

# Performance of Random Walks in One-Hop Replication Networks

Luis Rodero-Merino <sup>a,\*</sup>, Antonio Fernández Anta <sup>b</sup>, Luis López <sup>b</sup>,  
Vicent Cholvi <sup>c</sup>

<sup>a</sup>*Telefónica I+D, Madrid, Spain*

<sup>b</sup>*LADyR, GSyC, Universidad Rey Juan Carlos, Móstoles, Spain*

<sup>c</sup>*Universitat Jaume I, Castellón, Spain*

---

## Abstract

Random walks are gaining much attention from the networks research community. They are the basis of many proposals aimed to solve a variety of network-related problems such as resource location, network construction, nodes sampling, etc. This interest on random walks is justified by their inherent properties. They are very simple to implement as nodes only require local information to take routing decisions. Also, random walks demand little processing power and bandwidth. Besides, they are very resilient to changes on the network topology.

Here, we quantify the effectiveness of random walks as a search mechanism in *one-hop replication networks*: networks where each node knows its neighbors' identity/resources, and so it can reply to queries on their behalf. Our model focuses on estimating the expected average search time of the random walk by applying network queuing theory. To do this, we must provide first the expected average search length. This is computed by means of estimations of the expected average coverage at each step of the random walk. This model takes into account the *revisiting effect*: the fact that, as the random walk progresses, the probability of arriving to nodes already visited increases, which impacts on how the network coverage evolves. That is, we do not model the coverage as a memoryless process. Furthermore, we conduct a series of simulations to evaluate, in practice, the above mentioned metrics. Our results show a very close correlation between the analytical and the experimental results.

**Key words:** random-walk, look-ahead networks, average search time, average search length

---

---

\* Corresponding author. Tel: +34 913373928, fax: +34 913374272, postal address: Telefónica I+D, C/Emilio Vargas 6, C.P.:28043, Madrid, Spain.

*Email addresses:* rodero@tid.es (Luis Rodero-Merino), anto@gsyc.es (Antonio Fernández Anta), llopez@gsyc.es (Luis López), vcholvi@uji.es (Vicent Cholvi).

## 1 Introduction

*Random walks* are a mechanism to route messages through a network. At each hop of the random walk, the node holding the message forwards it to some neighbor chosen uniformly at random. Random walks have interesting properties: they produce little overhead and network nodes require only local information to route messages. In turn, this makes random walks resilient to changes on the network structure. Thanks to these features, random walks are useful for different applications, like routing, searching, sampling and self-stabilization in diverse distributed systems such as Peer-to-Peer (P2P) and wireless networks [1–10].

Past works have addressed the study of random walks. Some of this research has focused on the coverage problem, trying to find bounds for the expected number of hops taken by a random walk to visit all vertices (nodes) in a graph<sup>1</sup>  $G(C_G)$  [11–14]. Results vary from the optimal  $C_G$  of complete graphs  $\Theta(n \log n)$  [11] (where  $n$  is the number of vertices) to the worst case found in the lollipop graph  $\Theta(n^3)$  [15]. Barnes and Feige in [16] generalize this bound to the expected number of hops to cover a fraction ( $f < n$ ) of the vertices of the network, which they found is  $\Theta(f^3)$ . Other works, for example, are devoted to find bounds on the expected number of steps before a given node  $j$  is visited starting from node  $i$  ( $H_{i,j}$ ). For example, it is known that the upper bound for  $H_{i,j}$  is  $\Theta(n^3)$  [17]. Many of these results are based on the study of the properties of the transition matrix  $\mathbf{P}$  and adjacency matrix  $\mathbf{A}$  in spectral form [18].

The previous results are used in several works to discuss the properties of random walks in communication networks. Gkantsidis et al. [19] apply them to argue that random walks can simulate random sampling on P2P networks, a property that in their opinion justifies the ‘success of the random walk method’ when proposed as a search tool [3] or as a network constructing method [9]. Adamic et al. [20] study the search process by random walks in power-law networks applying the generating function formalism. This work seems deeply inspired by a previous contribution of Newman et al. [21], who study the properties (mean component size, giant component size, etc.) of random graphs with arbitrary degree distribution.

This paper introduces a study of random walks from a different perspective. It does not study the formal bounds in the amount of hops to cover the network. Instead, it tries to estimate the efficiency of the random walk as a search mechanism in communications networks, applying network queuing theory. It takes into account the bounded processing capacities of the nodes of the network and the load introduced by the search messages, that are routed using random walks. To obtain this load, we

---

<sup>1</sup> The term *time* to refer to the number of hops of the random walk (that is, its *length*) is usual in many previous works. Thus, for example,  $C_G$  is often denoted the *cover time*. However, in this work we will use the term *time* to refer to the *duration* of the random walk. To avoid confusion, from now on the term *time* will only denote the physical magnitude.

need to estimate first the average search length, which in turn is computed from the expected average coverage: the average number of different nodes covered at each hop of the random walk. A distinguishing feature of our work is that, as in the case of Adamic et al. [20], it deals with a scenario that has not been very exhaustively explored although, in our opinion, is quite interesting in the communications field: *one-hop replication networks*.

**One-hop Replication** One-hop replication networks (also called *lookahead networks* [22]) are networks where each node knows the identity of its neighbors and so it can reply on their behalf. Hence, to find a certain node by a random walk it suffices to visit any of its neighbors. This feature is present for example in social networks, where to find some person it is usually enough to locate any of her/his friends [20]. Also, certain proposals to improve the resource location process on P2P systems [2, 23] (some based on random walks) assume that each node knows the resources held by its neighbors, so to discover some resource (such as a file or a service) it suffices to visit any of the neighbors of the node(s) holding it.

In one-hop replication networks, when the random walk visits some node  $i$  we say it also *discovers* the neighbors of  $i$ . Hence, we will use two different terms to refer to the coverage of the random walk. We denote by *visited nodes* those that have been traversed by the random walk, and by *covered nodes* the visited nodes and their neighbors. See Figure 1 for an illustrative example.

**Previous Work and the Revisiting Effect** There is some research work related with the characterization of random walks in one-hop replication networks. In [24] the authors prove that in the power-law random graph the amount of hops for a random walk to discover the graph is sublinear (faster than coupon collection, with which the random walk is compared in [19]). Also, Manku et al. [22] study the impact of lookahead on P2P systems where searches are routed through greedy mechanisms. In another work, Adamic et al. [20] try to find analytical expressions for  $C_G$  the cover time of a random walk in power-law networks with two-hops replication. They detected divergences between the analytical predictions and the experimental results. The reason for such discrepancy, as the authors point out, is the *revisiting effect*, which occurs when a node is visited more than once. In small-world networks, where a small number of nodes are connected to other nodes far more often than the rest, it is quite common for random walks to visit often these highly connected nodes.

**Our Contributions** Although there is a plethora of interesting results about random walks, we have noticed that there are situations where current findings are not straightforward to apply, especially on communication networks with one-hop

replication. For example, in such networks, we can be interested on studying beforehand the expected behavior of the random walk to evaluate if it suits the system requirements. We characterize the random walk performance by four values:

- *The expected coverage.* Given by the expected number of visited and covered nodes of each degree  $k$  at each hop  $l$  of the random walk.
- *The expected average search length.* Expected length of searches in number of hops, assuming that the source and destination nodes of each search are chosen uniformly at random. Obtained from the coverage estimations.
- *The expected average search duration.* Expected time to solve searches. Obtained from the average search length, given the *processing capacity* of each node and the *load* on the network due to queries.
- *The maximum load that can be injected to the network* without overloading it.

In this work we provide a set of expressions that model the behavior of the random walk and give estimations for the three previous parameters. Our claim is that these expressions can be used as a mathematical tool to predict how random walks will perform on networks of arbitrary degree distribution. Then, we do not only address the coverage problem (i.e. to estimate the amount of nodes covered after each hop of the random walk), but we also apply queuing theory to model the response time of the system depending on the load. As we show, this approach allows to compute in advance important magnitudes, such the expected search duration or the maximum load that can be managed by the network before getting overloaded. Additionally, we find our model useful to study how certain features of the network impact on the performance of searches. For example we find that the best average search time is achieved only if the nodes with higher degrees have also greater processing capacities.

The expressions related with the estimation of covered nodes at each hop are the most complex part of the model. They must deal both with the one-hop replication feature and the revisiting effect. However, we should remark that the model can be trivially adapted to networks where the *one-hop replication* property does not hold, and the search finishes only when the node we are searching for is found (see the last paragraph in Section 2.4).

Likewise, it is easy to modify the model to a variation of the random walk where each node avoids sending back the message to the node it received it from at the previous hop. We denote this routing mechanism *avoiding random walks*, and we deem it interesting for two reasons. First, intuitively, it should improve the random walk coverage (we have confirmed this experimentally). Second, it can be implemented in real systems using only local information, just as the pure random walk (the sending node only needs to know from which neighbor the message came from).

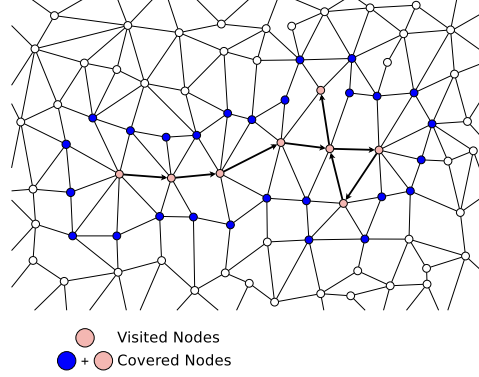


Fig. 1. Illustrative example of visited and covered nodes

A feature of our proposal is that it does not require the complete adjacency matrix  $A$ , that in some situations could be unknown. Instead, thanks to the randomness assumption we apply it only needs the degree distribution of the network to compute the metrics we are interested in. On the other hand, this work is focused on networks with good connectivity and where the nodes degrees are independent (see Section 2.1).

Another property of this model is that it takes into account the revisiting effect by modeling the coverage of the random walk at each hop  $l$  depending on the coverage at the previous hop  $l - 1$ . That is, the evolution of the coverage is not assumed to be a memoryless process, a simplification that can lead to errors as seen in [20].

The rest of the paper is organized as follows. Section 2 introduces our analysis of the coverage and average search length of random walks, along with some experimental evaluation. Section 3 is centered on obtaining the average search time of random walks. Finally, in Section 4, we state our conclusions and propose some potential future work.

## 2 Analysis of Random Walks

In this section, we analyze the behavior of random walks in arbitrary networks.

### 2.1 Model and Assumptions

We will represent networks by means of undirected graphs  $G = (V, E)$ , where vertices  $V$  represent the nodes and edges  $E \subseteq V \times V$  are the links between nodes. There are no links connecting a vertex to itself, or multiple edges between the same two vertices. This does not simplify our model, but makes it closer to real scenarios like typical P2P networks. We denote by  $|V| = n$  the number of nodes in the graph and

by  $n_k$  the number of nodes that have degree  $k$  (i.e., the number of nodes that have  $k$  neighbors,  $\sum_k kn_k = 2|E|$ ). For all vertices its degree  $k$  is lower than the size of the network  $n$ , as in typical real world networks (such as social and pure P2P networks) each node is connected to only a subset of the other vertices in the system<sup>2</sup>. We also denote by  $p_k$  the probability that some node in the network, chosen uniformly at random, has degree  $k$  (i.e.,  $p_k = n_k/n$ ). The average degree of a network is given by  $\bar{k} = \sum_k k p_k$ . For a given network, the distribution formed by the probabilities  $p_k$  (for all  $k$ ) is known as the *degree distribution* of such a network.

A random walk over  $G$  can be defined as a *Markov Chain* [15] process  $M_G$  where the transition matrix  $\mathbf{P} = [P_{ij}]$  is defined as:

$$P_{ij} = \begin{cases} \frac{1}{d(i)} & \text{if } (i, j) \in E. \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where  $P_{ij}$  is the probability of moving from node  $i$  to node  $j$ , and  $d(i)$  is the degree of node  $i$ .  $\mathbf{P}$  allows to study the probability of visiting each node at each hop  $l$ . This probability is expressed in the *state probability vector*,  $\mathbf{q}^l = (q_1^l, q_2^l, \dots, q_n^l)$ , where  $q_i^l$  represents the probability that the random walk visits node  $i$  at hop  $l$ . This probability evolves as  $\mathbf{q}^l = \mathbf{q}^{l-1} \mathbf{P}$ .

Assuming that  $G$  is connected and finite, then  $M_G$  is irreducible: any node can be reached from any other node, and the average path length between two any nodes is finite. Assuming also that  $G$  is non-bipartite, then we can state that  $M_G$  is aperiodic and so we are able to apply the *Fundamental Theorem of Markov Chains* [15]. This theorem states that in such graph  $M_G$  is *ergodic* and exists an unique state probability distribution  $\boldsymbol{\pi}$ , denoted the *stationary distribution*, such that  $\boldsymbol{\pi} \mathbf{P} = \boldsymbol{\pi}$ ,  $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_n)$ , where  $\pi_i$  is:

$$\pi_i = \frac{d(i)}{2|E|}. \quad (2)$$

Intuitively,  $\boldsymbol{\pi}$  represents the steady state of  $M_G$ . That is,  $\pi_i$  represents the probability that the node  $i$  is visited at any hop of the random walk once the stationary distribution has been reached. This probability is proportional to the degree of  $i$ ,  $d(i)$ .

---

<sup>2</sup> Some P2P networks like Napster have a central node that network members use to locate files. But those networks are not considered as pure P2P systems because they use a typical server-client architecture with a centralized topology to perform searches. They are regarded to have a “P2P” behavior only in the way files are shared. This work is rather focused on the decentralized topologies of pure P2P networks



**Mixing Rate and Conductance** We are interested on how fast the random walk converges to  $\pi$ , a magnitude that is called the *mixing rate* [18]. We require a fast convergence in order to be able to apply Equation 6.

The convergence rate is related with the eigenvalues of the transition matrix  $\mathbf{P}$ . A vector  $\vec{x}$  is an *eigenvector* of  $\mathbf{P}$  with *eigenvalue*  $\lambda$  iff  $\vec{x}\mathbf{P} = \lambda\vec{x}$  (so for example  $\pi$  is an eigenvector of  $\mathbf{P}$  with eigenvalue 1). It is well known [18] that  $\mathbf{P}$  has  $n$  real eigenvalues  $\lambda_0 = 1 > \lambda_1 \geq \dots \geq \lambda_{n-1} \geq -1$  (and in fact, if  $G$  is non-bipartite then  $\lambda_{n-1} > -1$ ). It is also known [25] that the convergence rate to  $\pi$  is governed by the second largest eigenvalue modulus of  $\mathbf{P}$ ,  $\max\{\lambda_1, |\lambda_{n-1}|\}$ . In most real world networks we can safely assume that  $\lambda_1 > |\lambda_{n-1}|$  [18, 19, 25]. The following holds for a random walk starting at node  $i$  [18]:

$$|\mathbf{P}_i^{(l)}(j) - \pi_j| \leq \sqrt{\frac{d(j)}{d(i)}} \lambda_1^l, \quad (3)$$

where  $\mathbf{P}_i^{(l)}$  is the distribution of the state of the random walk at hop  $l$ , when  $i$  is the initial state. Thus, we can expect a fast mixing for high values of the *spectral gap*  $1 - \lambda_1$ .

Now, the  $\lambda_1$  value is strongly related with the *conductance* of the network,  $\Phi_G$ . Informally, the conductance measures how well ‘connected’ the graph is. It is defined as follows. For  $S \subseteq V$ , the cutset of  $S$ ,  $C(S)$ , is the set of edges with one endpoint in  $S$  and the other endpoint in  $\bar{S}$ . The volume of  $S$ ,  $\text{vol}(S)$ , is defined as the sum of degrees of the nodes in  $S$ , i.e.,  $\text{vol}(S) = \sum_{i \in S} d(i)$ . Then the conductance of  $G$  is computed as:

$$\Phi_G = \min_{\substack{S \subseteq V \\ \text{vol}(S) \leq \text{vol}(V)/2}} \frac{|C(S)|}{\text{vol}(S)}. \quad (4)$$

The relationship between the conductance and the convergence is given by the following expression (*Cheeger’s inequality*) [18]:

$$\frac{\Phi_G^2}{2} \leq 1 - \lambda_1 \leq 2\Phi_G. \quad (5)$$

So a good conductance leads to high mixing rates, that is, the random walk state will converge quickly to the stationary distribution  $\pi$ . The intuition behind this fact is that in graphs with good conductance the random walk will be able to move to any region of the graph easily, whichever the origin node, and so it will evolve quickly to the equilibrium. We reason that high connectivity is to be expected in many real world networks (specially communication networks) and network models [26–28].

Therefore, we can assume that the probability that the node visited by the random walk has degree  $k$  at each hop of the random walk,  $P(k)$ , is also proportional to  $k$  and can be computed as:

$$P(k) = \sum_{\substack{i \in V \\ d(i)=k}} \frac{d(i)}{2|E|} = n_k \frac{k}{\sum_j j n_j} = \frac{k p_k}{\bar{k}}. \quad (6)$$

We will apply Equation 6 intensively for our analysis of the coverage. Of course, its correctness depends on the distance of the random walk to the stationary distribution, or how fast it converges to it. Another issue to be taken into account is the possible dependencies between successive steps of the random walk. Our analysis estimates the average number of nodes visited and covered by the random walk at a certain hop from the values estimated at the previous hop. The new estimation is done assuming that the random walk has statistical properties similar to the random sampling of nodes where the probability of choosing a certain node is proportional to  $k_i$ , *despite the apparent dependencies between consecutive hops*.

Also, the work by Gkantsidis et al. [19] shows the similarities between independent sampling and random walks, that we assume for our mean based analysis. As the authors state, in networks with good connectivity and expansion properties (which are strongly related to  $\lambda_1$ ) the random walk has a behavior close to independent sampling, being the probability of choosing some node proportional to its degree.

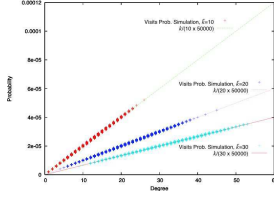
Besides, we have performed some experiments to verify the correctness of this hypothesis. The results, shown in Figure 2 confirm it is a valid assumption. Also, we would like to remark that the property expressed by Equation 6 is in fact assumed in previous works about random walks (e.g., [20, 21]) and backed by [19].

Another important issue we have tested is how ‘fast’ the random walk evolves to a state where the assumption of Eq. 6 holds. Figure 3 shows how the random behaves. It can be seen that, almost immediately after hop 0 (start node), the probability of reaching a node of degree  $k$  is  $P(k)$ .

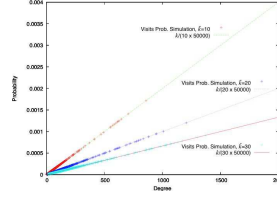
We should note that the good conductance property, that implies that the random walk can move from any node to any other node in few steps, discards some topologies such as cycles.

**Independence of Nodes Degrees** Finally, we assume that the degrees of neighbors are independent. That is, given any two connected nodes  $i$  and  $j$  ( $(i, j) \in E$ ) and any two degree values  $k_1$  and  $k_2$ , then  $P[d(i)=k_1 \mid d(j)=k_2] = P[d(i)=k_1] = p_{k_1}$ . This property holds in networks built by random mechanisms, like the ones used to built the ER and small-world networks we target in our experiments. To confirm that the degree independence assumption is valid we have run some experiments,



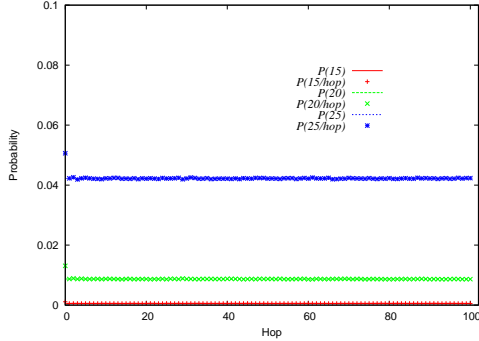


(a) Erdos-Renyi networks.

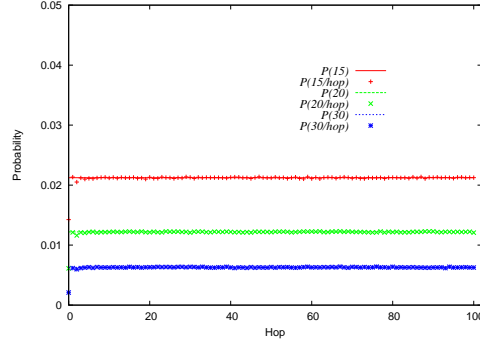


(b) Small-world networks.

Fig. 2. In these figures, we show the probability of a search message arriving at a particular node as a function of its degree. We have used both Erdos-Renyi and small-world (power-law) networks formed by 50,000 nodes, with different average node degrees (10, 20 and 30). The same experiments have been performed with networks formed by 25,000 and 100,000 nodes, and we found similar results. As it can be readily seen, the probability of a search message arriving at a particular node is proportional to the degree of the node.



(a) Erdos-Renyi network,  $\bar{k} = 30$ .



(b) Small-world network,  $\bar{k} = 10$ .

Fig. 3. These figures compare the probability  $P(k)$  of reaching a node of degree  $k$  as defined by the model, with the measured probability of reaching a node of degree  $k$  at each hop of the random walk. Both for ER and small-world networks the experimental results are averaged over three different networks with the same average degree and size ( $n = 50 \cdot 10^4$ ).

whose results are shown in Figure 4. These experiments aim to measure if the probability of reaching a node of degree  $k$  when following a random walk is affected by the degree  $k'$  of the node the random walk was in the previous hop ( $P(k/k')$ ). Our results lead to the conclusion that  $\forall k, k' P(k/k') = P(k)$ , that is,  $k'$  does not have an impact on  $k$ .

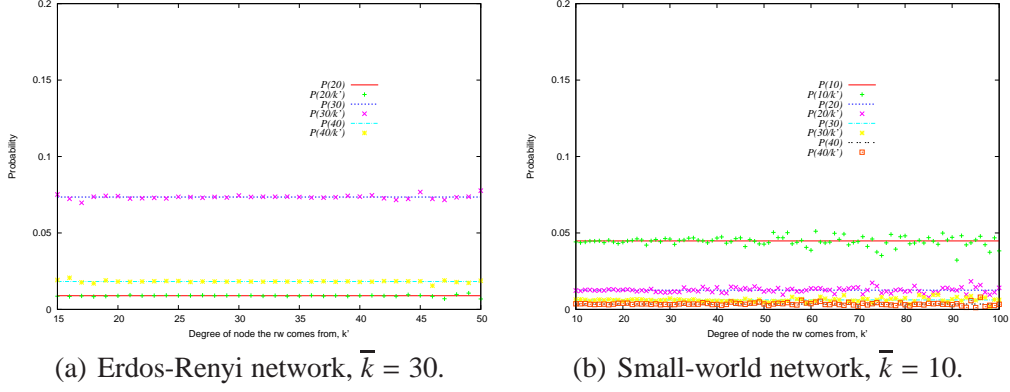


Fig. 4. These figures compare the probability  $P(k)$  of reaching a node of degree  $k$  as defined by the model, with the measured probability of reaching a node of degree  $k$  given that the rw comes from a node of degree  $k'$ ,  $P(k/k')$ . Both for ER and small-world networks the experimental results are averaged over three different networks with the same average degree and size ( $n = 10^5$ ).

We should note also that this property is not fulfilled in certain graphs like those built by preferential mechanisms where it is well-known that there is a correlation among neighbors degrees [29]. This could lead to certain deviations in mean-based analysis of the random walk (as our own).

In the following, we study how many different nodes are visited by a random walk as a function of its length (i.e., of the number of steps taken) and of the degree distribution of the chosen network. Subsequently, we extend this result to also consider the neighbors of the visited node. These metrics allow us to quantify how much of a network is being “known” throughout a random walk progress. Then, we turn our attention to provide an estimation of the average search length of a random walk. In the last subsection, we validate our analytical results by means of simulations. We assume that only the degree distribution  $p_k$  and the size  $n = |V|$  of the network are known.

## 2.2 Number of Visited Nodes

This metric represents the average number of different nodes that are visited by a random walk until hop  $l$  (inclusive), denoted by  $V^l$ . Note that nodes may each be visited more than once, but revisits are not counted.

To obtain  $V^l$ , we first calculate the average number of different nodes of degree  $k$  that are visited by a random walk until hop  $l$  (inclusive), denoted by  $V_k^l$ . We make a case analysis:

- When  $l = 0$  (i.e., in the source node): Since the source node of the random walk is chosen uniformly at random, then the probability of starting a random walk at

a node of degree  $k$  is  $p_k$ . Therefore,

$$V_k^0 = 1 \cdot p_k + 0 \cdot (1 - p_k) = p_k. \quad (7)$$

- When  $l = 1$  (i.e., at the first hop): Here we apply that the probability of visiting some node of degree  $k$  at any hop is given by  $P(k)$  (Equation 6). This is based on the assumption that the random walk behaves similarly to independent sampling despite dependencies between consecutive hops (based on [19], see Section 2.1). We deem this premise to be reasonable even at the first stages of the random walk, due to the high mixing rates found in the type of networks on which we focus our work (again, see Section 2.1). Recall that the experimental evaluation both of this assumption (Fig. 2) and of our model (shown in Section 2.5), seem to verify this. Thus, we have that

$$\begin{aligned} V_k^1 &= V_k^0 + P(k) \\ &= p_k + \frac{k p_k}{\bar{k}}. \end{aligned} \quad (8)$$

- When  $l > 1$ : we must take into account the probability of the random walk arriving at an already visited node. To compute such a probability, we define the following two values:
  - $P_v(k, l)$ : This represents the probability that, if the random walk arrives at a node of degree  $k$  at hop  $l$ , that node has been visited before. It can be obtained as follows:

$$P_v(k, l) = \frac{V_k^{l-2}}{n_k}. \quad (9)$$

Note that we put  $V_k^{l-2}$  instead of  $V_k^{l-1}$  because the node visited at hop  $l - 1$  can not be visited at hop  $l$  (no vertex is connected to itself).

- $P_b$ : This is the probability that at any given hop the random walk is moving back to the node where it came from<sup>3</sup>. Since any visited node has degree  $k$  with probability  $P(k)$ , then the random walk will go back through the same link from which it came with probability  $1/k$ . Therefore, we have:

$$P_b = \sum_k P(k) \frac{1}{k} = \frac{1}{\bar{k}}. \quad (10)$$

Using these probabilities,  $V_k^l$  can be written as

$$\begin{aligned} V_k^l &= V_k^{l-1} + P(k)(1 - P_b)(1 - P_v(k, l)) \\ &= V_k^{l-1} + \frac{k p_k}{\bar{k}} \left(1 - \frac{1}{\bar{k}}\right) \left(1 - \frac{V_k^{l-2}}{n_k}\right). \end{aligned} \quad (11)$$

<sup>3</sup> Here we can easily adapt the model to the *avoiding random walk*. If we don't want to consider the case of a random walk moving back to the node where it came from, it is enough to assign  $P_b = 0$ .

Finally, taking the results obtained in Equations 7, 8 and 11, we have that the total number of different nodes visited until hop  $l$  is

$$V^l = \sum_k V_k^l. \quad (12)$$

### 2.3 Number of Covered Nodes

This metric provides an estimation of the average number of different nodes *covered* by a random walk until hop  $l$  (inclusive), denoted by  $C^l$ . A node is covered by a random walk if such a node, or any of its neighbors, has been visited by the random walk.

To obtain  $C^l$ , we first calculate the number of different nodes of degree  $k$  covered at hop  $l$ , denoted by  $C_k^l$ .

- When  $l = 0$ :

$$\begin{aligned} C_k^0 &= p_k(1 + kP(k)) + \sum_{j \neq k} p_j j P(k) \\ &= V_k^0 + P(k) \bar{k}. \end{aligned} \quad (13)$$

The first term takes into account the possibility that the source node has degree  $k$ . The second term refers to the number of neighboring nodes (of the source node) of degree  $k$ . If the source node has degree  $j$  (which happens with probability  $p_j$ ) then, on average,  $j P(k)$  nodes of degree  $k$  will be covered, since each one of the  $j$  neighboring nodes of the source node will have degree  $k$  with probability  $P(k)$ .

- When  $l > 0$ : Given a link  $(v, w) \in E$ , we say that it has two endpoints, which are the two ends of the link. We denote the endpoint of the link at node  $v$  by  $v(w)$ , and similarly the endpoint of the link at node  $w$  by  $w(v)$ . We say that  $v(w)$  *hooks onto* node  $v$ . We also say that  $v(w)$  has been *checked* by a random walk if such a random walk has visited node  $w$ . These concepts are graphically explained in Fig. 5.

Now, let us denote by  $E^l$  the number of endpoints checked for the first time at hop  $l$ , and by  $P_u(k, l)$  the probability that these endpoints hook onto still uncovered nodes of degree  $k$ . Then,  $C_k^l$  (where  $l > 0$ ) can be written as follows:

$$C_k^l = C_k^{l-1} + P_u(k, l) E^l. \quad (14)$$

- To obtain  $E^l$ , we consider the number of different endpoints checked after hop  $l$  to be  $\sum_j j V_j^l$ . So, the number of endpoints checked for the first time at hop  $l$  is  $\sum_j (V_j^l - V_j^{l-1}) j$ . However, one of the endpoints hooks onto the node the random walk comes from (i.e., it cannot increase the amount of nodes that are covered). Thus:

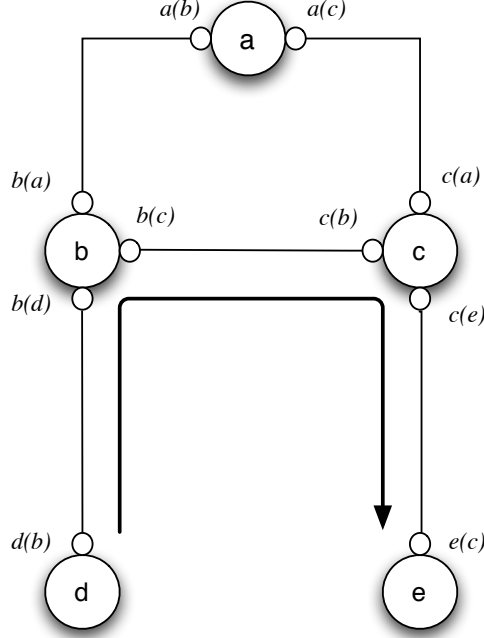


Fig. 5. The figure shows a simple graph formed by 5 nodes (named  $a$ ,  $b$ ,  $c$ ,  $d$  and  $e$ ) where there is a random walk that follows the path  $d-b-c-e$ . At each node, we represent the different "endpoints" that are *hooked* on that node by means of small circles. For instance, the endpoints  $a(b)$  and  $a(c)$  are said to be *hooked* onto node  $a$ . In the graph, when the random walk starts (at node  $d$ ), then endpoint  $b(d)$  is said to be *checked*. Similarly, when it visits node  $b$ , then endpoints  $d(b)$ ,  $a(b)$  and  $c(b)$  are said to be checked. The same mechanism applies when the random walk visits nodes  $c$  and  $e$ .

$$E^l = \sum_j (V_j^l - V_j^{l-1})(j-1). \quad (15)$$

- To obtain  $P_u(k, l)$ , on one hand we consider the overall number of endpoints hooking onto uncovered nodes of degree  $k$  just before hop  $l$  is  $k(n_k - C_k^{l-1})$ . On the other hand, the overall number of endpoints is  $\sum_j j n_j$ , and the overall number of checked endpoints until hop  $l-1$  (inclusive) is  $\sum_j j V_j^{l-1}$ . That is, the number of endpoints not checked just before hop  $l$  is  $\sum_j j n_j - \sum_j j V_j^{l-1}$ . Therefore, we can write:

$$P_u(k, l) = \frac{k(n_k - C_k^{l-1})}{\sum_j j n_j - \sum_j j V_j^{l-1}}. \quad (16)$$

Substituting Equation 15 and 16 into Equation 14, we have that

$$C_k^l = C_k^{l-1} + \left( \frac{k(n_k - C_k^{l-1})}{\sum_j j n_j - \sum_j j V_j^{l-1}} \right) \times \sum_j (V_j^l - V_j^{l-1})(j-1). \quad (17)$$

Finally, taking into account Equations 13 and 17, we have that the total number of

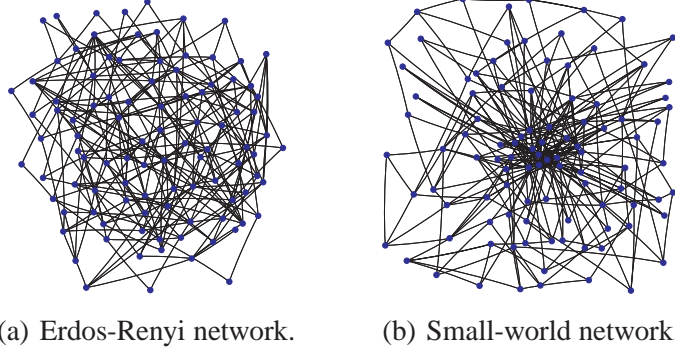


Fig. 6. In the Erdos-Renyi network most nodes have approximately the same number of links. In contrast, the small-world network is heterogeneous: the majority of the nodes have approximately the same number of links but a few nodes have a large number of them.

nodes covered after hop  $l$  is

$$C^l = \sum_k C_k^l. \quad (18)$$

#### 2.4 Average Search Length

Using the previous metric, we are now able to provide an estimation of the average search length of random walks, denoted by  $\bar{l}$ . Formally,  $\bar{l}$  is given by the following expression:

$$\bar{l} = \sum_{l=0}^{\infty} l P_f(l), \quad (19)$$

where  $P_f(l)$  is the probability that the search finishes at hop  $l$  (i.e., the probability that the search is successful at hop  $l$ , having failed during the previous  $l - 1$  hops). Let us define the *probability of success* at hop  $l$ , denoted by  $P_s(l)$ , as the probability of finding, at that hop, the node we are searching for.  $P_s(l)$  can be obtained as the relation between the number of new nodes that will be covered at hop  $l$ , and the number of nodes that are still uncovered at hop  $l$ . That is,

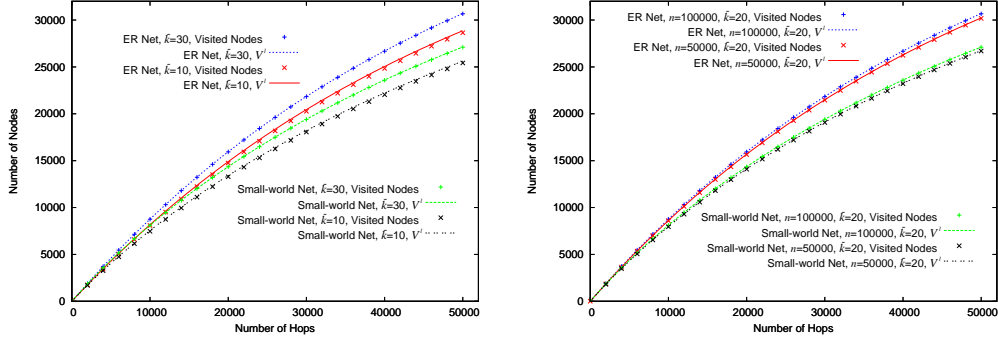
$$P_s(l) = \frac{C^l - C^{l-1}}{n - C^{l-1}}. \quad (20)$$

Now,  $P_f(l)$  can be obtained as follows:

$$P_f(l) = P_s(l) \prod_{i=0}^{l-1} (1 - P_s(i)) = \frac{C^l - C^{l-1}}{n}. \quad (21)$$

Therefore,  $\bar{l}$  can be written as





(a) Erdos-Renyi and Small-world;  $n = 5 \cdot 10^4$ ;  $\bar{k} = 10, 30$ . (b) Erdos-Renyi and Small-world;  $n = 5 \cdot 10^4, 10^5$ ;  $\bar{k} = 20$ .

Fig. 7. Visited nodes  $V^l$ .

$$\bar{l} = \frac{1}{n} \sum_{l=0}^{\infty} l (C^l - C^{l-1}). \quad (22)$$

## 2.5 Experimental Evaluation

We have run a set of experiments to evaluate the accuracy of the expressions presented in the previous subsections. The results obtained are presented in this section.

For our work, we consider two kinds of network: small-world networks (constructed as in [21]) and Erdos-Renyi networks (constructed as in [30]).

- *Small-world networks* [21,31]. In [32] it is shown that many real world networks present an interesting feature: each node can be reached from any other node in few hops. These networks are typically denoted small-world networks. The Internet, the Web, the Science collaboration graph, etc. are examples of real world networks that are consistent with this property. This kind of networks are also specially interesting for our work because here the revisiting effect commented in Section 1 is strongly present due to the uneven degree distribution. We build small-world networks using the mechanism described in [21], which leads to networks whose degree distribution follows a power-law distribution  $p_k \sim k^{-\alpha}$  (power-law networks).
- *Erdos-Renyi (ER) random networks* [30]. For two any nodes  $i, j \in V$  there is a constant probability  $c$  that they are connected ( $(i, j) \in E$ ). The resulting degree distribution is a binomial distribution  $p_k \sim \binom{n}{k} c^k (1-c)^{n-k}$ .

See Figure 6 for an illustrative example of both kinds of networks.

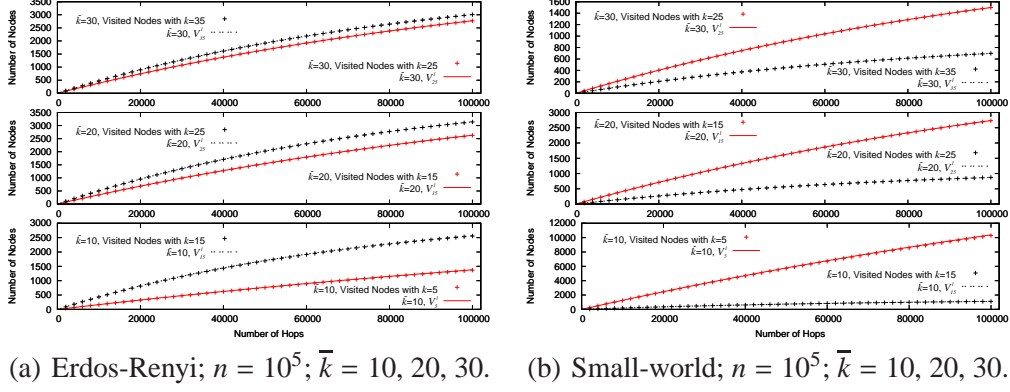


Fig. 8. Visited nodes  $V_k^l$ , for  $k = \bar{k} + 5$  and  $k = \bar{k} - 5$ .

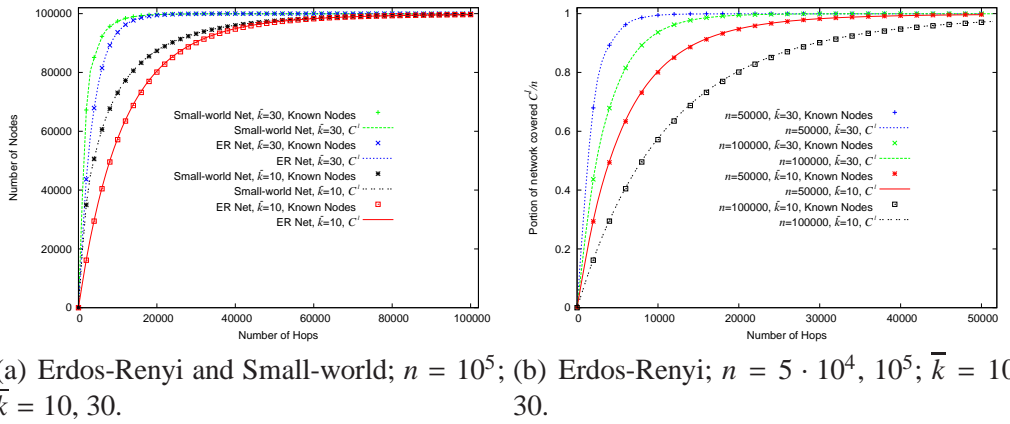


Fig. 9. Covered nodes  $C_k^l$ .

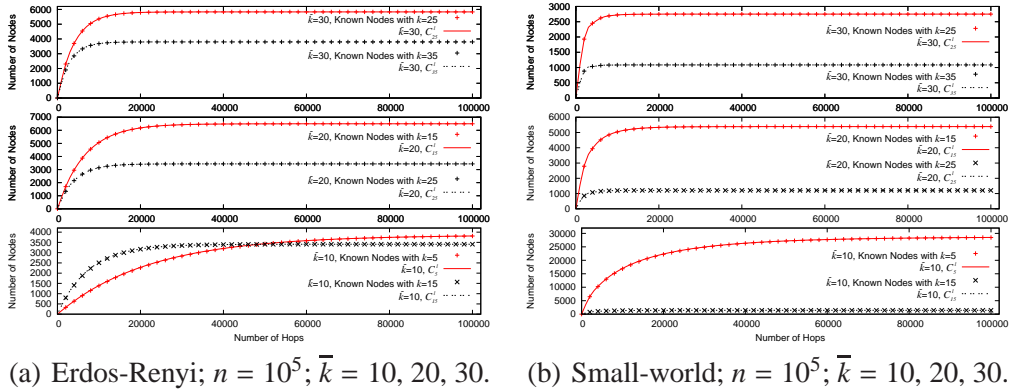


Fig. 10. Covered nodes  $C_k^l$ , for  $k = \bar{k} + 5$  and  $k = \bar{k} - 5$ .

**Number of Visited and Covered Nodes** Our first goal is to study the evolution of the network coverage by random walks in real networks.

The experiments were run on networks of two sizes,  $n = 5 \cdot 10^4$  and  $n = 10^5$  nodes. Networks were built using three different average degrees:  $\bar{k} = 10$ ,  $\bar{k} = 20$  and  $\bar{k} = 30$ . In each network we ran  $10^4$  random walks of length  $n = |V|$ . The source node

of each random walk was chosen uniformly at random. From the experiments, we obtained the average number of visited and covered nodes for each degree  $k$  at each hop  $l$ . Finally, for each network, we extracted its degree distribution  $n_k$  and apply the expressions described in the previous section to get a prediction of those values, given by  $V_k^l$  and  $C_k^l$ . Results are shown in Figures 7, 8, 9, and 10. For the sake of clarity, the experimental results are shown every 2000 hops in all figures. Model predictions, on the other hand, are drawn as lines.

Figure 7(a) shows the evolution of the number of visited nodes in ER and small-world networks of size  $n = 5 \cdot 10^4$  nodes, with two different average degrees  $\bar{k} = 10$  and  $\bar{k} = 30$ . We see that, although the length of the random walks is enough to potentially include all the nodes, only a fraction of them are visited. This happens because of the revisiting effect, and it is more evident when the number of hops increases, since the probability of revisiting grows with the number of hops. The revisiting effect is stronger in small-world networks than in random networks. The reason is the uneven distribution of the nodes degrees: there are some nodes with a very high degree that will be visited once and again by the random walk. Thus, the chances of finding new nodes at each hop are lowered faster in small-world networks than in ER networks. Also, we observe in Figure 7(a) that in networks of smaller  $\bar{k}$  the revisiting effect is stronger. Finally, Figure 7(b) shows the impact of the network size  $n$  on the amount of visited nodes. As expected, a greater  $n$  implies a lesser number of revisits for the same number of hops. In all cases, the prediction  $V^l$  of the total amount of different nodes visited is very close to the experimental results.

In Figure 8 we study the accuracy of the predictions of the amount of visited nodes of a particular degree  $k$  at each hop  $l$ ,  $V_k^l$ . We draw the results and predictions of degrees  $k = \bar{k} + 5$  and  $k = \bar{k} - 5$ , for  $\bar{k} = 10$ ,  $\bar{k} = 20$  and  $\bar{k} = 30$ . Again, it can be seen that the model predictions fit very well with the experimental results, despite the revisits and the different behavior observed for different degrees.

Figure 9 gives the results of the experiments run to study the coverage of the random walk. Figure 9(a) shows how the coverage grows faster in small-world networks than in ER networks for networks of the same average degree  $\bar{k}$ . This contrasts with the amount of visited nodes, that behave in the opposite way (see previous paragraphs). The reason is the presence of well-connected nodes, that are quickly visited during the first hops of the random walk and increase considerably the coverage because of the high amount of neighbors they have. For example, after 4000 hops, the random walk has covered about half of the small-world network with  $\bar{k} = 10$ , while in the ER network of the same  $\bar{k}$  the random walk only has covered close to 30% of the nodes. Moreover, we can see that the network average degree has also an important impact on the coverage. In both kind of networks the coverage grows faster when the average degree is higher. Besides, we observe that the difference of the coverage for both networks decreases more quickly for a higher  $\bar{k}$ . Figure 9(a) confirms the importance of the average degree, comparing the results

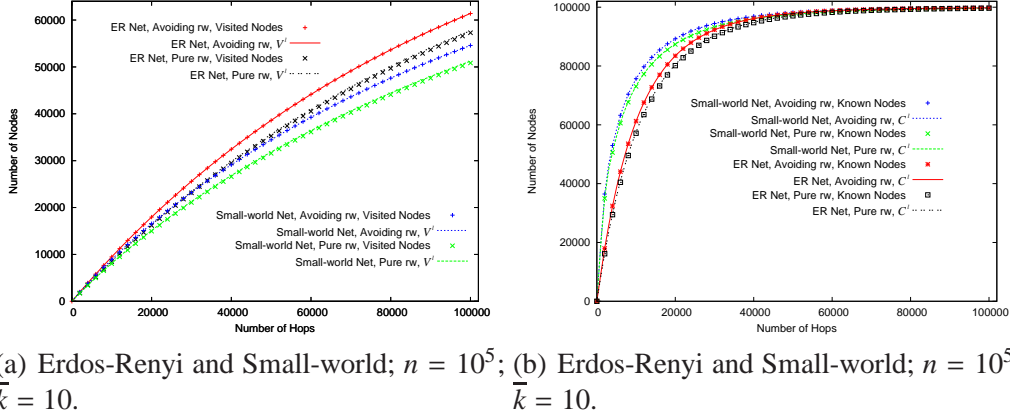


Fig. 11. Avoiding Random Walk, Visited and Covered Nodes  $V^l$  and  $C^l$ .

for networks of different size and  $\bar{k}$ . In addition, Figure 9(b) compares the results of the coverage for ER networks of different sizes and average degrees. As it could be expected, the networks of smaller size require less hops to be covered. We observe also that the average degree has an important influence on the coverage difference. The greater the average degree, the faster the coverage of both networks converges. In all cases, the  $C^l$  values given by the model predict very well how the coverage behaves and evolves.

Figure 10 allows to check the precision of the coverage predictions for different  $k$  values,  $C_k^l$ . As before, the values provided are very close to the experimental results, although the behavior of the coverage changes strongly depending on the kind of network and average degree.

Finally, we check the model accuracy for random walks that avoid the previous node, the *avoiding random walk*. As stated in Section 2.2, the avoiding random walk can be easily implemented by our model just by setting  $P_b = 0$  (see Equation 10). Results are shown in Figure 11. There we compare the coverage of pure and avoiding random walks in ER and small-world networks of size  $n = 10^5$  nodes and average degree  $\bar{k} = 10$ . Figure 11(a) confirms that, as expected, the avoiding random walk is able to visit a greater number of different nodes, as the revisiting effect is, to a certain degree, lessened. However, Figure 11(b) shows that this has little impact on the network coverage. We find that there is only a small increase on the amount of covered nodes when using avoiding random walks, for both kind of networks. Nonetheless, in all cases the  $V^l$  and  $C^l$  values given by the model are very close to real results.

**Average Search Length** For the experiments regarding the average search length we used networks whose sizes ranged from  $10^4$  to  $2 \cdot 10^5$  nodes. In each experiment we ran  $10^4$  searches, averaging the obtained results. At each search, two nodes (one corresponding to the source and the other to the destination) were chosen uniformly

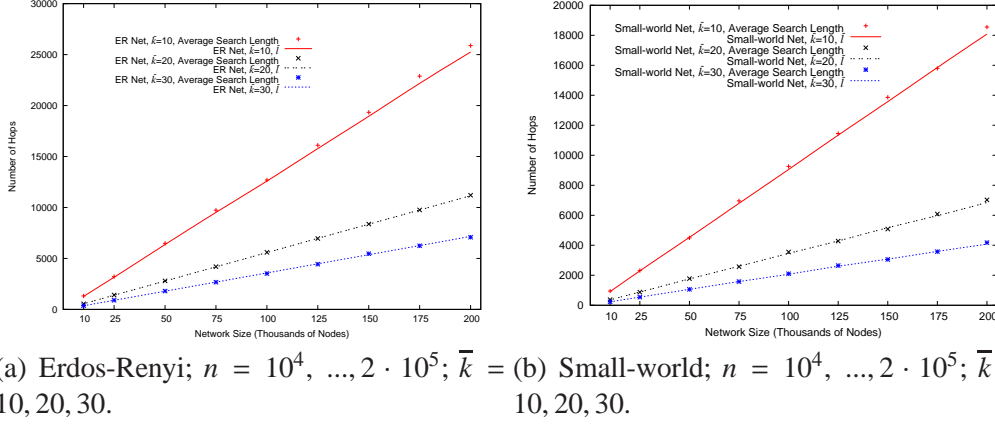


Fig. 12. Average Search Length  $\bar{l}$ .

at random. Starting from the source, a random walk traversed the network until the destination node was found (i.e., a neighbor of the destination is visited).

The first thing to note is that the average search length grows linearly with the network size in both ER and small-world networks. Besides, the average degree  $\bar{k}$  has an important effect on the results. The bigger the  $\bar{k}$ , the shortest the searches are. The reason is that a higher  $\bar{k}$  implies that at each hop more nodes of the network are discovered. Also, it can be observed in Figure 12 that the average search length is greater in ER networks than in small-world networks. This can be explained if we take into account that random walks, on average, cover more nodes in small-world networks than in ER networks (see Figures 9).

As in the previous experiments, Figure 12 also shows that our experimental results regarding the average search length correspond very close to the analytical results that were obtained.

At this point, we would like to note that, given the assumptions we made in our analytical model, it seems that the very good match achieved with the experimental results could only occur if these assumptions are correct. As a matter of fact, we have verified, in practice (see Figs. 2 and 4), that the type of networks we consider in this paper, indeed, fulfill our assumptions.

On the other hand, it is clear that if we take into account networks that do not fulfill some of our assumptions, then a certain mismatch should be expected. For instance, networks built by preferential mechanisms are known not to preserve the independence of degrees of neighbors [29]. Therefore, we should not aim for a very close correspondence between analytical and experimental results. We have performed the same experiments we ran for random and small-world networks regarding the average search length, but this time with networks built using the preferential attachment mechanism proposed by Barabási [31]. Now, we have observed that, as expected, in preferential networks our experimental results do not correspond very close to the analytical results (see Fig. 13(a)). Instead, the model seems to be con-

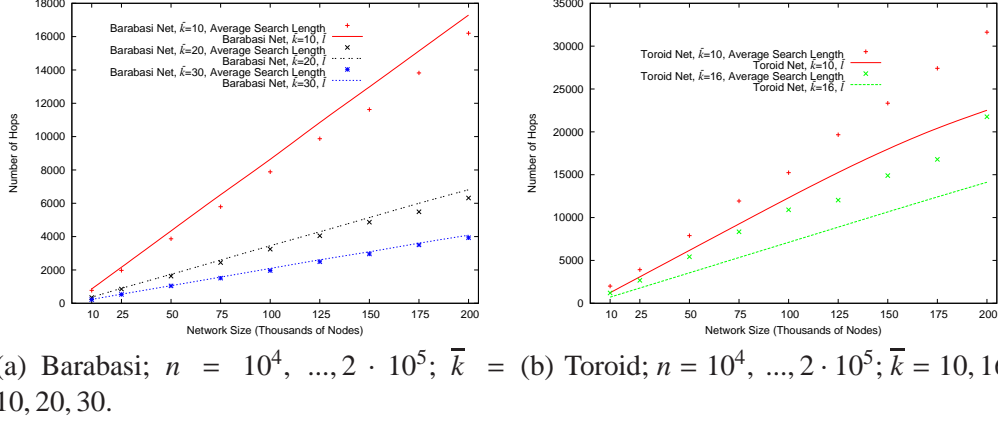


Fig. 13. Average Search Length  $\bar{l}$ , not pure random networks.

sistently pessimistic. Also, the error continuously grows with the network size.

Finally, we have tested the model against Toroidal networks of different average degrees  $\bar{k} = 10$  (5 dimensions) and  $\bar{k} = 16$  (8 dimensions). Our intention is to analyze networks which are not random at all. Results, which are shown in Fig. 13(b), show a very clear mismatch among the results predicted by the model and the actual performance of the random walk.

### 3 Duration of Searches by Random Walks

In this section, we present the second part of our model. Here we provide useful expressions that allow to predict the performance of random walks as a search tool, which is the main goal of this work. These expressions rely on the same estimation of the average search length (like the one described in the previous section), that is combined with Queuing Theory [33]. As a result, given the processing capacities and degrees of nodes, we are able to compute two key values:

- The *load limit*: the searches rate limit that the network can handle before saturation.
- The *average search time*: the average time it takes to complete a search, given the global load.

Also, we show how these expressions can be used to analyze which features a network should have so random walks have a better performance (i.e., searches are solved in less time). In particular, we focus on studying the relationship between degree and capacity distributions, showing that the minimum search time is obtained when nodes of higher capacities are also those of higher degrees.

In our analysis, networks are assumed to be *Jackson networks* [33]: the arrival of new searches into the network follows a Poisson distribution and the service at each



node is a Poisson process.

### 3.1 Searches Length and Load on Nodes

Our first step is to set the relationship between the average searches length and the system load. Each search is processed, on average,  $1 + \bar{l}$  times (once at the source node, and once at each step of the random walk). Using this, we can express the total load on all the nodes of the system,  $\lambda$ , as

$$\lambda = (1 + \bar{l}) \gamma, \quad (23)$$

where  $\gamma$  is the load injected in the system by new searches, that we assume to be known. Note that  $\lambda$  is composed of the new generated searches ( $\gamma$ ), plus the searches that move from one node to another, denoted by  $\gamma'$ . Hence,

$$\bar{l} = \frac{\gamma' + \gamma}{\gamma} - 1 = \frac{\gamma'}{\gamma}. \quad (24)$$

To compute the load on each particular node  $j$ ,  $\lambda_j$ , let us take into account that the probability that a random walk visits a node is proportional to the node's degree (see Section 2). This implies that, for each node  $j \in V$ , the load on node  $j$  due to search messages, denoted  $\gamma'_j$ , is proportional to its degree  $k_j$ . As a result, we have that there is a value  $\tau$  such that  $\gamma'_j = \tau k_j$ , for all  $j$ . Hence,  $\gamma' = \sum_j \gamma'_j = \tau d$ , where  $d$  is the sum of all degrees in the network (i.e.,  $d = \sum_k n_k k$ ). Therefore,

$$\tau = \frac{\bar{l} \gamma}{d}. \quad (25)$$

Assuming that all nodes generate approximately the same number of new searches ( $\gamma/n$ ), we can compute the average load at node  $j$  as

$$\lambda_j = \tau k_j + \frac{\gamma}{n} = \gamma \left( \frac{\bar{l} k_j}{d} + \frac{1}{n} \right). \quad (26)$$

where the first term represents the load due to search messages, and the second term to the searches generated at node  $j$ . Note that any other search generation rate model can be implemented just by changing the term  $\gamma/n$ .

### 3.2 Average Search Duration

In order to obtain the average search duration,  $T_r$ , we use *Little's Law* [33], which states that

$$r = \gamma \times T_r, \quad (27)$$

where  $r$  is the average number of *resident* searches in the network (i.e., searches that are waiting or being served), and  $\gamma$  is the average number of searches *generated* per unit of time (i.e., the arrival rate of searches). Observe that  $\gamma$  is assumed to be known. Hence, the challenge to compute  $T_r$  is to obtain  $r$ . Let  $r_j$  be the number of resident searches in node  $j$ . Then,  $r = \sum_j r_j$ .

To obtain  $r_j$ , we apply *Little's Law* again, this time individually to each node  $j$ :

$$r_j = \lambda_j \times T_r^j, \quad (28)$$

where  $T_r^j$  is the average search time at node  $j$  and  $\lambda_j$  is the average *load* at node  $j$ , which includes both searches generated at node  $j$  and searches due to messages from other nodes. Next we use that, by *Jackson's Theorem* [34] (recall we assume the network to be a Jackson network), each node  $j$  can be analyzed as a single M/M/1 queue with Poisson arrival rate  $\lambda_j$  and exponentially distributed service time with mean  $T_s^j$  (which can be computed from the node capacity, that we assume to be known). Then:

$$T_r^j = \frac{T_s^j}{1 - \rho_j}, \quad (29)$$

where  $\rho_j$  is the utilization rate and  $T_s^j$  is the average service time at node  $j$ . As  $\rho_j = \lambda_j T_s^j$ , we can write

$$T_r^j = \frac{T_s^j}{1 - \lambda_j T_s^j}. \quad (30)$$

Once we have  $\lambda_j$  and  $T_r^j$ , we can combine them to obtain

$$\begin{aligned}
T_r &= \frac{r}{\gamma} \\
&= \frac{\sum_j r_j}{\gamma} \\
&= \frac{1}{\gamma} \sum_j \lambda_j T_r^j \\
&= \frac{1}{\gamma} \sum_j \frac{\lambda_j T_s^j}{1 - \lambda_j T_s^j} \\
&= \sum_j \frac{T_s^j \left( \frac{\bar{l} k_j}{d} + \frac{1}{n} \right)}{1 - T_s^j \gamma \left( \frac{\bar{l} k_j}{d} + \frac{1}{n} \right)} \\
&= \sum_j \left( \frac{n d}{T_s^j (k_j \bar{l} n + d)} - \gamma \right)^{-1}.
\end{aligned} \tag{31}$$

That is, we have provided an expression that computes the *average search time* using the topology, the average service times of nodes, and the search arrival rate.

### 3.3 Load Limit

Implicitly, in our previous results it has been assumed that no node is overloaded (i.e.,  $\lambda_j < 1/T_s^j$  for all  $j$ ). Otherwise, the network would never reach a stable state. Thus, a key value for any network is its *load limit*: the minimum search arrival rate ( $\gamma$ ) that would overload the network, denoted by  $\gamma_o$ . Clearly,  $\gamma_o = \min_j \{\gamma_o^j\}$  being  $\gamma_o^j$  the minimum search arrival rate that would overload node  $j$ .

From Equation 26, we have that

$$\lambda_j = k_j \frac{\bar{l} \gamma}{d} + \frac{\gamma}{n}. \tag{32}$$

Also, since no node must be overloaded, it must be satisfied that

$$\lambda_j < \frac{1}{T_s^j}. \tag{33}$$

Combining Equation 32 with Equation 33 we have that, for each  $j$ , the following

Table 1  
Capacity distributions

Percentage of nodes	Processing capacity
20%	1
45%	10
30%	100
4.9%	1,000
0.1%	10,000

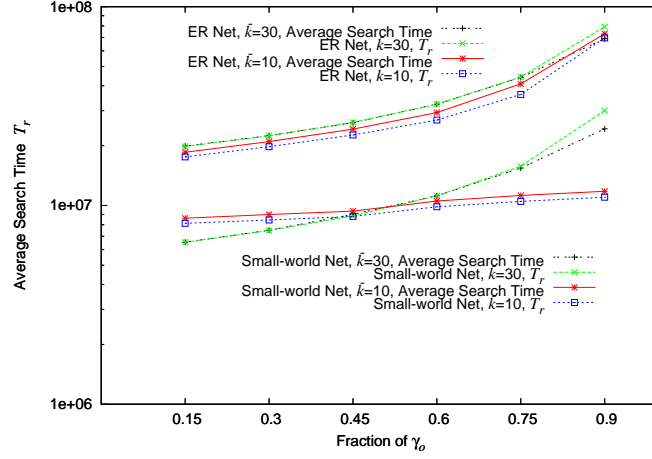


Fig. 14. Average Search Times. For the analytical values ( $T_r$ ), we used Equation 31, taking into account that  $T_s^j$  follows an exponential distribution with average  $\lambda_j$  (i.e.,  $T_s^j \sim \text{Exponential}(\lambda_j)$ ), where  $\lambda_j$  can be computed as the relation between the number of resources known and their processing capacity.

must hold:

$$\gamma < \frac{d n}{T_s^j (k_j \bar{l} n + d)}. \quad (34)$$

Therefore, the load limit for node  $j$  is

$$\gamma_o^j = \frac{d n}{T_s^j (k_j \bar{l} n + d)}, \quad (35)$$

and

$$\gamma_o = \min_j \left\{ \frac{d n}{T_s^j (k_j \bar{l} n + d)} \right\}. \quad (36)$$

### 3.4 Experimental Evaluation

**Average Search Duration** In this subsection, we present the results of a set of experiments addressed to evaluate, in practice, the accuracy of our model for the average search time. As in the previous experiments (Section 2.5), we conducted extensive simulations over ER and small-world networks. All networks are made up of  $10^4$  nodes.

In each experiment, nodes generate new searches following a Poisson process with rate  $\gamma/n$ , where  $\gamma$  is the global load on the network. When a node starts a search for a resource, it first checks whether it already knows that resource (i.e., if the node itself or any of its neighbors hold the resource). If so, the search ends successfully. Otherwise, a search message for the requested resource is created and sent to some neighbor node chosen uniformly at random. When a node receives a search message, it also verifies whether it knows the resource. If so, the search is finished. Otherwise, the search is again forwarded to another neighbor chosen uniformly at random. The experimental results are obtained by averaging the results that were obtained.

We used six different global loads ( $\gamma$ ):  $0.15 \times \gamma_o$ ,  $0.3 \times \gamma_o$ ,  $0.45 \times \gamma_o$ ,  $0.6 \times \gamma_o$ ,  $0.75 \times \gamma_o$  and  $0.9 \times \gamma_o$ , where  $\gamma_o$  is the minimum arrival rate that would overload the network (see Section 3.3). The distribution of the nodes search processing capacities  $c_i$  is derived from the measured bandwidth distributions of Gnutella [35] (see Table 1). Capacities are assigned so that nodes with a higher degree are given a higher capacity. All nodes are assumed to have the same number of resources  $w = 10,000$ . Each resource is held by one node, and all resources have the same probability of being chosen for search. The processing time at each node  $i$  follows an exponential distribution with an average service time computed as  $T_s^i = w k_i / c_i$ . This average is computed dividing the amount of resources checked for each search (the total amount of resources known,  $w(k + 1)$ , minus the resources of the node the search message came from,  $w$ ) by the node's capacity.

For each load, we measured the average search times experimentally for each network. Results are shown in Fig. 14. It can be seen that, as expected, the average search time always increases with the load, undergoing a higher growth when it approaches the maximum arrival rate. Furthermore, our experimental results show a very close correspondence with the analytical results that were obtained.

**Load Limit** We have computed the  $\gamma_o$  values for random and small-world networks with different average degrees. For each kind of network and average degree five networks were built with the capacity distribution presented in Table 1. Our goal was to observe the variation of the  $\gamma_o$  for networks of the same type and  $\bar{k}$ , and also to study the difference among the  $\gamma_o$  values depending on the network kind and average degree.

Results, which are shown in Figure 15, differ for random and small-world networks.

The first thing to note is that small-world networks can handle a greater load than random networks.

Small-world networks present variations of the  $\gamma_o$  values even for networks of the same average degree. Despite this variation, it is clear that the load limit tends to grow with the  $\bar{k}$ . The reason is that a greater  $\bar{k}$  implies a smaller global load for the same rate of queries injected to the system. Recall that the total load is given by  $(1 + \bar{l})\gamma$  (Equation 23) and that higher average degrees lead to lesser average searches lengths  $\bar{l}$  (Figure 12(b)). Hence, it is possible to perform more queries before overloading the network.

Erdos-Renyi networks however behave in a very different manner. They present very little variations of the  $\gamma_o$  values. And, more surprising, there is a small decrease of the load limit when the  $\bar{k}$  grows. This contrasts with the behavior of small-world networks. As it is shown in Figure 12(a), larger average degrees imply smaller average searches lengths and so a smaller global load. However, the  $\gamma_o$  that can be handled by the network does not change accordingly to this. The reason seems to be that in ER networks the load is more evenly distributed among nodes. This implies that low capacity nodes have to handle an important amount of searches. Besides, a greater average degree impacts on the average services times  $T_s$  of these nodes, as they know, and so they have to process, more resources per search. Hence, these nodes keep being the bottleneck of the network despite the smaller average search length, preventing the system to be able to handle a greater load.

However, it is important to recall that these results are also due to the capacity distribution used, and how it was distributed among the nodes. In small-world networks, if we assign low capacities to high degree nodes we can expect them to become bottlenecks of the network that force small  $\gamma_o$  values. In ER networks, adding more high capacity nodes could change the  $\gamma_o$  tendency so it would grow with the average degree. Exploring all these phenomena is beyond the scope of this paper.

### 3.5 *Optimal Relationship between Degree and Capacity Distributions*

In this section we show that, when there is a full correlation between the *capacity* of a node (i.e., the number of searches a node can process per time unit) and its degree, this leads to a minimal value of the average search time  $T_r$ .

Let us first state the relation we assume between the capacity  $c_j$  and the average service time  $T_s^j$  of a node  $j$ . We assume that the first is a parameter that does not depend on the degree or the number of resources known by the node, and only depends on the processor and network connection speeds. We assume that the second is a strictly increasing function of the node's degree  $f(k_j)$ . We assume that a node's service time is directly proportional to its degree and inversely proportional to its



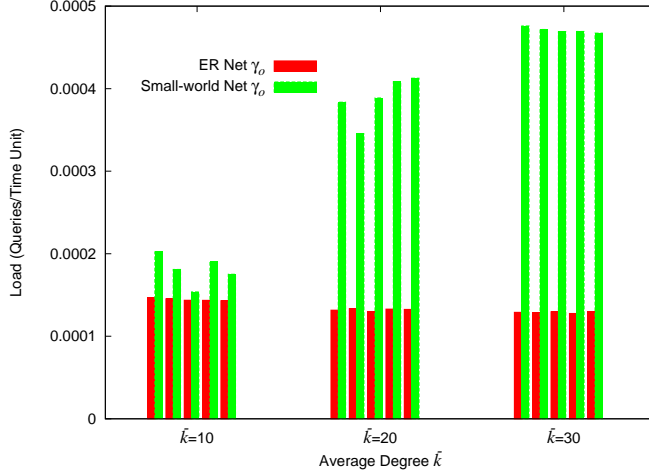


Fig. 15. Load limit  $\gamma_o$ . Five different networks are created for each network type (ER or small-world) and average degree  $\bar{k}$ . The resulting  $\gamma_o$  are shown grouped by  $\bar{k}$ .

capacity as follows:

$$T_s^j = \frac{f(k_j)}{c_j}. \quad (37)$$

Let us now consider a pair of nodes  $i, j \in V$ , such that  $k_j > k_i$  (so  $f(k_j) > f(k_i)$ ), and two possible positive capacities  $c_1$  and  $c_2$ , such that  $c_1 > c_2$ . We show that, if no other degree or capacity assignment changes, having  $c_j = c_1$  and  $c_i = c_2$  gives a smaller average search time,  $T_r$ , than the average search time  $T_r'$  with reverse assignment  $c_j' = c_2$  and  $c_i' = c_1$ .

Using Eq. 37, we obtain the following possible average service times:

$$T_{s,1}^j = \frac{f(k_j)}{c_1}; T_{s,1}^i = \frac{f(k_i)}{c_2}; T_{s,2}^j = \frac{f(k_j)}{c_2}; T_{s,2}^i = \frac{f(k_i)}{c_1}, \quad (38)$$

in which  $T_{s,1}$  are the service times obtained with the first capacity assignment and  $T_{s,2}$  are the service times obtained with the second. From the above equations, we have

$$\begin{aligned} T_{s,1}^j T_{s,1}^i &= \frac{f(k_i) f(k_j)}{c_1 c_2} \\ &= T_{s,2}^j T_{s,2}^i, \end{aligned} \quad (39)$$

and

$$\begin{aligned} T_{s,1}^i - T_{s,2}^i &= f(k_i) \frac{c_1 - c_2}{c_1 c_2} \\ &< f(k_j) \frac{c_1 - c_2}{c_1 c_2} \\ &= T_{s,2}^j - T_{s,1}^j. \end{aligned} \quad (40)$$

Let  $\lambda_i$  and  $\lambda_j$  be the loads on  $i$  and  $j$ . Since  $k_i < k_j$ , then  $\lambda_i < \lambda_j$ . Hence, from this and Eq. 40, we find that

$$\lambda_i T_{s,1}^i + \lambda_j T_{s,1}^j < \lambda_i T_{s,2}^i + \lambda_j T_{s,2}^j. \quad (41)$$

To compute the values  $T_r$  and  $T'_r$ , we use Eq. 31

$$T_r = \frac{1}{\gamma} \left( r_i + r_j + \sum_{h \neq i, h \neq j} r_h \right); \quad (42)$$

$$T'_r = \frac{1}{\gamma} \left( r'_i + r'_j + \sum_{h \neq i, h \neq j} r_h \right), \quad (43)$$

where  $r_i$  and  $r_j$  are obtained with the first capacity assignment and  $r'_i$  and  $r'_j$  with the second. Observe that  $r_h$  remains the same for any node  $h$  that is neither  $i$  nor  $j$ , because its degree, load, and capacity are just the same for both cases. Hence, if  $r_i + r_j < r'_i + r'_j$  then  $T_r < T'_r$ .

From Eqs. 28 and 30, we obtain that

$$\begin{aligned} r_i + r_j &= \frac{\lambda_i T_{s,1}^i}{1 - \lambda_i T_{s,1}^i} + \frac{\lambda_j T_{s,1}^j}{1 - \lambda_j T_{s,1}^j} \\ &= \frac{-2 \lambda_i \lambda_j T_{s,1}^i T_{s,1}^j + \lambda_i T_{s,1}^i + \lambda_j T_{s,1}^j}{1 + \lambda_i \lambda_j T_{s,1}^i T_{s,1}^j - (\lambda_i T_{s,1}^i + \lambda_j T_{s,1}^j)}, \end{aligned} \quad (44)$$

and

$$\begin{aligned} r'_i + r'_j &= \frac{\lambda_i T_{s,2}^i}{1 - \lambda_i T_{s,2}^i} + \frac{\lambda_j T_{s,2}^j}{1 - \lambda_j T_{s,2}^j} \\ &= \frac{-2 \lambda_i \lambda_j T_{s,2}^i T_{s,2}^j + \lambda_i T_{s,2}^i + \lambda_j T_{s,2}^j}{1 + \lambda_i \lambda_j T_{s,2}^i T_{s,2}^j - (\lambda_i T_{s,2}^i + \lambda_j T_{s,2}^j)}. \end{aligned} \quad (45)$$

Finally, applying Eqs. 39 and 41, we conclude that

$$r_i + r_j < r'_i + r'_j, \quad (46)$$

and hence

$$T_r < T'_r. \quad (47)$$

This proves that, for a given degree distribution, the best performance will be obtained by assigning the largest capacities to the nodes with the largest degrees. Note that we have found a condition that is necessary in order to attain the minimum possible  $T_r$ , once the degree distribution has been set. However, different degree distributions can obtain very different  $T_r$  values.

## 4 Conclusions

In this paper, we have presented an analytical model that allows us to predict the behavior of random walks. Furthermore, we have also performed some experiments that confirm the correctness of our expressions.

Some work can be carried out to complement our results. For instance, several random walks can be used at the same time, a situation that could be used to further improve the efficiency of the search mechanism. These random walks could run independently or, in order to cover separated regions on the graphs, coordinate among them in some way.

## References

- [1] C. Gkantsidis, M. Mihail, A. Saberi, On the random walk method for p2p networks, in: Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, Vol. 1, Hong Kong, 2004, pp. 148–159.
- [2] Y. Chawathe, S. Ratnasamy, N. Lanham, S. Shenker, Making Gnutella-like P2P systems scalable, in: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003), Karlsruhe, Germany, 2003, pp. 407–418.
- [3] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, in: Proceedings of the 16th International Conference on Supercomputing, New York, New York, United States, 2005, pp. 84–95.
- [4] Q. Lv, S. Ratnasamy, S. Shenker, Can heterogeneity make Gnutella scalable?, in: Revised Papers from the First International Workshop on Peer-to-Peer Systems, Cambridge, United States, 2002, pp. 94–103.
- [5] N. Bisnik, A. Abouzeid, Modeling and analysis of random walk search algorithms in p2p networks, in: Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P 2005), IEEE Computer Society, 2005, pp. 95–103.
- [6] I. Mabrouki, X. Lagrange, G. Froc, Random walk based routing protocol for wireless sensor networks, in: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools (ValueTools '07), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 1–10.
- [7] M. Alanyali, V. Saligrama, O. Savas, A random-walk model for distributed computation in energy-limited networks, in: Proceedings of the 1st Workshop on Information Theory and its Application, 2006.
- [8] S. Dolev, E. Schiller, J. L. Welch, Random walk for self-stabilizing group communication in Ad Hoc networks, IEEE Transactions on Mobile Computing 5 (7) (2006) 893–905.

- [9] C. Law, K.-Y. Siu, Distributed construction of random expander networks, in: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), Vol. 3, 2003, pp. 2133–2143.
- [10] N. Sadagopan, B. Krishnamachari, A. Helmy, Active query forwarding in sensor networks, *Ad Hoc Networks* 3 (2005) 91–113.
- [11] U. Feige, A tight lower bound on the cover time for random walks on graphs, *Random Structures and Algorithms* 6 (1995) 433–438.
- [12] J. D. Kahn, N. Linial, N. Nisan, M. E. Saks, On the cover time of random walks on graphs, *Journal of Theoretical Probability* 2 (1) (1989) 121–128.
- [13] D. Zuckerman, A technique for lower bounding the cover time, in: Proceedings of the 22nd annual ACM symposium on Theory of computing, 1990, pp. 254–259.
- [14] D. J. Aldous, Lower bounds for covering times for reversible markov chains and random walks, *Journal of Theoretical Probability* 2 (1) (1989) 91–100.
- [15] R. Motwani, P. Raghavan, *Markov Chains and Random Walks*, Cambridge University Press, 1995, Ch. 6, pp. 127–160.
- [16] G. Barnes, U. Feige, Short random walks on graphs, in: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing (STOC 93), ACM Press, 1993, pp. 728–737.
- [17] G. Brightwell, P. Winkler, Maximum hitting time for random walks on graphs, *Random Structures and Algorithms* 1 (1989) 263–276.
- [18] L. .Lovász, *Random walks on graphs: A survey*.
- [19] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks, in: Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, Vol. 1, Hong Kong, 2004, pp. 120–130.
- [20] L. A. Adamic, B. A. Huberman, R. M. Lukose, A. R. Puniyani, Search in power law networks, *Physical Review E* 64 (2001) 46135–46143.
- [21] M. E. J. Newman, S. H. Strogatz, D. J. Watts, Random graphs with arbitrary degree distributions and their applications, *Physical Review E* 64 (2) (2001) 026118–1,026118–17.
- [22] G. S. Manku, M. Naor, U. Wieder, Know thy neighbor’s neighbor: the power of lookahead in randomized p2p networks, in: Proceedings of the 36th annual ACM symposium on Theory of computing (STOC 2004), ACM Press, 2004, pp. 54–63.
- [23] V. Cholvi, P. A. Felber, E. W. Biersack, Efficient search in unstructured peer-to-peer networks, in: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, Barcelona, Spain, 2004, pp. 271–272.
- [24] M. Mihail, A. Saberi, P. Tetali, Random walks with lookahead in power law random graphs, *Internet Mathematics* 3.

- [25] A. Sinclair, Improved bounds for mixing rates of marked chains and multicommodity flow, in: *Lecture Notes in Computer Science, Proceedings of the 1st Latin American Symposium on Theoretical Informatics (LATIN 92)*, Vol. 583, Springer-Verlag, 1992, pp. 474–487.
- [26] A. Tahbaz-Salehi, A. Jadbabaie, Small world phenomenon, rapidly mixing markov chains, and average consensus algorithms, in: *Proceedings of the 46th IEEE Conference on Decision and Control*, IEEE Computer Society, 2007, pp. 276–281.
- [27] C. Gkantsidis, M. Mihail, A. Saberi, Conductance and congestion in power law graphs, *ACM SIGMETRICS Performance Evaluation Review* 31 (2003) 148–159.
- [28] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Graph structure in the web, *Computer Networks: The International Journal of Computer and Telecommunications Networking* 33 (2000) 309–320.
- [29] P. L. Krapivsky, S. Redner, Organization of growing random networks, *Physical Review E* 63 (2001) 066123.
- [30] B. Bollobás, *Random Graphs*, 2nd Edition, Cambridge University Press, 2001.
- [31] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, *Science* 286 (1999) 509,512.
- [32] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, *Reviews of Modern Physics* 74 (2002) 47–97.
- [33] W. Stallings, A practical guide to queuing analysis, 2000, available online, <ftp://shell.shore.net/members/w/s/ws/Support/QueuingAnalysis.pdf>.  
URL <http://williamstallings.com/StudentSupport.html>
- [34] G. Giambene, *Queuing Theory and Telecommunications: Networks and Applications*, Springer, 2005.
- [35] S. Saroiu, P. K. Gummadi, S. D. Gribble, A measurement study of peer-to-peer file sharing systems, in: *Proceedings of SPIE (Proceedings of Multimedia Computing and Networking 2002, MMCN'02)*, Vol. 4673, 2002, pp. 156–170.