Gossip Strategies for Service Composition

Angelo Furno, Eugenio Zimeo
University of Sannio
Department of Engineering
82100, Benevento - Italy
{angelo.furno, eugenio.zimeo}@unisannio.it

Abstract—**Unstructured peer-to-peer (P2P) architectures offer several benefits to implement semantic discovery and composition in future-generation service registries. However, their success strongly depends on the adoption of efficient techniques for disseminating semantic queries over the network. Gossip strategies significantly reduce the amount of messages with respect to flooding, but they need a predefined tuning of the effectual fanout to achieve good performance. In this paper, we compare typical gossip strategies with our proposal, which is able to dynamically exploit network knowledge to fulfil a selective choice of propagation paths in order to ensure high recall and further reduce the number of messages exchanged.**

We perform the comparison in a simulated environment to observe resolution time, recall and message overhead on large-size and evolving networks while searching for service compositions. We have adopted Bernoulli, Random Geometric and Scale-Free graphs to model different network topologies. The experimental results show that our approach is able to adapt to network changes and preserve high levels of recall. In particular, it reduces message overhead, with respect to both optimized flooding and the analysed gossip-based strategies, or improves the recall, whereas resolution time remains almost unchanged.

Keywords-**Peer-to-Peer Computing; Gossip algorithms; Query Forwarding; Service Discovery; Service Composition.**

I. INTRODUCTION

As the number of Web services deployed in the Internet grows, their discovery becomes a fundamental feature for potential clients. This is particularly true in the cloud context, where the "pay-per-use" economy model asks for efficient and scalable discovery techniques to find the services that better fit user requirements. Since the discovery attempt of a desired service does not always lead to a satisfying solution, service composition has to be exploited to improve the potential for discovering the best services. However, discovery by composition increases computational complexity and consumption of resources, and consequently it requires more efficient and scalable techniques to be effective.

Distributed and decentralized infrastructures based on peer-to-peer (P2P) architectural models are the best candidates to implement future-generation service registries [1]. These models are interesting also because they enable a new form of collaboration where the roles of consumers and providers are interchangeable and allow for cooperatively creating service compositions that satisfy consumers' queries.

Decentralized registries ensure high functional and non-functional scalability: **(1)** by using a P2P registry, each organization is responsible for its own services; **(2)** several discovery processes, running in parallel over the network peers, can efficiently explore large repositories.

In our previous paper [2], we have proposed a technique for improving the performance of cooperative discovery of service compositions in P2P networks with superpeers. Any peer of an initially unstructured service network can publish semantically described services in a local registry and collaborate for distributed discovery of both atomic and composite services. Discovery queries, containing the semantic specification of the goal services to find, are quickly and efficiently disseminated through the network of peers. Our solution strongly reduces the number of messages exchanged, if compared with flooding, due to two main contributions: **(1)** topology restructuring whenever a new service composition is found; **(2)** efficient query dissemination on the connectivity graph.

In this paper, we focus on the second contribution and assume a completely unstructured P2P network (i.e., no super-peers). Based on the use of a P2P simulator, we compare our technique with a selection of gossip protocols [3]–[5], commonly used to disseminate information over large-scale and very dynamic networks more efficiently than flooding [6]–[8]. These protocols exploit graph redundancy to reduce message propagation by configuring the fanout (the number of outgoing links to use for information propagation).

The remaining part of the paper is organized as follows. Section II presents the main research efforts related to this work; Section III briefly introduces the approach used for P2P service discovery and composition; Section IV describes the algorithms for gossip-based query forwarding; Section V reports on the configurations used in our simulations as regards delays, algorithms and topologies for modelling large-scale networks; in Section VI, the evaluation scenarios and the experimental results are discussed; finally, Section VII concludes the paper and highlights future work.

II. RELATED WORK

Service discovery and composition often relies on centralized registries and discovery engines [9]. To improve scalability, dynamicity and robustness, in recent years, some researchers [10], [11] have exploited DHT-based networks (e.g., Chord [12]). However, structured P2P approaches suffer from high churn overhead, strong provider-dependency and complexity of hash functions to implement semantic matching. Unstructured P2P networks are acquiring growing consensus [1], [2], [13], [14] for supporting semantic service discovery and composition, due to their flexibility, fault tolerance and ease to implement semantic matching. However, they introduce potential overhead because of the huge amount of messages generated by flooding-based forwarding techniques (e.g., [1], [13], [14]), which may cause high routing costs and low scalability [15].

In unstructured and large P2P networks, gossip protocols [3]–[5] have been proposed as a solution to implement effective message broadcast, supported by their ease to deploy, high reliability, and scalability [6], [7]. Their probabilistic nature may significantly reduce the amount of exchanged messages, if the application context does not require 100% of reliability. Thus, these techniques have been widely used for information dissemination over the Internet [16], [17] (e.g., multi-player games, video streaming application), or exploited in wireless ad hoc and sensor networks [4], [18], [19].

Different gossip techniques have been proposed in the literature, like *Fixed-fanout Gossip* (i.e., GossipFF) [3], [17], [20], *Probabilistic Edge Gossip* (i.e., GossipPE) [4], [19] and *Probabilistic Broadcast Gossip* (i.e., GossipPB) [5], [18], based on the criterion to limit the outgoing links to use for information forwarding over well-known random graphs, typically used to model large-scale random topologies (e.g., *Bernoulli* [21], *Random Geometric* [22], and *Scale-Free* [23]). They present different performance figures with reference to the network topologies considered.

In [8], the authors use a generic parameter (*effectual fanout*) for comparing GossipFF, GossipPE and GossipPB strategies. They observe, by means of simulation, the trade-offs among dissemination reliability, message complexity and latency, with various kinds of input over Bernoulli, Random Geometric and Scale-Free topologies.

However, the adoption of gossip strategies for P2P service composition is still in its infancy since very few works address the problem of composition efficiency [2]. In this paper, we leverage on our previous experience on reducing message complexity for service discovery and composition to propose a dynamic gossip-based strategy that is able to adapt the fanout to the characteristics of the underlying network topology.

III. Cooperative Service Discovery and Composition

We adopt an unstructured P2P network, whose peers are involved in the concurrent execution of protocols for: **(1)** semantic service publishing in a local repository; **(2)** discovery of services satisfying user goals, allowing for collaborative and distributed composition; **(3)** query forwarding to efficiently propagate service requests in the network; **(4)** anti-entropy gossiping, as part of the forwarding strategy, to disseminate knowledge about the network structure.

Each node of the P2P network is a computer, a virtual machine or any other software/hardware device that is able to execute the protocols above. On the same node, there can be, in general, multiple peer processes. Peer links form an unstructured overlay network, called *connectivity graph*. It can be a random mesh, a ring, a tree or any other complex graph

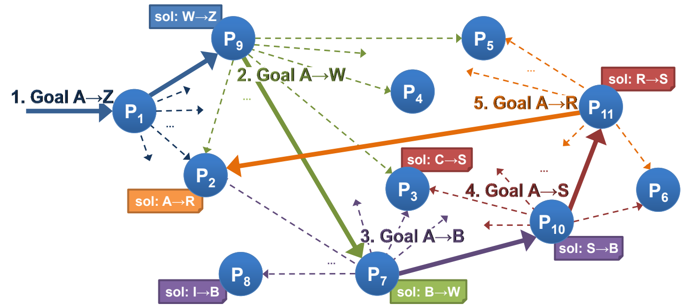


Fig. 1: Cooperative P2P composition to solve the abstract goal $A \rightarrow Z$.

and could be associated to physical proximity or low latency paths among the network nodes.

Our discovery-by-composition technique is based on the independent exploration of each peer's local service repository, by means of semantic matchmaking capabilities. If the discovery of atomic services satisfying the specified requested goal fails, the peers explore their local registry to find partial solutions, according to a backward search technique. In other words, since the service goal is specified as a transition from an initial state X to a final state Y (i.e., $X \rightarrow Y$), only service solutions matching the final state are considered (e.g. $K \rightarrow Y$) and a new goal request is issued to fill the gap (i.e., $X \rightarrow K$). Thus, the forwarding mechanism to gossip the new goal query is enacted. Partial solutions, related to the same initial query, are composed together according to a distributed merging process to generate the final complete solution.

Fig. 1 reports an example of this strategy to solve the $A \rightarrow Z$ abstract service goal, requested by peer P_1. $A \rightarrow R$, $R \rightarrow S$, $S \rightarrow B$, $B \rightarrow W$ and $W \rightarrow Z$ represent the solving services, published on the local repository of the different peers of the network (P_2, P_{11}, P_{10}, P_7 and P_9).

More details about our collaborative P2P composition strategy and its evaluation over some network topologies with reference to optimized flooding can be found in [2]. Query forwarding on the connectivity graph relies on our dynamic gossip-based strategy that will be described and compared with other gossip strategies in the next sections.

IV. Gossip-based query forwarding

Cooperative service discovery and composition require efficient mechanisms to spread goal queries over the P2P network. In [2], we proposed an approach, namely *probabilistic forwarding*, to reduce message overhead when performing service composition in unstructured P2P networks, with respect to flooding.

Algorithm 5 reports on our solution. Algorithms 1 to 4 describe three well-known gossip protocols (*GossipFF* [3], [17], [20], *GossipPE* [4], [19] and *GossipPB* [5], [18]), commonly used to disseminate information over large-scale networks. These protocols are able to reduce message overhead by limiting the number of neighbours for propagation, but they exploit statically defined thresholds, which usually do not change during execution.

Algorithm 1 describes the general gossip framework for spreading information over large-scale networks. We have used this protocol as an alternative to our probabilistic forwarding to diffuse queries containing the specification of the desired semantic services among the peers of the network. To start information dissemination, the query source sends a message to all of its neighbours (lines 2 and 3). When one node receives a message, which has not been previously received, it is simply re-transmitted according to the specific gossip strategy, otherwise the message is discarded. In the following, we consider a large-scale simulated P2P network P_N comprised of N sites $s_1, s_2, \ldots, s_N$. Node s_i's neighbourhood is denoted as Λ_i and $V_i = |\Lambda_i|$ represents its degree. The Gossip() procedure at line 10 may be implemented as [8]: **(1)** *GossipFF*, **(2)** *GossipPE*, and **(3)** *GossipPB*. All these algorithms receive the message to gossip and one strategy-specific $parameter$ (i.e. $fanout$, p_e or p_v), which is used to control the dissemination of the received message and whose value, decided at configuration time, is the same for all the nodes.

Algorithm 1 Generic Gossip algorithm

1. **procedure** BROADCAST(msg)
2.   **for all** $s_j \in \Lambda_i$ **do**
3.   Send(msg, s_j);
4.   **end for**
5. **end procedure**

6. **procedure** RECEIVE(msg)
7.   **if** $msg \notin MsgHistory$ **then**
8.   Deliver(msg);
9. $msgHistory \leftarrow msgHistory \bigcup \{msg\}$;
10.   Gossip(msg, $parameter$); $\triangleright$ *Gossip* procedure and *parameter* are decided at configuration time
11.   **end if**
12. **end procedure**

In GossipFF (Algorithm 2), node s_i sends its message (msg) to a fixed number (denoted as $fanout$) of randomly selected nodes in Λ_i (lines 6-8). If $fanout \geq V_i$, s_i transmits msg to all of its neighbours (lines 2, 3 and 11, 12). Particularly, if $fanout \geq max\{V_1, V_2, \ldots V_N\}$, Algorithm 2 is a pure flooding algorithm.

Algorithm 2 Fixed Fanout Gossip (at s_i)

1. **procedure** GOSSIPFF(msg, $fanout$)
2.   **if** $fanout \geq V_i$ **then**
3. $toSend \leftarrow \Lambda_i$;
4.   **else**
5. $toSend \leftarrow \emptyset$;
6.   **for** $f = 1 \rightarrow fanout$ **do**
7.   random select $s_j \in \Lambda_i \setminus toSend$;
8. $toSend \leftarrow toSend \bigcup s_j$;
9.   **end for**
10.   **end if**
11.   **for all** $s_j \in toSend$ **do**
12.   Send(msg, s_j);
13.   **end for**
14. **end procedure**

In GossipPE (Algorithm 3), site s_i randomly chooses those edges over which msg is transmitted according to a fixed probability p_e (lines 2-4, in which the *Random()* procedure generates a random number in the interval $[0, 1]$). When $p_e = 1$ for all sites, we obtain the flooding algorithm.

Unlike Algorithm 3, in GossipPB (Algorithm 4), each site, except the source, diffuses msg to all its neighbours with fixed probability p_v (lines 2-3). In particular, when $p_v = 1$ this protocol becomes the flooding algorithm.

Algorithm 3 Probabilistic Edge Gossip (at s_i)

1. **procedure** GOSSIPPE(msg, p_e)
2.   **for all** $s_j \in \Lambda_i$ **do**
3.   **if** $Random() \leq p_e$ **then**
4.   Send(msg, s_j);
5.   **end if**
6.   **end for**
7. **end procedure**

Algorithm 4 Probabilistic Broadcast Gossip (at s_i)

1. **procedure** GOSSIPPB(msg, p_v)
2.   **if** $Random() \leq p_v$ **then**
3.   Broadcast(msg);
4.   **end if**
5. **end procedure**

A. Efficient probabilistic forwarding over P2P unstructured networks

Algorithm 5 implements our forwarding mechanism. Each peer exploits this propagation technique on the connectivity graph to spread a locally submitted query or to forward the queries received from one of its neighbours.

Algorithm 5 Propagation over the connectivity graph (at s_i)

procedure PROPAGATEQUERYTONEIGHBOURS($query$)
    $\tau_{Groups} \leftarrow$ EvaluateGroupsThreshold()
    $\tau_{Density} \leftarrow$ EvaluateDensityThreshold()
    $\tau_{Hops} \leftarrow$ EvaluateHopsThreshold()
    $\tau \leftarrow \omega_{Groups} * \tau_{Groups} + \omega_{Density} * \tau_{Density} + \omega_{Hops} * \tau_{Hops}$;
    $f \leftarrow \lceil \tau \cdot \lambda \rceil$; $\triangleright$ λ = number of s_i's neighbours
    $count \leftarrow 0$;
    $i \leftarrow 0$;
    $newForwarded \leftarrow \{\}$;
    $selectedNeighbours \leftarrow \emptyset$;
    while $i < f \wedge count < \lambda$ **do**
        random select $s_j \in \Lambda_i \setminus selectedNeighbours$;
        $selectedNeighbours \leftarrow selectedNeighbours \bigcup s_j$;
        if s_j != $query.sender \wedge s_j \notin query.forwardedPeers$ **then**
            $newForwarded \leftarrow \{newForwarded | s_j\}$;
            $i \leftarrow i + 1$;
        end if
        $count \leftarrow count + 1$;
    end while
    $query.forwardedPeers \leftarrow newForwarded$;
    for all $s_i \in newForwarded$ **do**
        Send($query$, s_i);
    end for
end procedure

In a traditional flooding approach, every message received from a peer is forwarded to all of its neighbours. As a simple variation of the traditional flooding, we consider in our evaluation the *optimized flooding*. In this variation, the query message maintains the list of neighbours to which it has already been forwarded. In order to keep low the size of the message and the complexity of the forwarding mechanism, each query only stores the set of neighbours forwarded by its sender. Whenever the receiving peer has to decide about propagation, the list is considered to exclude local neighbours that already received the query from the sender. It is then replaced with the updated information about the actual new forwarded peers.

Algorithm 5 exploits a propagation threshold, namely τ, limiting the number of neighbours considered for query forwarding. The propagation threshold is the fraction of neighbours to select for propagation. It is dynamically computed anytime propagation has to be performed, by using up-to-date network information. Variable f represents the maximum number of neighbours to contact for query propagation (i.e., *fanout*). As in the optimized flooding strategy, received query messages contain the list of the peers forwarded by the sender. By excluding common neighbours that already received the query from the sender, up to f neighbours are randomly selected among the whole neighbourhood and stored as the new list of propagated peers in the query message. Finally, the message is forwarded to them.

We consider three different kinds of network information for the dynamic evaluation of τ: **(1)** Availability of relevant service composition overlays; **(2)** Global density of the network; **(3)** Number of peers (hops) crossed by the goal query from the source to the current peer. The information above makes it possible to distinguish three contributions to threshold τ, which we denote as τ_{Groups}, $\tau_{Density}$ and τ_{Hops}. τ is evaluated as a weighted (the weights ω_{Groups}, $\omega_{Density}$ and ω_{Hops} are defined at configuration time) and normalized (weights sum up to 1) sum of these three contributions. Each contribution is a threshold itself, is defined in the range $[0, 1]$ and is dynamically computed when the `PropagateQueryToNeighbours` algorithm is executed by a peer, using the currently available data. Additional thresholds could be considered in our forwarding framework, at the cost of increased overhead for evaluating τ.

Since in this paper the focus is on comparing the performance of different gossip-based forwarding strategies for service composition in the context of different network topologies with dynamic properties (i.e., changing network density), we refer to our previous work [2] for more details about τ_{Groups} and τ_{Hops} evaluation. Conversely, $\tau_{Density}$ computation, which we also improved with respect to our previous implementation, is described in the next Subsection.

B. Density Threshold ($\tau_{Density}$)

$\tau_{Density}$ is computed by means of an anti-entropy gossip protocol, performed by every peer of the P2P network to know the network density, defined as the average number of neighbours on the connectivity graph (i.e., average peer degree): **(1)** each peer s_i stores a local approximation of the average number of neighbours in the network as its $state_{s_i}$. The initial value is chosen as the number of s_i's neighbours on the connectivity graph; **(2)** each peer s_i performs a random selection of the neighbour s_j to gossip with; **(3)** when receiving the gossip information $state_{s_i}$ from neighbour s_i, peer s_j updates its state to the value: $(state_{s_i} + state_{s_j})/2$. This state update converges to the global state average (see [24]).

In order to know how much the current state is a reliable representation of the global network density, we incrementally compute, on each peer, the standard deviation (σ) of the local density information (δ), at any state update. At the beginning of the gossip protocol, we assume an infinite standard deviation. Hence, we introduce a parameter ($\bar{\delta}$), defined as:

$$\bar{\delta} = \begin{cases} \delta - \sigma, & \text{if } \sigma < \delta \\ 1, & \text{otherwise} \end{cases}, \tag{1}$$

which we use to evaluate the $\tau_{Density}$ threshold. The rationale behind the computation of $\tau_{Density}$ is that the denser is the network (i.e., higher $\bar{\delta}$), the less propagations will be necessary for the query to reach the various peers in the network, because of the presence of many alternative paths. By using its current $\bar{\delta}$ value, each peer computes $\tau_{Density}$ as:

$$\tau_{Density} = \frac{(K/\bar{\delta})}{(\bar{\delta} + (K/\bar{\delta}) - 1)}, \tag{2}$$

with $K \geq 1$. The $(K/\bar{\delta})$ factor (*inverse delta factor*) allows for better controlling density-based message reduction. The presence of this factor transforms a simple parametric hyperbole into an auto-tuning one. In case the locally estimated average density decreases (i.e., the network becomes sparser), the inverse delta factor will allow for a softer reduction of the number of messages with respect to a simple hyperbole, in order not to lose solutions. Conversely, if the network becomes denser, lower values of $\tau_{Density}$ will cause more messages to be cut. K is a dumping factor, tuned according to the specific topological properties of the connectivity graph.

We designed our gossip-based anti-entropy protocol to work only during the inactive phases of our system (i.e., when there are no service requests being processed on the peers) in order not to introduce message/computational overhead during the stages of P2P discovery. However, since the gossip message elaboration overhead is typically low, peers also include network density information within the messages regularly exchanged during query forwarding (i.e., piggybacking).

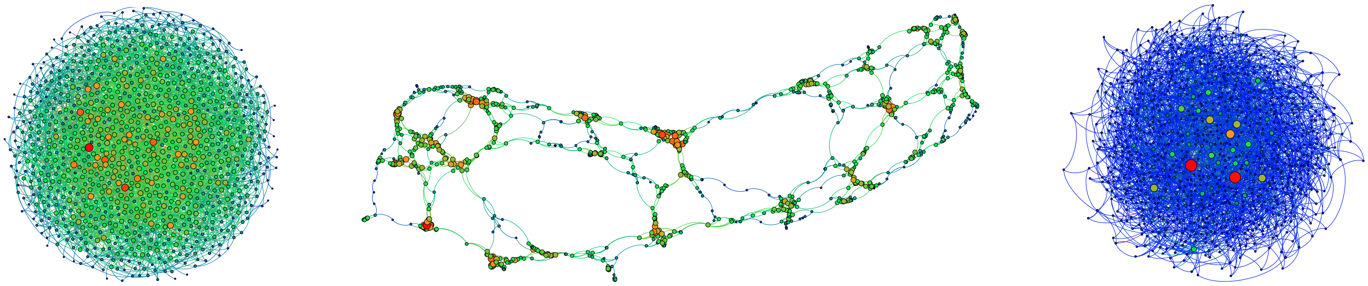
To dynamically compute the value of $\tau_{Density}$ in Algorithm 5, peers only need to access local up-to-date information, without additional message or computational complexity. In addition, we assume in this paper the absence of composition overlay networks (i.e., τ_{Groups} always equal to 1). Thus, the overall τ computation in Algorithm 5 requires negligible time and no additional message overhead, if compared with the other gossip strategies (Algorithms 1 to 4).

V. Simulation Test Bed

The probabilistic forwarding strategy and the gossip algorithms described in Section IV have been evaluated over large P2P networks, by exploiting the PeerSim [25] simulator.

In our experiments, we focused on measuring three performance indexes:

- *Recall*: system's ability to find simple or complex solutions to a goal request, when services published on peers' repositories make it possible to satisfy it. In the following experiments, the recall information is computed as the ratio of the number of solutions found by our system during the simulation cycles and the number of existing ones. Specifically, only one solution is present in the network at each simulation cycle. Therefore, recall is evaluated with respect to the number of simulation cycles;
- *Message overhead*: the number of request and response messages exchanged among the peers in the network to find solutions to a goal request. Messages are counted



(a) Bernoulli Graph (b) Random-geometric Graph (c) Scale-free Barabasi-Albert Graph

Colour scale from minimum to maximum node degree

Fig. 2: Examples of different P2P Network topologies: 1000 nodes and average degree per node ≈ 6

considering the whole network until the solution to the query under test is received;

- *Resolution time*: the time required for the submitter to receive the solution to the requested goal.

To make simulations consistent with realistic usage scenarios, we introduced three kinds of delay:

- *Semantic elaboration delay*: models the delay introduced by a realistic semantic matching between the query received by the peer and a service description available on that peer. For backward partial resolution, we consider half this delay, since only post-conditions are compared. In simulations, we have used $400ms$ for local complete solution and $200ms$ for partial backward solution. These delays have been computed by measuring the average time required to solve a number of queries with some of the matchmakers used in the S3 contest, the annual contest on Semantic Service Selection [26]. In particular, we focused on ISeM [27], since it offers matching capabilities based on IOPE descriptions;
- *Transmission delay (t_d)*: models the delay for placing a message from the application layer (peer sending a message) on the network abstraction layer (the node), when no multicast communication is available. If a peer on node A has to send at time t_0 a message to peers on nodes B, C and D, the message for B will be sent at t_0, for C at time $t_0 + t_d$ and for D at time $t_0 + 2 \cdot t_d$. A $1ms$ delay has been used in simulations;
- *Network latency*: is the delay for the simulated protocol used for message exchanges among the network nodes. It represents the time required for a message sent from a node to reach the destination one. In simulations, network latency has been modelled as a uniform random variable in the range $[10ms, 130ms]$. This range refers to latency measured (by using the ping application) on the Internet when sending small/medium messages to very distant destinations ($130ms$) or very close ($10ms$) ones;

Each node of the simulated network hosts one single peer process (therefore, the terms peer and node will be used interchangeably in the following). Each peer is able to publish, discover and compose services. Service descriptions are published on randomly selected peers before the beginning of each simulation cycle in order to evaluate system behaviour with respect to different assignments of the services to the peers.

Several configurations have been considered in simulations to evaluate the performance of the proposed algorithms, both as regards the topologies used for the connectivity graph and in relation to the parameters of the forwarding algorithms.

A. Connectivity graph configuration

The P2P network is initialized with a specific number of nodes and a connectivity graph.

To perform our simulations we considered three topologies: Bernoulli (or Erdős-Rényi) $B(N, p_N)$ [21], Random Geometric $G(N, \rho)$ [22], and Scale-Free graphs $S(N, m)$ [23], which are typically used to model peer-to-peer systems [20], wireless sensor networks [5] and ad hoc networks [4], respectively.

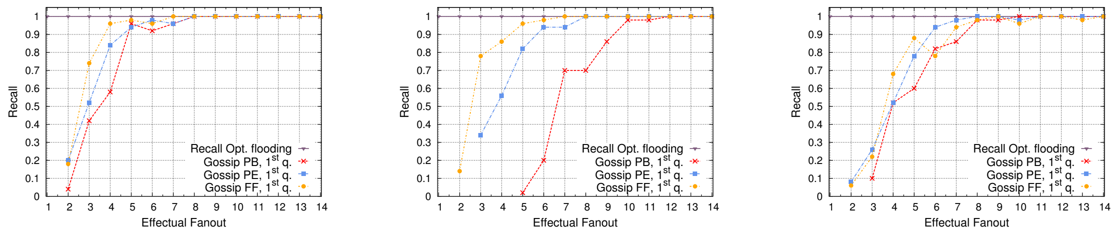
Figures 2a, 2b and 2c give a visual representation of each different topology. They have been obtained by analysing some of the connectivity graphs used in simulations with the Gephi graph visualization tool [28]. Nodes with degrees larger than the average are shown with larger size and hot colours, while the ones with smaller degrees are presented with smaller size and cold colours. Edges are depicted with the same colour of the node they depart from. The figures appear consistent with the theoretical definition of these topologies. In the case of the Bernoulli graph, most of the nodes present the average degree (green nodes in the figure), while there are few nodes with very high or low degrees. In the random geometric graph, instead, we may observe many high-density zones, containing nodes that are spatially close. Nodes belonging to different areas have few connections to one another because of the large spatial distance among them. Finally, according to a power law distribution, in the Scale-Free graph there are very few nodes with the highest connectivity (*hubs*), while the majority of the network has the lowest density (*peripheries*).

As in [8], we have considered topologies composed of $N = 1000$ sites. In Table I, we report the values of the parameters used to configure the three connectivity graph topologies.

Topology	Parameters
$B(N, p_N)$	$p_N = 0.014$
$G(N, \rho)$	$a = 7500, b = 3000, \rho = 330$
$S(N, m)$	$m_0 = 9\ (m_0 - clique), m = 7$

TABLE I: Topology parameters

This configuration is related to the case of a mean degree approximately equals to 14 ($\bar{V} \approx 14$). In the simulations, we have also considered other values for the mean degree



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

Fig. 3: Recall of gossip strategies for different topologies as a function of the effectual fanout

and changed the parameters according to the properties of the corresponding topology (see [8] for details).

B. Probabilistic Forwarding Algorithm Configuration

As reported in Table II, the query propagation mechanism (Algorithm 5) has been configured in order to give relevance to the knowledge about density available in the P2P network.

Query Propagation Parameters	
ω_{Groups}	0.1
$\omega_{Density}$	0.8
ω_{Hops}	0.1
K	$[70, 100]$

TABLE II: Parameters for the probabilistic forwarding algorithm

This is due to our intention of evaluating our forwarding algorithm with respect to other gossip strategies only by considering the topological properties of the network. To this purpose, we strongly limited the effects of the mechanisms for caching already found solutions, based on peer grouping, and for stopping query propagation based on traversed hops counting (see [2] for details).

Regarding the K dumping factor in $\tau_{Density}$ evaluation, values in the range [70, 100] showed good performance in our simulations. The results reported in Section VI are related to the mean value 85.

VI. Evaluation

In the following evaluation, we refer to the *main simulation scenario*: one random peer requests the discovery of a specific service goal (i.e., transition from state X to state Y); each peer hosts one service in its repository and there is only one composite solution in the network, specifically, a chain of 10 services published on different peers of the network. Variations of this main scenario are detailed in the text.

To configure the different gossip algorithms described in Section IV and compare them with our probabilistic forwarding strategy, we have used the effectual fanout parameter F_{eff}, defined in [8]. It enables the accurate analysis of the behaviour of a gossip algorithm over a topology and simplifies the theoretical comparison of different gossip algorithms on this topology. For a fixed topology and gossip algorithm, the effectual fanout characterizes the mean dissemination power of infected sites. In the following, we report on the definition of F_{eff} for the GossipPE, GossipPB, and GossipFF algorithms.

$$F_{eff} = \begin{cases} p_e \cdot \bar{V} & \text{GossipPE} \\ p_v \cdot \bar{V} & \text{GossipPB} \\ \sum_{k=1}^{fanout-1} P(k) \cdot k + \sum_{k=fanout}^{N-1} P(k) \cdot fanout & \text{GossipFF} \end{cases} \quad (3)$$

Given the effectual fanout, we have derived the values of the parameters used for the configuration of the specific gossip strategy (i.e., p_v, p_e and $fanout$), by using the inverse formulas of the ones in Eq. 3. Therefore, in the following results, we refer to a particular gossip strategy by its name and the chosen effectual fanout value.

For each specific topology of the connectivity graph (i.e., one of Bernoulli, Random Geometric or Scale-Free), the network size has been fixed to 1000 nodes, while the average node degree has been increased or decreased according to the specific scenario. Each degree configuration has been simulated over 50 cycles, before changing the graph topology parameters in order to have a new average degree. At each cycle, the services composing the solution have been shuffled over the peers of the network and the same query has been issued by one random submitter. Resolution times are presented as the average over the different values collected at the end of each simulation cycle, together with min-max confidence intervals. Recall is presented as the ratio between number of found solutions and number of available solutions (i.e., 50) in the range $[0, 1]$. Percentage difference of exchanged messages has been computed according to the formula:

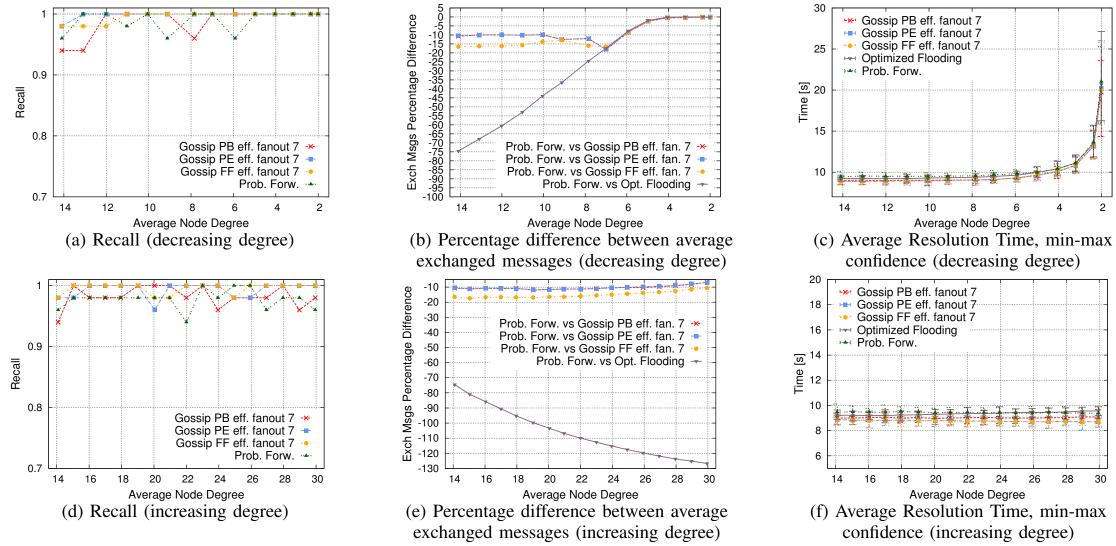
$$\%_{diff} = \frac{M_{\text{prob forw}} - M_{\text{other forw}}}{(M_{\text{prob forw}} + M_{\text{other forw}})/2} * 100 \quad (4)$$

where $M_{\text{forw strategy}}$ represents the average number of exchanged messages required to find the solution in the network when using the specified *forwarding strategy*.

A. Effectual fanout for gossip strategies

The first experiment was related to the analysis of the gossip strategies over Bernoulli, Random Geometric and Scale-Free topologies when searching for a 10-service composite solution distributed in the network. In particular, Fig. 3 graphically reports on the recall values observed after 50 simulation cycles performed over the networks of 1000 nodes, with an average node degree equal to 14, a reasonable value for large scale networks [8].

The results confirm that GossipFF is particularly good in terms of infection capability on $G(N, \rho)$, but not on $S(N, m)$ (in several cases it has the lowest recall), and that all the algorithms have almost the same infection power on $B(N, p_N)$ [8]. The experiment was also useful to choose the effectual fanout to use in the comparison of the different gossip strategies with our probabilistic forwarding technique, detailed in the next subsections. Since the number of messages increases when using larger effectual fanouts, we decided to consider an



(a) Recall (decreasing degree)

(b) Percentage difference between average exchanged messages (decreasing degree)

(c) Average Resolution Time, min-max confidence (decreasing degree)

(d) Recall (increasing degree)

(e) Percentage difference between average exchanged messages (increasing degree)

(f) Average Resolution Time, min-max confidence (increasing degree)

Fig. 4: Simulation results for Bernoulli graph with 1000 nodes: comparison with gossip and optimized flooding

effectual fanout equal to 7: this value assures an almost optimal recall (greater than 93% for at least two gossip strategies over the three topologies) at the lowest message overhead.

B. Simulations over Bernoulli Graph

Figures 4a, 4b and 4c graphically reports on the experimental results related to the first variation of the main simulation scenario: starting from an initial average node degree of 14, we progressively decreased it, by selectively removing edges, in order to preserve topological properties, to the smallest possible value to have the graph still connected (i.e., ≈ 2). This scenario simulates a typical situation for P2P networks in which many links disappear because of address changes or congestion.

The probabilistic forwarding strategy presents a stable recall (Fig. 4a), higher than 95% for all the average node degrees in the range $[14, 2]$ and is comparable to those related to the gossip strategies having effectual fanout equal to 7. This is due to the ability of our density threshold mechanism to adapt to network changes, by recognizing the need for more neighbours to be forwarded (i.e. by increasing the dynamic fanout), due to density decrease. In fact, the gossip protocol used to compute the average node degree quickly converges to the new lower value. Thus, the peers will use an increased value for $\tau_{Density}$ (see equation 2) and select more neighbours for propagation. Therefore, even the partial solutions distributed over less connected areas of the Bernoulli topology can be reached and complete composite solutions can be created from them.

This effect can be better appreciated in Fig. 4b, which focuses on percentage difference of the average message overhead between our probabilistic forwarding and optimized flooding or gossip strategies. When the Bernoulli graph is denser, the density threshold cuts up to 20% of the messages produced by the gossip strategies. Message reduction has the largest absolute value when the average degree is equal to 7, where the gossip strategies degenerate into a flooding approach. If the average degree is lower, message reduction decreases in order to not lose solutions, down to the case in which the flooding approach is the only viable approach to achieve high recall in a very sparse network (average degree is lower than 3).

Regarding resolution time (Fig. 4c), the overhead of the probabilistic forwarding is negligible with respect to the other approaches.

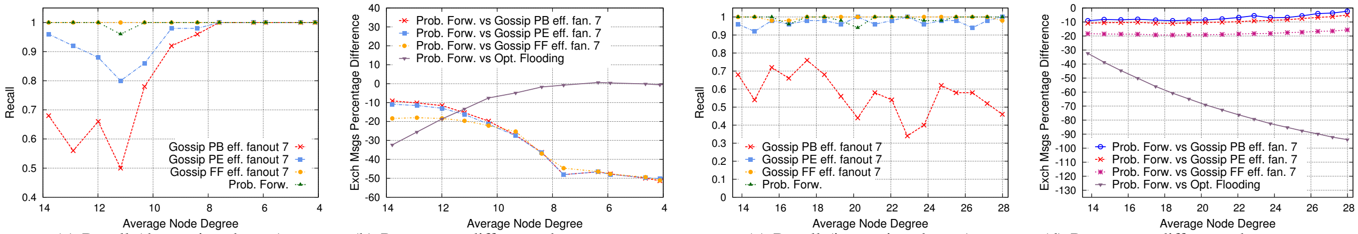
Figures 4d, 4e and 4f are related to the second variation of the main simulation scenario: the average node degree of the Bernoulli graph is progressively increased from 14 to 30. This scenario simulates the situation for P2P networks in which new links appear because of neighbour discovery or network decongestion. These simulations aim at verifying the capability of the probabilistic forwarding to maintain its reduced message overhead (Fig. 4e) and high recall levels (Fig. 4d) even on denser network, where flooding produces a significant growth in the number of exchanged messages (which is proportional to the number of edges in the graph), while gossip strategies with fixed fanout maintain a constant message overhead. In Fig. 4e, the message overhead percentage difference between our probabilistic strategy and optimized flooding increases (in absolute value) as the node average degree increases. With respect to gossip forwarding, it tends to be stable on values comprised in the range 10-20%, up to very high levels of average degree (i.e., ≈ 28). This is due to the asymptotical behaviour of the $\tau_{density}$ function, while message overhead for gossip strategies becomes constant after the average node degree becomes larger than the effectual fanout.

As in the previous scenario, resolution time (Fig. 4f) is almost unaffected by the choice of the forwarding strategy.

C. Simulations over Random Geometric Graph

The same scenarios considered in the case of Bernoulli graphs have been evaluated over the Random Geometric topologies (Fig. 5).

Figures 5a and 5b are related to the scenario of decreasing average node degree, while Figures 5c and 5d are related to the one of increasing average. The probabilistic forwarding strategy, as in the case of the Bernoulli topology, is effectively



(a) Recall (decreasing degree)
(b) Percentage difference between average exchanged messages (decreasing degree)
(c) Recall (increasing degree)
(d) Percentage difference between average exchanged messages (increasing degree)

Fig. 5: Simulation results for Random Geometric graph with 1000 nodes: comparison with gossip and optimized flooding

able to autonomously adapt to network density variations by reducing message overhead with respect to the gossip strategies (Figures 5b and 5d) and preserving high levels of recall (greater than 94% in all the cases, see Figures 5a and 5c).

An important aspect to point out is related to the selective choice of neighbours, described in Algorithm 5 and characterizing both the probabilistic forwarding and the optimized flooding: the received query is never sent back to the neighbours that have been forwarded at the previous step of propagation. Differently from the Bernoulli graph, the Random Geometric topology provides high-density areas in the network, where groups of neighbours are strongly connected to one another (see Fig. 2). Because of this property, by keeping track of the already forwarded neighbours in the query, as in our approach, it is possible to significantly reduce the message overhead both in the case of probabilistic forwarding and when using optimized flooding, with respect to the three different gossip strategies, which are not optimized in this sense.

However, as shown by the curves related to the comparison between probabilistic forwarding and optimized flooding, reported in Figures 5b and 5d, the effect of this optimization is mainly evident in the case of lower average degree. In fact, when the number of neighbours per node is relatively small, a larger overlap exists between previously forwarded nodes and new neighbours to be forwarded.

Resolution times are not reported due to space limitations. As for the Bernoulli topology, curves are comparable for the different forwarding strategies and their values increase in the range $[10s, 40s]$, in the case of decreasing degree, being more stable in the range $[10s, 20s]$, in the case of increasing degree.

D. Simulations over Scale-Free Graph

In the case of a Scale-Free graph, using an effectual fanout equal to 7 makes gossip strategies less effective, with respect to the previous topologies, especially if the average node degree is higher than 8 (Figures 6a and 6c). This arises from the peculiarities of the scale-free topology: very few nodes (*hubs* - red vertices in Fig. 2c) have the largest number of links in the network and most nodes have very few connections (*peripheries*). By cutting the number of forwarded neighbours at the network hubs, it is possible that entire zones of the networks are never reached by the goal query, while, if the message reduction is applied at a peripheral site, it may happen that hubs are not forwarded and the query never spreads outside the neighbourhood where it has been issued. Therefore, it may easily happen that solutions are not discovered. This problem is even more relevant in the case of resource aggregation (i.e., service composition), where multiple queries are issued by progressively discovering partial solutions.

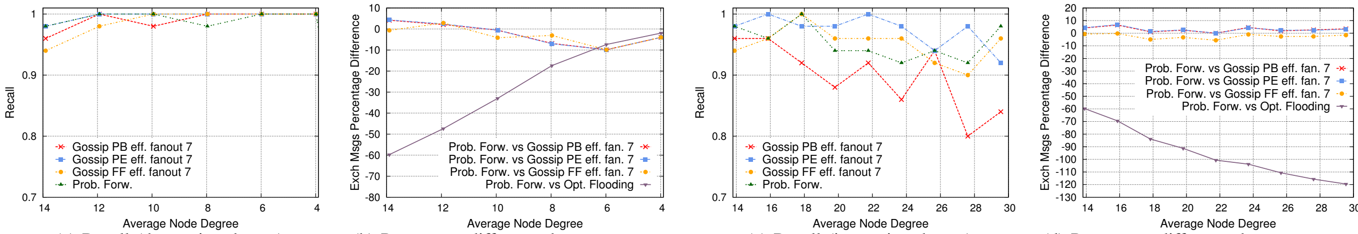
In the worst cases, our strategy is at least as effective as GossipFF forwarding in both the evaluation scenarios; also, in many cases it exhibits higher recall levels than all the other strategies, as can be seen in Figures 6a and 6c. This is due to the ability of our forwarding mechanism to recognize the need for more messages to be injected in the network, if compared with the Bernoulli or Random Geometric topologies, because many nodes have very low degrees and the anti-entropy gossip protocol executed to compute the average network density slowly converges. This is reflected in the lower (in absolute values) negative percentage differences of Figure 6b (regarding average degrees lower than 10) and in some positive values in Figures 6b and 6d (regarding average degrees greater than 10). The positive values indicates that, in some cases, our forwarding strategy inject more messages than the gossip ones, which are limited by their fixed fanout, allowing for higher levels of recall (i.e., more solutions found).

Resolution times (not shown in the figures) increase with a low exponential trend in the range $[8s, 11s]$, in the case of decreasing degree (from 14 to 4). Our probabilistic forwarding strategy is slower than the gossip ones according to an offset of about $+0.5s$. When average increases from 14 to 30, resolution times are stable in the same range with the same offset of $+0.5s$ for the probabilistic forwarding.

VII. Conclusion

We have compared, in the context of collaborative service composition, our probabilistic forwarding algorithm with optimized flooding and three well-known gossip-based strategies over unstructured P2P networks. Gossip-based techniques reduce message exchange for query propagation by exploiting graph redundancy through a parameter, called fanout. Our (informed) technique for service composition adopts network knowledge to guide routing and reduce query diffusion: when no or limited domain-specific knowledge is present in the network, the knowledge about the topology of the connectivity graph is exploited (e.g., average density) to avoid or limit redundant paths.

The comparison has been performed by exploiting a simulator to observe resolution time, recall and message overhead on large-size P2P networks. Two different scenarios have been considered regarding average node degree changes. Bernoulli, Random Geometric and Scale-Free graphs have been adopted



(a) Recall (decreasing degree)
(b) Percentage difference between average exchanged messages (decreasing degree)
(c) Recall (increasing degree)
(d) Percentage difference between average exchanged messages (increasing degree)

Fig. 6: Simulation results for Scale-Free graph with 1000 nodes: comparison with gossip and optimized flooding

to model different kinds of random networks. The experimental results show that our approach can adapt to network changes and preserve high levels of recall, without manual setup of any fanout parameter. In particular, it reduces message overhead, with respect to both optimized flooding and the analysed gossip-based strategies, or improves the recall, while resolution time remains almost unchanged in all the cases.

We are currently working on improving our forwarding strategy, by taking into account the topological properties of the peer's neighbourhood: if the forwarding peer has a reduced set of neighbours, with respect to the average degree, the number of neighbours to forward should be higher; in case of high density nodes, message reduction should take into account information about past propagation of the query in the neighbourhood in order to not cut messages over unexplored paths. We also plan to evaluate our strategy over other topologies (i.e., tree-like structures, rings) and to consider different evaluation scenarios (i.e., changes in network size, multiple overlay networks).

References

[1] V. Grassi, M. Marzolla, and R. Mirandola, "QoS-aware fully decentralized service assembly," in *Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press, 2013, pp. 53–62.

[2] A. Furno and E. Zimeo, "Efficient cooperative discovery of service compositions in unstructured P2P networks," *21th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2013)*, vol. 0, pp. 58–67, 2013.

[3] P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight probabilistic broadcast," *ACM Transactions on Computer Systems (TOCS)*, vol. 21, no. 4, pp. 341–374, 2003.

[4] B. Garbinato, D. Rochat, and M. Tomassini, "Impact of scale-free topologies on gossiping in ad hoc networks," in *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*. IEEE, 2007, pp. 269–272.

[5] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Transactions on Networking (ToN)*, vol. 14, no. 3, pp. 479–491, 2006.

[6] D. Ustebay, R. Castro, and M. Rabbat, "Efficient decentralized approximation via selective gossip," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 805–816, 2011.

[7] R. Friedman, D. Gavidia, L. Rodrigues, A. C. Viana, and S. Voulgaris, "Gossiping on MANETs: the beauty and the beast," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 5, pp. 67–74, 2007.

[8] R. Hu, J. Sopena, L. Arantes, P. Sens, and I. Demeure, "Fair comparison of gossip algorithms over large-scale random topologies," in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*. IEEE, 2012, pp. 331–340.

[9] L. Bevilacqua, A. Furno, V. di Carlo, and E. Zimeo, "A tool for automatic generation of WS-BPEL compositions from OWL-S described services," in *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on*, sept. 2011, pp. 1–8.

[10] O. D. Sahin, C. E. Gerede, D. Agrawal, A. E. Abbadi, O. H. Ibarra, and J. Su, "SPiDeR: P2P-based Web Service Discovery," in *Proceedings of the Third International Conference Service-Oriented Computing (ICSOC 2005)*. Springer, Dec. 2005, pp. 157–169.

[11] Z. Zhengdong, H. Yahong, L. Ronggui, W. Weiguo, and L. Zengzhi, "A P2P-based semantic web services composition architecture," in *e-Business Engineering, 2009. ICEBE'09. IEEE International Conference on*. IEEE, 2009, pp. 403–408.

[12] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4. ACM, 2001, pp. 149–160.

[13] M. Bisignano, G. Di Modica, and O. Tomarchio, "JaxSON: A semantic P2P overlay network for web service discovery," in *Services-I, 2009 World Conference on*. IEEE, 2009, pp. 438–445.

[14] M. Gharzouli and M. Boufaida, "PM4SWS: A P2P model for semantic web services discovery and composition," *Journal of Advances in Information Technology*, vol. 2, no. 1, pp. 15–26, 2011.

[15] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: search methods," *Computer networks*, vol. 50, no. 17, pp. 3485–3521, 2006.

[16] S. Ferretti and G. D'Angelo, "Multiplayer online games over scale-free networks: a viable solution?" in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST, 2010, p. 5.

[17] D. Frey, R. Guerraoui, A.-M. Kermarrec, B. Koldehofe, M. Mogensen, M. Monod, and V. Quéma, "Heterogeneous gossip," in *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag New York, Inc., 2009, p. 3.

[18] B. Blywis, M. Gunes, F. Juraschek, and S. Hofmann, "Gossip routing in wireless mesh networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*. IEEE, 2010, pp. 1572–1577.

[19] C.-C. Shen, Z. Huang, and C. Jaikaeo, "Directional broadcast for mobile ad hoc networks with percolation theory," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 4, pp. 317–332, 2006.

[20] A.-M. Kermarrec, L. Massoulié, and A. J. Ganesh, "Probabilistic reliable dissemination in large-scale systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 248–258, 2003.

[21] P. Erdos and A. Rényi, "On the evolution of random graphs," *Bull. Inst. Internat. Statist*, vol. 38, no. 4, pp. 343–347, 1961.

[22] M. Penrose, *Random geometric graphs*. Oxford University Press Oxford, 2003, vol. 5.

[23] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[24] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Trans. Comput. Syst.*, vol. 23, pp. 219–252, August 2005.

[25] A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator," in *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*, September 2009, pp. 99–100.

[26] Annual International Contest S3 on Semantic Service Selection. [Online]. Available: http://www-ags.dfki.uni-sb.de/~klusch/s3/

[27] M. Klusch and P. Kapahnke, "iSeM: Approximated reasoning for adaptive hybrid selection of semantic services," in *The semantic web: Research and applications*. Springer, 2010, pp. 30–44.

[28] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *International AAAI Conference on Weblogs and Social Media*, 2009.