Interconnection of large scale unstructured P2P networks: modeling and analysis

Vincenzo Ciancaglini[1], Rossano Gaeta[2], Riccardo Loti[2], and Luigi Liquori[1]

[1] LogNet Team, INRIA Sophia Antipolis Méditerranée, Sophia Antipolis, France
first.last@inria.fr,
[2] Dipartimento di Informatica, Università di Torino, Torino, Italy
first.last@di.unito.it

Abstract. Interconnection of multiple P2P networks has emerged as a viable solution to increase system reliability and fault-tolerance as well as to increase resource availability. In this paper we consider interconnection of large scale unstructured P2P networks by means of special nodes (called *Synapses*) that are co-located in more than one overlay. Synapses act as *trait d'union* by forwarding a query to all the P2P networks they belong to. Modeling and analysis of the resulting interconnected system is crucial to design efficient and effective search algorithms and to control the cost of interconnection. To this end, we develop a generalized random graph based model that is validated against simulations and it is used to investigate the performance of search algorithms for different interconnection costs and to provide some insight in the characteristics of the interconnection of a large number of P2P networks.

Keywords: interconnection; peer-to-peer; modeling; random graphs.

1 Introduction

The last decade has seen the rise of peer to peer networks with a variety of applications, such as file sharing, resource lookup, real time services (VoIP and P2P-TV), up to the most recent research in SmartGrids. Common issues which affect all P2P systems, such as scalability, fault tolerance and security, arise from the different peculiarities each class of applications might expose. Increasing the locality properties of such systems, be it geographical, semantic, network, or social-based locality, is one of the most valued approaches to face such challenges: by grouping together peers representing users, and increasing their connections with one another, one can improve scalability, fault tolerance, and security (consider the possible creation of a "circle of trust" amongst nearby peers).

In this paper, we consider the interconnection of large-scale unstructured P2P networks by means of special nodes called *Synapses* [1], which are co-located in more than one network, and act as *connectors* by sending or forwarding a query to some or all the P2P networks they belong to. Modeling and analysis of the resulting interconnected system is crucial to design efficient and effective search algorithms and to control the cost of network interconnection. Yet, simulation and/or prototype deployment based analysis can be very difficult - if not impossible - due to the size of each component (we consider large scale systems that can be composed of millions of nodes) and to the complexity arising from the interconnection of several such complex systems.

Our contribution

To overcome this strong limitation, we develop a generalized random graph based model to represent the topology of one unstructured P2P network, the partition of nodes into Synapses, the probabilistic flooding based search algorithms, and the resource popularity. We validate our model against simulations and prove that its predictions are reliable and accurate. We use the model to investigate the performance and the cost of different search strategies in terms of the probability of successfully locating at least one copy of the resource and the number of queries as well as the interconnection cost. We also gain interesting insights on the dependency between interconnection cost and statistical properties of the distribution of Synapses. Finally, we show that thanks to our model we can analyze the performance of a system composed of a large number of P2P networks. To the best of our knowledge, this is the first paper on model-based analysis of interconnection of large scale unstructured P2P networks[3].

The paper is organized as follows: Section 2 describes our system, Section 3 presents the mathematical derivation of the generalized random graph model we develop, Section 4 contains model validation through simulation, as well as model exploitation to study the performance of three search algorithms, Section 5 discusses related works, and in Section 6, we draw conclusions and outline ongoing activities that extend the current work.

2 System description

In this paper, we focus on unstructured P2P networks where peers organize into an overlay network by establishing application level connections among them. The topological properties of an overlay network are represented by the number of connections of any of its participants. To this end, we describe an overlay by means of the degree distribution $\{p_k\}$ that can be interpreted as the probability that a randomly chosen peer has k connections in the overlay ($\sum_{k=1}^{\infty} p_k = 1$).

We consider a set of X unstructured P2P networks that are interconnected thanks to a subset of peers that belong to multiple overlays (these special peers are denoted as *Synapses*). Any peer may then belong to $i \in \{1, \ldots, X\}$ overlays: we denote i as the *Synapse degree* of a peer. The interconnected system is then described by $\{s_i\}$ ($i \in \{1, \ldots, X\}$) where s_i is the fraction of peers belonging to i overlays ($\sum_{i=1}^{X} s_i = 1$).

[3] This paper is the full version of a two pages poster paper presented in [2]

Table 1: Paper notation.

Parameter	Description
X	Number of interconnected P2P networks.
p_k	Fraction of peers with k connections in an overlay.
s_i	Fraction of peers belonging to i overlays.
$p_f(i)$	Probability to forward a query for peers that belong to i overlays.
α	Average fraction of nodes owning a copy of a resource
TTL	Query time-to-live.

The search algorithm we consider is *flooding-based*. A peer starting a search sends *queries* to a randomly chosen subset of its one-hop neighbors. These nodes forward the queries to a randomly chosen subset of their one-hop neighbors, excluding the query originator, and so on until the maximum number of allowed hops, i.e. the query time-to-live (TTL). We also consider a variation of this search algorithm where a query is not forwarded by peers that own a copy of the resource. We focus on probabilistic versions of this general algorithm where any peer flips a coin before sending or forwarding a query to a specific neighbor. We allow the weight of this coin to be dependent on the Synapse degree of a peer; hence, a peer that belongs to i overlays sends/forwards a query to a particular neighbor with probability $p_f(i)$ $(i \in \{1, \ldots, X\})$. Please note that $\{p_f(i)\}$ $(i \in \{1, \ldots, X\})$ *is not* a probability distribution hence in general $\sum_{i=1}^{X} p_f(i) \neq 1$.

The goal of a search is to localize at least one resource related to the key we are looking for. There could be more replicas of the same resource hosted by different peers for two reasons: a resource is popular and/or is owned by peers located in different P2P networks. We represent resource popularity by $0 \leq \alpha \leq 1$, the average fraction of nodes that globally hold a copy of a given resource, and interpret it as the probability that a randomly chosen peer owns a copy of the resource.

All of the notation is summarized in Table 1 and a simple schema of interconnection through synapses is depicted in Fig. 1.

3 System model

This section illustrates the random graph modeling approach to represent one overlay topology, the interconnection of X P2P networks, the search algorithm, and resource popularity as described in Section 2.

3.1 One overlay topology

Each P2P network is organized into an overlay that we model as a generalized random graph whose degree distribution is $\{p_k\}$ that can be interpreted as the probability that a randomly chosen peer has k connections in the overlay. The

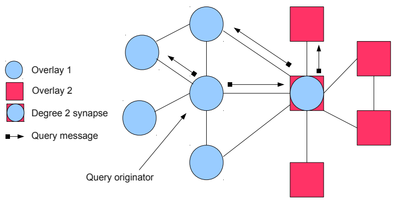


Fig. 1: Example of two P2P interconnected networks ($X = 2$) and one degree 2 synapse that belongs to both.

random graph degree distribution is a probability distribution therefore we consider its probability generating function (henceforth denoted as p.g.f.) that is equal to $G_0(z) = \sum_{k=0}^{\infty} p_k\, z^k$. To correctly characterize the neighborhood of a randomly chosen peer we also need to characterize the probability distribution of the number of connections of a peer reached by randomly choosing an *edge* of the overlay. This probability is proportional to the degree of the peer (kp_k) and it can be proved that its p.g.f. is given by $\frac{\sum_k kp_k z^k}{\sum_k kp_k} = z\frac{G_0'(z)}{G_0'(1)}$ where $G_0'(z)$ denotes the first derivative of $G_0(z)$ with respect to z and $G_0'(1)$ yields the average value of distribution $\{p_k\}$. Finally, to characterize the number of connections *excluding* the edge we chose we obtain the p.g.f. $G_1(z) = \frac{G_0'(z)}{G_0'(1)}$. Starting from Equations defining $G_0(z)$ and $G_1(z)$ we can compute the p.g.f. for the number of two hops neighbors of a randomly chosen peer as $G_0(G_1(z))$. Similarly, the p.g.f. for three hops neighbor is given by $G_0(G_1(G_1(z)))$, and so on.

For a detailed overview on analyzing generalized random graphs using generating functions, we refer the reader to [3].

3.2 Interconnection of multiple P2P networks

To interconnect multiple overlays we consider some peers as Synapses nodes: these peers belong to multiple P2P networks hence the interconnected system can be modeled by the probability distribution $\{s_i\}$ (with $i \in \{1, \ldots X\}$). The elements of this distribution describe the fraction of nodes belonging to multiple P2P networks: s_i is the fraction of nodes that belong to k P2P networks. Its p.g.f. is given by $F(z) = \sum_{i=0}^{\infty} s_i\, z^i$. If we consider one of the X P2P networks including the Synapse nodes then the p.g.f. for the number of connections of a randomly chosen peer can be written as

$$M(z) = s_1 G_0(z) + s_2 G_0^2(z) + \ldots + s_X G_0^X(z) = F(G_0(z))$$

that is, if the chosen node is a degree 1 synapse (this event has probability s_1) then the number of connections is represented by $G_0(z)$. If the node is a degree

2 synapse (this event has probability s_2), then the number of connections is represented by the sum of two independent random variables whose p.g.f. is $G_0(z)$; it is well-known that the generating function of the sum of two independent random variables is equal to the product of the respective generating functions yielding the $G_0^2(z)$ factor in the equation for $M(z)$. The same reasoning is valid for synapses whose degree is greater than 2.

A similar expression can be written for the neighborhood of a node reached by following one randomly chosen edge *excluding the selected edge*:

$$N(z) = s_1 G_1(z) + s_2 G_1(z) G_0(z) + \ldots + s_X G_1(z) G_0^{X-1}(z) = \frac{G_1(z)}{G_0(z)} F(G_0(z))$$

If we denote as $N_t(z)$ the p.g.f. for the probability distribution of the number of neighbors t hops away from a randomly chosen node we have that: $N_1(z) = M(z)$, and $N_2(z) = M(N(z))$, and $N_3(z) = M(N(N(z)))$, and so on. From these p.g.f. the average number of neighbors can be computed by evaluating their first derivative w.r.t. z in $z = 1$.

As such, each probability distribution $\{s_i\}$ induces an *interconnection cost* that we define as the average number of P2P networks a randomly chosen node belongs to:

$$f = F'(1) \tag{1}$$

3.3 Search algorithm

To model a flooding-based search in the interconnected system, we consider the set of probabilities $\{p_f(i)\}$, where $i \in \{1, \ldots X\}$. A peer belonging to i overlays sends/forwards a query to a particular neighbor with probability $p_f(i)$, where $i \in \{1, \ldots X\}$). Therefore, $\{p_f(i)\}$ *is not* a probability distribution.

We denote as q_h the probability that h first hop neighbors received a query from the peer that started the search. If the peer belongs to i overlays, it sends a query to one of its neighbors with probability $p_f(i)$. Therefore, the number of neighbors that receive the query follows a binomial distribution with parameter $p_f(i)$. Therefore, it is well known that the probability distribution $\{q_h\}$ has p.g.f. given by [3]:

$$Q(z) = s_1 G_0(1+p_f(1)(z-1)) + \ldots + s_X G_0^X(1+p_f(X)(z-1)) = \sum_{i=1}^{X} s_i G_0^i(1+p_f(i)(z-1)).$$

Similarly, for the p.g.f. of the probability distribution describing the number of queries sent by a node reached by following a randomly chosen edge, we obtain:

$$R(z) = \sum_{i=1}^{X} s_i \, G_1(1 + p_f(i)(z-1)) \, G_0^{i-1}(1 + p_f(i)(z-1)) \tag{2}$$

If we denote as $Q_t(z)$ the p.g.f. for the probability distribution of the number of neighbors t hops away from a randomly chosen peer that received a query, we have that: $Q_1(z) = Q(z)$, $Q_2(z) = Q(R(z))$, and $Q_3(z) = Q(R(R(z)))$, etc. As a special case, we may consider constant forwarding probabilities, *i.e.* $p_f(i) = p_f, \forall i \in \{1, \dots X\}$. In this case, we would obtain: $Q(z) = M(1+p_f(z-1))$ and $R(z) = N(1 + p_f(z-1))$. Since the p.g.f. of the probability distribution of the sum of independent random variables is given by the product of the corresponding p.g.f., the total number of queries generated by a search issued by a randomly chosen peer is described by: $T(z) = \prod_{t=1}^{TTL} Q_t(z)$ yielding the *average number of queries*

$$m = T'(1). \tag{3}$$

3.4 Hit probability

We model resource popularity by $0 \leq \alpha \leq 1$ that is the average fraction of peers that globally hold the given resource. We interpret this parameter as the probability that a randomly chosen node owns a copy of the resource.

If we denote as w_h the probability that h first hop neighbors hold a copy of the requested resource *and* received a query from a peer that belongs to i overlays we note that the number of such neighbors follows a binomial distribution with parameter $\alpha p_f(i)$. If we denote as $H_t(z)$ the p.g.f. for the probability distribution of the number of neighbors t hops away from a randomly chosen peer that received a query *and* hold a copy of the requested resource then we have that: $H_1(z) = Q_1(1+\alpha(z-1))$, $H_2(z) = Q_2(1+\alpha(z-1))$, $H_3(z) = Q_3(1+\alpha(z-1))$, and so on. Therefore, the total number of search hits is described by a probability distribution whose p.g.f. is given by: $H(z) = \prod_{t=1}^{TTL} H_t(z)$ yielding the search *hit probability*

$$p_{hit} = 1 - H(0) \tag{4}$$

3.5 A variation of the search algorithm

To model a search algorithm where peers that own a copy of the resource do not forward a query message it suffices to redefine $R(z)$ in Equation 2. In particular, when a peer owns a copy of the resource the number of its neighbors that receive the query is equal to 0: this happens with probability α. In Equation 5 this is represented by the term α that can be written as $\alpha p_0 z^0$ with $p_0 = 1$. With probability $1 - \alpha$ Equation 2 holds, therefore we obtain the p.g.f. of the probability distribution describing the number of queries sent by a node reached by following a randomly chosen edge as:

$$R(z) = \alpha + (1-\alpha) \sum_{i=1}^{X} s_i G_1(1 + p_f(i)(z-1)) G_0^{i-1}(1 + p_f(i)(z-1)) \tag{5}$$

The definition of $Q_t(z)$, and $T(z)$, and m remains unchanged.

4 Results

In this section, we will first show the results of the model validation, performed via a heavily multi-threaded simulator, written in Erlang, that reproduces, in terms of message routing, the exact behavior of a system described by our model. Also, we will show the results of some broad system evaluations made possible by the use of our model to compute metrics that would otherwise, if performed by means of simulations, require too much in terms of simulation time and computational power.

In our analysis, we consider different routing policies that can be employed in our scenarios, modeled by defining the $p_f(i)$ mentioned in Section 2. Those are:

- $p_f(i) = \frac{1}{i}$, henceforth referred to as *1/i*, i.e. the probability of selecting a neighbor is inversely proportional to the number of overlays a node is connected to. This routing tends to maintain a constant number of messages, but "flattens" the interconnected topology, not allowing synapse nodes to exploit the extended neighborhood.
- $p_f(i) = min(1, \frac{z_{max}}{zi})$, henceforth referred to as $zmax$, where $z = E[\{p_k\}]$ is the average number of neighbors for a node based on the current degree distribution and z_{max} is a system parameter, specified upon design, indicating the upper bound for the average number of forwarded messages. This policy allows for a better exploitation of Synapse nodes, while still finely limiting the number of messages in the system. In our evaluations, z_{max} has been set to $2z$, twice the average number of neighbors per node.
- $p_f(i) = 1$, henceforth referred to as *flood*, i.e. a routing where every node selects forwards a message to *every* neighbor, regardless of the number of connected overlays.

In both simulations and evaluations, the individual overlays have been modeled following the neighbors degree distribution measured in [4] from real world applications and used already in [5], in order to have an accurate overlay model.

4.1 Model validation

In order to evaluate the accuracy of our model in predicting the performance indexes of a real network, we validated the obtained results by means of simulation. The simulator employs standard statistical procedures to estimate 68% and 95% confidence intervals for the p_{hit} and m indexes defined in Section 3.

Simulation methodology The simulator has been developed from scratch in Erlang [6]. The choice of Erlang has been driven by its native multi-threading capabilities and inter-process communication model based on the message passing paradigm embedded in the language, thus allowing for a rapid implementation of an accurate network model made of node processes running independently and exchanging messages with one another. Each process has a list of other processes it can exchange messages with, that constitutes its neighborhood.

We consider N_s independent realizations for the interconnected overlay topologies (in our experiments $N_s = 30$); each interconnected topology is used to obtain one realization of m and p_{hit}. The h^{th} realization is obtained as follows:

- We first generate a new topology, made of X overlays interconnected by synapse nodes, using as input parameter the number of nodes $N = 500000$, the nodes degree distribution $\{p_k\}$ [4], and the $\{s_i\}$ to be validated;
- From the generated topology file, the simulator instantiates N node processes and assigns each the corresponding list of neighbors;
- One or more resources are then seeded in the system, according to their respective popularity α, by sending a `PUT(value)` message to $N\alpha$ random nodes;
- Separate worker processes take care of sending a query message `SEARCH(value,TTL)` to each node process in the network.
- Meanwhile, a listener process receives then the responses, either the resource being found or the TTL being reached, and of computes the statistics.

Topology generation The generation of a network made of interconnected overlays mainly consists of generating first X individual overlay topologies, and then connecting them by "merging" nodes from different overlays in one Synapse node, thus creating nodes with extended neighborhoods spanning across all the connected overlays. In order to generate X random graphs with a specified degree $\{p_k\}$ we relied on the algorithm presented in [7], that provides short generation times while guaranteeing the respect of the specified degree.

Validation results The first validation we performed was conducted for a system with only one overlay ($X = 1$). For the sake of brevity we only show the results for the *flood* routing strategy, $\alpha = 0.0001$, and $TTL = 3$. Table 2 shows the model is very accurate and faithfully predicts results when compared to the simulation output.

We then validated various scenarios with a higher number of interconnected overlays ($X = 4$), at $TTL = 3, 4$ and with different values of α, different routing policies and different distributions $\{s_i\}$. We considered the distribution for the degree of synapses summarized in Table 3.

Figure 2 (left) shows a comparison between the computed p_{hit} for different values of α and the corresponding simulation results, while Table 4 summarizes the same comparison for m. The results show that both performance metrics fall within the confidence interval of the simulation results.

Furthermore, we validate the system against the alternative search algorithm detailed in Section 3.5. For the sake of brevity, we are showing results only for S^2 since the same conclusions can be drawn for S^1 and S^3. Figure 2 (right) shows both p_{hit} and m against different values of α, since with this algorithm the number of message is dependent of the resource popularity. Even in this

Table 2: m for different s_i distributions: comparison between model and simulation.

	Model	**Simulation (95% C.I.)**
p_{hit}	0.3733	0.373552 ± 0.003852
m	4822.63	4821.57 ± 0.0498

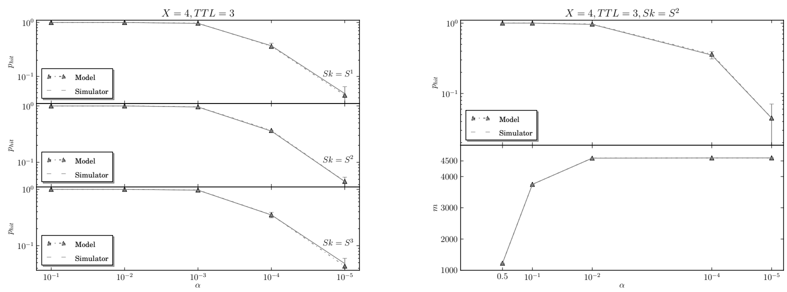


Fig. 2: p_{hit} for different α and s_i (left) and alternative search algorithm (right).

scenario, the model results fall within the confidence interval estimated by the simulator.

Therefore, we can safely conclude that our model is accurate in predicting the behavior of the performance indexes in a broad range of different scenarios. Furthermore, while simulations required hours of CPU time to complete solving our model took less than one second with a solver implemented in C.

4.2 Model exploitation

After validating the model we conducted a few analysis to show its usefulness in the design phase of the interconnection of several peer-to-peer networks.

Comparison of different routing policies A first evaluation concerns the choice of a specific routing policy in the system, i.e. the definition of different $p_f(i)$. In this case, we want to compare for values of α down to 10^{-6}, the performances in terms of p_{hit} and m for the distribution of degree of synapses S^1 (results for the other two distributions suggested similar considerations and are omitted for the sake of brevity), $X = 10$, and $TTL = 3$. Please note that to achieve a reliable measurement via simulation for $\alpha = 10^{-6}$ we would need to conduct complex simulations (at least 1000000 nodes) for a long simulation time (ideally each of them to be queried individually for multiple topology realizations).

Figure 3 show the values of p_{hit} for the 3 different policies and different resource popularities, while Figure 4 depicts the average number of messages for the 3 policies in the case of propagation of queries up to TTL hops (Figure 4b) and for the query propagation that stops when reaching a node holding a copy of the resource (Figure 4a) modeled in Section 3.5. In the former case, the number of messages is independent of the resource popularity while in the latter

Table 3: Definition of the $\{s_i\}$ distributions used for validation.

S^1	$s_1 = 0.7, s_2 = 0.1, s_3 = 0.1, s_4 = 0.1$
S^2	$s_1 = 0.4, s_2 = 0.3, s_3 = 0.2, s_4 = 0.1$
S^3	$s_1 = 0.1, s_2 = 0.2, s_3 = 0.3, s_4 = 0.4$

case we note that reduction of the number of query messages can be obtained for popular resources, i.e., for $\alpha > 0.01$.

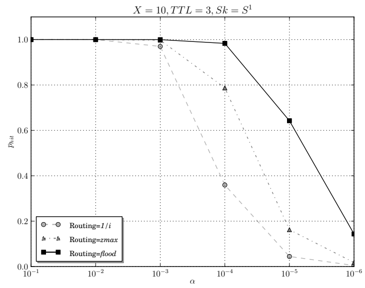


Fig. 3: Routing policies comparison: p_{hit} for different resource popularities α.

In this case, the model allows for a simple cost/benefit evaluation, based on the expected popularity of a resource. For one, we can notice an almost tenfold increase in the number of messages between the *zmax* and the *flood* policy, to which it does not correspond a proportional increase in the p_{hit}.

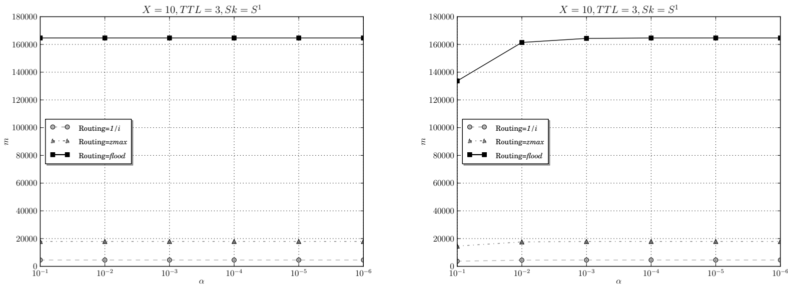
f-cost based evaluation In a cost/benefit analysis of the interconnected system, we consider p_{hit} as our benefit metric whereas m and f are considered as costs. Another kind of evaluation we performed consists of fixing the f cost and analyzing which distributions $\{s_i\}$ lead to better performances (p_{hit}) and minimum cost (m).

To this end we considered all distributions $\{s_i\}$ that can be defined for $X = 5$ where the individual probabilities are non-zero multiple of 0.05. We considered 3 values of f (namely, $f = 2, 3, 4$) and compared the performances of every distribution $\{s_i\}$ with given f for $TTL = 2$. Again, please note that this analysis would have required days of CPU time to be completed by means of simulation since even with a coarse granularity in the definition of $\{s_i\}$ (0.05) we tested hundreds of different distributions. This analysis required only a few seconds to complete with our model.

Figures 5a and 5b show a subset of these distributions (each point in the graph corresponds to a particular distribution $\{s_i\}$). We only plotted the ones

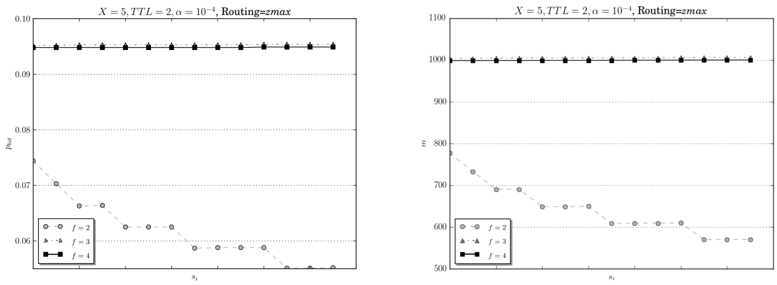
Table 4: m for different s_i distributions: comparison between model and simulation.

	Model	**Simulation (95% C.I.)**
S^1	4598.02	4596.77 ± 2.38
S^2	4701.82	4700.96 ± 0.49
S^3	4449.57	4453.58 ± 3.41



(a) Query propagation for TTL hops *(b)* Query propagation of Section 3.5

Fig. 4: Average number of messages for different routing policies.



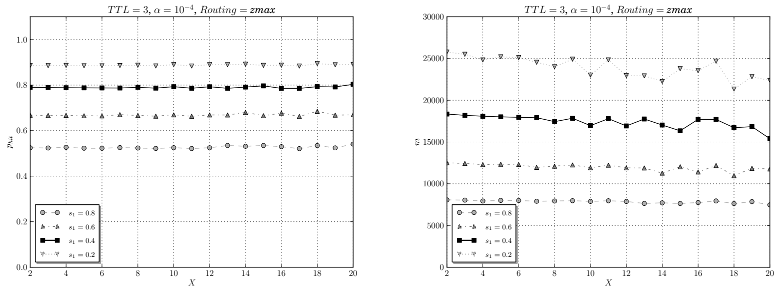
(a) Hit probability p_{hit} *(b)* Number of messages m

Fig. 5: s_i comparison at different f.

with the highest p_{hit}; it appears that the interconnection cost f alone is not directly bound to an increase in performances. There are, as a matter of fact, different configurations with $f = 3$ that perform equally (sometimes very slightly better) than those with a $f = 4$. Furthermore, within the configuration with $f = 2$ some are better than others in terms of performance and costs. Nevertheless, a clear relation exists between message cost m and p_{hit}: the larger the average number of messages the higher the p_{hit}.

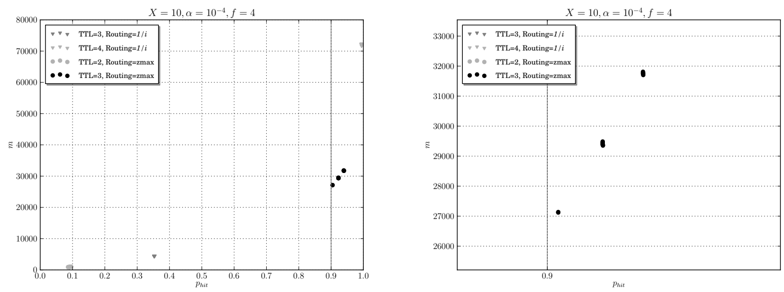
The behavior shown in the figures can be explained as following: the routing policy *zmax* limits the number of messages that can be issued by a node to z_{max}, which is set in our evaluations to $2z$. Therefore, increasing the number connections in the interconnected system (f) beyond certain values does not lead to a significant performance increase. That is why we observe a proportionally higher increase in the p_{hit} from $f = 2$ to $f = 3$ than from $f = 3$ to $f = 4$.

Effects of granularity Another aspect we analyze is a performance comparison as the number of overlays to interconnect increases. In this case we chose



(a) p_{hit} vs. X *(b)* m vs. X

Fig. 6: Performance evaluation with different numbers of overlay X.



(a) Overall view *(b)* Zoomed view

Fig. 7: Distribution of different routing policies with fixed f.

to analyze the behavior of the *zmax* routing policy, in a system with $TTL = 3$ and $\alpha = 0.0001$, for an increasing number of overlays (X) and for different distributions $\{s_i\}$, characterized by an increasing percentage of non-synapse nodes s_1, while the remainder of the distribution is equally distributed across the remaining s_i.

Figures 6a and 6b show four different configurations, with an increasing number of non-synapse nodes in the system. The parameter s_1 indicates the share of non synapses nodes, while the remaining part $(1 - s_1)$ is equally distributed among the remaining $X - 1$ values, i.e., $s_i = \frac{1-s_1}{X-1}$ for $1 < i \leq X$. It can be noted that at each given ratio of synapses vs non-synapses nodes the system behavior is roughly the same regardless the number of overlays. The efficiency is still tightly bound to the number of messages and both increase as s_1 decreases.

System design with minimum requirements Thanks to the high number of different configurations that can be evaluated with our model in a relatively short

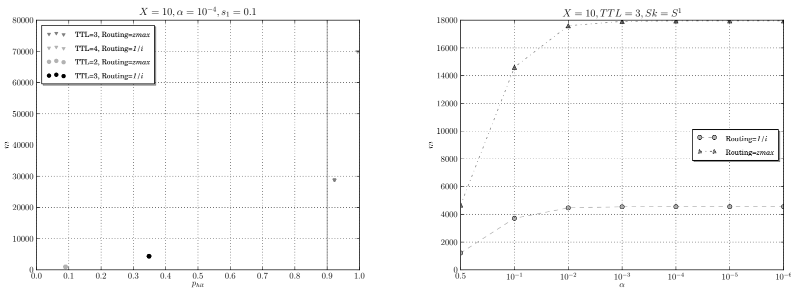


Fig. 8: Distribution of different routing policies with fixed s_1 (left) and message evaluation at different α, for different routing policies (right).

time, we conduct a further analysis to support the design of the interconnection of several peer-to-peer networks.

For instance, we set the number of overlays X and the resource popularity α; by setting a bound for the minimum desired p_{hit}, we can compare different routing policies and TTL values and find the one that minimizes the average number of messages m.

Figures 7 and 8 (left) show a classification of distributions $\{s_i\}$ for two different routing policies and two different TTL values with respect to p_{hit} and m for $X = 10$ and $\alpha = 0.0001$ (each point in the graphs represents a particular distribution $\{s_i\}$). In the first case (Figure 7), we decided to fix a cost factor and set $f = 4$, whereas in the second case (Figure 8 left), the fixed factor is the ratio of expected non-synapse nodes in the system s_1. We are able to discriminate immediately those distributions $\{s_i\}$ that do not satisfy the imposed criteria of having $p_{hit} > 0.9$. We also discriminate among those that do the distributions $\{s_i\}$ that minimize the number of messages m, as shown in Figure 7b.

Routing without propagation We briefly present some evaluation results based on the model variation presented in Section 3.5. In the first version of our model, the routing of a message is assumed to continue until the TTL expires, regardless of a resource being found or not. This leads to an $H_t(z)$ able to describe different cases, such as the probability of finding *multiple* copies of a resource. However the system is not optimal message-wise. In case we are interested only in the first hit of a search query, and we want to optimize the number of messages employed, with the variant of $R(z)$ described in 3.5 we are able to evaluate the system under the conditions that the routing in a node stops whenever a resource is found.

Figure 8 (right) shows the trend of m for different α, and two routing policies for $X = 10$, $TTL = 3$, and distribution S^1. While the number of messages was unrelated to the resource popularity before, here we see that, as routing stops upon first hit, the more popular a resource, the lower the number of messages per query.

5 Related work

Inter-cooperation of network instances has been identified in [8, 9] as one of the future trends in the current Internet architecture development. When discussing logical networks, various techniques to achieve inter-communication among them have been presented.

Synergy [10] is an architecture for the inter-cooperation of overlays which provides a cooperative forwarding mechanism of flows between networks in order to improve delay and throughput performances. Co-located nodes are, in the authors' opinion, good candidates for enabling such mechanisms and reduce traffic.

With a similar goal, authors in [11] propose algorithms tailored to file sharing applications, enabling a symbiosis between different overlays networks. They present hybrid P2P networks cooperation mechanisms and provide interesting observations on the appropriate techniques to perform network join, peer selection, network discovery, etc. Their simulations showed the effect of the popularity of a cooperative peer on the search latency evaluation, that is the more a node has neighbors, the better, as well as the effect of their caching mechanism which reduces (when appropriately adjusted) the load on nodes (but interestingly does not contribute to faster search).

Authors in [12] model an interconnected system by considering spaces with some degree of *intersection* between one another. They focus on different strategies to find a path between two overlays, and compare various routing policies analyzing which trade-offs lead to the best results. Trade-offs are considered in terms of number of messages, number of hops to find a result and state overhead. They provide a comparative analytical study of the different policies. They show that with some dynamic finger caching and with multiple gateways tactfully laid out in order to avoid bottlenecks due to the overload of a single gateway, they obtain good performances. Their protocol focuses on the interconnection of DHTs, while we focus on unstructured overlays.

Finally, [13] studies the co-existence of multiple overlay networks, namely Pastry and an unstructured overlay that uses a gossip protocol to improve its performance.

6 Conclusions and future work

In this paper we considered interconnection of large scale unstructured P2P networks through co-located nodes called synapses: these nodes send/forward a query to all the P2P networks they belong to. We developed a generalized random graph based model to represent the topology of one unstructured P2P network, the partition of nodes into synapses, the probabilistic flooding based search algorithms, and the resource popularity. We validated our model against simulations and proved that its predictions are reliable and accurate. The model allowed the analysis of very large and complex systems: we believe that simulation and/or prototype deployment based analysis would be unfeasible in this case.

We are currently working to further extend our model in several directions. In particular, we are generalizing equations to represent heterogeneous topologies and resource availability. As a consequence, we are also extending the analysis to more refined partition of synapses, i.e., to consider the fraction of nodes that belong to a specific set of P2P networks. Furthermore, we are extending the model to represent nodes availability due to churning. Last but not least, we are generalizing the model to represent interconnection of both unstructured and structured P2P networks.

References

1. L. Liquori, C. Tedeschi, L. Vanni, F. Bongiovanni, V. Ciancaglini, and B. Marinkovic, “Synapse: A scalable protocol for interconnecting heterogeneous overlay networks,” in *Proc. of Networking ’10*, 2010.
2. V. Ciancaglini, R. Gaeta, L. Liquori, and R. Loti, “Modeling and analysis of large scale interconnected unstructured p2p networks,” in *Proc. of ICPADS ’12*, 2012.
3. M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Random graphs with arbitrary degree distributions and their applications,” *Phys. Rev. E*, vol. 64, p. 026118, Jul 2001.
4. 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.
5. R. Gaeta and M. Sereno, “Generalized probabilistic flooding in unstructured peer-to-peer networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 2055–2062, 2011.
6. Erlang programming language website. [Online]. Available: http://www.erlang.org/
7. F. Viger and M. Latapy, “Efficient and simple generation of random simple connected graphs with prescribed degree sequence,” in *Computing and Combinatorics*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3595, pp. 440–449.
8. M. Siekkinen, V. Goebel, T. Plagemann, K. Skevik, M. Banfield, and I. Brusic, “Beyond the Future Internet–Requirements of Autonomic Networking Architectures to Address Long Term Future Networking Challenges,” in *International Workshop on Future Trends of Distributed Computing Systems*, 2007, pp. 89–98.
9. A. Fonte, M. Curado, and E. Monteiro, “Interdomain Quality of Service Routing: Setting the Grounds for the Way Ahead,” *Annals of Telecommunications*, vol. 63, no. 11, pp. 683–695, 2008.
10. M. Kwon and S. Fahmy, “Synergy: an Overlay Internetworking Architecture,” in *Proc. of International Conference on Computer Communications and Networks*, 2005, pp. 401–406.
11. K. Junjiro, W. Naoki, and M. Masayuki, “Design and Evaluation of a Cooperative Mechanism for Pure P2P File-Sharing Networks,” *IEICE Trans Commun (Inst Electron Inf Commun Eng)*, vol. E89-B, no. 9, pp. 2319–2326, 2006.
12. P. Furtado, “Multiple Dynamic Overlay Communities and Inter-space Routing,” *Lecture Notes in Computer Science*, vol. 4125, p. 38, 2007.
13. B. Maniymaran, M. Bertier, and A. Kermarrec, “Build one, get one free: Leveraging the coexistence of multiple p2p overlay networks,” in *Distributed Computing Systems, 2007. ICDCS’07. 27th International Conference on.* IEEE, 2007, pp. 33–33.