Multiple Random Walks to Uncover Short Paths in Power Law Networks

Bruno Ribeiro[1], Prithwish Basu[2], and Don Towsley[1]

[1]University of Massachusetts Amherst
{ribeiro, towsley}@cs.umass.edu

[2]Raytheon BBN Technologies
pbasu@bbn.com

Abstract—Developing simple distributed algorithms to allow nodes to perform topology discovery and message routing using incomplete topological information is a problem of great interest in network science. Consider the following routing problem in the context of a large power law network G. A small number of nodes want to exchange messages over short paths on G. These nodes do not have access to the topology of G but are allowed to crawl the network subject to a budget constraint. Only crawlers whose paths cross are allowed to exchange topological information. In this work we study the use of random walks (RWs) to crawl G. We show that RWs have the ability to find short paths and that this bears no relation to the paths that they take. Instead, it relies on two properties of RWs on power law networks:

- The RW ability to observe a sizable fraction of the network edges;
- The near certainty that two distinct RW sample paths cross after a small percentage of the nodes have been visited.

We show promising simulation results on several real world networks.

I. INTRODUCTION

Developing simple distributed methods to allow nodes to perform topology discovery and message routing using incomplete topological information is a problem of great interest in network science. In this work we focus on network topologies characterized by highly variable node degrees and short network diameters. Social and some technological networks (e.g., peer-to-peer (P2P) overlay networks) commonly fulfill these characteristics. Moreover, nodes ultimately need to find short paths in the network by collecting limited topological information.

Consider the following routing problem in the context of a large network G. A small set of h nodes $U = \{u_1, \ldots, u_h\}$ are looking for short paths on G over which to exchange messages. Routing consists of two phases: Topology discovery and shortest path calculation. During the topology discovery phase each node must partially crawl the network in order to partially discover the network topology. Nodes have a limited crawling budget B and only crawlers whose paths intersect are allowed to exchange topological information. Upon finishing the topology discovery phase nodes in U find short paths to other nodes in U using the sampled topology. Our results do not assume a particular degree distribution
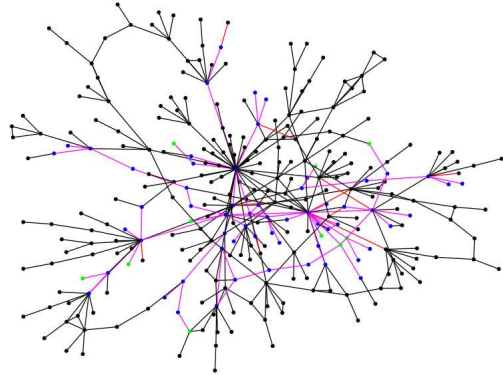
Fig. 1. Illustration of a RW sample path. Green nodes are the starting nodes, the blue nodes are nodes visited by the random walkers, and the purple edges are the edges used by the walkers to explore the graph which is a subset of $E(B, i)$.

but they apply particularly to power law networks. Simple topology discovery algorithms such as crawling the network using breadth-first search (BFS) are ill suited for the task at hand. The combination of short network diameter and large degree nodes can quickly exhaust the sampling budget B when BFS is used.

In this work each of the h nodes looking for short paths initiate a random walk (RW) on G. We consider the main cost of routing to be the cost of discovering a node, i.e., the cost of querying a node, and that nodes store the identity of all of their neighbors. We show RWs can find short paths, especially when the degree distribution is highly skewed. Our finding can be explained in the light of two RW properties when the network has a highly skewed degree distribution and high network conductance (network conductance is measured by the graph's Cheeger Constant [7]):

- The ability of a RW to observe a sizable fraction of the network edges (an edge is observed when at least one of its endpoints are visited by the RW); and
- The high probability that two RWs starting at different nodes cross after visiting a small number of nodes.

We provide promising simulation results on several real world networks. Applications of our work include the discovery of short paths between online social network users and aiding routing algorithms that use a social network to deliver messages between users (one such algorithm is Bubble Rap [5]).

A. Framework

In what follows we formalize the model. Consider an undirected graph $G = (V, E)$ with n nodes and m edges. Start h random walkers from nodes $U = \{u_1, \ldots, u_h\}$. We assume each walker takes B steps. Each walker leaves *breadcrumbs* so that one can trace its path back towards the initial node. Let $X(t, i) = (X_1^{(i)}, \ldots, X_t^{(i)})$, $i = 1, \ldots, h$, denote the sequence of nodes seen by the i-th walker at step t. Let $S(t, i)$, $i = 1, \ldots, h$, denote the set of nodes visited by the i-th walker at time t. Let $E(B, i)$ denote the set of edges that have at least one of its endpoints in $S(B, i)$. Upon exhausting its budget RW, i outputs $X(B, i)$, $S(B, i)$, and $E(B, i)$.

B. Naive Routing

Note that if two RWs i and j cross each other then $S(B, i) \cap S(B, j) \neq \emptyset$, $i, j = 1, \ldots, h$ (i.e., RW i must have found RW j's breadcrumbs or vice-versa). Without loss of generality assume walker i found j's breadcrumbs. A naive way to construct a path between u_i and u_j is to route messages retracing all RW steps using the breadcrumbs.

C. Drawbacks of Naive Routing

The naive algorithm presented above is quite inefficient. Random walks are not particularly good at finding short paths (as seen in Section V); unless there are few paths between pairs of nodes, e.g., G is a tree. A random walk with erased loops, also known as a loop-erased random walk, samples all possible spanning trees of G with equal probability [13]. Although loop erased RW results in shorter paths between u_i and u_j than *Naive Routing*, the improvement is unlikely to have great impact on the end result.

D. The True Power of RWs in Finding Short Paths

The ability of RWs to find short paths bears no relation to the paths that they take. Instead, it relies on RW properties often present in power law networks, namely large degree variability. We propose a model that allows us to obtain closed form approximations of the number of edges and vertices discovered by a RW. The main results found on Section II are summarized here:

- $|E(B, i)| \approx \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} B$, where $B \ll n$, $i = 1, \ldots, h$, and $\langle k \rangle$ and $\langle k^2 \rangle$ are the first and second moment of the node degrees in original graph G, respectively.
- Under mild conditions two RW sample paths cross, with high probability.

These observations motivate our use of RWs to uncover short paths on G.

E. Outline

The outline of this paper follows. Section II characterizes the subgraph obtained by the i-th random walker and the probability that two RW sample paths cross. Section III presents the Random Walk Short Path (RWSP) algorithm which uses RWs to uncover short paths on G. Section IV presents our results on the performance of RWSP on real world networks. Finally, Section V presents the related work.

Network	nodes	edges	$\langle k \rangle$	q
Flickr [10]	1.7M	31.1M	18.1	943.4
AS Net. [3]	26.5K	107K	4	279.24
Livejournal [10]	5.2M	9.8M	18.9	154
Enron [1]	37K	368K	10	140
Gnutella [12]	62.5K	296K	4.7	10.6
Power Grid [15]	5K	13K	2.7	2.9

TABLE I
SUMMARY OF NETWORKS

II. RW SUBGRAPH PROPERTIES

In this section we show that a random walk is able to collect many edges from G by visiting a relatively small number of nodes. Let n be the number of nodes and $B = \beta n$ be the RW budget. Let $E(B, i)$ denote the set of edges that have at least one of its endpoints visited by the i-th RW after B steps. We will see that $|E(B, i)|$ is much larger than B if node degrees have large variance. In what follows we assume $n \gg 1$ and $B \ll n$.

We avoid the intricacies of the interaction between the RW and the graph topology by assuming that G is drawn uniformly at random from the ensemble of all possible graphs with the same degree sequence as G. This is the so-called configuration model [8] and provides us with a simplified model of G that has a simple topology structure. The configuration model assumption is widely used in the network science literature. It allows us to obtain closed form approximations of the measures of interest that match well our simulations over real world networks (as seen later in this section). These are the main results of this section (for $B \ll n$):

- $|S(B, \cdot)| \approx B$, i.e., the set of nodes visited by the RW scales linearly with B. The RW tends to collect only new nodes at the beginning of the walk.
- $|E(B, i)| \approx qB$, where $q = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}$, and $\langle k \rangle$ and $\langle k^2 \rangle$ are the first and second moment of the node degrees in original graph G, respectively.
- We provide conditions under which our algorithm (denoted RWSP) finds path between any pair of nodes (u_i, u_j), $\forall i, j$, with high probability.

A network with large highly skewed degree distribution has large q. For instance, in the Flickr social photo sharing website $q = 943.4$, in the Enron email network this value is $q = 140$, and in the Livejournal blog posting social network this value is $q = 154$. Table I presents values of q for several real world networks. In such networks the edges in $E(B, i)$ spans a sizable fraction of the edges of G even when $\beta = B/n$ is small.

A. RW observes a sizable fraction of the edges of G.

In what follows we find a closed approximation formula for the number of edges in $E(t, i)$, $t \leq B$. Some of the following equations are valid for any RW starting at a node in U. Hence, in what follows we drop parameter i from our notation when there is no ambiguity, i.e., in equations that are valid for any of the h RWs we write $E(t)$ in place of $E(t, i)$ and $S(t)$ in place of $S(t, i)$. Let $S(t)$ denote the set of nodes visited by

a RW at time step t. As we are interested in large graphs, we assume $n \gg 1$. Let $\langle k \rangle$ and $\langle k^2 \rangle$ denote the first and second moments of the degree distribution of G, respectively. Let $\tau = t/n$ and $n_e(\tau) = \langle |E(\tau n)| \rangle$ be the average number of edges with an endpoint in $S(\tau n)$, where $\langle \cdot \rangle$ is the expectation operator. We provide a closed form mean field approximation for $n_e(\tau)$ when $\tau \ll \langle k \rangle^2 / \langle k^2 \rangle$ in Section II-C:

$$n_e(\tau) = 2m \left(1 - \exp \left(-\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2} \tau \right) \right). \qquad (1)$$

and

$$\langle |S(\tau n)| \rangle = \frac{n \langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \left(1 - \exp \left(-\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2} \tau \right) \right).$$

Figures 2 and 3 compare our mean field approximation in (1) with simulation results for the Gnutella and Flickr networks. The Gnutella network cannot be considered a power law network as its maximum degree is quite small (one hundred), but our approximation works for any degree distribution not only for power law distributions. Please refer to Table I for characteristics of these two networks.
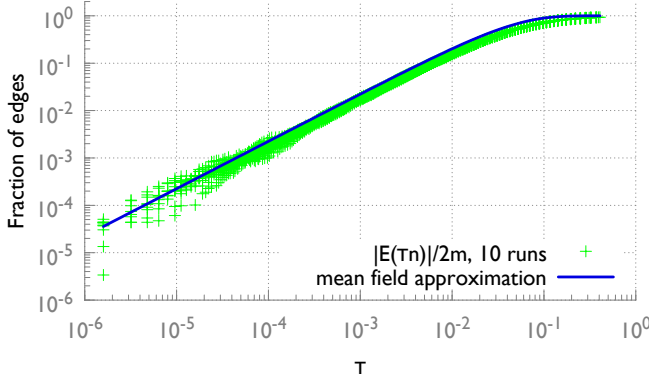

Fig. 2. $n_e(\tau)$ **in the Gnutella network.** This graph shows the fraction of edges in $E(\tau n)$ predicted by our mean field approximation against real simulation values. The simulation results of 10 runs are shown in a scatter plot of green crosses. The mean field approximation in equation (1) is shown as a blue line. In the Gnutella network $q = 10.6$, which is not as large as in a "true power law" network.

To see how $n_e(\beta)$ grows with $\beta = B/n$ let's decompose (1) into its Taylor series expansion

$$n_e(\beta) = 2m \left(\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2} \beta + O(\beta^2) \right)$$
$$= \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \beta n + O(\beta^2), \qquad (2)$$

as $\langle k \rangle = 2m/n$. For $\beta \ll 1$, (2) yields

$$n_e(\beta) \approx \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \beta n. \qquad (3)$$

In a power law network as $n \to \infty$, the probability that a node has degree k satisfies

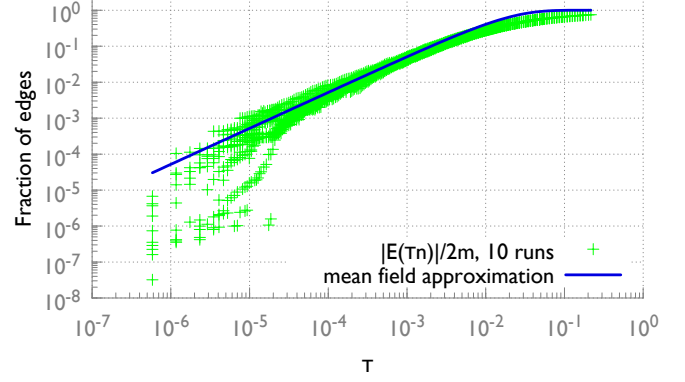$$p_k \propto (\alpha - 1)k^{-\alpha}. \qquad (4)$$

Fig. 3. $n_e(\tau)$ **in the Flickr network.** Same as Figure 2 over the Flickr network. In the Flickr network $q = 943.4$.

For power law degree distributions the approximation in (3) yields

$$n_e(\beta) = \begin{cases} \infty & \text{if } \alpha \le 3 \\ \beta n (\alpha - 3)^{-1} & \text{if } \alpha > 3. \end{cases}$$

While $n_e(\beta) < \infty$ in any real world network, the above equation provides an invaluable hint that $n_e(\beta)$ can be quite large in real life. Indeed, in the Flickr network we observe

$$n_e^{\text{flickr}}(\beta) \approx 943.4 \times 1,700,000 \times \beta,$$

which means that by the end a random walk, collects on average $943.4\beta n$ distinct edges. Table I shows the value of q for several real world networks. Before we proceed to derive equation (1), we show that under certain conditions a RW starting at u_i, $i = 1, \ldots, h$, visits a node that is also visited by RW j, $\forall j \ne i$, with high probability.

B. Two RW sample paths cross with high probability.

In what follows we show that under certain conditions, RWs starting at nodes u_i and u_j, $\forall i, j$, cross with high probability. Assume, without loss of generality, that RW i completes before we start RW j. The intuition behind our result is that for sufficiently large $|E(B, i)|/2m$, $S(B, i)$ functions as an attractor for the j-th RW, such that with high probability RW j hits the set $S(B, i)$ before B steps. The above reasoning, however, cannot be applied to a RW that has a long transient; for instance a RW on a graph shaped like dumbbell likely will not satisfy our conditions.

We are interested in the hitting time of RW j into the set $S(B, i)$. Let $X(B, j) = \{X_1^{(j)}, \ldots, X_B^{(j)}\}$ be the set of nodes visited by RW j. Then

$$P[X \cap S(B, i) = \emptyset]$$

denotes the probability that RW j and RW i do not meet.

Unfortunately obtaining hitting times to $S(B, i)$ in closed form is a hard problem. Hence, we provide the following approximation, which assumes that exists $\Delta \ll B$ such that $\lambda_2^\Delta \ll |E(B, i)|/2m$, where λ_2 is the second largest

eigenvalue of the RW transition probability matrix (see Lovasz [7]). A sufficient condition for small λ_2 is for G to have a large network conductance (more details about the relationship between network conductance and λ_2 can be found in Lovasz [7]). The probability that RW j starts at node $u \in V$ and reaches node $v \in S(\beta n, i)$ (with degree k_v) in Δ steps is [7]

$$P[X_{w+\Delta}^{(j)} = v | X_w^{(j)} = u] = \frac{k_v}{2m} + O(\lambda_2^{\Delta_n}).$$

Thus, the probability that RW j starts at some node $X_w^{(j)}$ and reaches a node in the set $S(\beta n, i)$ is approximately

$$P[X_{w+\Delta}^{(j)} \in S(\beta n, i) | X_w^{(j)}] \approx |E(B, i)|/2m, \qquad (5)$$

as we assume that the $O(\lambda_2^{\Delta_n})$ term is negligible in respect to $|E(B, i)|/2m$. Note that the probability that RW j at steps $1, \Delta + 1, 2\Delta + 1, \ldots$ does not hit $S(B, i)$ is higher than this probability considering all $1, 2, \ldots, B$ steps. Thus,

$$P[X \cap S(B, i) = \emptyset] \leq P[\{X_1, X_{\Delta+1}, \ldots\} \cap S(B, i) = \emptyset].$$

Application of (5) and the strong Markov property yields

$$P[X \cap S(B, i) = \emptyset] \leq \left(1 - c\frac{n_e(\beta)}{2m}\right)^{\lfloor \beta n/\Delta \rfloor}. \qquad (6)$$

Substituting (2) into (6) yields

$$P[X \cap S(B, i) = \emptyset] \leq \left(1 - c\left(\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2}\beta + O(\beta^2)\right)\right)^{\lfloor \beta n/\Delta \rfloor}.$$

As $n \gg 1$ and for $\langle k^2 \rangle < \infty$, (6) yields

$$P[X \cap S(B, i) = \emptyset] \ll 1.$$

Hence, if the condition stated in (5) is satisfied then RWs i and j must meet with high probability. In what follows we provide a derivation of (1).

C. Mean field analysis of the number of edges in $E(t)$.

Let $\tau = t/n \ll \langle k \rangle^2/\langle k^2 \rangle$ and let $n_e(\tau) = \langle |E(\tau n)| \rangle$ be the average number of edges in $S(\tau n)$. As G follows a configuration model, by following an edge, a RW reaches a node v with degree k_v with probability proportional to k_v. Hence, node v has average degree larger than $\langle k \rangle$. This is a classical case of the inspection paradox, which here means that the probability of sampling a node v with degree k_v is proportional to k_v and thus the average degree of a sampled node is $\langle k_v \rangle = \langle k^2 \rangle/\langle k \rangle$. If $v \notin S(t)$ we are adding at most $k_v - 1$ edges to $E(t)$, (the minus one is a consequence of the need to remove the edge the RW followed to reach v which was already in $E(t)$). If G was a random configuration graph, as $t \ll n$ and $n \gg 1$, the probability that any of the $k_v - 1$ edges are already in $E(t)$ is negligible. Thus, by adding v we are adding $q = \langle k_v \rangle - 1$ edges to $E(t)$.

Assume that the RW samples edges uniformly at random, we can write the following expression for n_e

$$n_e(\tau + 1/n) - n_e(\tau) = qP[v \notin S(t, i)] = q\left(1 - \frac{n_e(\tau)}{2m}\right),$$

where $n_e(\tau)/2m$ is the probability that a randomly sampled edge is in $E(t)$. Multiplying both sides of the above equation by n and letting $n \to \infty$ yields

$$\frac{n_e(\tau + 1/n) - n_e(\tau)}{1/n} = nq\left(1 - \frac{n_e(\tau)}{2m}\right)$$

$$\frac{dn_e(\tau)}{d\tau} = \frac{q}{\langle k \rangle}(2m - n_e(\tau))$$

with the boundary condition $n_e(0) = 0$. The solution to the above differential equation is

$$n_e(\tau) = 2m(1 - e^{-q\tau/\langle k \rangle}),$$

concluding our derivation.

Noting that $\langle |S(\tau n)| \rangle = n_e(\tau)/q$ and following the above steps one derives

$$\langle |S(\tau n, i)| \rangle = \frac{n}{\langle k^2 \rangle - \langle k \rangle}\left(1 - \exp\left(-\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2}\tau\right)\right).$$

We omit the above derivation for conciseness as it closely follows the one shown for (1).

III. THE RANDOM WALK SHORT PATH (RWSP) ALGORITHM

.

The RWSP algorithm is as follows: Consider h independent RWs, numbered from 1 to h. Let $W(i) \subseteq \{1, \ldots, i-1, i+1, \ldots, h\}^{h-1}$ denote the indices of the set of RWs that have crossed the path of RW i, $i = 1, \ldots, h$. Let $C(i)$ denote the set of nodes that have been visited by RW i and at least one other RW $j \neq i$. Initialize $W(i) = \emptyset$ and $C(i) = \emptyset$. For each walker $i = 1, \ldots, h$ do:

- Start walker i at node $u_i \in U$ with budget B.
- At time step t update the "explored" nodes $S(t, i)$ and edges $E(t, i)$ discovered by RW i.
- If the t-th visited node, x_t, has already been visited by RWs with indices W' and $i \neq W'$ do:
  1) $W(i) \leftarrow W(i) \cup W'$;
  2) $C(i) \leftarrow C(i) \cup \{x_t\}$;
  3) $\forall w \in W', W(w) \leftarrow W(w) \cup \{i\}$;
  4) $\forall w \in W', C(w) \leftarrow C(w) \cup \{x_t\}$.
- Proceed to the next step $t \leftarrow t + 1$ until $t = B$.
- At the end of walk nodes that found each other trade the topology each node discovered, G'. This exchange can be facilitated by the contact nodes in the set C.
- Let $G^\star(i) = \bigcup_{r \in W(i) \cup \{i\}} G'(B, r)$ denote the union of all subgraphs known to u_i. Compute the minimum spanning tree $T(i)$ on $G^\star(i)$ starting from u_i.

Node u_i uses $T(i)$ to route its messages to nodes u_j on G, $\forall j \in W(i)$. Figure 1 shows a snapshot of the algorithm. In what follows we present our simulation results.

IV. Results

This section presents our results when simulating RWSP on real world networks. Table I summarizes the datasets used in our simulations. Figure 4a show the fraction of RW shortest path lengths on $G^\star$ defined in Section III (X axis) between $h = 4$ nodes vs. the true shortest path lengths (Y axis) [blue/gray matrix]. We choose $h = 4$ walkers because as we increase k our results get better when B is kept constant. We also evaluate different values of h, $h \in \{2, 6, 8, 16\}$, rescaling B such that for a constant $\beta > 0$, $B = \beta n / h$. The results for different values of h are similar to those presented here, thus due to space constraints we omit them.

In Figure 4a the yellow bars next to the Y axis show the true all pairs shortest path length distribution. For instance, in LiveJournal's most of the shortest paths are between 5 and 6 hops. The RW sampling budget is $B = 2.5\%$ of the nodes in the graph. A total of $10,000$ runs were used to produce the averages seen in the graph. The axis marking INF refers to the fraction of times that a node could not reach another node (due to disconnected components). In all runs the RWSP was able to find nodes in the same component (we condition the RWSP to start in the giant component). We also tested RWSP over other networks with similar parameters.

Our results are very promising. For Livejournal (Figure 4a) more than 60% of the RWSP paths of any given length are the shortest paths. Moreover, more than 90% of the RWSP paths of any given length are within one hop of the shortest paths.

We also test RWSP on Flickr, AS graph, a snapshot of the Gnutella network, Enron email dataset with similar results. On Flickr (Figure 4b) ($B = 0.0125\,n$) more than 85% of the RWSP paths of any given length are the shortest paths; also, all paths are within 2 hops of the shortest paths. On Gnutella (Figure 4c) ($B = 0.0125\,n$) network we observe more than 60% of RWSP paths of all lengths to be within 2 hops of the shortest paths. For the AS graph (Figure 4d) (RWSP with $B = 0.05\,n$), more than 65% of RWSP paths of all lengths are the shortest paths; the majority ($> 95\%$) of RWSP paths now counting over all lengths are the shortest paths. On Enron (Figure 4e) ($B = 0.05\,n$) we have a result as good as the one for the AS graph.

Figure 4f shows that the only graph in which the RWSP does not perform well, the Power Grid network. The Power Grid network is not a power law graph and has the largest diameter of any of the previous networks, thus, we expected RWSP to perform poorly. Consulting Table I we observe that $(\langle k^2 \rangle - \langle k \rangle)/(\langle k \rangle) = 2.9$ is small and thus G' spans only a small fraction of the original graph, making the task of finding a long path close to impossible.

V. Related Work

In the context of computer networking, random walks have been used to deliver messages in ad-hoc wireless networks but suffer from excessive delays due to long hitting times [14], [9]. The social network structure has been explored to deliver messages between users in Bubble Rap [5]. Wireless *ad-hoc* and sensor networks are unlikely to benefit from RWSP as we showed in our previous work [4]. In [4] we show that for network topologies with node degrees concentrated around the mean r, $H_{max} \approx \frac{n \log n}{r^2}$. A corollary to the above is applicable to random geometric networks at the critical radius [11]; in that regime, $r \approx \log n$, and therefore, $H_{max} = O(n)$. Dumbbell shaped networks also present a challenge to RWSP as the worst case hitting times could have a $O(n^3)$ scaling law [6] . However, the hitting time to a *subgraph* drops sharply when the size of the subgraph increases in comparison to the hitting time to a specific node [2], and thus RWSP can be applicable even in some scenarios where the node-to-node hitting time is large.

References

[1] Y. Yang B. Klimmt. Introducing the enron corpus. In *CEAS conference*, 2004.

[2] Prithwish Basu and Saikat Guha. Effect of limited topology knowledge on opportunistic forwarding in ad hoc wireless networks. In *WiOpt*, pages 237–246. IEEE, 2010.

[3] Caida as graph from november 5 2007.

[4] Chi-Kin Chau and P. Basu. Analysis of latency of stateless opportunistic forwarding in intermittently connected networks. *IEEE/ACM Transactions on Networking*, 19(4):1111–1124, 2011.

[5] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 241–250. ACM, 2008.

[6] L. Lovasz. Random walk on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:1–46, 1993.

[7] Laszlo Lovasz. Eigenvalues of graphs (http://www.cs.elte.hu/~lovasz/eigenvals-x.pdf).

[8] Tomasz Luczak. Sparse random graphs with a given degree sequence. *Random Graphs*, 2:165–182, 1992.

[9] Issam Mabrouki, Xavier Lagrange, and Gwillerm Froc. Random walk based routing protocol for wireless sensor networks. In *Proc. of the 2nd ValueTools*, pages 71:1–71:10, 2007.

[10] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proc. of the IMC*, October 2007.

[11] Sanatan Rai. The spectrum of a random geometric graph is concentrated. *Journal of Theoretical Probability*, 20(2):119–132, 2007.

[12] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 2002.

[13] Oded Schramm. Scaling limits of loop-erased random walks and uniform spanning trees. *Israel J. Math.*, 118:221–288, 2000.

[14] Sergio D. Servetto and Guillermo Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *Proc. of the WSNA*, pages 12–21, 2002.

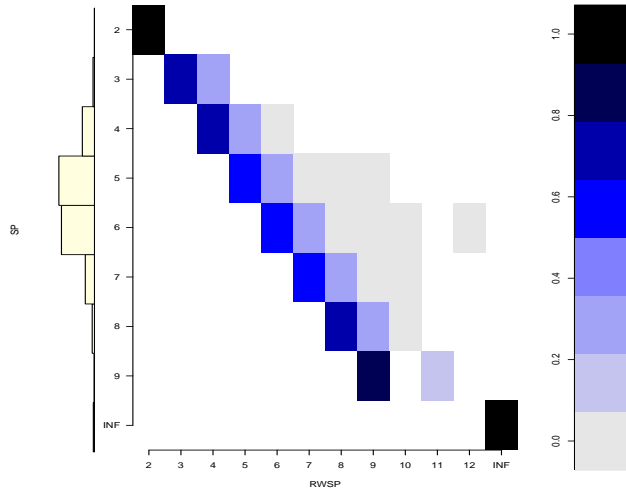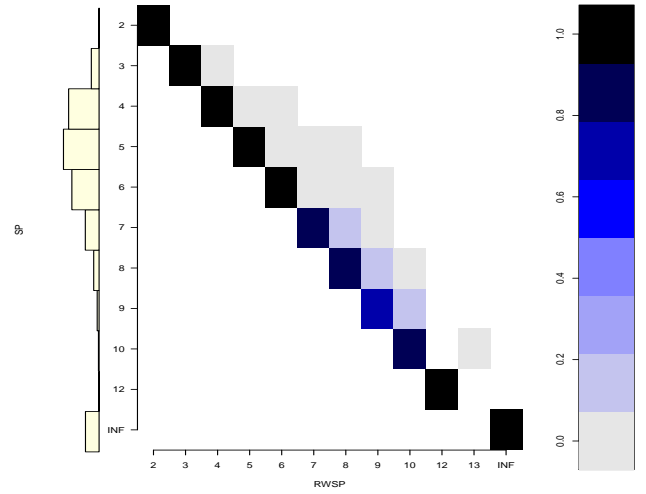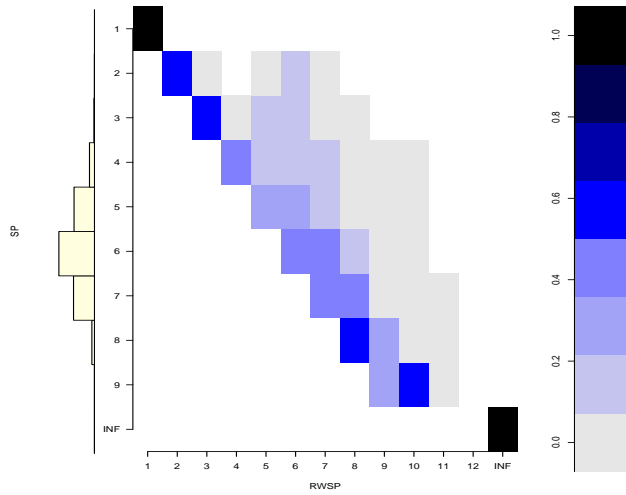[15] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.
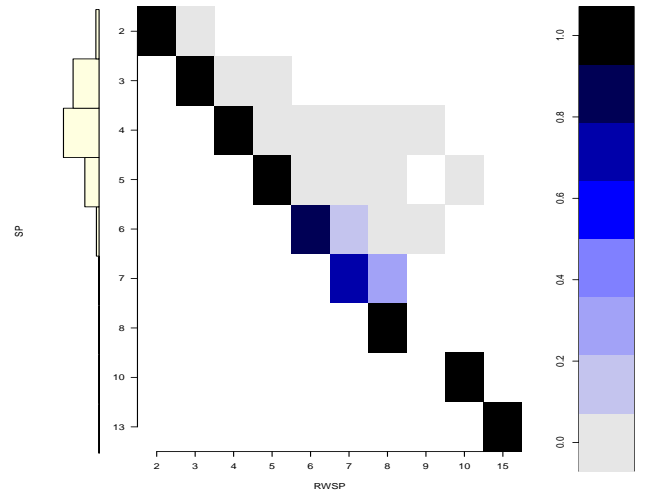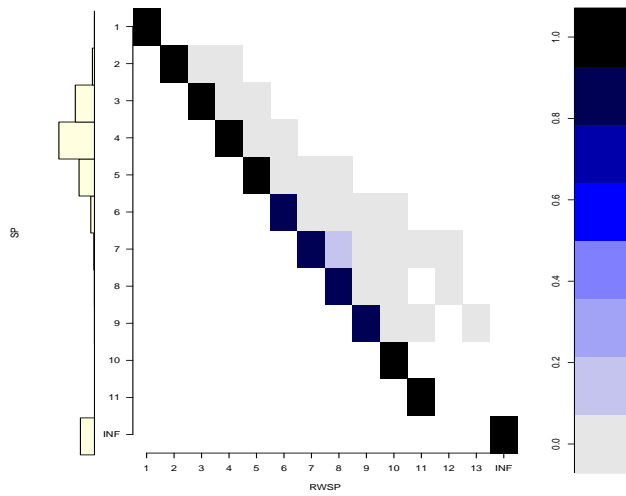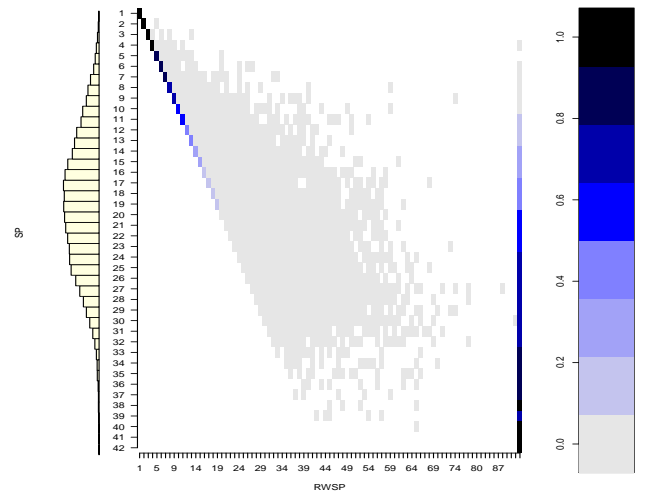
(a) LiveJournal ($h = 4$, $B = 0.025\,n$)

(b) Flickr ($h = 4$, $B = 0.025\,n$)

(c) Gnutella ($h = 4$, $B = 0.05\,n$)

(d) AS graph ($h = 4$, $B = 0.0125\,n$)

(e) Enron ($h = 16$, $B = 0.0125\,n$)

(f) Power Grid ($h = 4$, $B = 0.05\,n$)

Fig. 4. **RWSP vs. Shortest Paths.** h walkers and sampling budget B. Fraction of RW shortest path lengths (X axis) between nodes vs. true shortest path lengths (Y axis) [blue/gray matrix]. Yellow bars next to Y axis show true shortest path lengths distribution. The axis marking INF refers the fraction of times that a node could not reach another node (due to disconnected components).