R. Gaeta, M. Sereno
Generalized probabilistic flooding in unstructured peer-to-peer networks
IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS
(2011) 22
DOI: 10.1109/TPDS.2011.82

Generalized probabilistic flooding in unstructured peer-to-peer networks

Rossano Gaeta, Matteo Sereno

Abstract—In this paper we propose a generalization of the basic flooding search strategy for decentralized unstructured peer-to-peer (P2P) networks. In our algorithm a peer forwards a query to one of its neighbors using a probability that is a function of the number of connections in the overlay network of both. Moreover, this probability may also depend on the distance from the query originator. To analyze the performance of the proposed search strategy in heterogeneous decentralized unstructured P2P networks we develop a generalized random graph (GRG) based model that takes into account the high variability in the number of application level connections that each peer establishes, and the non-uniform distribution of resources among peers. Furthermore, the model includes an analysis of peer availability, i.e., the capability of relaying queries of other peers, as a function of the query generation rate of each peer. Validation of the proposed model is carried out comparing the model predictions with simulations conducted on real overlay topologies obtained from crawling the popular file sharing application Gnutella. Performance of the proposed strategy is investigated in a few example scenarios.

1 Introduction

Peer-to-peer (P2P) paradigm has emerged as new model for distributed networked services and applications. P2P applications have been deployed in many different areas, such as distributed grid computing [1], storage [2], web cache [3], Internet telephony [4], streaming [5], [6], conferencing [7], content distribution [8], [9], and so on. But file sharing applications are perhaps the most popular P2P applications: many different file sharing systems, such as Gnutella, Kazaa, Edonkey, Emule, BitTorrent, exist and collect million of users. These type of applications are characterizing a great fraction of the Internet traffic nowadays and several statistics on IP traffic have recently put in evidence that P2P traffic is starting to dominate the bandwidth in certain segments of the Internet.

In a P2P-based application participants are termed as *peers* and play the dual role of both provider and requester of a service. Services are the location and transfer of (part of) a *resource* that can be owned by several peers thus defining the resource *popularity*. Peers organize themselves in an overlay (logical) network on top of the physical network. Each peer establishes application level connections only to a subset of known peers (its *neighbors*). Management of the overlay network is done at the application level: different management schemes define different classes of P2P networks.

In this paper we consider searching in heterogeneous *decentralized unstructured* P2P networks [10]: peers join and leave the application at their own will in an uncoordinated fashion and a central index for resource location is absent. Each peer is only responsible for maintaining a local index of the resources it owns and it is willing to provide to others. When a peer needs to locate a resource it sends out request messages (*queries*) to its neighbors. There are two main approaches for locating a resource in unstructured decentralized P2P networks: flooding and random walk. In random walk based search strategies peers forward a query message (termed as *walker*) to one randomly chosen neighbor at each step although several walkers can be employed in parallel to increase the probability of successfully locating a resource (*hit probability*). In flooding based search strategies, when a peer requests a resource it sends queries to all its neighbors. This collection of neighbors may then forward the query to their neighbors (excluding, of course, the neighbor that sent the original request). These neighbors may then propagate the query to their neighbors and so on up to a certain predefined maximum level. Hence, resource location is performed by flooding the network with resource-location request packets.

Heterogeneity of Internet users (different hardware, operating systems, application software, connection bandwidth and availability, activity time, etc) reflects on P2P networks as well. It has been found out by several authors [11], [12], [13], [14], [15] that the distribution of peer session times in P2P-based file-sharing applications follows a power-law, i.e., it exhibits high variability. It follows that the number of direct application-level connections to other peers (*the degree of a peer*) of participants show high variability [16], [17], [13], [14], [15]. Furthermore, it has also been found out that resource distribution among peers varies greatly [12], [13], [18], [15]. It follows that taking heterogeneity of P2P-based applications into account is crucial for system design and evaluation. In this paper we develop a mathematical model to analyze the effect of these sources of heterogeneity in P2P networks on the number of messages required to discover a resource and on the hit probability. We also propose and analyze a generalization of the

- *Rossano Gaeta and Matteo Sereno are with the Dipartimento di Informatica, Università di Torino, Torino, Italia.*
E-mail: {rossano,matteo}@di.unito.it

flooding search strategies that exploits the advantages of heterogeneity to decrease the average amount of overhead traffic while increasing the hit probability for a resource.

We consider the problem of analyzing searching techniques in heterogeneous decentralized unstructured P2P networks very important for two reasons:

1) despite the growing popularity of BitTorrent-like P2P protocols unstructured P2P networks, e.g., the current version of Gnutella [19],remain extremely popular these days as witnessed by recent measurements reporting that the number of different participants in snapshots taken in a 7-10 minutes period is in the order of $2,000,000$ peers (corresponding to both ultra-peers and leaves) [17], [15]. Therefore we believe that applicability of the results we obtain is not limited.
2) although most of the traffic generated by P2P applications is for data transfer, search still represents a non-negligible communication and processing burden on peers.

In the model we develop the overlay network is represented by means of Generalized Random Graphs (GRG) [20], [21], [22]. This choice is motivated by the observation that at any point in time a snapshot of the overlay network can be represented by a finite graph of size N where a vertex represents a peer and application-level connections among peers are modeled as edges. Although P2P networks are highly dynamic and result in a constantly and *randomly* changing topology, if we assume that the time scale of search operations is much shorter than the time scale of the P2P network topology evolution, we can reasonably assume that at any instant in time the snapshot of the P2P network topology can be viewed as an instance of a finite graph of size N. Furthermore, the GRGs allow the degree distribution of a randomly chosen node to be described by *any arbitrary* probability distribution thus allowing the inclusion of high variability in the number of application level connections.

Furthermore, we assume a resource is characterized by a popularity defined as the probability that a randomly chosen peer owns a copy of the resource. Again, we define a degree-dependent probability observing that highly connected peers are often attractors to other peers because they are the repository of a large number of resources.

The model also describes the availability of peers, i.e., the probability a peer is not overloaded and it is able to relay queries to (subset of) its neighbors. To this end we represent a peer by a simple finite buffer queue whose loss probability is used to compute availability as a function of the number of connections.

To the best of our knowledge, this is the first paper on analytical models of P2P networks that includes all these heterogeneity issues in the model derivation and analysis as well as the limited processing capacities of peers that result in possible overloading.

Thanks to the model we developed we also propose and analyze in a few scenarios a generalization of the flooding search strategies that we call *generalized probabilistic flooding*. As pointed out by several authors [23], [24], [25], the straightforward flooding strategy suffers from poor granularity, i.e., the amount of message overhead exponentially increases as a function of only one message parameter, named *time-to-live (TTL)*. We propose a probabilistic flooding strategy where a peer decides to forward a query to its neighbors using a probability p_f that is a function of its degree. Furthermore, a peer accepts an incoming query with a probability p_r that is a function of its degree. To reduce the message overhead both probabilities may depend on the distance from the query originator, as well. In this setting a query from a k_f degree peer to one of its k_r degree neighbor is forwarded with probability $p_f(k_f, d) \cdot p_r(k_r, d+1)$ where $d < TTL$ is the distance (in terms of number of hops) of the forwarding peer from the query originator.

The model analysis is carried out by deriving the generating function of the distribution of the number of query messages sent throughout the network starting from a query originator. The generating function is successively refined to account for the probability of successfully finding a resource whose popularity is known. The solution is developed with the simplifying assumption that the number of newly visited peers at each step of the flooding process are described by independent random variables. Equivalently, we assume that the clustering coefficient[1] of the graph representing the overlay network tends to 0 as the size of the overlay network tends to ∞. We test the validity of the independence assumption in Section 4.1 where the model predictions are thoroughly validated against simulations conducted on real snapshots of Gnutella that we gathered by developing a distributed crawler [15] inspired by the work in [17].

This paper differs from our previous work on this subject [27], [28]; the new results of this paper are:

- the definition of a general flooding algorithm that can be parametrized by the distance of the query message from the originator and by the degree of both forwarding and receiving nodes;
- the model exploitation to evaluate in a few scenarios the quantitative impact of heterogeneity of peer characteristics on the performance of flooding based search strategies. The finding is that heterogeneity must be included in models of P2P networks: Section 4.2 shows that the assumption of uniform resource distribution results in an underestimation of the hit probability that is more marked in the low variance degree distributions;
- modeling of the limited processing capacities of peers that results in the possibility for a peer to be

[1] There are several mathematical definitions of what the clustering coefficient of a graph is. Intuitively, it expresses the average probability that two neighbors of a node are neighbors themselves. The interested reader may refer to [26] for a detailed discussion on this topic.

overloaded and thus not able to forward other peers queries;

- the analysis of an example of generalized probabilistic flooding in heterogeneous scenarios.

The paper is organized as follows: Section 2 summarizes previous work on this subject, Section 3 contains the technical part of the paper where the generating functions describing the probability distribution of the performance indexes we consider are derived. Section 4 presents results obtained from the model solution: the validity of the proposed model is discussed by extensively comparing its predictions against simulation on real overlay networks topologies in Section 4.1, some results showing the impact of heterogeneity on the hit probability as well as on the number of messages are presented in Section 4.2, characteristics of the generalized probabilistic flooding in a few scenarios are analyzed in Section 4.3. Section 5 summarizes the paper content and outlines future developments.

2 Related work

Until today, the development of analytical models to gain insight of the behavior of P2P-based applications has mainly focused on homogeneous assumptions on the behavior and characteristics of the application participants. Two noticeable exceptions are the works in [29] and [30] that take into account the bandwidth diversity problem in BitTorrent-like file-sharing applications. These studies deeply differs from ours since in the type of systems they consider there is no need to locate a resource (the location of a tracker process suffices to be able to start downloading a file). Furthermore, none of the following papers has dealt with the limited processing capacities of peers that results in the possibility for a peer to be overloaded and thus not able to forward other peers queries.

Several papers have analyzed the behavior of P2P-based applications by means of analytical models. The paper in [24] explores alternatives (expanding rings and random walks) to the classical flooding search strategies. The authors also evaluate different network topologies and resource replication strategies as to provide better performance. This paper differs from ours since one of its objectives was the analysis of the replica distributions to minimize query load. When alternatives to the classical flooding are analyzed uniform resource distribution is considered and peers availability is neglected. Furthermore, the evaluation of search strategies has been carried out by means of simulation.

The work in [27] exploits generalized random graphs to represent overlay networks but only focuses on simple homogeneous scenarios where peers are always available, resources are uniformly distributed among peers, and flooding-based search strategies are probabilistic but uniform across peers. Furthermore, validation of the model predictions is only done by simulations. This work has been extended in [28] that contains the design and exploitation of a Gnutella crawler as well as an analysis of the measurements obtained with this tool. The paper also develops a simple model of probabilistic flooding that is validated on the snapshots obtained from the crawler.

The analysis of probabilistic flooding to disseminate information in unstructured P2P networks so that global outreach and reduced message overhead are achieved has been the subject of several papers. In [31] and [32] random graph theory is exploited to bound and compute values of the network-wide forwarding probability that ensures global outreach with high probability. In [33] a probabilistic heuristic to disseminate information is defined and analyzed on Poisson and Power-law random graphs. The proposed algorithm is also compared to probabilistic flooding.

The work in [34] exploits the theory of random graphs to prove properties of a generalization of the search that combines flooding and random walks. The authors discuss the properties of normalized flooding on classes of random graphs and evaluate the more general hybrid strategy by simulation on several types of topologies.

The authors of [25] focus on the analysis of resource replication strategies and analyze them under the assumption that flooding stops as soon as a node holding a copy is found. They also analyze replication strategies where replicas are uniformly distributed among peers. In [35], [36] they generalize their results to investigate the relations between the number of replicas of each resource and the query request rate for that resource.

The idea of exploiting peer heterogeneity to achieve better performance is not new, see for instance [37] and [38]. For random walk based strategies the work in [39] introduces a number of local search strategies that utilize high degree nodes in power-law graphs and that have costs scaling sub-linearly with the size of the graph. The authors use GRGs and the generating function analysis technique to demonstrate the utility of these strategies on the Gnutella peer-to-peer network. This paper differs from ours since it is based on the use of random walk; furthermore, it only exploits degree properties of the query receiver and assumes uniform resource distribution.

The work in [23] quantifies the effectiveness of random walks for searching and construction of unstructured P2P networks. It also compares flooding and random walk by simulations on different network topologies where resource are uniformly distributed and peer availability is not considered.

The authors of [40] introduce a scalable searching protocol for locating contents in random networks with heavy-tailed degree distributions. The algorithm is able to find any content in the network with probability one with a time complexity $O(\log N)$ (N is the network size), and a number of messages that scales sub-linearly with respect to N. The analysis of the size of the giant connected component of a random graph with heavy tailed degree distributions under bond percolation is at the heart of their main results.

The paper [41] proposes a search algorithm that exploits k random walkers for resource discovery. Nodes keep information on each query they process per neighbor. This information is then used to probabilistically select a neighbor to forward a walker for a specific object. The procedure continues until all k walkers have terminated. A walker terminates with a success if a node holds a copy of the resource while it ends with a failure if the walker TTL has expired.

The work in [42] proposes a query routing mechanism for unstructured P2P networks where the participant peers build probabilistic routing tables, constructed and maintained through an exchange of updates among immediate neighbors in the overlay. The proposed routing mechanism uses the information of these routing tables to forward search queries. Availability of peers is not included in the analysis and the topology considered for the analysis is a regular random graph.

3 The model

In this section we describe the behavior of peers in the network and we derive the generating function of the probability distribution of the number of queries sent throughout the network starting from a peer that does not have a copy of a resource and that issues a request for it. From this generating function we obtain the average number of query messages as well as the hit probability for a query. Finally, we derive the availability distribution, i.e., the probability that a peer is not overloaded by query traffic.

3.1 Peers' behavior

Queries are originated by peers that set the time-to-live attribute to an integer value denoted as *TTL*. A peer that receives a query decreases the *TTL* by one. If the *TTL* reaches the value zero then the query is not forwarded. Each peer manages a query buffer (whose size we denote as B) that is used to store incoming queries that have to be processed. Queries in the buffer are processed at a rate that we denote as μ. Processing a query involves decreasing its *TTL*, searching through the peer resources to look for a match, and forwarding the query (depending on the *TTL*) to its neighbors according to a particular search algorithm. A query is not forwarded back to the peer that sent it. A peer that finds a match for a query forwards it anyway to increase the hit probability. Queries are enqueued as long as the buffer is not full. Arrival of queries to be forwarded that find the buffer full are discarded (and hence not forwarded). Peers generate their own queries at a given rate λ. Queries originated by a peer always preempt waiting queries in the query buffer and are always forwarded no matter how full the query buffer is. According to this system description, peers may not be able to store an incoming query in their query buffer. The probability that a query can be inserted in the query buffer is called *availability* and it is denoted as a_k for a peer with k connections in the overlay.

When a peer has to forward a query to one of its neighbors it does so with a probability that is a function of the degrees (the number of connections in the overlay) of both and of their distances from the query originator. Distances are expressed in number of hops and for the forwarding peer it must be less than TTL. We assume that the distance of the query originator from itself is equal to 0. It follows that the strategy we propose requires neighbors to periodically exchange only the information on the number of application level connections they established. Therefore our strategies employ an extremely limited form of lookahead that significantly reduces the message overhead in lookahead-based search strategies [43], [44].

The popularity of a resource is expressed as the probability that a randomly chosen peer owns a copy of it. This probability is a function of the peer degree so to model a scenario where resources are non-uniformly allocated. We denote the probability that a copy of the resource is owned by a peer with k connections in the overlay as γ_k.

3.2 Generalized random graphs

We represent the overlay network by means of Generalized Random Graphs (GRG) [20], [21], [22]. GRGs are defined by the degree probability distribution $\{p_k\}$, i.e. the probability that a randomly chosen node has exactly k undirected edges emanating from it. Edges are selected independently and uniformly over the space of possible edges, constrained by the degree distribution. It can be shown that starting from the following two basic generating functions

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k, \quad G_1(x) = \frac{G_0'(x)}{G_0'(1)} \tag{1}$$

it is possible to derive the generating functions for the probability distribution of the number of neighbors at any distance from a randomly chosen node ($G_0'(x)$ denotes the first derivative of $G_0(x)$ with respect to x). Generating function $G_0(x)$ describes the distribution of the number of neighbors (the degree distribution) of a randomly chosen node while $G_1(x)$ describes the degree distribution of the node reached by following one end of a randomly chosen *edge* (with the edge one is following excluded). It is possible to show that the generating function for the number of neighbors two hops away from a randomly chosen node is given by $G_0(G_1(x))$, the number of neighbors three hops away from a randomly chosen vertex is given by $G_0(G_1(G_1(x)))$, and so on. Furthermore, if we consider a randomly chosen node whose degree is equal to k and if we assume that edges are independently marked with probability p_f then the probability that y out of k edges are marked is distributed according to a binomial distribution $B(k, p_f)$.

The generating function of the distribution of the number of marked edges is then obtained as

$$\begin{aligned}\sum_{y=0}^{\infty}\sum_{k=y}^{\infty} p_k \binom{k}{y} p_f^y (1-p_f)^{(k-y)} x^y &= \\ \sum_{k=0}^{\infty} p_k \sum_{y=0}^{k} \binom{k}{y} (xp_f)^y (1-p_f)^{(k-y)} &= \\ \sum_{k=0}^{\infty} p_k (1-p_f+xp_f)^k &= \\ G_0(1+(x-1)p_f).\end{aligned}$$

Please note that the generating functions characterization of the topological properties of GRGs is correct only asymptotically as the size of the network increases (see [20], [21], [22] for details).

3.3 The basic model

The scenario we consider is then characterized by the following probability sets:

- $\{p_k\}$ is the probability distribution describing the degree of a randomly chosen node;
- $\{a_k\}$ is the set of probabilities that represent the availability of peers, i.e., a_k is the probability that a k degree node is not overloaded by query messages;
- $\{\gamma_k\}$ is the set of probabilities that describe the resource allocation, i.e., γ_k is the probability that a k degree node owns a copy of the resource;
- $\{p_f(k_f, d)\}$ where $0 \leq d < TTL$ represents the probability that a k_f degree node whose distance from the query originator is d forwards the query message to its neighbors (whose distance from the query originator is $d+1$).
- $\{p_r(k_r, d)\}$ where $0 < d \leq TTL$ is the probability that a k_r degree node whose distance from the query originator is d agrees to receive an incoming query message (that could be successively discarded if its query buffer is full upon arrival).

Please note that only $\{p_k\}$ is a probability distribution for which it holds that $\sum_{k=0}^{\infty} p_k = 1$; all the other probability sets are not probability distributions.

From the degree distribution $\{p_k\}$ we obtain the average degree of a randomly chosen node as $\overline{k} = \sum_{k=0}^{\infty} kp_k$. Assume we choose a random edge: such an edge arrives at a node with probability proportional to the degree of that node, and the node therefore has a probability distribution of degree proportional to kp_k. In particular, we denote as $e_k = \frac{kp_k}{\sum_{k=0}^{\infty} kp_k}$ the probability that a randomly chosen edge leads to a k degree node.

The fraction of nodes that are potential query originators is given by $q_o = \sum_{k=0}^{\infty} p_k(1-\gamma_k)$. We also need to define the probability that a node that is reached by following one end of a randomly chosen edge and that is d hops away from the query originator agrees to receive a query message (that could be successively discarded if its query buffer is full upon arrival). It is defined as $p_r(d) = \sum_{k=0}^{\infty} e_k p_r(k,d)$.

We define the generating function of the number of messages that a randomly chosen peer that originates a query would send if information about the degree of its neighbors is not exploited as

$$M(x) = \sum_{k=0}^{\infty} \frac{p_k(1-\gamma_k)}{q_o}(1+(x-1)p_f(k,0))^k. \tag{2}$$

A similar quantity is required for a node that is reached by following one end of a randomly chosen edge and that is $d > 0$ hops away from the query originator. This generating function is given by

$$N(x,d) = \sum_{k=0}^{\infty} e_k\{1+[(1+(x-1)p_f(k,d))^{k-1}-1]p_r(k,d)a_k\} \tag{3}$$

where the product $p_r(k,d)a_k$ represents the condition that a node forwards queries only if it agrees to receive it (defined by probabilities $p_r(k,d)$) and its query buffer is not full (described by probabilities a_k). This node would send a query to its neighbors with probability $p_f(k,d)$ without exploiting information on their degree. It would do so to all its neighbors except the one connected to the edge we chose (this is accounted for by the $k-1$ exponent for the power of x in Equation (3)): along this edge the query reached the node and we assume a query is never returned back. Please note that the algorithm in Section 3.1 can be easily adapted not to forward a query if a peer holds a copy of the requested resource. In this case, Equations (2) and (3) should be properly redefined to account for the different search protocol.

By properly combining Equations (2) and (3) we can derive the generating function of the probability distribution of the number of messages sent by a randomly chosen query originator to its neighbors as

$$QM_1(x) = M(1+(x-1)p_r(1)).$$

The generating function of the probability distribution of the number of messages received by nodes two hops away from the query originator is obtained as

$$QM_2(x) = M(N(1+(x-1)p_r(2),1))$$

and, in general, for nodes t hops away from the query originator we obtain

$$QM_t(x) = M(N(N \dots N(1+(x-1)p_r(t),t-1)\dots,2),1)).$$

Assuming that the random variables representing the number of peers that receive a query to forward at each step of the query diffusion process are independent we write the generating function for the total number of nodes that have received a copy of the query out to distance TTL as[2]

$$Q(x,TTL) = \prod_{t=1}^{TTL} QM_t(x), \tag{4}$$

[2] the generating function of the sum of independent random variables is given by the product of the single generating functions.

Starting from Equation (4) it is possible to compute the probability distribution of the total number of messages required to perform a query. It would require the computation of the probabilities $m_s = \frac{1}{s!}Q^{(s)}(0, TTL)$ where $Q^{(s)}$ denotes the s^{th} derivative of function Q with respect to x. In this paper we only focus on the average number of messages sent throughout the P2P network that we define as

$$\overline{m} = Q'(1, TTL). \tag{5}$$

Queries are processed only by nodes that are not overloaded. In this case, the resource might be owned by a k degree node with probability γ_k. We compute the average probability that a node that is reached by following one end of a randomly chosen edge and that is d hops away from the query originator, agrees to receive a query, is not overloaded, and owns a copy of the resource as $p_{own}(d) = \sum_{k=0}^{\infty} e_k a_k p_r(k, d)\gamma_k$. We can derive the generating function of the probability distribution of the number of available neighbors that received the query issued by a query originator and that own a copy of the resource as

$$H_1(x) = M(1 + (x - 1)p_{own}(1)),$$

and, in general, for nodes t hops away from the query originator we obtain

$$H_t(x) = M(N(N \dots N(1+(x-1)p_{own}(t), t-1) \dots, 2), 1)).$$

We define the hit probability for locating a resource among nodes that are d hops away from the query originator as

$$p_{hit}(d) = 1 - H_d(0)$$

and the overall hit probability as

$$p_{hit} = 1 - \prod_{d=1}^{TTL} (1 - p_{hit}(d)). \tag{6}$$

3.4 The complete model

In Section 3.3 we assumed that the set $\{a_k\}$ describing the probability that a k degree node is not overloaded are known. Obviously, this probability depends on several factors such as the query buffer parameters, the query generation and processing rates, the topological features of the overlay networks, and the search algorithm. Therefore we observe that the set $\{a_k\}$ depends on the number of queries that in turn depend on $\{a_k\}$ through Equations (2) and (3). This observation naturally leads to devise a fixed point iteration algorithm for the computation of the set $\{a_k\}$ starting from the other system parameters. The iteration is represented by Algorithm 1 where all dependencies on the probability set $\{a_k\}$ have been highlighted.

Algorithm 1 Fixed point algorithm to compute the probability set $\{a_k\}$ and all performance indexes.

```
Initialize $\{p_k\}, \{\gamma_k\}$ and $TTL$
for $d$ = 1 to $TTL$ do
    Initialize $\{p_f(k, d)\}, \{p_r(k, d)\}$
end for
Initialize $a_k = 1$ for all $k$
repeat
    for $d$ = 1 to $TTL$ do
        Compute $p_{own}(d, \{a_k\}), p_r(d)$
    end for
    Compute $p_{hit}(\{a_k\}), \overline{m}(\{a_k\}), \overline{p}_{loss}(\{a_k\})$
    for all $k$ do
        $\Lambda_k = 0$
        for $d$ = 1 to $TTL$ do
            $\Lambda_k = \Lambda_k + \overline{s}(d, k, \{a_k\})(1 - \overline{p}_{loss}(\{a_k\}))^{d-1}$
        end for
        $\Lambda_k = \Lambda_k \lambda$
        $a'_k = 1 - loss(\Lambda_k, \mu, B)$
        $\epsilon_k = \frac{|a_k - a'_k|}{a_k}$
        $a_k = a'_k$
    end for
until max$\{\epsilon_k\} < \epsilon$
print $p_{hit}(\{a_k\}), \overline{m}(\{a_k\}), \overline{p}_{loss}(\{a_k\})$
```

The key observation is that the higher the number of connections a peer maintains in the overlay network the higher the average number of queries it is asked to relay and the higher the probability these queries will overload it. Therefore we need to compute the average rate of arrival of queries for a k degree node p. Clearly this depends on the rate of query generation of a single peer (i.e., the parameter λ) and on the number of sources that generate queries whose distance from p is less than or equal to TTL. In turn, this number depends on the search algorithm, i.e., probability sets $\{p_f(k, d)\}$ and $\{p_r(k, d)\}$. Assuming an initial value for the probability set $\{a_k\}$ we can compute the average probability of discarding queries throughout the entire overlay network as

$$\overline{p}_{loss} = \sum_{k=0}^{\infty} p_k(1 - a_k). \tag{7}$$

We assume homogeneity of peers and therefore we consider $\overline{p}_{loss}$ as the query discarding probability for any peer of the overlay network.

The second step is to characterize the number of query originators that, due to the particular search algorithm, could involve a k degree peer in the query diffusion process. Consider a randomly chosen k degree node p and the queries originated by its direct neighbors. If a direct neighbor p' of p issues a request for a resource then p agrees to accept the query (that could be successively discarded if the query buffer of p is full upon arrival) with probability $p_r(k, 1)$. If a peer p'' whose distance from p is equal to two hops generates a query then p agrees to accept the query (relayed by an intermediate peer) with probability $p_r(k, 2)$ and so on. In general, the generating function for the number of nodes at distance d that originate queries that could be accepted by p is given by

$$R(x, d, k) = (1 + (x - 1)p_r(k, d))^k. \tag{8}$$

We need a generating function to represent the number of sources for which a node that is reached by following one end of a randomly chosen edge agrees to accept their queries. This generating function is similar to that of Equation (3). In fact, it is given by

$$S(x, d) = \sum_{k=0}^{\infty} e_k\{1+[(1+(x-1)p_r(k, d))^{k-1}-1]p_f(k, d)a_k\} \tag{9}$$

where the product $p_f(k, d)a_k$ represents the fact that the node is not overloaded and has forwarded the queries along the edge we randomly chose. It agrees to receive queries originated by peers that are d hops away with probability $p_r(k, d)$. It would do so to all its neighbors except that connected to the edge we chose (this is accounted for by the $k - 1$ exponent for the power of x in Equation (9)).

To derive the generating function for the number of query originators that could involve a k degree peer in the query diffusion process we must take into account the probabilities that query originators use to decide whether to send a query to their neighbors. In particular, we derive the generating function of the probability distribution of the number of messages actually received by a randomly chosen degree k node from query originator among its direct neighbors as

$$QR_1(x, k) = R(1 + (x - 1)p_f(0), 1, k).$$

The generating function of the probability distribution of the number of messages actually received by a randomly chosen degree k node from query originator among nodes two hops away is given by

$$QR_2(x, k) = R(S(1 + (x - 1)p_f(0), 1), 2, k)$$

and, in general, for nodes t hops away we obtain

$$\begin{aligned} QR_t(x, k) &= R\big(S(S \dots S(1 + (x - 1)p_f(0), 1) \dots, \\ &\quad \dots t - 2), t - 1), t, k\big). \end{aligned}$$

It follows that the average number of sources at distance d for a k degree peer is given by $\overline{s}(d, k) = QR'_d(1, k)$. Each source generates queries at rate λ. Queries originated at distance d from a k degree node must travel along $d-1$ intermediate nodes; all these nodes discard queries with probability $\overline{p}_{loss}$ therefore the arrival rate of queries as seen by a k degree node is equal to $\lambda\overline{s}(d, k)(1-\overline{p}_{loss})^{d-1}$. Since all query originators up to distance TTL provide query arrivals the overall aggregated query arrival rate is given by

$$\Lambda_k = \lambda \sum_{d=1}^{TTL} \overline{s}(d, k)(1 - \overline{p}_{loss})^{d-1}. \tag{10}$$

The last step involves the computation of the new a_k (the availability of a k degree peer) under a query arrival process whose average rate is equal to Λ_k. We assume that a peer can be represented as a $M/M/1/B$ queue. This choice is motivated by the possibility of using a closed form formula for the loss probability thus making the fixed point iteration extremely efficient. On the other hand, the query arrival process to a k degree peer is actually the composition of a large number of independent query generation processes whose limiting behavior can be approximated by a Poisson process as in a $M/M/1/B$ queue. It follows that the new value for the peer availability is given by $a'_k = 1-loss(\Lambda_k, \mu, B)$ where $loss(\Lambda_k, \mu, B)$ denotes the loss probability of a $M/M/1/B$ queue that is obtained as

$$loss(\Lambda_k, \mu, B) = \begin{cases} \dfrac{(1 - \rho_k)\rho_k^B}{1 - \rho_k^{B+1}}, & \text{if } \rho_k < 1, \\ 1.0, & \text{otherwise} \end{cases}$$

where $\rho_k = \frac{\Lambda_k}{\mu}$.

Once a new set of values for the peers availability is computed the iteration repeats and stops when the maximum relative error between successive iterations falls below a predefined accuracy threshold ϵ.

4 Results

In this section we present validation results to evaluate the accuracy of Equations (5) and (6). We compare their outcomes with results obtained from simulations on real overlay topologies captured by crawling the highly popular P2P-based file sharing applications Gnutella. We also exploit the model to assess the impact of the search algorithm on network congestions as well as the effect of heterogeneity on the performance of flooding-based search strategies.

4.1 Model validation

We compared the predictions based on the numerical solution of Equations (5) and (6) with those obtained from simulations. To keep the complexity of simulation experiments low we do not simulate the message exchange dynamics; rather we assume all peers are not overloaded and always relay query messages to neighbors according to the chosen search algorithm.

The simulator employs standard statistical procedures to estimate 95% confidence intervals for $\widehat{msg}$ and $\hat{p}_{hit}$, the simulated average number of queries and hit probabilities, respectively. We consider N_{exp} overlay topologies: each topology is used to obtain one realization of $\widehat{msg}$ and $\hat{p}_{hit}$. The i^{th} realization is obtained in the following way:

- in the initialization phase, the i^{th} overlay topology (a graph instance) comprising N_i nodes is read from an input file.
- For each k-degree node probability γ_k is used to randomly set the *hold_resource* flag to true. We denote as $\mathcal{N}_q$ the subset of nodes that do not have a copy of the resource and let N_q denote its cardinality.

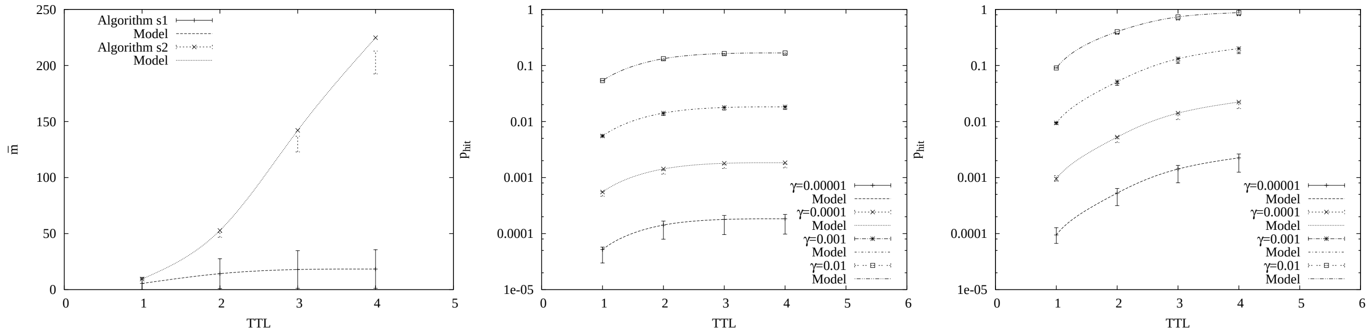

Fig. 1. Comparison of Equations (5) (left graph) and (6) (middle graph refers so $s1$ and right graph refers to $s2$) against simulations on the Gnutella ultra-peer overlay networks.

- each node $r \in \mathcal{N}_q$ is considered as the root of a modified breadth-first visit whose depth is equal to TTL. During the graph visit a k_r-degree node connected to a k_f-degree available node is visited with a probability given by $p_f(k_f, d) \cdot p_r(k_r, d+1)$ where $d < TTL$ is the distance from r. If we denote as $msg_r(d)$ the number of nodes visited that are d hops away from r and let $msg_r = \sum_{d=1}^{TTL} msg_r(d)$ denote the overall number of nodes visited starting from r then we compute $\widehat{msg}^{(i)} = \frac{1}{N_q} \sum_{r \in \mathcal{N}_q} msg_r$.
- we also compute $\hat{p}_{hit}^{(i)}$ as the average fraction of the msg_r nodes that own a copy of the requested resource.

Inspired by [17], we designed and implemented a distributed crawler of the Gnutella overlay network to gather snapshots of a real system to use to validate our model [15]. The average time for the crawler to collect one snapshot is about ten minutes. For our model validation we considered $N_{exp} = 30$ snapshots whose size ranged from 253,625 to 339,822 ultra-peers. All results have been computed for $TTL = 4$ to limit the exponential growth of the CPU time required to obtain estimates from the simulator. In fact, simulations required about an hour to complete while the numerical model solution is almost instantaneous. Simulations with $TTL > 4$ require considerably more CPU time while the model solution complexity is practically constant.

All results that we present for the model validations have been obtained by classifying peers as low, average, and high degree peers. The possible degree-based partitions are countless but we decided to adopt the following: we were inspired by the Gnutella protocol specification [19] that suggests that each ultra-peer should connect to $5-30$ other ultra-peers therefore we consider as low degree peers those whose degree is less than 5, average peers those that keep from 5 to 30 connections, and high degree peers all the remaining. We considered uniform resource distributions ranging from $\gamma = 10^{-5}$ to $\gamma = 10^{-2}$. Table 1 summarizes the main simulation parameters for the model validation.

parameter	value
N_{exp}	30
TTL	4
γ	$[10^{-5}, 10^{-2}]$

TABLE 1
System parameters for the simulations.

We consider two search algorithms that exploit both peers degree and distance from the query originator: the first strategy (*s1*) is defined by

$$p_f(k, d) = \begin{cases} 0.5^d, & \text{if } 0 < k < 5, \\ 0.75^d, & \text{if } 5 \leq k \leq 30 \\ 1.0^d, & \text{otherwise} \end{cases}$$

and

$$p_r(k, d) = \begin{cases} 0.2^d, & \text{if } 0 < k < 5, \\ 0.35^d, & \text{if } 5 \leq k \leq 30 \\ 0.5^d, & \text{otherwise} \end{cases}$$

while the second strategy (*s2*) is defined as

$$p_f(k, d) = \begin{cases} 1.0^d, & \text{if } 0 < k < 5, \\ 0.75^d, & \text{if } 5 \leq k \leq 30 \\ 0.5^d, & \text{otherwise} \end{cases}$$

and

$$p_r(k, d) = \begin{cases} 0.4^d, & \text{if } 0 < k < 5, \\ 0.6^d, & \text{if } 5 \leq k \leq 30 \\ 0.8^d, & \text{otherwise} \end{cases}$$

To test the accuracy of the model we developed we considered both the case of a strategy where the probability of forwarding queries increases as the node degree increases (*s1*) and a strategy that corresponds to the case where this probability decreases for highly connected nodes (*s2*).

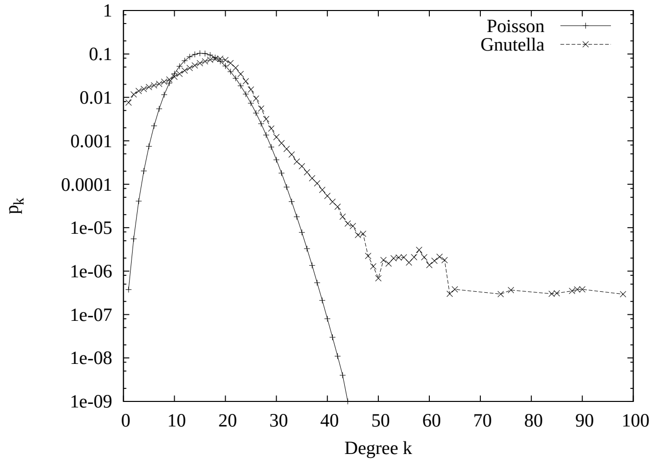

Fig. 2. Degree distributions $\{p_k\}$ for the model exploitation.

For the validation on snapshots of Gnutella we computed the empirical degree distribution by averaging the degree distribution of the single snapshots, i.e., $\frac{1}{N_{exp}} \sum_{i=1}^{N_{exp}} p_k^{(i)}$ where $p_k^{(i)}$ is the fraction of nodes of the i^{th} snapshot whose degree is equal to k. We denote as $\{p_k^{gnu}\}$ the empirical degree distribution for the Gnutella ultra-peer overlay network whose average is equal to 15.79. We then used $\{p_k^{gnu}\}$ as input to the model we developed and we computed the values of Equations (5) and (6).

Figure 1 depicts the comparison between the model prediction and the simulation outcome. We can observe that the accuracy of the model predictions is high if compared to simulation of search strategies on topologies obtained from measurements of the Gnutella ultra-peer overlay network. The model seems to slightly overestimate the number of messages for $TTL = 4$ in the case of strategy $s2$. The reason is that the model is accurate under the assumption of $N \rightarrow \infty$ we believe that considering large values for TTL makes the model predictions less accurate with respect to simulation estimates due to the finite size of the snapshots we use. Nevertheless, we believe the validation results we obtained are very important and confirm the accuracy of the model predictions even in this very tough setting for any analytical model.

Varying the resource distribution would only change the values of p_{hit}: we performed several experiments using the two strategies we considered and obtained very close matching for the hit probabilities. We also obtained very good agreement for several other search strategies, i.e., functions $p_f(k,d)$ and $p_r(k,d)$. Also in some of these cases, the model slightly overestimates the total number of messages for large values of TTL.

4.2 Model exploitation

In this section we evaluate the impact of heterogeneity of the number of connections and the resource distribution among peers on the performance of simple probabilistic flooding ($p_f(k,d) = p_f = 0.5$, $p_r(k,d) = 1$). We set $B = 100$ and $\mu = 1$ for the query buffer management.

To this end we consider the topology we used for validation that we obtained from crawling the Gnutella network (degree distribution $\{p_k^{gnu}\}$) and a topology characterized by a Poisson degree distribution whose average is equal to the average of $\{p_k^{gnu}\}$. Figure 2 shows the two distributions we considered.

We analyze a particular probability set describing the distribution of resources among peers $\{\gamma_k\}$ and compute the value of p_{hit}, $\overline{m}$, and $\overline{p}_{loss}$ in the case of non-uniform distribution and in the case of uniform distribution whose average probability value is $\overline{\gamma} = \sum_{k=1}^{\infty} \gamma_k p_k$. In particular, we consider

$$\gamma_k = \begin{cases} 10^{-5}, & \text{if } 0 < k < 5, \\ 10^{-4}, & \text{if } 5 \leq k \leq 20 \\ 10^{-3}, & \text{otherwise} \end{cases}$$

We denote this resource distributions as rd_1 and we obtain $\overline{\gamma}^{Poisson} = 0.000098$ and $\overline{\gamma}^{gnu} = 0.000100$.

Figure 3 depicts the hit probability (Equation (6), left graph), the average number of messages (Equation (5), middle graph), and the average query discard probability (Equation (7), right graph) for increasing query generation rates λ. First of all we observe that the overlay network reaches a congestion point for a particular value of λ. This is highlighted by the value of p_{hit} that drops to very small values after the query generation rate of each peer exceeds the critical value $\lambda = 0.0001$. This phenomenon can be explained by observing that when λ increases the overall query arrival rate for k degree peer Λ_k increases until the query buffer load factor ρ_k reaches the value 1. This is the symptom that the peer is overloaded and all queries are discarded (hence not forwarded). High degree peers are the first to become overloaded and this makes the hit probability decrease because in rd_1 the higher the degree the higher the value of γ_k. Furthermore, when peers with a large number of connections stop functioning as query forwarder the hit probability decreases since a large number of peers is not probed.

The second observation is that the average number of messages in the Gnutella case is higher than the Poisson case before congestion occurs. This is due to the different variance of the two degree distributions despite the same average value. In fact, the Gnutella distribution shows a tail that extends up to the value $k = 98$ while the Poisson distribution actually is cut at $k = 44$. On the other hand, when congestion has set in the Gnutella network is less resilient to the overloading of the mostly connected nodes that reduces the average number of circulating queries. This also reflects on the hit probability: before congestion occurs the hit probability in Gnutella network is greater than the hit probability in the Poisson case (0.76 vs 0.51 in the heterogeneous case and 0.40 vs 0.34 in the homogeneous case) but during congestion the opposite is true.

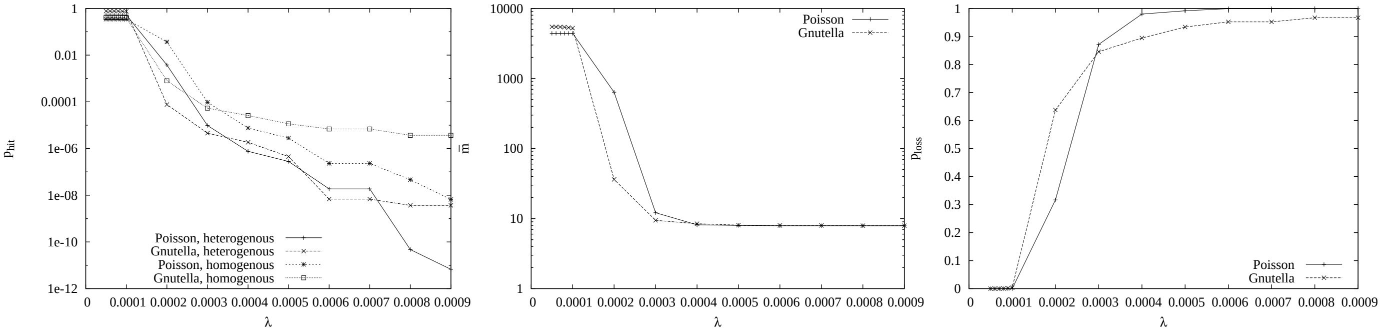

Fig. 3. Values of p_{hit} (left graph), $\overline{m}$ (middle graph), and $\overline{p}_{loss}$ (right graph) for $TTL = 4$ and increasing query generation rates.

The third point is the behavior of $\overline{p}_{loss}$; as λ increases also the average query discarding probability increases. From Equation (10) we observe that Λ_k increases because λ increases. In this regime the network reduces to a set of un-cooperating peers; most of them cannot relay others queries but do generate their own queries that are not able to spread throughout the network that is mostly populated by overloaded nodes.

The last observation is for model developers: assuming a homogeneous distribution model for the resource replication leads to an underestimation of the hit probability with respect to the heterogeneous case before congestion occurs (at least in the case of rd_1). This is an important issue that must be taken into account when developing models of a real system.

As a final remark, a similar behavior has been observed for other values of TTL (we tried $1 \leq TTL \leq 6$) and for other resource distributions where the higher the degree of the peer the higher the probability of storing a replica of the resource. When we consider values of $TTL > 4$ than congestion shows up for much lesser values of λ.

4.3 An example of generalized probabilistic flooding

In Section 4.2 we made several interesting observations on the behavior of the network as the query generation rate increases. In this section we provide an evaluation of a particular search algorithm in case of heterogeneous resource allocation and the Gnutella topology. A complete characterization of the search algorithms would require a sensitivity analysis where a wide spectrum of possibilities for all the probability distributions is considered. Here we limit our analysis to a few scenarios and we leave a thorough investigation for future developments. The scenario we consider is the same analyzed in Section 4.2. Here we consider a search algorithm defined as

$$p_f(k,d) \quad = \quad \begin{cases} 1.0, & \text{if } d = 0, \\ min(1, \frac{k_f}{k}), & \text{otherwise} \end{cases}$$

and

$$p_r(k,d) \quad = \quad \begin{cases} 1, & \text{if } d \leq 1, \\ min(1, \frac{k_r}{k}), & \text{otherwise} \end{cases}$$

The rationale behind this definition is the following: it must not be possible that a query originator ends up with no query messages sent to its direct neighbors, i.e., $p_f(k,d) = 1$ for any k and for $d = 0$. On the other hand, it must be assured that the direct neighbors of a query originator agree to receive the query, i.e., $p_r(k,d) = 1$ for any k and for $d \leq 1$. This constraint is necessary since otherwise there is a nonzero probability that a query originator does not spread a query for the requested resource. The values of k_f and k_r define the average number of queries forwarded (agreed to receive) by a k degree peer. This search algorithm aims at limiting the number of queries forwarded and received by high degree peers while guaranteeing the low degree ones a high utilization of their relaying capacity.
Figure 4 depicts the hit probability (Equation (6), left graph), the average number of messages (Equation (5), middle graph), and the average query discard probability (Equation (7), right graph) for increasing query generation rates λ and for $k_r = 15$ that is less than the average degree of the Gnutella topology, i.e., 15.79. It can be observed that the network reaches congestion for all the values of k_f we considered. As k_f increases the critical value for λ decreases thus accelerating the onset of congestion. If the algorithm must be designed to sustain a maximum value for the query generation rate then the choice of the maximum k_f can be carried out by exploiting the model we developed. The modeling framework we defined could also be used to design a search algorithm that meets constraints on the maximum average number of messages ($\overline{m}_{max}$) and on the minimum hit probability p_{min}. In our example, it could be used to derive the set $\{(k_f, k_r, TTL) : p_{hit} \geq p_{min} \wedge \overline{m} \leq \overline{m}_{max}\}$. The choice of the optimal values can be done based on the maximum value of λ before congestion

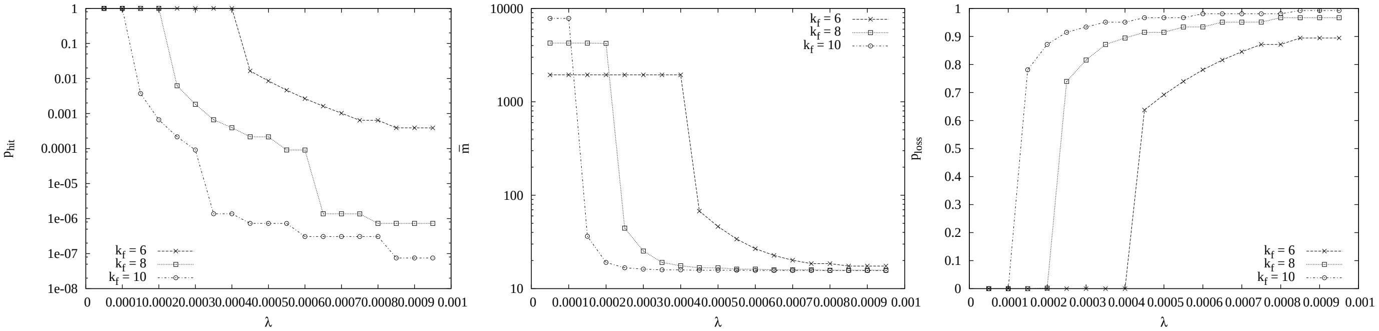

Fig. 4. Values of p_{hit} (left graph), $\overline{m}$ (middle graph), and $\overline{p}_{loss}$ (right graph) for $TTL = 4$ and increasing query generation rates.

kicks in.

5 Conclusions and Future developments

In this paper we analyzed the impact of heterogeneity in P2P-based applications on the number of queries sent throughout the network by peers that request a resource and on the hit probability for the search process. We also analyzed the congestion of the network when the search algorithm is able to overload peers with limited processing capacities. To this end, we developed an analytical model exploiting generalized random graphs to represent the overlay network and incorporating the dependence of peers availability and non-uniform resource distribution by considering probabilities that depend on the nodes degree. We also exploited a simple queuing model to compute the peers availability as a function of the search algorithm. We thoroughly validated the model that showed good agreement with the predictions obtained by simulations on real overlay networks obtained from crawling a popular P2P-based file-sharing applications. To this end, we developed a distributed crawler inspired to previous work on this subject that is able to gather Gnutella 2 snapshot in a few minutes.

We observed interesting behavior of a simple probabilistic flooding algorithm that leads the network to congestion. We also showed that neglecting heterogeneity leads to rather different results even in this simple settings. Furthermore, we provided an example of definition of a complex search algorithm that could be easily analyzed by means of our techniques to find optimal parameters setting.

Future developments of the current work are currently underway: first of all we are working to obtain a complete characterization of the strategy. It requires a sensitivity analysis where a wide spectrum of possibilities for all the probability distributions must be considered. We are also working to extend the model to avoid nodes that have the requested resource to continue flooding the query. Another natural step of the current research is to conduct a delay analysis of generalized probabilistic flooding. To this end, the $M/M/1/B$ queuing model we defined can be easily exploited. Furthermore, we want to extend the model to include different classes of nodes, e.g., to represent the different ISPs the peers belong to in order to quantify the search traffic through an ISP peering point, and to model the correlations structure among nodes degree, i.e., the fraction of edges that connect degree k nodes to degree k' nodes.

References

[1] "Entropia," Tech. Rep., http://www.entropia.com.

[2] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-Area Cooperative Storage with CFS," in *Proc. of the 18-th ACM Symposium on Operating Systems Principles (SOSP '01)*, Banff, Canada, 2001.

[3] S. Iyer, P. Rowstron, and P. Druschel, "Squirrel: a Decentralized Peer-to-Peer Web Cache," in *Proc. of ACM Symposium on Principles of Distributed Computing (PODC '02)*, Monterey, CA, USA, 2002.

[4] "Global Index (GI)," Tech. Rep., URL `http://www.skype.com/skype_p2pexplained.htm`.

[5] S. Iyer, P. Rowstron, and P. Druschel, "SplitStream: High-Bandwidth Multicast in Cooperative Environments," in *Proc. of the 19-th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, USA, 2003.

[6] X. Zhang, J. Liu, B. Li, and T. P. Yum, "DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming," in *Proc. IEEE Infocom 2005*. Miami, FL, USA: IEEE Comp. Soc. Press, 2005.

[7] Y.-H. Chu, S. Rao, , and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, 2002, special Issue on Networking Support for Multicast.

[8] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An Efficient Mechanism for Content Distribution in a Peer-to-Peer (P2P) Network," Microsoft Research, MSR-TR-2004-100, Tech. Rep., 2004.

[9] B. Cohen, "BitTorrent protocol specification," in *First Workshop on Economics of Peer-to-Peer Systems (P2P '03)*, Berkeley, CA, USA, June 2003.

[10] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-Peer File Sharing System," in *Proceedings of INFOCOM 2003*, San Francisco, USA, Apr 2003.

[11] D. Stutzbach and R. Rejaie, "Characterizing Churn in Peer-to-Peer Networks," University of Oregon, CIS-TR-2005-03, Tech. Rep., 2005.

[12] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 219–232, Apr 2004.

[13] P. K. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19)*, Bolton Landing, NY, USA, Oct 2003.

[14] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts," *Multimedia Systems Journal*, vol. 8, no. 5, Nov 2002.

[15] R. Bolla, R. Gaeta, A. Magnetto, M. Sciuto, and M. Sereno, "A measurement study supporting p2p file-sharing community models," *Computer Networks*, vol. 53, no. 4, pp. 485 – 500, 2009.

[16] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network," in *In proceedings of IEEE 1st International Conference on Peer-to-peer Computing (P2P2001)*, Linkoping Sweden, Aug 2001.

[17] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems," in *Proc. of the ACM SIGCOMM Internet Measurement Conference*, Berkeley, CA, USA, 2005.

[18] S. Zhao, D. Stutzbach, and R. Rejaie, "Characterizing Files in the Modern Gnutella Network: A Measurement Study," in *Proc. of SPIE/ACM Multimedia Computing and Networking*, San Jose, CA, USA, 2006.

[19] T. Klingberg and R. Manfredi, "Gnutella 0.6," http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html, Tech. Rep.

[20] M. E. J. Newman, "The spread of epidemic disease on networks," *Physical Review E*, vol. 66, 2002.

[21] ——, "The structure and function of networks," *Computer Physics Communications*, vol. 147, pp. 40–45, 2002.

[22] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, 2001.

[23] C. Gkantsidis, M. Mihail, and A. Saberi, "Random Walks in Peer-to-Peer Networks," in *IEEE INFOCOM 2004*, Hong Kong, China, Mar 2004.

[24] Q. Lv, E. Cohen, K. Li, and S. Shenker, "Search and Replication in unstructured peer-to-peer network," in *International Conference on Supercomputing (ICS02)*, 2002.

[25] S. Tewari and L. Kleinrock, "Analysis of Search and Replication in Unstructured Peer-to-Peer Networks," in *Proc. 2005 ACM SIGMETRICS Conference*, Banff, Canada, Jun (6-10) 2005, poster Presentation.

[26] M. Angeles Serrano and M. Boguna, "Tuning clustering in random networks with arbitrary degree distributions," *Physical Review E*, vol. E 72, 2005.

[27] R. Gaeta, G. Balbo, S. Bruell, M. Gribaudo, and M. Sereno, "A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks," *Performance Evaluation*, vol. 62, no. 1–4, pp. 1–16, 2005, proceedings of the conference *Performance 2005*.

[28] R. Gaeta and M. Sereno, "On the evaluation of flooding-based search strategies in peer-to-peer networks," *Concurr. Comput. : Pract. Exper.*, vol. 20, no. 6, pp. 713–734, 2008.

[29] F. Lo Piccolo, G. Neglia, and G. Bianchi, "The Effect of Heterogeneous Link Capacities in BitTorrent-Like File Sharing Systems," in *Proceedings of International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P 2004)*, Volendam, The Netherlands, 2004.

[30] F. Clévenot-Perronnin, P. Nain, and K. W. Ross, "Multiclass P2P networks: Static resource allocation for service differentiation and bandwidth diversity," *Performance Evaluation*, vol. 62, no. 1–4, pp. 32–49, 2005, proceedings of the conference *Performance 2005*.

[31] K. Oikonomou and I. Stavrakakis, "Performance analysis of probabilistic flooding using random graphs," in *Proc. First International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2007)*, Helsinki, Finland, June 2007.

[32] S. Crisostomo, U. Schilcher, C. Bettstetter, and J. Barros, "Analysis of probabilistic flooding: How do we choose the right coin?" in *Proc. IEEE International Conference on Communications*, Dresden, Germany, June 2009.

[33] A. O. Stauffer and V. C. Barbosa, "Probabilistic heuristics for disseminating information in networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 425–435, 2007.

[34] C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid Seach Schemes for Unstructuired Peer-to-Peer Networks," in *Proc. IEEE Infocom 2005*. Miami, FL, USA: IEEE Comp. Soc. Press, 2005.

[35] S. Tewari and L. Kleinrock, "Optimal Search Performance in Unstructured Peer-to-Peer Networks With Clustered Demands," in *Proc. 2006 ICC Conference*, Istanbul, Turkey, June 2006.

[36] ——, "Proportional Replication in Peer-to-Peer Networks," in *Proc. IEEE Infocom 2006*, Barcelona, Spain, Apr (23-29) 2006.

[37] Y. Chawathe, S. Ratnasamy, B. L., N. Lanhaui, and S. Shenker, "Making Gnutella-like P2P Systems Scalable," in *Proc. of ACM Sigcomm 2003*, Karlsruhe, Germany, Aug 2003.

[38] Q. Lv, S. Ratnasamy, and S. Shenker, "Can Heterogeneity Make Gnutella Scalable?" in *Proc. of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002)*. Cambridge, MA, USA: Springer, LNCS no. 2429, 2002.

[39] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in Power-Law Networks," *Physical Review E*, vol. 64, 2001.

[40] N. Sarshar, V. Roychowdury, and P. Boykin, "Percolation search algorithm, making unstructured p2p networks scalable," in *Proc. of the Fourth IEEE P2P 2004*. Zurich, Switzerland: IEEE Computer Society Press, 2004.

[41] D. Tsoumakos and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," in *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*. Washington, DC, USA: IEEE Computer Society, 2003, p. 102.

[42] A. Kumar, J. Xu, and E. W. Zegura, "Efficient and Scalable Query Routing for Unstructured Peer-to-Peer Networks," in *Proc. IEEE Infocom 2005*. Miami, FL, USA: IEEE Comp. Soc. Press, 2005.

[43] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," in *Proc. of the ACM SIGCOMM*, Pittsburg, PA, USA, Aug 2002.

[44] M. Mihail, A. Saberi, and P. Tetali, "Random Walks with Lookahead in Power Law Random Graphs," Georgia Institute of Technology, Atlanta, GA, USA, Tech. Rep., 2004.

Rossano Gaeta Rossano Gaeta was born in Torino in 1968. He received a degree in Computer Science in 1992 and a Ph.D. degree in Computer Science in 1997 both from the University of Torino. He is currently an associate professor at the Computer Science Department of University of Torino. He has co-authored 60 papers published in international journals and presented in leading international conference and he participated to the program committee of several conferences. He has been recipient of the Best Paper Award at the following conferences: 14-th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2006, and 26th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation (PERFORMANCE), 2007. His research interests include the use of formal models for dependability analysis of time critical systems, the performance evaluation of peer-to-peer computing systems and large scale sensor networks, and the use of rateless codes in peer-to-peer applications.

Matteo Sereno Matteo Sereno was born in Nocera Inferiore, Italy. He received his Laurea degree in Computer Science from the University of Salerno, Italy, in 1987, and his Ph.D. degree in Computer Science from the University of Torino, Italy, in 1992. He is currently full professor at the Computer Science Department of the University of Torino. His current research interests are in the area of performance evaluation of computer systems, communication networks, peer-to-peer systems, sensor networks, queueing networks and stochastic Petri net models.