On Probabilistic Flooding Search over Unstructured Peer-to-Peer Networks

Spiridoula V. Margariti
Department of Computer Engineering
T.E.I. of Epirus, Arta, Greece, GR-47100
and
Department of Computer Science & Engineering
University of Ioannina, Ioannina, Greece, GR-45110
`smargari@cse.uoi.gr`

Vassilios V. Dimakopoulos
Department of Computer Science & Engineering
University of Ioannina, Ioannina, Greece, GR-45110
`dimako@cse.uoi.gr`

Abstract

Probabilistic flooding has been proposed as a means of reducing the excessive message overheads induced by plain flooding in unstructured peer-to-peer network search. We propose here Advanced Probabilistic Flooding (APF), a novel strategy which operates differently from other known strategies. In particular, the decision of a node to propagate a message (or not) is based on both the popularity of resources and the hop distance from the node that initiated the query. The latter is used to estimate the number of nodes reached by the query message. Based on these parameters we adjust the forwarding probability at the time a node receives the query message so as to reduce the duplicate message overhead while maintaining a high probability of query success. The primary goal of our approach is to minimize the cost of search associated with excessive message transmissions. Our claims are supported by detailed experiments in various network topologies.

1 Introduction

Locating a desired resource or searching is a key challenge for a wide range of complex networks including the internet, social networks, biological networks and peer-to-peer (p2p) networks [29]. Here we focus on p2p networks, which can be categorized into two broad classes structured and unstructured. Lv et al [22] classifies them in centralized, decentralized structured, and decentralized unstructured while Park et al [28] further distinguishes unstructured p2p systems as centralized unstructured, hybrid unstructured and decentralized (or pure) unstructured.

Search in centralized networks utilizes a central directory while decentralized structured ones provide strict rules for data placement and resource discovery. Hybrid p2p topologies support both centralized (in the form of e.g. super nodes) and decentralized (access and exchange resources) organization and search functionality. Decentralized unstructured systems have arbitrary network topologies, resource placement and search. In this work, we concentrate on decentralized unstructured or simply *unstructured networks*.

Unstructured peer-to-peer (p2p) networks consist of a large population of networked computers that offer resources and operate in a fully decentralized manner. Such systems are usually quite large and highly dynamic and each node (or computer) has only information for a small subset of the other participating nodes. A key challenge is that of locating a desired resource. In the absence of information about node and resource placement, these systems use mainly flooding and its variants to provide search facilities. In flooding, the node (peer) that initiates the search sends a query message to all its neighbors. Any neighbor that does not know about the resource propagates the message to all its neighbors, and so on, until the resource is discovered or some termination conditions are reached.

Although flooding is a fundamental strategy for resource discovery in several types of networks, it is responsible for overloading the network with a large number of messages [9]. By overloading we mean the unnecessary query message transmissions that occur from many rebroadcasts, particularly in highly interconnected networks. These redundant retransmissions, named *duplicate messages*, are multiple copies of the same query that arrive to a node by following different paths. Studies in various contexts [26, 22] demonstrate the seriousness of the problem and highlight that rebroadcasts should be used with caution.

As a consequence, flooding-based search strategies are generally fast but produce excessive traffic, a large portion of which is redundant.The objective of many works is to provide models and techniques that are capable of moderating the redundant traffic. They employ network attributes and properties in order to reduce unnecessary message transmissions. They commonly use a *time-to-Live* or *TTL* parameter, in order to limit the forwarding of the query message beyond a predefined number of hops (steps). In traditional flooding the TTL value is fixed and determines the search space while in the Expanding Ring strategy [22] sequences of increasing TTL values are used. Initially, TTL is set to a small value and it is increased by one after every unsuccessful search. In [1] a TTL selection policy is proposed which allows each node to adjust the TTL value based on local information. Close to Expanding Ring, [5] establishes a relationship between TTL and the probability distribution of the location of the resources and derives sequences of TTL using dynamic programming if the probability distribution is known; otherwise randomized strategies are employed. Although this condition termination is well known and widespread for search mechanisms in p2p networks, a relation between TTL value and success probability is still lacking.

Probabilistic search, which is the subject of this paper, is a class of search strategies which try to alleviate the deficiencies of flooding. They exploit a property or an estimation of the network to make a probabilistic decision whether a message should be forwarded to another node or not. More specifically, each node that receives a query message, forwards it to each of its neighbors with some probability

p_f, termed *forwarding probability.* Probabilistic strategies may either be oblivious (e.g. [13, 22, 14]) or use predefined (e.g. [17, 23]) or tunable values (e.g. [1, 7, 27, 11]) for deciding which links to follow during the message forwarding process.

In this paper we propose a novel probabilistic flooding strategy, termed Advanced Probabilistic Flooding (APF). In our method, the forwarding probability is a function of a) the distance from the query initiator, b) the popularity of the desired resource and c) the number of a node's neighbors (node degree). The forwarding probability decreases according to the distance from the originating node. As such, our strategy manages to stop redundant flooding paths (also known as *overshooting*), improving flooding performance and minimizing unnecessary costs due to redundant messages. In addition, because the forwarding probability depends on resource popularity, the flooding extend is adaptive, and varying depending on the resource asked for, which adds to the efficiency of our scheme. In order to support our results, we conduct detailed simulation experiments that evaluate the performance of the proposed strategy. We also compare with other known probabilistic flooding schemes, and show that our proposal results in superior performance which in many cases matches the speed and success rates of pure flooding, while at the same time eliminating its duplicate message overheads.

The rest of the paper is organized in the following manner. In the next section we survey related work. In Section 3 we summarize the system model and our basic assumptions about the search procedure. Section 4 presents our probabilistic flooding strategy and elaborates on the required calculations. In Section 5 we focus on the evaluation of the proposed strategy through simulation experiments. In addition we extend our strategy to networks with tunable clustering coefficients and special topologies, such as random power-law ones. Finally, Section 6 summarizes the paper and discusses future work.

2 Related work

A number of strategies have been proposed to ameliorate the efficiency of flooding. Some of these strategies strive to limit the extend of flooding deterministically, while others try achieve this in a probabilistic manner. Deterministic schemes, such as iterative deepening [34], local indices based search [34], and the attenuated bloom filter [30], use an estimation of prior searches in order to forward the query. In *local indices*, each node keeps the collected file indices of peers within a predefined distance. If a search request is out of a node's knowledge, this node performs a flooding search. In *iterative deepening*, also known as *expanding ring*, the querying node issues a sequence of flooding searches with increasing depth limits (i.e. TTL values) until the resource is found. An attenuated bloom filter of depth d stored at each node, summarizes resources probably available on all nodes d-hops away. During the search process the next neighbor is determined by examining the attenuated Bloom filters of the current node.

In this paper, we are interested in probabilistic flooding schemes. Probabilistic search strategies may exploit a property or an estimation of the network to make a probabilistic decision whether a message should be forwarded to another node or not. Whenever a node receives a query message, it propagates it to (a subset of) its neighbors with some forwarding probability, p_f. One may classify these strategies according to the nature of the forwarding decision. A class of probabilistic strategies, which we term oblivious, forward the query message to randomly selected neighbors. Random walkers [22, 13], belong to this category and abandon flooding altogether. In the standard random walk, when a node receives a message and does not hold the desired resource, it forwards the query message to a randomly chosen neighbor. Alternatively, one can employ multiple walkers in parallel to locate a resource [22]. The adaptivity of the walker termination conditions leads to improved performance with respect to duplicate messages. Although random walks are known for their simplicity and low overhead in comparison with flooding, they usually result in larger response times [13, 22, 14]. The authors of [13] underline the properties of normalized flooding and propose a hybrid search scheme that combines flooding and random walks. However, these works do not address the problem of excessive traffic and high response time.

The works in [17] and [23] represent probabilistic strategies that use fixed, predefined values for the forwarding probability. In [17] a query message is forwarded to a fixed portion of neighbors (the authors used a predefined value of $p_f = 0.5$) while a value of $p_f = 0.6$ was chosen by Makino et al in [23] for exchanging routing information in power-law networks. Depending on the characteristics of the p2p network, these strategies may reduce the message overheads as compared to pure flooding, but may also suffer from poor success rates, resulting in lower overall performance. In contrast, in our case the forwarding probabilities are adjustable.

Finally, a third class of strategies use tunable forwarding probability values. In these strategies, the forwarding probability can be the same for all nodes [1, 17, 23, 7, 27] or varying [32, 35, 11]. Using percolation theory Banaei-Kashani and Shahabi [1] introduce probabilistic flooding and discuss how the forwarding probability can be set to a certain critical value in order to eliminate overhead, but without explicit guarantees regarding the probability of success. In both [1] and [7] the value of p_f is tuned so that query messages reach all nodes with high probability, while Oikonomou el al [27] derive a relationship between the value of p_f and a desired node coverage level. They set p_f to a constant value between 0.25 and 0.5. According to simulation results an appropriate value for p_f is 0.4 but it increases the termination time. Chrisostomo et al [8] are interested in a p_f that is capable of reaching all networks nodes. It can be given or it can be defined close to 0.5.

In Adaptive Probabilistic Search (APS) [33], the probabilistic forwarding of multiple walkers is estimated using knowledge from past searches. Similarly, SPUN [16] is based on prior knowledge of successful paths to probabilistically select the best subset of neighbors to forward the query. Adaptive Resource-based Probabilistic Search (ARPS) [35] utilizes different forwarding probabilities for different resources, depending on estimations of resource popularity. In addition, when applied to power-law networks each node adjusts this forwarding probability further, according to its degree, while in [32], the forwarding probability depends on both the sending and receiving nodes' degree. Close to [32], in generalized probabilistic flooding studied by [12], p_f is a considered a function of the distance from of query initiator and the degree of both the forwarding and the receiving nodes. Finally, Gaeta et al [11] considered the effect of a forwarding probability which is decreasing exponentially with the distance from the originator, as a tradeoff with the maximum allowable value of TTL.

In our proposed APF scheme, the forwarding probability decreases according to the distance from the originating node but unlike [11] the rate of decay is not exponential; in contrast, it is based on an estimation of the current node coverage. Moreover, it is not only degree-based as in [32] or decaying popularity-based as in [35]. Unlike the works discussed above where the decision which controls the search process is based on just a portion of the involved parameters, the proposed APF strategy utilizes all of them.

3 Preliminaries

3.1 The Network

In unstructured p2p systems participating peers form an overlay network which provides connectivity among them. We model peers as nodes in an undirected graph with a total of N nodes (vertices). Connections among the peers are represented by edges incident with the corresponding nodes in the graph. We consider search in overlay topologies using a random graph-theoretic model. We focus on well-known classes of these networks such as random and power law graphs and random graphs with tunable clustering coefficient.

Random graphs were introduced by Erdös and Rényi [10] and they are used widely to study topological properties of real networks. The two basic, but closely related models of random graphs are: the *uniform* and the *binomial*. In the first model, the graph is chosen uniformly at random among all graphs $G(N, M)$, i.e. graphs with N nodes and M edges. In the binomial model each of the $N(N-1)/2$ possible edges that connect pairs of vertices is present with probability p. This gives an average node degree of $\overline{d} = pN$.

Another topological class of interest is based on the preferential attachment model, that many real-life networks seem to follow. The main property of these networks is that their degree distribution obeys some power law, i.e. the probability that a node is connected to k other nodes is $\sim k^{-\alpha}$, where α a constant. Such a model was studied by Barabasi and Albert [29], observing that a new node prefers to connect with other nodes according to their degree. This means that, the probability that a new node will be connected to node i depends on the degree of node i node (say d_i), such as

$$p(d_i) = \frac{d_i}{\sum_j d_j}.$$

where the sum in the denominator is taken over all pre-existing nodes in the network. Finally, we consider random graphs with tunable clustering coefficients. Clustering is a common characteristic found in many real networks such as social, biological and technological networks [24, 31] and it expresses the property that two neighbors of a node v may also be neighbors themselves. The *clustering coefficient* (*cc*) of the network is measure of the clustering property and is defined as [24]:

$$cc = \frac{3 \times N_\Delta}{\sum_i \binom{d_i}{2}}$$

where d_i is the degree of a node i and N_Δ is the number of triangles (i.e. triads of nodes which are neighbors to each other) present in a network.

3.2 The Search Process

In a p2p network, peers submit queries for locating a copy of a resource they are interested in. If there exist $r \geq 1$ replicas of a particular resource, then the *popularity* of the resource is defined as $q = r/N$, that is the ratio of peers that possess it. It is vital that search strategies in decentralized unstructured p2p systems be *cost effective* and *successful*. By cost effectiveness we mean that the produced traffic to spread a query over the network is within reasonable limits. A search is considered successful if it discovers at least one replica of the desired resource.

In (pure) flooding, the peer that poses the query contacts all its neighbors. Any neighbor that does not know about the resource forwards the query message to its neighbors, which in turn forward it to their own neighbors and so on until a peer is contacted which possesses the desired resource or some other termination condition is met.

It is a well known fact that during this process a large portion of the generated traffic is redundant, i.e. the query message is re-transmitted, possibly multiple times, to already visited nodes. Due to this undesirable situation, *duplicate detection mechanisms* (DDMs) need to be employed in order to limit unnecessary duplicate messages. Such a mechanism is used for example in Gnutella where each query message is assigned a globally unique identifier (GUID) field [18]. When a peer receives a message, it stores its GUID in a local query cache and keeps it there for some time. If the peer receives the same query message again (i.e. the same GUID), it simply discards it, avoiding unnecessary retransmissions.

It should be clear that such a mechanism can never be perfect in the sense that a duplicate message may always appear long after its GUID was removed from a peer's cache, for example due to delays in the underlying physical network. Nevertheless, this simple mechanism is quite powerful and manages to eliminate most of the redundant traffic. Notice, however, that even by using a perfect DDM, there are certain duplicate messages that cannot be prevented. They concern copies of the query message that arrive at the same node through different paths in the network. All this duplicate detection mechanism can do is stop propagating them *after* they arrive. In what follows, we assume that independently of the variation of flooding used, an ideal DDM is in effect, unless otherwise stated.

A typical termination condition of the search process is the TTL parameter, which limits the length of the paths followed by a query message. A drawback of flooding-based search, is that if the resource is found in one path, the other paths followed by the flooded message continue to evolve until TTL expires, as there is no way to inform them that the resource has already been found. The purpose of our work is to further reduce this overhead. As in other randomized flooding strategies, we indirectly add one more termination condition: the forwarding probability, p_f. An intermediate peer may stop propagating the query further to its neighbors with probability $1 - p_f$. In contrast to other strategies, this probability is not constant; the forwarding of the message is a local peer decision which takes into account the distance from query initiator and an estimation of whether the query is already answered. The details are presented in the following section.

4 APF: A New Probabilistic Flooding Strategy

Consider a node u that has received a query for a particular resource. If u does not possess the resource, according to probabilistic flooding, it has to decide whether to forward it to a particular neighbor with some probability p_f. The main idea behind our strategy is that the query message should be propagated further *only if the query has not been answered yet*; that is p_f should be equal to the probability that the required resource has not been discovered by the flooding procedure up to that point. Thus p_f should be a function of t, the distance from the query initiator, or equivalently the "step" of the search.

Suppose that node u receives the query at step t, and that by step t there are N_t nodes that have received the query. Node u will decide whether to forward the message any further by estimating the probability that the resource has been found up to step t. The number of nodes that have not been visited yet is given by $N - N_t$. If there exist r replicas of the required resource, then the probability that the resource has not been found as of step t is given by the probability that all those r replicas have been placed among the $N - N_t$ remaining nodes (which clearly have to be at least r in number, otherwise the resource has already been found).

Because in unstructured p2p systems there is no correlation between the overlay topology and the placement of data, we may make the simplifying but accurate assumption of independent replica placements. As a result, the probability that a replica is placed in one of the not-yet-visited nodes is given by:

$$\frac{N - N_t}{N} = 1 - \frac{N_t}{N}.$$

Consequently, the probability that all r replicas are placed among these nodes is given by:

$$\left(1 - \frac{N_t}{N}\right)^r.$$

This is the probability that none of the already visited nodes is an owner of the resource. In such a case, node u should indeed forward the query and this is exactly what we do in our APF algorithm.

Summarizing the above discussion, in APF any node that receives the query message at step t, and does not possess the required resource, propagates it to its neighbors with a forwarding probability

$$p_f(t) = \left(1 - \frac{N_t}{N}\right)^{qN}, \tag{1}$$

where $q = r/N$ is the popularity of the resource and N_t is the number of distinct nodes that have received the query by step t. We let $p_f(0) = 1$, which in effect forces the initiator to transmit the query to all its neighbors.

In order for a node to calculate the forwarding probability of (1), it must known about:

(a) the popularity of the resource (q) and

(b) the total number of visited nodes by step t (N_t).

We consider (b) in the next section. Regarding (a), in what follows we assume that popularities are known. Otherwise they can be estimated at each node using techniques that monitor the local traffic. Zhang et al [35] use a mechanism to estimate the popularity of resources that exploits the feedback from previous searches by updating a local table with estimated popularities. After forwarding a query, the peer decreases the estimated popularity by a factor β. Positive answers increase the resource popularity according to the distance from the intermediate node to the resource owner. Bisnik and Abouzeid [2] suppose that each node maintains a popularity table for each resource on the system. The estimation of resource popularity is based on feedback from the most recent searches. If there is no knowledge or estimation mechanism, a peer should use a small value for q, in order to be on the conservative side.

4.1 Estimation of coverage

Consider now N_t, which is the coverage, i.e. the number of distinct nodes that have received the query message, by step t. If we denote by n_i the number of new peers contacted at step i, $0 \leq i \leq t$, of the search process, then clearly:

$$N_t = \sum_{i=0}^{t} n_i, \tag{2}$$

where $N_0 = n_0 = 1$ as the initiator node is considered to be aware of the query just before the first step. As a result, our problem is reduced to estimating the number of new nodes met at each step of the procedure.

An approximation of the average number of peers in distance i from a given peer is given by Newman et al [25]. In particular, if p_k is the probability that a node has degree k, then the generating function for the probability of vertex degrees is given by:

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k.$$

The generating function for the degrees of vertices reached by following a random edge is denoted as $G_1(x)$ and can be seen that $G_1(x) = G_0'(x)/G_0'(1)$, where G_0' is the first derivative of G_0. It is then derived that the average number of visited nodes in any distance i is given by:

$$[G_1'(1)]^{i-1} G_0'(1).$$

As Chandra et al pointed out [4, 3], this approach does not correspond exactly to the number of distinct nodes reached by flooding because of “cross” and “back” edges in the overlay network. Back/cross edges are links that lead to the formation of cycles. The authors then refine the model by estimating the cross edge probability and back edge probability at any distance i from a given node.

We follow a similar but simpler approach to estimate the number of visited nodes i steps away from the query initiator, which also takes into account the fact that in probabilistic flooding not all possible neighbors are contacted. Below we assume a random graph in the Erdös and Rényi sense; a refinement for power-law random graphs will be derived later.

Let n_i be the expected number of unique contacted nodes that lie in distance i from the query initiator. Consider the message transmissions during step i. Because we have assumed that a duplicate detection mechanism is in effect, the only peers that forward the query during this step are the n_{i-1} nodes in distance $i-1$ from the initiator for, otherwise, there would exist more than n_{i-1} nodes in that distance. Since a node should not transmit the message back to the neighbor that delivered it, the nodes in question will only transmit to new nodes in distance i. Given that the average node degree is $\overline{d}$, then from the $(\overline{d}-1)n_{i-1}$ message transmissions, only a portion of

$$1 - \frac{N_{i-1}}{N}$$

will be delivered to new nodes. This is because, out of the N possible destinations of a messages, the N_{i-1} have already been visited by time $i-1$. Considering also the fact that because of randomized flooding, each of the transmissions of nodes in distance $i-1$ occurs with probability equal to the forwarding probability, $p_f(i-1)$, we obtain:

$$n_i = (\overline{d}-1)n_{i-1}\left(1 - \frac{N_{i-1}}{N}\right)p_f(i-1), \tag{3}$$

In conclusion, (1)–(3) constitute a set of recursive equations which can be used to obtain the desired forwarding probability at any step of the procedure.

5 Evaluation

In this section we evaluate the proposed strategy through simulation experiments. We have constructed a peer-to-peer message-level network simulator, implemented in C, which is able to generate various random graph topologies, such as uniform random, random regular, random with tunable clustering coefficients and power-law graphs according to user-supplied parameters. It can also generate random graphs with a given degree distribution and clustering coefficient according to the algorithms of Heath and Parikh [15]. The simulator also places R resources, based on the *uniform* or the *proportional* replica distribution schemes [6]. In uniform replication, every resource has the same number of replicas while in proportional replication the number of replicas of each resource is proportional to the query probability. These have been shown to be among the worst replication strategies with respect to search performance [6] and have been chosen so as to stress the search algorithms we consider here.

After the topology construction, for each simulation run nodes of the system are selected in random and are marked as owners of a resource replica, according to the given placement policy and the resource popularities. Then one of the peers, chosen uniformly randomly, initiates a search query. The search is limited by a TTL (time-to-live) parameter, t, which gives the maximum allowed number of

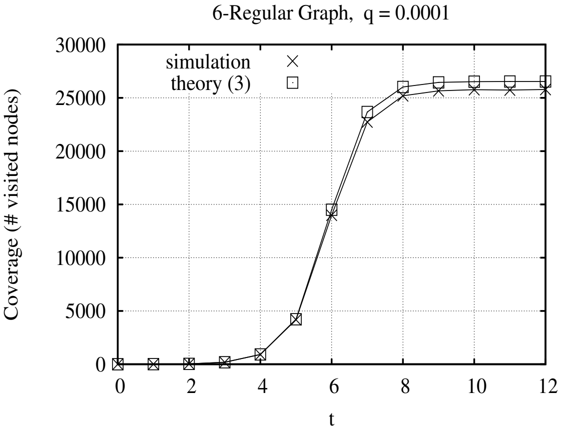


Figure 1: Coverage in a random 6-regular network of 100,000 nodes.

steps / path length. During the search detailed statistics are kept, including the total number of messages, the number of duplicate messages, the number of visited nodes, etc. If at least one replica of the resource is found within the t steps, the query is considered *successful*, otherwise it is unsuccessful. We run each experiment at least 1000 times and average the results. The portion of runs that resulted in successful queries gives the *probability of success* for the particular value of t. For each simulation run the network and the resource placement remains unchanged.

In order to make a comparative study, in our simulator we have implemented most known probabilistic flooding strategies, as presented in the introduction. Here we include results for representative algorithms which include plain flooding, modified BFS [17] and ARPS [35], along with our proposed APF.

5.1 Random Graphs

We have examined a large number of random and random regular graphs. In a random graph the degree of vertices varies according to the probability of inclusion of an edge (p), giving an average degree of $\overline{d} = pN$. In a random d-regular graph all nodes have the same degree, i.e. $\overline{d} = d$.

Before we present comparative results, we demonstrate the effectiveness of our approach. In Figs. 1–2 we have considered a random 6-regular network. For the plot in Fig. 1 we issue queries for a resource with popularity $q = 10^{-4}$. The curves show the number of covered nodes, as obtained from the simulation, and as predicted by (2), showing the accuracy of the estimation. In Fig. 2 we plot the forwarding probability for different values of resource popularity. It can be seen that the less popular the resource, the higher the forwarding probability in order to force propagating the query to more nodes. Another important observation is the fact that the shape of the curves are the exact opposite of the coverage curve in Fig. 1. The forwarding probability is high in the initial steps and later becomes low; the transition occurs at approximately the same step where coverage is becoming

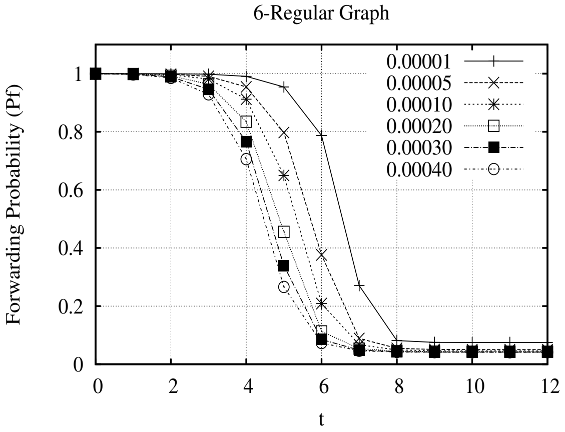


Figure 2: Forwarding probability curves for different popularity values in a random 6-regular network of 100,000 nodes.

Table 1: Simulation network parameters (ER random network, 100,000 nodes)

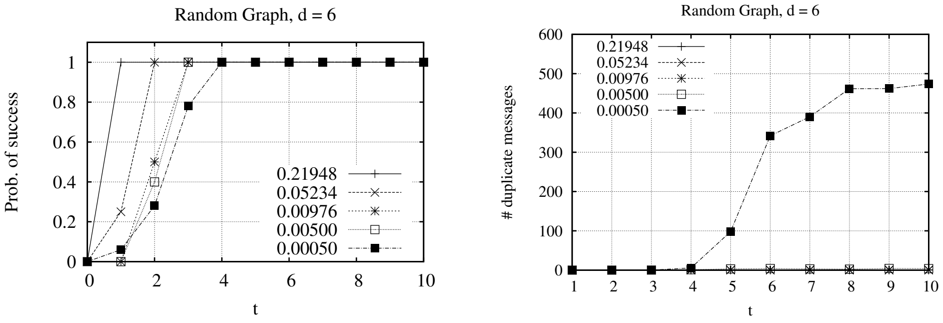
$\overline{d}$	q	flooding	mBFS	ARPS
5	5×10^{-5}	$p_f = 1$	$p_f = 0.5$	$p_f = 0.9$
6-regular	3×10^{-4}	$p_f = 1$	$p_f = 0.5$	$p_f = 0.8$
7 with $cc = 0.14$	5×10^{-5}	$p_f = 1$	$p_f = 0.5$	$p_f = 0.9$

high. We believe this plays an important role in the performance of our strategy.

In Fig. 3, we measure the efficiency of APF using proportional replication on a random graph of $N = 100{,}000$ nodes and average degree equal to $\overline{d} = 6$. We assume that each query i follows a Zipf-like distribution ($\propto i^{-\alpha}$). Also resource popularities follow Zipf-like distributions ($q_i \propto 1/i^{-\alpha}$) with a given α value. For this example, the value of α is set to 0.8, while the resulting resource popularities ranged from approximately 0.22 to 10^{-4}. The plot shows that the amount of duplicate messages is negligible while high success rate is achieved. Even if the resource popularity is very low (e.g. 0.0005), the number of duplicate messages also remains in low levels. The reason is that each peer exploits local information in addition to its distance from the originating query node (TTL value).

Next we mainly use uniform replication as both uniform and proportional have similar results, the search space is independent from query rate [6] and uniform replication is fair in terms of equitable allocation of replicas. In the following, we present experiments with random and random regular p2p networks of $N = 100{,}000$ nodes. The average node degree varies from 5 to 7 and the resource popularity ranges as $5 \times 10^{-5} \leq q \leq 10^{-3}$, that is resources have from 5 to 100 replicas for the particular network size we consider, i.e. not very popular, so as to stress the search procedure. Table 1 summarizes the simulation parameters.

Figs. 4–5 present the performance of the proposed strategy in comparison with the others. In Fig. 4 we have considered a random graph with an average node



(a)

(b)

Figure 3: Probability of success (a) and duplicate messages (b) in a random network of 100,000 nodes with proportional replication and varying resource popularity ($0.21948 \leq q \leq 0.0005$).

degree of 5, where we search for a rare resource with popularity $q = 5 \times 10^{-5}$. We plot both the probability of success and the message overhead, as the number of duplicate message transmissions. In Fig. 4(a) we observe that most strategies manage to achieve 100% success probability with a TTL value of 7 hops, where most of the nodes have been visited. At the same time, Fig. 4(b) shows that this is achieved with minimal duplicate message overheads for APF. Modified BFS has an even smaller duplicate message cost but this is due to the low success rate it achieves as seen in Fig. 4(a).

The situation is similar in Fig. 5 where all strategies have better performance due to the higher popularity of the required resource. Even in this case, however, the superiority of our scheme is apparent since it combines the effective success rate with almost no duplicate message overheads, in contrast to all the other methods.

Next we consider random graphs with given clustering coefficient. Because the clustering coefficient, cc, represents the probability that two nodes with a common neighbor are also neighbors themselves, only a fraction $1 - cc$ of the edges a node uses at any step i will lead to new (unvisited) nodes. Consequently we modify (3) as follows in order to account for the given clustering coefficient:

$$n_i = (\overline{d} - 1) n_{i-1} (1 - cc) \left(1 - \frac{N_{i-1}}{N}\right) p_f(i-1), \tag{4}$$

Thus for this type of networks we use equations (4) instead of (3).

In Fig. 6 we present results for a random graph (Poisson-distributed degrees) with clustering coefficient $cc = 0.14$ while the Table 1 summarizes the other network parameters for this simulation. The results for the probability of success (Fig. 6(a)) and the number of duplicate messages (Fig. 6(b)) are in agreement with what we observed in the other networks we considered. APF, with the modification of (4), behaves quite efficiently.

In conclusion the behavior of APF is quite impressive, both in terms of success rate and in terms of overhead. It manages to approach the performance of pure

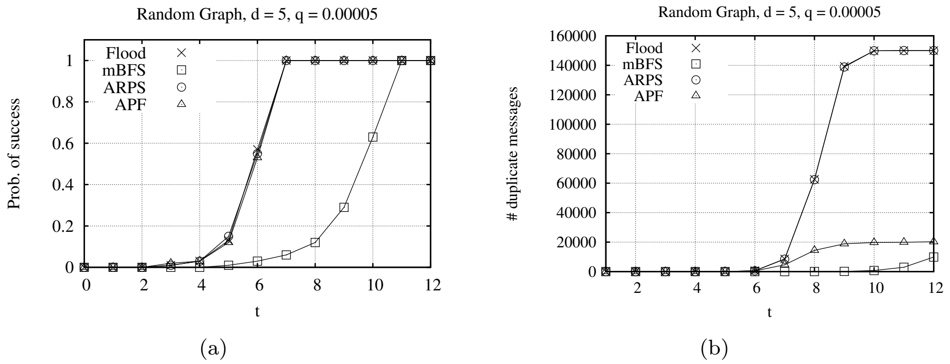


Figure 4: Probability of success (a) and duplicate messages (b) in a random network of 100,000 nodes and $r = 5$.

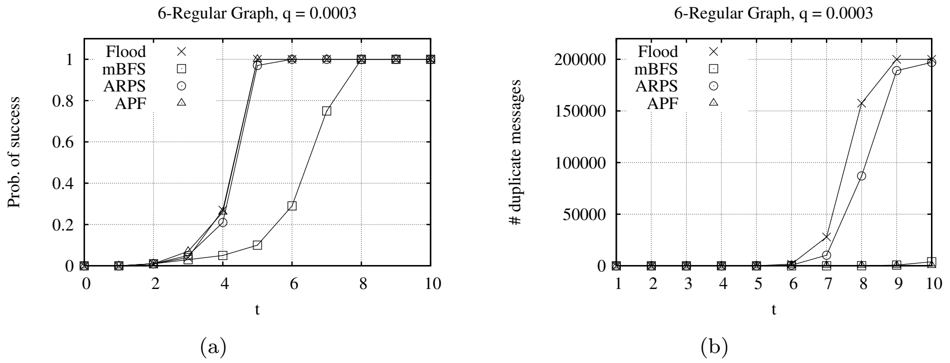


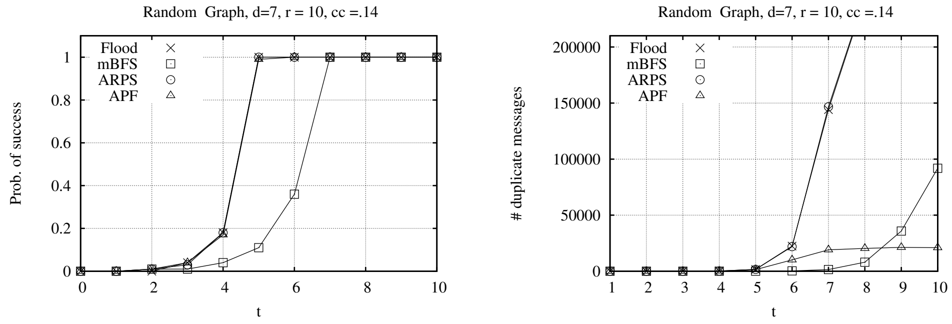
Figure 5: Probability of success (a) and duplicate messages (b) in a random 6-regular network of 100,000 nodes and $q = 0.0003$.

flooding but without incurring its cost. As an additional note, we remind the reader that we have assumed that an ideal duplicate detection mechanism is in effect as discussed in Section 3.2. We have conducted identical experiments with the DDM disabled; while the probability of success is not affected, the number of duplicate messages is vastly increased in all algorithms but APF, accentuating the benefits of our strategy even more.

5.2 Application to other networks

In this section we assess the performance of our approach on networks other than the uniform random ones. We consider power-law networks and in particular we generate such networks with 100,000 nodes using the Barabasi-Albert (BA) model [29], starting from two nodes and adding one new node at each step with $m = 2$ links.

Deploying the APF algorithm over this topology, we observed that, while the



(a)

(b)

Figure 6: Probability of success (a) and duplicate messages (b) in a random network of 100,000 nodes, $\overline{d} = 7$ and clustering coefficient $cc = 0.14$.

Table 2: Simulation parameters of other networks

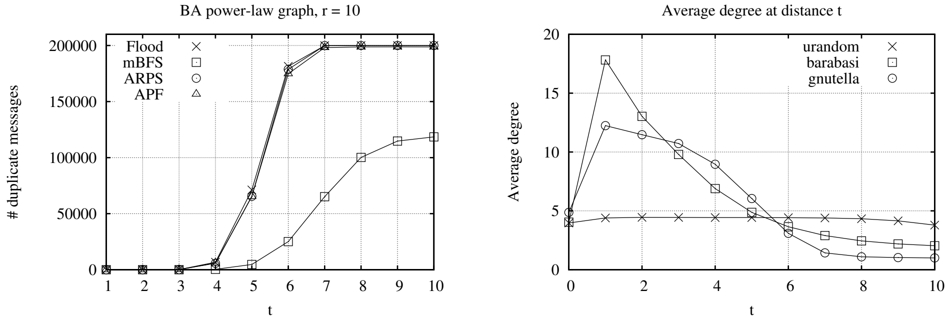
$\overline{d}$	graph	nodes	q	flooding	mBFS	ARPS
4	BA	100000	0.00005	$p_f = 1$	$p_f = 0.5$	$p_f = 0.9$
4	BA	100000	0.001	$p_f = 1$	$p_f = 0.5$	$p_f = 0.7$
4.9	gnutella	26518	0.0037	$p_f = 1$	$p_f = 0.5$	$p_f = 0.7$
23	slashdot	82168	0.00006	$p_f = 1$	$p_f = 0.5$	$p_f = 0.9$

probability of success remains at high levels, the number of duplicate messages increases substantially. This is depicted in Fig. 7(a) for a network of $\overline{d} = 4$, where APF seems to have no difference from pure flooding. The explanation lies on the structure of of this topology that results in varying average node degrees, $\overline{d}_t$, in distance t from the originating node, which are different from the average degree ($\overline{d}$) of the network overall. In Fig. 7(b) we have plotted the average degree of nodes at distance $t = 1, 2, \ldots, 10$ from the originating node as observed in simulation sessions for three different networks: a random graph with $\overline{d} = 4$, a BA power-law graph with $\overline{d} = 4$ and a real Gnutella snapshot with $\overline{d} = 4.9$, which will be discussed below. Unlike the random graph for which the average node degree remains almost constant at any distance t, in the other two networks it varies significantly with t.

Because in power-law networks the vast majority of nodes have small degree and very few nodes are highly connected, the average network degree ($\overline{d}$) is very different from the average degree of the visited nodes at each step of the procedure. If the average node degree, $\overline{d}_t$, in distance t from the originating node was known then our strategy would be applicable by using $\overline{d}_t$ instead of $\overline{d}$ in our model (eq. (3)):

$$n_t = (\overline{d}_t - 1)n_{t-1}\left(1 - \frac{N_{t-1}}{N}\right)p_f(t-1), \tag{5}$$

In order to verify this claim, we performed the following experiment: we determine $\overline{d}_t$, by averaging the degree of nodes at distance t, as seen by the simulation of the pure flooding strategy. We then utilize the measured quantities in (5) and ap-



(a) (b)

Figure 7: (a) Duplicate messages when searching in a BA power-law network of 100,000 nodes, $\overline{d} = 4$ and $r = 10$. (b) Average degree of neighbors at distance t from the originating nodes in different topologies.

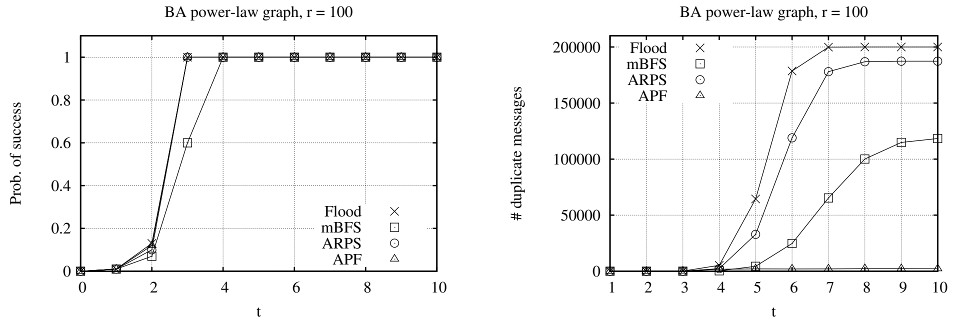
ply APF using (5) instead of (3). The results are given in Figs. 8–9. The simulation parameters are summarized in Table 2.

The simulation results in Figs. 8–9 show the duplicate message reduction when searching for a resource with $r = 100$ and $r = 5$ replicas respectively. Using knowledge about how the average node degree varies with respect to the distance, APF manages to reduce again the search cost as it eliminates the duplicate messages without decreasing the response time or the query success rate.

In Fig. 10–11 we apply the same idea for two "real"-world graphs, namely snapshots of the gnutella p2p and the slashdot social networks, as obtained by the Stanford Network Analysis Platform [19]. More specifically we consider a trace of a 2002 gnutella network with 26518 nodes and 65369 edges (Fig. 10) and and a trace of the Feb. 2009 slashdot network with 82168 nodes (Fig. 11). Other simulations parameters are described in Table 2. As above, we obtain the value for $\overline{d}_t$ form the simulation of full flooding. We performed same experiments over these real topologies. The conclusions remain the same; in any case our approach results in a significant reduction of duplicate messages. As it seen in the figures, it does not affect the probability of success, which is kept at the same levels as in flooding.

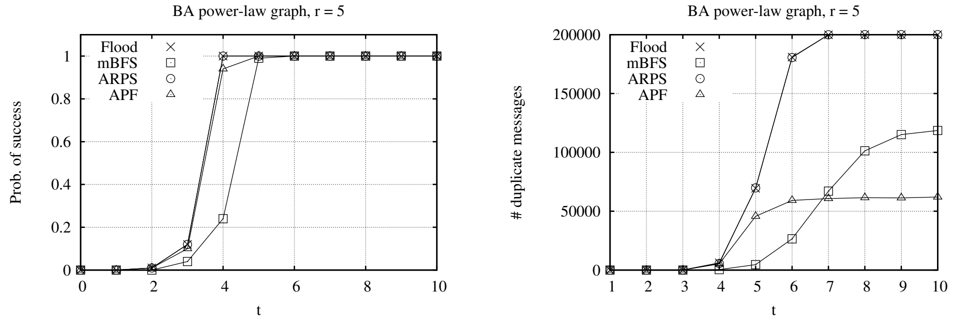
6 Discussion and Future Work

In this work, we present APF, a novel probabilistic flooding strategy that can be employed for query routing in unstructured p2p networks. Each node that receives the query message decides whether to propagate it any further with probability p_f which is based on an estimate of the nodes that are covered at each step, and the popularity of the requested resource. This is in contrast to other strategies that either keep a constant forwarding probability or decrease it obliviously according to the distance from the originating node. We assess experimentally the performance of APF through detailed simulations, in a variety of network topologies which include



(a) (b)

Figure 8: Probability of success (a) and duplicate messages in a BA power-law network of 100,000 nodes and $r = 100$ replicas.



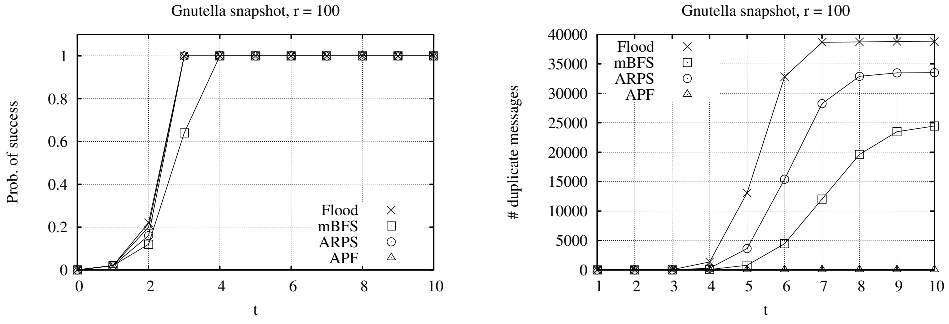
(a) (b)

Figure 9: Probability of success (a) and duplicate messages in a BA power-law network of 100,000 nodes and $r = 5$.

uniform random graphs, power-law graphs and real network snapshots, with or without a clustering coefficient. Comparing it with other known approaches, we show that our strategy results in high success probabilities (almost the same as full flooding) while at the same time enjoying quite low duplicate message counts.

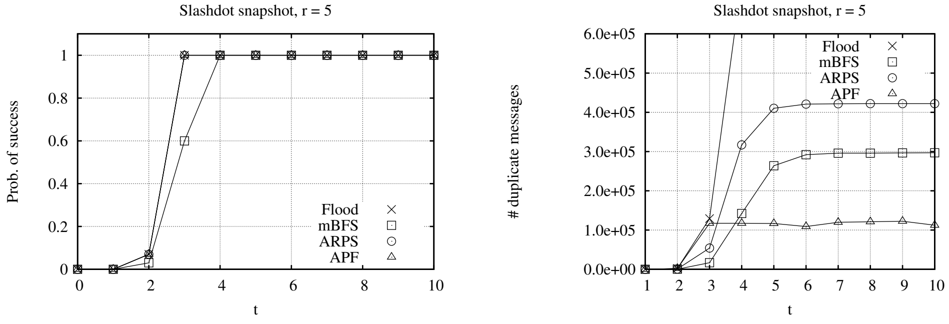
We are currently working on enhancing the applicability of APF in power-law random graphs; we work on determining approximate values for the average node degree, $\overline{d}_t$, in distance t from the originator of the query, both analytically and algorithmically. Another step in this research, is to investigate whether our algorithm can be combined with other existing search techniques. This seems to be possible because in our strategy each node uses a local estimation, independently of the other peers.

As part of our on-going work, we are currently considering highly dynamic networks and studying the impact that the addition and deletion of nodes (churn) in the p2p network may have on the performance of APF. The behavior of each



(a) (b)

Figure 10: Probability of success (a) and duplicate messages in gnutella snapshot of 26518 nodes and $r = 100$ replicas.



(a) (b)

Figure 11: Probability of success (a) and duplicate messages in slashdot snapshot of 82168 nodes and $r = 5$ replicas.

node is determined by its lifetime and the connections which are kept alive with others nodes, i.e. its neighbors. Connections can be removed or replaced depending on new node arrivals and old node departures. Although the recent models of networks evolution presume constant average degree and slowly growing diameter, Leskovec [20], has shown, based on empirical observation, that the average degree is increasing while the average distance between nodes is decreasing and a giant component emerges.

Our approach is based on two fundamental network parameters, the average node degree and the distance from the query initiator. In this paper we have assumed a constant average degree but for the case of dynamic networks we are working on the estimation of the average degree as a function of time. We expect that as the average distance decreases and a giant component covers a significant fraction of nodes, the accuracy of our model increases and the performance of APF improves accordingly. In addition, previous works in mobile ad hoc networks (MANETs), e.g. [21], strengthen our expectation that in dynamic networks, probabilistic flooding seems to be the most suitable search strategy.

References

[1] Banaei-Kashani, F., Shahabi, C.: Criticality-based analysis and design of unstructured peer-to-peer networks as "complex systems". In: Proceedings of the 3st International Symposium on Cluster Computing and the Grid, CCGRID '03, pp. 351–. IEEE Computer Society, Washington, DC, USA (2003)

[2] Bisnik, N., Abouzeid, A.: Modeling and analysis of random walk search algorithms in p2p networks. In: Hot Topics in Peer-to-Peer Systems, 2005. HOT-P2P 2005. Second International Workshop on, pp. 95 – 103 (2005). DOI 10.1109/HOT-P2P.2005.13

[3] Chandra, J., Ganguly, N.: On coverage bounds of unstructured peer-to-peer networks. Advances in Complex Systems **14**(4), 611–633 (2011)

[4] Chandra, J., Shaw, S.K., Ganguly, N.: Analyzing network coverage in unstructured peer-to-peer networks: A complex network approach. In: Networking, pp. 690–702 (2009)

[5] Chang, N.B., Liu, M.: Controlled flooding search in a large network. IEEE/ACM Trans. Netw. **15**(2), 436–449 (2007)

[6] Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. SIGCOMM Comput. Commun. Rev. **32**(4), 177–190 (2002). DOI 10.1145/964725.633043

[7] Crisóstomo, S., Schilcher, U., Bettstetter, C., Barros, J.a.: Analysis of probabilistic flooding: how do we choose the right coin? In: Proceedings of the 2009 IEEE international conference on Communications, ICC'09, pp. 2080–2085. IEEE Press, Piscataway, NJ, USA (2009)

[8] Crisóstomo, S., Udo, S., Christian, B., Joao, B.: Probabilistic flooding in stochastic networks: Analysis of global information outreach. Computer Networks **56**(1), 142 – 156 (2012). DOI 10.1016/j.comnet.2011.08.014

[9] Dimakopoulos, V.V., Pitoura, E.: On the performance of flooding-based resource discovery. IEEE Trans. Parallel Distrib. Syst. **17**(11), 1242–1252 (2006). DOI http://dx.doi.org/10.1109/TPDS.2006.161

[10] Erdös, P., Rényi, A.: On random graphs, I. Publicationes Mathematicae (Debrecen) **6**, 290–297 (1959). URL http://www.renyi.hu/~{}p_erdos/Erdos.html#1959-11

[11] Gaeta, R., Balbo, G., Bruell, S.C., Gribaudo, M., Sereno, M.: A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks. Perform. Eval. **62**(1-4), 1–16 (2005)

[12] Gaeta, R., Sereno, M.: Generalized Probabilistic Flooding in Unstructured Peer-to-Peer Networks. IEEE Trans. Parallel Distrib. Syst. **2**(12), 2055–2062 (2011)

[13] Gkantsidis, C., Mihail, M., Saberi, A.: Hybrid search schemes for unstructured peer-to-peer networks. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 3, pp. 1526 – 1537 vol. 3 (2005). DOI 10.1109/INFCOM.2005.1498436

[14] Gkantsidis Christos, M.M., Amin, S.: Random walks in peer-to-peer networks. In: INFOCOM (2004)

[15] Heath, L.S., Parikh, N.: Generating random graphs with tunable clustering coefficients. Physica A: Statistical Mechanics and its Applications **390**(2324), 4577 – 4587 (2011). DOI 10.1016/j.physa.2011.06.052

[16] Himali, D.R., Prasad, S.K.: Spun: A p2p probabilistic search algorithm based on successful paths in unstructured networks. 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum **0**, 1610–1617 (2011). DOI http://doi.ieeecomputersociety.org/10.1109/IPDPS.2011.316

[17] Kalogeraki, V., Gunopulos, D., Zeinalipour-Yazti, D.: A local search mechanism for peer-to-peer networks. In: Proceedings of the eleventh international conference on Information and knowledge management, CIKM '02, pp. 300–307. ACM, New York, NY, USA (2002). DOI http://doi.acm.org/10.1145/584792.584842

[18] Klemm, A., Lindemann, C., Vernon, M.K., Waldhorst, O.P.: Characterizing the query behavior in peer-to-peer file sharing systems. In: Proc. IMC '04, 4th ACM SIGCOMM conference on Internet measurement, pp. 55–67. Sicily, Italy (2004)

[19] Leskovec, J.: Stanford network analysis project (2011). URL http://snap.stanford.edu/index.html

[20] Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Trans. Knowl. Discov. Data **1**(1) (2007). DOI 10.1145/1217299.1217301

[21] Li, S., Lou, L., Hong, L.: Directional probabilistic broadcast in wireless mobile ad hoc networks. In: Proceedings of the 2013 International Conference on Computational and Information Sciences, ICCIS '13, pp. 1421–1424. IEEE Computer Society, Washington, DC, USA (2013). DOI 10.1109/ICCIS.2013.375

[22] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: ICS, pp. 84–95 (2002)

[23] Makino, N., Arakawa, S., Murata, M.: A flooding method for exchanging routing information in power-law networks. In: APCC 2005, Asia-Pacific Conference on Communications, pp. 812 –816 (2005). DOI 10.1109/APCC.2005.1554175

[24] Newman, M.E.J.: Random Graphs with Clustering. Physical Review Letters **103**(5), 058,701 (2009). DOI 10.1103/PhysRevLett.103.058701

[25] Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E **64**(2), 026,118 (2001)

[26] Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, pp. 151–162. ACM, New York, NY, USA (1999). DOI 10.1145/313451.313525. URL http://doi.acm.org/10.1145/313451.313525

[27] Oikonomou, K., Kogias, D., Stavrakakis, I.: Probabilistic flooding for efficient information dissemination in random graph topologies. Comput. Netw. **54**, 1615–1629 (2010). DOI http://dx.doi.org/10.1016/j.comnet.2010.01.007

[28] Park H. Ratzin R. I., .v.d.S.M.: Peer-to-peer networks: Protocols, cooperation and competition. In: .X.N. C. Zhu Y. Li (ed.) Streaming Media Architectures, Techniques, and Applications: Recent Advances, pp. 262–294. Hershey (2011)

[29] Reka, A., Barabási: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47–97 (2002). URL http://arxiv.org/abs/cond-mat/0106096

[30] Rhea, S., Kubiatowicz, J.: Probabilistic location and routing. In: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, pp. 1248–1257 vol.3 (2002). DOI 10.1109/INFCOM.2002.1019375

[31] Serrano Ángeles, M., Boguñá, M.: Tuning clustering in random networks with arbitrary degree distributions. Phys. Rev. E **72**, 036,133 (2005). DOI 10.1103/PhysRevE.72.036133

[32] Stauffer, A., Barbosa, V.: Probabilistic heuristics for disseminating information in networks. IEEE/ACM Transactions on Networking **15**(2), 425 –435 (2007). DOI 10.1109/TNET.2007.892877

[33] Tsoumakos, D., Roussopoulos, N.: Adaptive probabilistic search for peer-to-peer networks. In: Proceedings of the 3rd International Conference on Peer-to-Peer Computing, P2P '03, pp. 102–. IEEE Computer Society, Washington, DC, USA (2003)

[34] Yang, B., Garcia-Molina, H.: Improving search in peer-to-peer networks. In: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02), ICDCS '02, pp. 5–. IEEE Computer Society, Washington, DC, USA (2002). URL http://dl.acm.org/citation.cfm?id=850928.851859

[35] Zhang, H., Zhang, L., Shan, X., Li, V.: Probabilistic search in p2p networks with high node degree variation. In: ICC '07, IEEE International Conference on Communications, 2007, pp. 1710 –1715 (2007). DOI 10.1109/ICC.2007.286