.

[43] a montré que la dépendance d'arêtes sur $\mathcal{S}(N, m)$ est très faible, c'est-à-dire, du même ordre que $\mathcal{B}(N, p_N)$. Toutefois, par rapport à $\mathcal{B}(N, p_N)$ et $\mathcal{G}(N, \rho)$, sa variance des degrés est très importante.

De plus, par rapport aux deux autres graphes, $\mathcal{S}(N, m)$ possède le diamètre le plus petit à cause des *hubs* qui créent les raccourcis [17]. Le diamètre de $\mathcal{B}(N, p_N)$

est petit avec les cliques rares, tandis que dans $\mathcal{G}(N, \rho)$, le diamètre est le plus grand.

6.3 Algorithmes de gossip

La dissémination d'informations dans un réseau à grande échelle s'appuie sur l'algorithme de gossip générique illustré par l'algorithme 8. Initialement, la source envoie son message à *tous* ses voisins (lignes 61 et 62). Un site retransmet le message reçu en appelant la procédure Gossip() (ligne 67) à condition qu'il ne l'ait jamais reçu. Sinon, le message est abandonné. Les sites qui reçoivent le message au moins une fois sont appelés **sites infectés**, tandis que ceux qui ne reçoivent aucun message sont les **sites isolés**.

Algorithm 8: Algorithme de gossip générique

```
60 Broadcast (⟨msg⟩)
61     foreach s_j ∈ Λ_i do
62         Envoyer (⟨msg⟩, s_j) ;

63 Recevoir (⟨msg⟩)
64     if msg ∉ msgHistory then
65         Livrer(⟨msg⟩) ;
66         msgHistory ← msgHistory ∪ {⟨msg⟩} ;
67         Gossip(⟨msg⟩,paramètres) ;
```

Il y a trois grandes classes d'algorithmes de gossip pour implémenter la procédure Gossip() : (1) le gossip avec fanout fixé (*GossipFF*), (2) le gossip avec choix d'arêtes probabiliste (*GossipPE*) et (3) le gossip basé sur une diffusion probabiliste (*GossipPB*). Nous considérons également l'algorithme *GossipDD* qui est une amélioration de *GossipPB* dans le graphe $\mathcal{S}(N, m)$.

```
68 /* fanout : nombre de voisins sélectionnés */
69 GossipFF (⟨msg⟩,fanout)
70     if fanout ⩾ V_i then
71         toSend ← Λ_i
72     else
73         toSend ← ∅
74         for f = 1 to fanout do
75             aléatoirement selectionner s_j ∈ Λ_i/toSend
76             toSend ← toSend ⋃ s_j
77     foreach s_j ∈ toSend do
78         Envoyer (⟨msg⟩, s_j)
```

Algorithm 9: Gossip avec fanout fixé (dans s_i)

Dans *GossipFF* (Algorithme 9), le site s_i envoie le message msg à un nombre fixé de sites, noté $fanout$, qui sont aléatoirement sélectionnés dans Λ_i (lignes 74-76). Notons que si $fanout \geqslant V_i$, s_i transmet msg à tous ses voisins (lignes 70 et 71). Particulièrement, si $fanout \geqslant \max\{V_1, V_2, \cdots V_N\}$, Algorithme 9 devient un

algorithme de flooding.

Dans les algorithmes suivants, `Random()` génère un nombre aléatoire dans l'intervalle $[0, 1]$.

```
79 /* p_e : probabilité d'utiliser l'arête                    */
80 GossipPE (⟨msg⟩,p_e)
81     foreach s_j ∈ Λ_i do
82         if Random() ⩽ p_e then
83             Envoyer (⟨msg⟩, s_j)
```

Algorithm 10: Gossip avec choix d'arêtes probabiliste (dans s_i)

Dans *GossipPE* (Algorithme 10), chaque site choisit avec une probabilité p_e les arêtes sur lesquelles msg est retransmis (voir ligne 82). Notons que si $p_e = 1$ pour tous les sites, nous obtenons un algorithme de flooding.

```
84 /* p_v : probabilité de broadcast                          */
85 GossipPB (⟨msg⟩,p_v)
86     if Random() ⩽ p_v then
87         foreach s_j ∈ Λ_i do
88             Envoyer (⟨msg⟩, s_j)
```

Algorithm 11: Gossip basé sur une diffusion probabiliste (dans s_i)

Dans *GossipPB* (Algorithme 11), chaque site, excepté la source, diffuse msg à tous ses voisins avec une probabilité p_v (ligne 86). Si $p_v = 1$, ce protocole devient un algorithme de flooding.

```
89 /* d : seuil de degré                                     */
90 /* p_high : probabilité de retransmission pour les sites avec un
      grand degré                                            */
91 /* p_low : probabilité de retransmission pour les sites avec un
      faible degré                                           */
92 GossipDD (⟨msg⟩,d,p_high, p_low)
93     if V_i > d then
94         if Random() ⩽ p_high then
95             foreach s_j ∈ Λ_i do
96                 Envoyer (⟨msg⟩, s_j)
97     else
98         if Random() ⩽ p_low then
99             foreach s_j ∈ Λ_i do
100                Envoyer (⟨msg⟩, s_j)
```

Algorithm 12: Gossip dépendant du degré (dans s_i)

GossipDD (Algorithme 12) essaie d'améliorer la performance de *GossipPB* sur le graphe $\mathcal{S}(N, m)$ en distinguant deux types de sites. Les sites avec un plus grand

degré retransmettent msg avec une forte probabilité p_{high}, et ceux dont le degré est petit le retransmettent avec une faible probabilité p_{low}, où $p_{high} > p_{low}$. La décision du niveau du degré pour un site dépend du *seuil de degré d* (ligne 93).

6.4 Métriques de performance

Dans le contexte de la dissémination d'informations, les métriques suivantes sont couramment utilisées dans les littératures [65, 86, 92] pour l'évaluation des performances. **La Complexité en Messages, notée M :** mesure le nombre de messages envoyés (ou reçus, car il n'y pas de perte de message) par chaque site $M = \frac{\Omega}{N-1}$ où Ω est le nombre total de messages échangés pendant la dissémination.

La Fraction de Sites Infectés, notée α : définit le pourcentage de sites dans le système qui ont reçu le message généré par la source à la fin de la dissémination.

La Fiabilité, notée R : définit le pourcentage de messages générés par la source, qui ont été reçus par tous les sites. La fiabilité égale à 100% indique que l'algorithme réussit à diffuser tout message généré par la source à tous les autres sites dans le système, ce qui assure *l'atomicité* [86].

La Latence, notée L : mesure le nombre de sauts nécessaire pour diffuser un message à tous les destinataires, c'est-à-dire, la longueur du chemin le plus long parmi les chemins les plus courts de la source à tous les sites qui ont reçu le message.

6.5 État de l'art

De nombreuses recherches ont étudié la performance des quatre algorithmes sur les trois topologies en termes de la fiabilité, la message complexity et la latence.

Dans [86], les auteurs étudient la fiabilité de la dissémination d'informations en appliquant l'algorithme *GossipFF* dans $\mathcal{B}(N, p_N)$. En supposant que le *fanout* de tous les sites dans le système est inférieur au nombre de ses voisins, cet article conclut que pour qu'un système avec N sites atteigne la fiabilité R, il faut fixer $fanout = -\ln\left(\frac{-\ln(R)}{N}\right)$. Des études reposant sur des simulations [47, 52], aboutissent à des résultats similaires. Cependant, ils n'ont considéré que *GossipFF* dans $\mathcal{B}(N, p_N)$.

La performance de *GossipPB* dans $\mathcal{G}(N, \rho)$ est discutée et implémentée dans [15, 65], alors que [31] a théoriquement analysé *GossipPB* dans $\mathcal{B}(N, p_N)$. La performance de *GossipPB* dans $\mathcal{G}(N, \rho)$ est également étudiée de façon théorique dans [145], dans le but de choisir p_v afin d'atteindre une forte fiabilité. À partir de la discussion dans [113] sur la fiabilité, en s'appuyant sur la propriété de percolation dans $\mathcal{B}(N, p_N)$, les auteurs proposent une expression asymptotique du nombre moyen de messages et de la latence moyenne nécessaire pour atteindre l'atomicité. Néanmoins, par rapport à nos recherches, leurs efforts se sont focalisés sur la performance d'un algorithme de gossip dans une topologie aléatoire précise. Sur les deux topologies aléatoires $\mathcal{S}(N, m)$ et $\mathcal{B}(N, p_N)$, la latence d'une version modifiée de *GossipPB*, qui permet tout site de renvoyer le message plusieurs fois à un de

ses voisins avec certaine probabilité, est analysée par le modèle SIS (Susceptible-Infection-Susceptible) dans [58].

Dans [61], les auteurs montrent que la performance des quatre algorithmes de gossip dans $\mathcal{S}(N, m)$ est meilleure que dans $\mathcal{B}(N, p_N)$.

GossipPE présente de meilleures performances que *GossipPB* dans $\mathcal{G}(N, \rho)$, en fonction de la taille du système (ou le degré du site) dans [132]. En plus, *GossipFF*, *GossipPE* et *GossipPB* dans $\mathcal{S}(N, m)$ sont comparés de la même manière dans [51]. Dans [126], le choix entre *GossipPE* et *GossipPB* dépend des contraintes des différentes applications dans $\mathcal{G}(N, \rho)$. Néanmoins, pour une fiabilité donnée, il n'existe pas de méthode générale pour obtenir les gains quantitatifs pour tous les algorithmes dans les différents graphes en termes de la message complexity. Nous présentons donc un nouveau paramètre générique : le *fanout effectif*, qui exprime la puissance moyenne de dissémination des sites infectés et permet les comparaisons équitables des algorithmes de gossip.

6.6 Fanout effectif

Le nombre de messages retransmis des trois algorithmes dépend des paramètres d'entrée (p_v, p_e ou $fanout$) qui sont très différents. Dans le but de faire une comparaison équitable entre ces algorithmes de gossip sur les topologies décrites dans la section 6.2, nous introduisons un nouveau paramètre : le ***fanout effectif*** noté F_{eff}. Ce paramètre permet des analyses précises du comportement d'un algorithme de gossip sur une topologie. Il simplifie la comparaison des différents algorithmes sur une topologie. Ainsi, pour un algorithme et une topologie donnés, F_{eff} caractérise la puissance moyenne de dissémination des sites infectés. Pour une topologie fixée, il est possible de déduire les paramètres d'entrée à partir du fanout effectif.

Pour les algorithmes *GossipPE*, *GossipPB* et *GossipFF*, nous définissons respectivement que :

$$F_{eff}^{GossipPE} = p_e \cdot \bar{V} \tag{6.1}$$

$$F_{eff}^{GossipPB} = p_v \cdot \bar{V} \tag{6.2}$$

$$F_{eff}^{GossipFF} = \sum_{k=1}^{fanout-1} P(k) \cdot k + \sum_{k=fanout}^{N-1} P(k) \cdot fanout \tag{6.3}$$

$\mathbf{U_h}$ et $\mathbf{I_h}$ définissent respectivement le nombre espéré de sites qui ne sont pas infectés après h itérations des algorithmes (h sauts) et le nombre espéré de sites nouvellement infectés entre la $(h-1)^{\text{ième}}$ et $h^{\text{ième}}$ itération, pour $1 \leqslant h \leqslant L$ où L est la latence. Ainsi, $U_0 = N - 1$, $I_0 = 1$, et $U_L = (1 - \alpha)N$.

Les variables U_h et I_h sont liées de la manière suivante :

$$I_h = U_{h-1} - U_h,\ 1 \leqslant h \leqslant L \tag{6.4}$$

Theorem 7. *Pour les trois algorithmes de gossip sur les trois topologies à grande échelle ($N \gg 1$), la message complexity $M \approx \alpha F_{eff}$.*

Proof. Comme il n'y a aucune perte de messages, le nombre total de messages reçus par chaque site est égal au nombre de messages transmis. À chaque saut h, un site renvoie F_{eff} messages à ses voisins. Le nombre espéré de messages transmis au saut h est alors $F_{eff} \cdot I_h$.

Considérant tous les sauts et l'équation (6.4), nous obtenons le nombre total de messages reçus pour tous les sauts :

$$\Omega = \sum_{h=1}^{L} F_{eff} \cdot I_h = F_{eff} \cdot \sum_{h=1}^{L} I_h = F_{eff} \cdot (N - 1 - U_L)$$

Par la définition de la message complexity, $M = \frac{(N-1-U_L)}{N-1} \cdot F_{eff} = \frac{(\alpha N-1)}{N-1} \cdot F_{eff}$. Comme N est très grand, on obtient $M \approx \alpha F_{eff}$. □

En effet, en exploitant la mesure dans [29] du nombre de transmissions redondantes qui entraînent l'envoi du message reçu à un site infecté, on pourrait obtenir le même résultat.

Corollary 8. *Pour atteindre une forte fiabilité pour les trois algorithmes de gossip sur les trois topologies, à grande échelle, la message complexity $M \approx F_{eff}$.*

Proof. Pour qu'une forte fiabilité soit atteinte, (par exemple, heuristiquement, plus de 95% de sites *totaux* sont infectés en moyenne à la fin d'une dissémination), α doit être très proche de 100% et, selon le théorème 7, le résultat est obtenu. □

6.7 Conclusion

Dans ma thèse, nous avons comparé et cherché à améliorer les algorithmes de gossip dans diverses topologies aléatoires. Notre méthodologie a été d'étudier l'impact des propriétés propres à chaque graphe pour proposer des solutions efficaces adaptées à chacune des topologies. Dans la suite, j'envisage de continuer avec la même approche en améliorant nos premiers résultats dans le cadre des graphes scale-free.

6.7.1 Perspectives à court terme

Dans un premier temps je veux approfondir les premiers résultats obtenus dans les topologies scale-free. Ainsi j'envisage deux nouveaux travaux complémentaires :

1 - Évaluation sur des traces réelles Une des limites de nos résultats expérimentaux réside dans l'utilisation du modèle Barabási-Albert pour générer des graphes du type scale-free. Si cette approche est largement utilisée dans la littérature, il induit quelques effets de bord comme l'existence d'un degré minimal égal à m. J'envisage donc de compléter notre étude en intégrant dans notre simulateur des graphes réels issus de traces obtenues sur des réseaux sociaux. Ces nouvelles

expériences permettront entre autre de mesurer l'impact de la présence de quelques sites ayant un degré inférieur à m.

2 - Modélisation sur la fiabilité des algorithmes Les premières études sur les comparaisons de performances ont montré l'impact des propriétés de topologie sur la fiabilité. Cependant, nous n'avons pas encore déduire mathématiquement la relation entre la fiabilité et le fanout effectif. Ainsi, une modélisation sur la fiabilité des algorithms dans les topologies aléatoires sera cherchée et proposée.

6.7.2 Perspectives à long terme

À plus long terme, j'envisage d'utiliser les résultats obtenus sur d'autres topologies et plus particulièrement sur les graphes géométriques dans le but de proposer des algorithmes optimisés pour ce type de topologies. Nous pourrons par exemple chercher à exploiter la propriété de la *dépendance d'arêtes* présente dans ce type de graphe. Tout comme les solutions proposées dans le cadre du graphe scale-free on s'attachera à n'utiliser que la connaissance locale des sites. Parmi les pistes envisagées : la détection de ponts, de zones peu denses, d'effet de bordures, etc.. De plus, mes anciennes recherches sur le codage source-canal conjoint pourront être utilisée pour récupérer les données, si les dernières sont perturbées dans le canal de transmission, par example, les réseaux sans fil.

Index

Bibliography

[1] Omnet++: Discrete event simulation system *http://www.omnetpp.org/*. (Cited on pages 46, 68 and 73.)

[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. (Cited on pages 13 and 38.)

[3] J.-D. Abdulai Abdulai, M. Ould-Khaoua, L.M. Mackenzie, and A. Mohammed. Neighbour coverage: A dynamic probabilistic route discovery for mobile ad hoc networks. *SPECTS*, pages 165–172, 2008. (Cited on pages 12, 13 and 38.)

[4] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. Technical report, INRIA, Oct. 2002. (Cited on pages 23, 24 and 27.)

[5] M. Aizenman and D. Barsky. Sharpness of the phase transition in percolation models. *Communications in Mathematical Physics*, 108(10):489–526, 1987. (Cited on page 30.)

[6] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, Jan. 2002. (Cited on pages 5, 14, 18, 46, 67, 68, 83 and 87.)

[7] X. An and R. Hekmat. Probabilistic-based message dissemination in ad-hoc and sensor networks using directional antennas. *International Conference on Mobile Adhoc and Sensor Systems*, pages 432–438, 2009. (Cited on pages 5, 12, 13 and 17.)

[8] C. Avin. Distance graphs: From random geometric graphs to bernoulli graphs and between. *DIALM-POMC*, pages 71–78, 2008. (Cited on pages 17, 18 and 87.)

[9] S. Baehni, Patrick Th. Eugster, and R. Guerraoui. Data-aware multicast. *International Conference on Dependable Systems and Networks (DSN)*, pages 233 – 242, Jun. 2004. (Cited on page 34.)

[10] R. Bakhshi, D. Gavidia, W. Fokkink, and Maarten V. Steen. An analytical model of information dissemination for a gossip-based wireless protocol. *Symposium on Foundations of Computer Science*, pages 230–242, 2009. (Cited on page 13.)

[11] F. Banaei-Kashani and C. Shahabi. Criticality-based analysis and design of unstructured peer-to-peer networks as "complex systems". *International Symposium on Cluster Computing and the Grid*, pages 351–358, May 2003. (Cited on page 35.)

[12] Z. Bar-Yossef, R. Friedman, and G. Kliot. Rawms - random walk based lightweight membership service for wireless ad hoc networks. *International symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 238–249, 2006. (Cited on pages 34 and 42.)

[13] A. Barrat and M. Weigt. On the properties of small-world network models. *Eur. Phys. J. B*, 13:547–560, 2000. (Cited on page 15.)

[14] Kenneth P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17:41–88, 1998. (Cited on pages 34 and 42.)

[15] B. Blywis, M. Günes, F. Juraschek, and S. Hofmann. Gossip routing in wireless mesh networks. *ISPIMRC*, pages 1572–1577, 2010. (Cited on pages 4, 9, 28, 36, 39, 42, 64, 85 and 90.)

[16] B. Blywis, M. Glines, F. Juraschek, P. Schmidt, and P. Kumar. Des-sert: A framework for structured routing protocol implementation. *IFfP Wireless Days*, 2009. (Cited on pages 11 and 39.)

[17] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24:5–34, Jan. 2004. (Cited on pages 20 and 87.)

[18] F. Bonnet, A.-M. Kermarrec, and M. Raynal. Small-world networks: From theoretical bounds to practical systems. *Principles of Distributed Systems*, 4878:372–385, 2007. (Cited on page 39.)

[19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. *INFOCOM*, 3:1653 – 1664, Mar. 2005. (Cited on page 34.)

[20] M. Bradonjic, R. Els'asser, T. Friedrich, T. Sauerwald, and A. Stauer. Effcient broadcast on random geometric graphs. *ACMSIAM Symposium on Discrete Algorithms (SODA)*, pages 1412–1421, 2010. (Cited on page 18.)

[21] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.*, 85(25):5468–5471, Dec. 2000. (Cited on page 32.)

[22] J. Cartigny, F. Ingelrest, and D. Simplot. Rngrelay subset flooding protocols in mobile ad-hoc networks. *International Journal of Foundations of Computer Science*, 14(2):253–265, 2003. (Cited on pages 25 and 27.)

[23] J. Cartignyh, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Localized lmst and rng based minimum energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, pages 1–16, 2005. (Cited on pages 3, 25 and 27.)

[24] N. Carvalho, U. Minho, J. Pereira, U. Minho, R. Oliveira, L. Rodrigues, and U. Lisboa. Emergent structure in unstructured epidemic multicast. *International Conference on Dependable Systems and Networks*, pages 481–490, 2007. (Cited on page 23.)

[25] R. Chandra, V. Ramasubramanian, and Kenneth P. Birman. Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. *International Conference on Distributed Computing Systems (ICDCS)*, (6940057):275–283, April 2001. (Cited on page 22.)

[26] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. *Theor. Comput. Sci.*, 412(24):2602–2610, 2011. (Cited on pages 14 and 39.)

[27] F. Chung and L. Lu. The diameter of sparse random graphs. *Adv. in Appl. Math*, 26:257–279, 2001. (Cited on page 17.)

[28] IEEE Computer Society LAN MAN Standards Committee. Wireless lan medium access control (mac) and physical layer (phy) specifications. *ANSI/IEEE Standard*, 1999. (Cited on page 37.)

[29] D. E. Cooper, P. Ezhilchelvan, and I. Mitrani. Encounter-based message propagation in mobile ad-hoc networks. *Journal of Ad hoc Networks (Elsevier)*, 7, 2009. (Cited on pages 48 and 92.)

[30] P. Costa, M. Migliavacca, Gian P. Picco, and G. Cugola. Introducing reliability in content-based publish-subscribe through epidemic algorithms. *International Workshop on Distributed Event-Based Systems*, pages 1–8, 2003. (Cited on page 22.)

[31] S. Crisóstomo, U. Schilcher, C. Bettstetter, and J. Barros. Analysis of probabilistic flooding: How do we choose the right coin? *ICC*, pages 1–6, Jun. 2009. (Cited on pages 64 and 90.)

[32] G. D'Angelo and S. Ferretti. Simulation of scale-free networks. *International Conference on Simulation Tools And Techniques*, pages 1–10, 2009. (Cited on pages 4, 39, 41 and 42.)

[33] S. E. Deering and D. R. Cheriton. Multicast routing in datagram internetworks and extended *lans*. *ACM Transactions on Computer Systems*, 8:85–110, 1990. (Cited on page 20.)

[34] X. Défago, A. Schiper, and P. Urbán. Totally ordered broadcast and multicast algorithms: A comprehensive survey. *ACM Computing Surveys*, 36:2004, 2000. (Cited on pages 12 and 26.)

[35] 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. *Symposium on Principles of distributed computing (PODC)*, (536870):1–12, 1987. (Cited on pages 2, 11, 20 and 22.)

[36] M. Deshpande, B. Xing, I. Lazardis, B. Hore, N. Venkatasubramanian, and S. Mehrotra. Crew: A gossip-based flash-dissemination system. *International Conference on Distributed Computing Systems*, page 45, 2006. (Cited on pages 34 and 42.)

[37] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwrigh. Geographic gossip: Efficient aggregation for sensor networks. *International ACM/IEEE Symposium on Information Processing in Sensor Networks (IPSN)*, pages 69–76, April 2006. (Cited on page 34.)

[38] B. Doerr, M. Fouz, and T. Friedrich. Asynchronous rumor spreading in preferential attachment graphs. *SWAT*, pages 307–315, 2012. (Cited on page 22.)

[39] B. Doerr, M. Fouz, and T. Friedrich. Why rumors spread so quickly in social networks. *Commun. ACM*, 55(6):70–75, 2012. (Cited on pages 5 and 39.)

[40] V. Drabkin, R. Friedman, G. Kliot, and M. Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. Technical report, Computer Science Department, Technion - Israel Institute of Technology, 2006. (Cited on pages 37 and 42.)

[41] D. Dubhashi, C. Johansson, O. Häggström, A. Panconesi, and M. Sozio. Irrigating ad hoc networks in constant time. *Symposium on Parallelism in algorithms and architectures (SPAA)*, pages 106–115, 2005. (Cited on pages 34 and 42.)

[42] C. P. Dwivedi, S. Sharma, and V. Sharma. A survey on gossip-based energy conservation for wireless ad hoc network routing. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 2:31–36, 2012. (Cited on page 38.)

[43] N. Eggemann and S.D. Noble. The clustering coefficient of a scale-free random graph. *Discrete Applied Mathematics*, 159(10):953–965, 2011. (Cited on page 87.)

[44] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae*, 6:290–297, 1959. (Cited on pages 17, 30 and 33.)

[45] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(17):17–60, 1960. (Cited on pages 5, 14, 46 and 87.)

[46] P. T. Eugster and R. Guerraoui. Probabilistic multicast. *International Conference on Dependable Systems and Networks (DSN)*, pages 313–324, 2002. (Cited on page 34.)

[47] P. T. Eugster, R. Guerraoui, S. B. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. *ACM Transaction on Computer Systems*, 21:341–374, 2003. (Cited on pages 4, 28, 33, 42, 64, 85 and 90.)

[48] P. T. Eugster, R. Guerraoui, and P. Kouznetsov. δ-reliability: A probabilistic measure of broadcast reliability. *International Conference on Distributed Computing Systems*, pages 24–26, Mar. 2004. (Cited on page 11.)

[49] P. T. Eugster, R. Guerraoui, A. m. Kermarrec, and L. Massoulié. From epidemics to distributed computing. *IEEE Computer*, 37(5):60–67, May 2004. (Cited on pages 10 and 33.)

[50] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29:251–262, Aug. 1999. (Cited on page 32.)

[51] Stefano Ferretti and Gabriele D'Angelo. Multiplayer online games over scale-free networks: a viable solution? *ICST*, 2010. (Cited on pages 4, 41, 42, 65 and 91.)

[52] D. Frey, R. Guerraouia, A.-M. Kermarrec, B. Koldehofe, M. Mogensen, M. Monod, and V. Quéma. Heterogeneous gossip. *International Conference on Middleware*, pages 42–61, 2009. (Cited on pages 4, 28, 34, 42, 64, 85 and 90.)

[53] R. Friedman, V. Drabkin, G. Kliot, and M. Segal. On reliable dissemination in wireless ad-hoc networks. *IEEE Transactions on Dependable and Secure Computing*, 8(6):866–882, 2011. (Cited on pages 5, 38, 42 and 82.)

[54] R. Friedman and A. C. Viana. Gossiping on manets: The beauty and the beast. *Operating Systems Review*, 41(5):67–74, Oct. 2007. (Cited on pages 4, 28, 45 and 85.)

[55] A. Fronczak, P. Fronczak, and J. A. Hołyst. Mean-field theory for clustering coefficients in barabási-albert networks. *Phys. Rev. E*, 68(4):046126, oct 2003. (Cited on page 19.)

[56] A. Fronczak, P. Fronczak, and J. A. Hołyst. Average path length in random networks. *Phys. Rev. E*, 70:056110, 2004. (Cited on page 19.)

[57] A. El Gamal, J. Mammen, E. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. *INFOCOM*, 1, Mar. 2004. (Cited on page 39.)

[58] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. *INFOCOM*, pages 1455–1466, 2005. (Cited on pages 16, 36, 64 and 91.)

[59] Ayalvadi J. Ganesh, A.-M. Kermarrec, and L. Massoulié. Peer-to-peer lightweight membership service for large-scale group communication. *Networked Group Communication*, 2233:44–55, 2001. (Cited on page 34.)

[60] Ayalvadi J. Ganesh, A.-M. Kermarrec, and L. Massoulié. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, 52:139–149, 2003. (Cited on page 34.)

[61] B. Garbinato, D. Rochat, and M. Tomassini. Impact of scale-free topologies on gossiping in ad hoc networks. *NCA*, pages 269–272, 2007. (Cited on pages 4, 28, 33, 39, 41, 42, 64, 85, 86 and 91.)

[62] P.B. Godfrey and D. Ratajczak. Naps: Scalable, robust topology management in wireless ad hoc networks. *Information Processing in Sensor Networks (IPSN)*, pages 443–451, Apr. 2004. (Cited on page 38.)

[63] G. Grimmett. *Percolation*. Springer, 1989. (Cited on pages 4, 31, 35, 62 and 74.)

[64] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, April 1998. (Cited on page 24.)

[65] Z.J. Haas, J.Y. Halpern, and L. Li. Gossip-based ad hoc routing. *INFOCOM*, 3:1707–1716, 2002. (Cited on pages 2, 4, 5, 11, 14, 17, 28, 30, 33, 36, 38, 42, 46, 64, 85, 86 and 90.)

[66] C. Herrera and P.J. Zufiria. Generating scale-free networks with adjustable clustering coefficient via random walks. *Network Science Workshop*, pages 167–172, 2011. (Cited on page 68.)

[67] R. Van Der Hofstad. *New Perspectives in Stochastic Geometry*. Oxford University Press, 2010. (Cited on page 30.)

[68] P. Holme and B.J. Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, 2002. (Cited on page 68.)

[69] X. Hou and D. Tipper. Gossip-based sleep protocol (gsp) for energy efficient routing in wireless ad hoc networks. *WCNC*, pages 1305–1310, 2004. (Cited on page 38.)

[70] R. Hu, X. Huang, M. Kieffer, O. Derrien, and P. Duhamel: . :. Robust critical data recovery for mpeg-4 aac encoded bitstreams. *ICASSP*, pages 397–400, 2010. (Cited on page 83.)

[71] R. Hu, M. Kieffer, and P. Duhamel. Protocol-assisted channel decoding. *Signal Processing Lettre*, 19:483–486, 2012. (Cited on page 83.)

[72] R. Hu, J. Sopena, L. Arantes, P. Sens, and I. Demeure. Fair comparison of gossip algorithms over large-scale random topologies. *International Symposium on Reliable Distributed Systems*, pages 331–340, Oct. 2012. (Cited on pages 71, 78 and 81.)

[73] R. Hu, J. Sopena, L. Arantes, P. Sens, and I. Demeure. Efficient dissemination algorithm for scale-free topologies. *International Conference on Parallel Processing*, Oct. 2013. (Cited on page 82.)

[74] Q. Huang, Y. Bai, and L. Chen. Efficient lightweight broadcasting protocols for multi-hop ad-hoc networks. *International symposium on personal, indoor and mobile radio communications*, pages 1–5, 2006. (Cited on pages 39 and 42.)

[75] Tsung-Chuan Huang, Yen-Pang Lin, and Lung Tang. Neighbor-aware gossip-based broadcasting scheme for wireless sensor networks. *CMC*, pages 293–297, 2010. (Cited on pages 13 and 38.)

[76] F. Ingelrest and D. Simplot-Ryl. Localized broadcast incremental power protocol for wireless ad hoc networks. *Wireless Networks*, 14(3):309–319, 2008. (Cited on pages 25 and 27.)

[77] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouitim, and L. Viennot. Optimized link state routing. *draft-ietf-manet-olsr-06.txt, http://www.ietf.org/*, 2000. (Cited on page 23.)

[78] Márk Jelasit. *Self-Organising Software: From Natural to Artificial Adaptation.* Number 139-162. Springer, 2011. (Cited on pages 2, 10, 21 and 22.)

[79] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and Maarten V. Steen. The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. *ACM/IFIP/USENIX international conference on Middleware*, pages 79–98, 2004. (Cited on page 34.)

[80] M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst.*, 23:219–252, August 2005. (Cited on page 33.)

[81] H. Jeong and Y. Yoo. Dynamic probabilistic flooding algorithm based-on neighbor information in wireless sensor networks. *International Conference on Information Networking*, pages 340–345, Feb. 2012. (Cited on page 13.)

[82] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson. A simple protocol for multicast and broadcast in mobile ad hoc networks. *Internet Draft: draft-ietf-manet-simple-mbcast-01.txt*, 2001. (Cited on pages 2, 10 and 85.)

[83] A. Jüttner and A. Magi. Tree based broadcast in ad hoc networks. *Mob. Netw. Appl.*, 10(5):753–762, 2005. (Cited on page 25.)

[84] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. *Symposium on Foundations of Computer Science*, pages 565–574, November 2000. (Cited on page 22.)

[85] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. *Journal of the ACM (JACM)*, 51:943–967, November 2004. (Cited on page 35.)

[86] A.-M. Kermarrec, L. Massoulié, and A.J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE TPDS*, 3:248–258, Mar. 2003. (Cited on pages 2, 4, 5, 12, 13, 14, 17, 22, 28, 33, 42, 46, 47, 52, 64, 74, 85, 86 and 90.)

[87] A. Keshavarz-Haddad, V. J. Ribeiro, and R. H. Riedi. Color-based broadcasting for ad hoc networks. *Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt)*, pages 49–58, Apr. 2006. (Cited on pages 4, 12, 13, 25 and 27.)

[88] M. B. Khalaf, A. Y.Al-Dubai, and W. Buchanan. A new adaptive broadcasting approach for mobile ad-hoc networks. *Conference on Wireless Advanced*, pages 1–6, 2010. (Cited on page 13.)

[89] D. Kim and N. Maxemchuk. A comparison of flooding and random routing in mobile ad hoc network. *New York Metro Area Networking Workshop*, 2003. (Cited on pages 24 and 27.)

[90] A. Kini, V. Veeraraghavan, N. Singhal, and S. Weber. Smartgossip: an improved randomized broadcast protocol for sensor networks. *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 210–217, April 2006. (Cited on pages 12, 13 and 22.)

[91] P. Kyasanur, R. R. Choudhury, and I. Gupta. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. *International Conference on Mobile Adhoc and Sensor Systems*, pages 91–100, Oct. 2006. (Cited on page 38.)

[92] J. Leitao, J. Pereira, and L. Rodrigues. Epidemic broadcast trees. *SRDS*, pages 301–310, 2007. (Cited on pages 2, 3, 11, 12, 16, 22, 46 and 90.)

[93] Marc Lelarge. Efficient control of epidemics over random networks. *SIGMETRICS/Performance*, pages 1–12, 2009. (Cited on page 10.)

[94] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2004. (Cited on pages 38, 39 and 42.)

[95] F. Li and I. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. *IEEE Conference on Local Computer Networks*, pages 193–202, 2001. (Cited on page 23.)

[96] 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. (Cited on pages 3, 23 and 27.)

[97] J. Lipman, P. Boustead, J. Chicharo, and J. Judge. Resource aware information dissemination in ad hoc networks. *International Conference on Networks*, pages 591–596, 2003. (Cited on page 3.)

[98] J. Lipman, P. Boustead, and J. Judge. Neighbor aware adaptive power flooding in mobile ad hoc networks. *International Journal of Foundations of Computer Science*, 14(2):237–252, 2003. (Cited on pages 25 and 27.)

[99] 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. (Cited on page 85.)

[100] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *TRANSACTIONS ON MOBILE COMPUTING*, pages 111–122, 2002. (Cited on pages 23 and 27.)

[101] W. Lou and J. Wu. A reliable broadcast algorithmwith selected acknowledgements in mobile ad hoc networks. *GLOBECOM*, 2003. (Cited on page 26.)

[102] J. Luo, Patrick Th. Eugster, and J.-P. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad-hoc networks. *INFOCOM*, 3:2229–2239, 2003. (Cited on pages 9, 11 and 36.)

[103] W. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. *ACM ICS*, pages 84–95, 2002. (Cited on page 10.)

[104] I. S. Lysiuk and Z. J. Haas. Controlled gossiping in ad hoc networks. *WCNC*, pages 1–6, 2010. (Cited on page 39.)

[105] R. Meester and R. Roy. *Continuum Percolation*, volume Cambrige Tracts in Mathematics. Cambridge University Press, 1996. (Cited on page 29.)

[106] E. L. Merrer, A.-M. Kermarrec, and L. Massoulié. Peer to peer size estimation in large and dynamic networks: A comparative study. *International Symposium on High Performance Distributed Computing*, pages 7–17, 2006. (Cited on page 33.)

[107] M. Mihail, C. Papadimitriou, , and A. Saberi. On certain connectivity properties of the internet topology. *Journal of Computer and System Sciences*, pages 28–35, 2006. (Cited on page 16.)

[108] M. J. Miller, C. Sengul, and I. Gupta. Exploring the energy-latency tradeoff for broadcasts in energy-saving sensor networks. *Int. Conference on Distributed Computing Systems*, pages 17–26, 2005. (Cited on pages 13, 35 and 38.)

[109] G. Neglia, G. Reina, and S. Alouf. Distributed gradient optimization for epidemic routing: a preliminary evaluation. *IFIP Wireless Days*, pages 1–6, Dec. 2009. (Cited on pages 13, 38 and 42.)

[110] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003. (Cited on pages 15, 16, 18 and 46.)

[111] S.-Y. Ni, Y.-C. Tseng, Y.-S.Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *International Conference on Mobile Computing and Networking*, 8:151–162, 1999. (Cited on pages 2, 10, 39 and 68.)

[112] Ryotaro Oda, Tomoyuki Ohta, and Yoshiaki Kakuda. An efficient on-demand hierarchical routing protocol based on autonomous clustering for mobile ad hoc networks. *ISADS*, pages 180–187, 2007. (Cited on pages 12, 25 and 27.)

[113] K. Oikonomou, D. Kogias, and I. Stavrakakis. Probabilistic flooding for efficient information dissemination in random graph topologies. *Computer Networks*, 54:1615–1629, July 2010. (Cited on pages 36, 64 and 90.)

[114] P. Orlik, J. Zhang, B. Bhargava, G. Ding, G. Ding, Z. Sahinoglu, and Z. Sahinoglu. Reliable broadcast in zigbee networks. *SECON*, 2005. (Cited on page 22.)

[115] E. Pagani and G. P. Rossi. Reliable broadcast in mobile multihop packet networks. *Mobicom*, pages 34–42, 1997. (Cited on page 26.)

[116] V. Paruchuri, A. Durresi, D. Dash, and R. Jain. Optimal flooding protocol for routing in ad hoc networks. *Wireless Communications and Networking Conference*, 2003. (Cited on pages 24 and 27.)

[117] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of Internet: A Statistical Physics Approach*. Cambridge University Press, 2004. (Cited on page 69.)

[118] W. Peng and X. C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. *Workshop on Mobile and Ad Hoc Networking and Computing*, pages 129–130, 2000. (Cited on pages 24 and 27.)

[119] M.D. Penrose. *Random Geometric Graphs*, volume 5. Oxford University Press, oxford studies in probability edition, May 2003. (Cited on pages 5, 14, 17, 46 and 87.)

[120] T.K. Philips, S.S. Panwar, and A.N. Tantawi. Connectivity properties of a packet radio network model. *IEEE Transactions on Information Theory*, pages 1044–1047, 1989. (Cited on page 17.)

[121] B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47:213–223, 1987. (Cited on page 22.)

[122] J. Planès, S. Bord, and J. Fraysse. Continuous percolation in organic conducting blends. *Physica Status Solidi (B), Basic Research*, 230:289–293, 2002. (Cited on page 29.)

[123] A. Polynikisa. Random walks and scale-free networks. Master's thesis, University of York, UK, 2006. (Cited on page 19.)

[124] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report RR-3898, INRIA, 2000. (Cited on pages 23, 24 and 27.)

[125] S. Rajeswari and Dr.Y.Venkataramani. An adaptive energy efficient and reliable gossip routing protocol for mobile adhoc networks. *IJCTE*, 2:5, Oct. 2010. (Cited on page 13.)

[126] V. Raman and I. Gupta. Performance tradeoffs among percolation-based broadcast protocols in wireless sensor networks. *Workshop on Wireless Ad hoc and Sensor Networking*, pages 158–165, 2009. (Cited on pages 35, 42, 65 and 91.)

[127] Robbert V. Renesse. Scalable and secure resource location. *Hawaii International Conference on System Sciences*, 4:4012, Jan. 2000. (Cited on page 35.)

[128] A. M. Reynolds, G. A. Sword, S. J. Simpson, and D. R. Reynolds. Predator percolation, insect outbreaks, and phase polyphenism. *Current Biology*, 19(1):20–24, Dec. 2008. (Cited on page 29.)

[129] L. Rodrigues, U. De Lisboa, S. Handurukande, J. Pereira, U. Do Minho, R. Guerraoui, and A. M. Kermarrec. Adaptive gossip-based broadcast. *DSN*, pages 47–56, 2003. (Cited on pages 34 and 42.)

[130] E. Schoch, S. Dietzel B. Bako, , and F. Kargl. Dependable and secure geocast in vehicular networks. *ACM International Workshop on Vehicular Inter-Networking (VANET)*, September 2010. (Cited on page 39.)

[131] E. T. Seppälä, A. M. Pulkkinen, and M. J. Alava. Percolation in three-dimensional random field ising magnets. *Phys. Rev. B*, 66(14):144403, 2002. (Cited on page 29.)

[132] 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. (Cited on pages 4, 28, 33, 35, 42, 65, 85, 86 and 91.)

[133] S.-T. Sheu, Y. Tsai, and J. Chen. A highly reliable broadcast scheme for ieee 802.11 multi-hop ad hoc networks. *ICC*, 1:610–615, 2002. (Cited on page 26.)

[134] M. Sirivianos. Overlay multicast in mobile ad-hoc networks using araneola. 2009. (Cited on page 34.)

[135] A. O. Stauffer and V. C. Barbosa. Dissemination strategy for immunizing scale-free networks. *Phys. Rev. E*, 74:056105, Nov 2006. (Cited on page 10.)

[136] A. O. Stauffer and V. C. Barbosa. Probabilistic heuristics for disseminating information in networks. *IEEE/ACM Transactions on Networking*, 15:425–435, 2007. (Cited on pages 18 and 82.)

[137] D. Stauffer and A. Aharony. Introduction to percolation theory. *Crc Pr Inc*, July 1994. (Cited on pages 31 and 32.)

[138] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *Transactions On Parallel and Distributed Systems*, 13(1):14–25, 2002. (Cited on pages 24 and 27.)

[139] Ivan Stojmenovic and Jie Wu. Broadcasting and activity-scheduling in ad hoc networks. *Ad Hoc Networking*, pages 205–229, 2004. (Cited on pages 3, 24 and 27.)

[140] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. Technical Report 248, Rutgers University, 2000. (Cited on pages 24 and 27.)

[141] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980. (Cited on pages 3, 25 and 27.)

[142] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. 2011. arXiv:1102.2166. (Cited on page 79.)

[143] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *Transactions on Computers*, pages 545–557, 2003. (Cited on pages 3, 23 and 27.)

[144] D. Ustebay, R. Castro, and M. Rabbat. Efficient decentralized approximation via selective gossip. *Journal of Selected Topics in Signal Processing*, PP(99):1, 2011. (Cited on pages 4, 28, 45 and 85.)

[145] G. Vakulya and G. Simo. Energy efficient percolation-driven flood routing for large-scale sensor network. *IMCSIT*, 3:877–883, 2008. (Cited on pages 4, 28, 39, 42, 64, 85 and 90.)

[146] J. van den Berg, R. Meester, and D. G. White. Dynamic boolean models. 1997. (Cited on page 30.)

[147] S. Verma and W. T. Ooi. Controlling gossip protocol infection pattern using adaptive fanout. *International Conference on Distributed Computing Systems*, pages 665–674, 2005. (Cited on pages 11, 34 and 42.)

[148] S. Voulgaris, D. Gavidia, and Maarten V. Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *Journal of Network and Systems Management*, 13, 2005. (Cited on page 34.)

[149] 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. (Cited on page 69.)

[150] I. Webman, J. Jortner, and M. H. Cohen. Critical exponents for percolation conductivity in resistor networks. *Phys. Rev. B*, 16(6):2593–2596, Sep. 1977. (Cited on page 29.)

[151] J. Wu. An enhanced approach to determine a small forward node set based on multipoint relays. *Vehicular Technology Conference*, 4:2774–2777, 2003. (Cited on page 24.)

[152] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *DIAL Intertional Workshop Discrete Algorithms and Methods for Mobile Computing and Communicaiton*, 3:7–14, 1999. (Cited on pages 3, 24 and 27.)

[153] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems Journal*, 3:63–84, 2001. (Cited on pages 3, 24 and 27.)

[154] Ya Xu, John Heidemann, and Deborah Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Technical report, USC/Information Sciences Institute, Oct. 2000. (Cited on page 13.)

[155] S. Y. Yan. *Number Theory for Computing*. Springer-Verlag, Berlin, Germany, 2nd edition, 2002. (Cited on page 18.)

[156] Q. Zhang and D. P. Agrawal. Performance evaluation of leveled probabilistic broadcasting in manets and wireless sensor networks. *Simulation*, 81(8):533–546, 2005. (Cited on pages 37 and 38.)

Epidemic dissemination algorithms in large-scale networks: comparison and adaption to topologies

Abstract: Information dissemination (broadcast) is essential for numerous distributed applications. This must be efficient, which limits the message redundancy, and ensures high reliability as well as low latency. We consider here the distributed algorithms that benefitting from the properties of the underlying topologies. Nonetheless, these properties and the parameters in the algorithms are heterogeneous. Thus, we should find a method to fairly compare them. First of all, we study the probabilistic protocols for information dissemination (gossip) executed over three random graphs. The three graphs represent the typical topologies of large-scale topologies: Bernoulli graph, the random geometric graph, and scale-free graph. In order to fairly compare their performance, we propose a new generic parameter: effectual fanout. For a given topology and algorithm, the effectual fanout characterizes the mean dissemination power of infected sites. Furthermore, it simplifies the theoretical comparison of different algorithms over one topology. After having understood the impact of topologies and algorithms on the performance, we propose an efficient reliable algorithm for scale-free topologies.

Keywords: Information dissemination, distributed algorithms (Gossip), large-scale networks, random topologies, performance comparison, reliability, message complexity, latency.

Algorithmes de dissémination épidémiques dans les réseaux à grande échelle : comparaison et adaptation aux topologies

Abstraire : La dissémination d'informations (broadcast) est essentielle pour de nombreuses applications réparties. Celle-ci doit être efficace, c'est à dire limiter la redondance des messages, et assurer forte fiabilité et faible latence. Nous considérons ici les algorithmes répartis profitant des propriétés des topologies sous-jacentes. Cependant, ces propriétés et les paramètres dans les algorithmes sont hétérogènes. Ainsi, nous devons trouver une manière pour les comparer équitablement. D'abord, nous étudions les protocoles probabilistes de dissémination d'informations (gossip) exécutées sur trois graphes aléatoires. Les trois graphes représentent les topologies typiques des réseaux à grande-échelle : le graphe de Bernoulli, le graphe géométrique aléatoire et le graphe scale-free. Afin de comparer équitablement leurs performances, nous proposons un nouveau paramètre générique : le fanout effectif. Pour une topologie et un algorithme donnés, le fanout effectif caractérise la puissance moyenne de la dissémination des sites infectés. De plus, il simplifie la comparaison théorique des différents algorithmes sur une topologie. Après avoir compris l'impact des topologies et les algorithmes sur les performances , nous proposons un algorithme fiable et efficace pour la topologie scale-free.

Mots-clés : Dissémination d'information, algorithmes répartis (Gossip), réseaux à grande-échelle, topologies aléatoires, comparaison de performance, fiabilité, complexité de message, latence.