A Least-Resistance Path in Reasoning about Unstructured Overlay Networks*

Giorgos Georgiadis and Marina Papatriantafilou

Department of Computer Science and Engineering, Chalmers University of Technology, S-412 96 Göteborg, Sweden
Fax: +46-31-7723663
{georgiog,ptrianta}@chalmers.se

Abstract. Unstructured overlay networks for peer-to-peer applications combined with stochastic algorithms for clustering and resource location are attractive due to low-maintenance costs and inherent fault-tolerance and self-organizing properties. Moreover, there is a relatively large volume of experimental evidence that these methods are efficiency-wise a good alternative to structured methods, which require more sophisticated algorithms for maintenance and fault tolerance. However, currently there is a very limited selection of appropriate tools to use in systematically evaluating performance and other properties of such non-trivial methods.

Based on a well-known association between random walks and resistor networks, and building on a recently pointed-out connection with peer-to-peer networks, we tie-in a set of diverse techniques and metrics of both realms in a unifying framework. Furthermore, we present a basic set of tools to facilitate the analysis of overlay properties and the reasoning about algorithms for peer-to-peer networks. One of the key features of this framework is that it enables us to measure and contrast the local and global impact of algorithmic decisions in peer-to-peer networks. We provide example experimental studies that furthermore demonstrate its capabilities in the overlay network context.

1 Introduction

A commonly used classification of peer-to-peer resource-sharing networks distinguishes them in *structured* and *unstructured*, based on their approach to the resource-lookup problem [1] and whether they use a distributed data structure to navigate the network. An unstructured approach like flooding is easy to implement but can also be the cause of quite challenging problems such as high bandwidth load. However, several research results [2,3,4,5,6] indicate that a promising way to fix such shortcomings is through statistical methods such as the *random walker* technique, where the query messages are considered to perform random walks on the overlay. An added bonus of these techniques are the inherent, "built-in" *fault-tolerance* properties they exhibit, being resilient to failure of links or nodes.

Recently many authors pointed out the usefulness of non-trivial random walkers [7,8,9] in construction and querying of *interest-based networks*, where nodes sharing the same interests are closer together than others. However, these studies currently lack

* Research supported by the Swedish Research Council, contract number 60465601.

H. Sips, D. Epema, and H.-X. Lin (Eds.): Euro-Par 2009, LNCS 5704, pp. 483–497, 2009.

a unifying framework for the comparative study of the various random walks on different overlay networks. It is also the case that many significant results for non-trivial random walks come in context with physics-related phenomena (cf. [10] and references therein), and therefore cannot be applied directly in an overlay network context. However, there is a strong connection between random walks and electric circuit theory, especially *resistor networks* [11]. By focusing on unstructured overlays that use random walks to perform queries[1], we can translate them in terms of elementary electric circuit terms and then use established tools to do the analysis in a simplified manner. Bui and Sohier in [12,13] have made significant first steps in exploring this connection to resistor networks, along with other authors, albeit for simpler walks (e.g. [14] and references therein).

In this work we aim at connecting, refining and adapting distinct techniques that involve generalized random walks and electric networks in a peer-to-peer network context, as well as providing a methodology to help address, in a way both analytical and systematic, questions such as: What is the relation between random walks and the underlying overlay? Which random walks are better suited for which networks? How can we improve a random walk and which metrics can we use to evaluate its performance? Which applications are favored by various random walks and in which context? How can we model known and recurring problems of peer-to-peer networks using the electric network paradigm?

We expand previous work on the connection among random walks and electric circuits and we propose a framework to facilitate the analysis of random walk properties in the context of topologies. Central in this framework is the notion of a biased random walk upon locally stored information, which enables the use of relevant measures of interest for network navigation. We call this framework REPO: resistor-network-based analysis of peer-to-peer overlays. Using REPO it is possible to study a broad set of topics such as content replication efficiency, overlay construction for interest-based networks, load balancing, the effect of failures, as well as local policies, bounded resource handling issues and more.

Along with our framework, we present a basic set of tools that demonstrate its intuitiveness and we complement these tools with an illustrative study that addresses several of the key questions mentioned above. In particular, we (i) identify scenarios that differentiate a random walker's behavior in different peer-to-peer overlays, (ii) reason about options for improvement of the random walker's query efficiency and the underlying network's robustness, (iii) connect local policies to global phenomena observed in peer-to-peer overlays using resistor networks, (iv) address recurring problems in peer-to-peer networks using electric theory metrics, (v) suggest novel, analytic ways of employing established techniques in a peer-to-peer context. Let us note that the framework is expressed in a way that enables it to be used in various contexts regarding the overlay network and statistical methods, and can also address construction issues e.g. of interest-based networks [8].

Other Related Work. In [2], Adamic et al present an algorithm that performs queries with a random walker on an overlay following a power-law degree distribution,

[1] In the rest of the text the terms "query" and "random walker" are used interchangeably.

directing it at every step to the highest degree neighbor. Using mean field analysis they estimate the mean number of steps needed for the query but the fact that they use deterministic random walks prohibits them from using randomized techniques to speed up the lookup process, also causing obvious overloading problems to the highest degree nodes. Lv et al [15] perform non-deterministic, simple random walks using multiple walkers and by applying content replication along the lookup path they aim at speeding up consecutive lookups but they focus only on extensive simulations. In [3], Ata et al study experimentally a biased random walk similar to what we use to illustrate REPO, along with a version that directs the walker probabilistically towards the lowest connected nodes, and find cases where both of them are useful. Gkantsidis et al [4] use simple random walks both for querying and overlay construction, and show that they can decrease significantly the cost of flooding in many practical cases of unstructured overlay networks. Fraigniaud et al in [8] are the first, to the extent of our knowledge, to use a non-trivial random walk to build an overlay network. By using a deterministic random walk that makes transitions based on the node's degree in order to locate information, they create links between the issuer of the query and the answering node, resulting in an *interest network* where nodes are connected based on their interests. This deterministic random walk (see also [2]) may have some overloading problems, as mentioned by the authors; when used in interest-based overlay construction, it creates a network with heavy tail degree distribution similar to a power-law network and the deterministic walkers can overload the (relatively) few high degree nodes. However, the experimental study in [8] shows that each lookup succeeds in roughly logarithmic number of steps over the network size, indicating that a biased random walk can be quite efficient when used as a search strategy.

Structure of the Paper. Theoretical background concepts are presented in Section 2, including elements of graph theory, electric circuit theory and linear algebra needed in our analysis. The REPO framework is presented in Section 3, along with a basic set of tools and examples of theoretical analysis using the ideas therein. Section 4 presents a set of experimental studies based on the REPO framework, concerning a key set of topics of interest on peer-to-peer networks, such as content replication efficiency, tolerance to failures and effect of resource limitations. The final section presents conclusions and directions of future work.

2 Theoretical Background

Graph-Related Concepts. Let the overlay network be an undirected, weighted graph $G\,(V, E)$, $|V| = n, |E| = m$, with weights w_{ij} for every edge $e = (i, j)$. We will write Γ_i to denote the set of immediate neighbors of node i. Each node i has degree $k_i = |\Gamma_i|$ and *weighted degree* $d_i = \sum\limits_{j \in \Gamma_i} w_{ji}$ the sum of weights over all edges adjacent to that node. The *connected components* of G are its maximal subgraphs, where a path exists between every pair of nodes in the same connected component.

The *adjacency matrix* A of a weighted graph has as elements $a_{ij} = w_{ij}$, and $a_{ij} = 0$ if there is no edge between i and j. The *Laplacian matrix* L of a weighted graph is the matrix $L = D - A$, where A is the adjacency matrix and D a diagonal matrix with

Table 1. Connections of peer-to-peer application notions and random walk concepts with resistor network terms

Peer-to-peer application	**Random walk concept**	**Resistor network term**	**Description**
Navigation	Edge weight	Conductance	The walker uses edge weights at every node to make a preferential choice for its next transition.
	Mean number of crossings	Current	When a unit current is injected into node a and exits at node b, the current at every edge e is the mean number of crossings of e by the walker at the direction of the current, when he starts at node a and before he reaches b [20].
Query efficiency	Return probability	Potential	When a unit voltage is applied between nodes a and b ($v_a = 1, v_b = 0$), the potential at every node is the probability for the walker to reach node a before reaching node b, when starting from that node [20].
	Mean commute time	Effective resistance	The effective resistance between any two points on the resistor network is proportional to the commute time of the walker between the same points [21].
	Mean first-passage time	None	The mean time needed for the walker to reach a destination node for the first time, for a fixed source-destination pair. Although there is no metric directly associated to the mean first-passage time, we can compute it using the Laplacian matrix of the graph.
Robustness	Node/edge connectivity	Algebraic connectivity	The algebraic connectivity is a lower bound for both node and edge connectivity of a graph [18].

values $d_{ii} = \sum_{k=1}^{n} w_{ik}$ [16].We are mostly interested in the inverse of the Laplacian matrix, and although as singular matrix it has no usual inverse, we can use its Moore-Penrose generalized inverse (more on these matrices can be found in [16]). The second smallest eigenvalue of L is called *algebraic connectivity* and is a measure of how well connected and robust to failures is the corresponding graph. Some of its most well known properties [17] are that (i) the graph is disconnected if and only if its algebraic connectivity is zero and that (ii) the multiplicity of zero among the eigenvalues of the graph equals the number of connected components. Additionally, many simple operations that can be applied on a network have well documented effects on the algebraic connectivity [18]. For example, it is known that the formation of hypernodes leads to networks with higher algebraic connectivity [19].

Random Walks and Resistor Networks. We consider a discrete time random walk on the weighted graph with transition probability $p_{ij} = \frac{w_{ij}}{\sum_{k \in \Gamma_i} w_{ik}}$ from node i to j following edge (i, j). The *mean first-passage time* $m(i, j)$ of node j is the mean number of steps a walker needs to reach node j for the first time, starting at i. Another metric of interest is the sum of $m(i, j)$ and $m(j, i)$, called *mean commute time* $c(i, j)$ between i and j.

For the definitions of elementary terms of electric circuit theory such as voltage, current, resistance and conductance, we point the reader at related books (e.g. [22]).

We are mostly interested in the *effective resistance* R_{xy} between two points x and y in a circuit, which is defined as the resistor we must use if we want to replace the entire circuit between these two points, in order to get the same current for the same difference in potential between these two points. The *effective conductance* C_{xy} is defined as the inverse of the effective resistance.

There is an important correspondence between graphs and electric circuits: a *resistor network* [11] is defined as a weighted undirected graph which corresponds to an electric circuit with the following characteristics: every graph node is treated as a checkpoint in a circuit, useful for measurements of potential, and every graph edge as a resistor with conductance equal to the edge's weight. A resistor bridges two checkpoints, as usual in electric circuits, if and only if there is an edge between the corresponding nodes. A simple example of this correspondence can be found in figure 1.

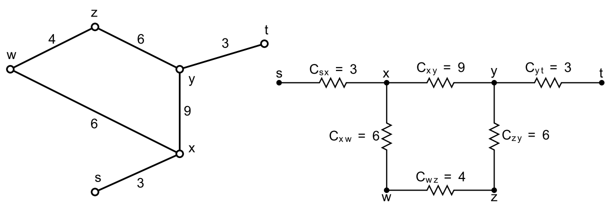

Fig. 1. A simple weighted graph and its corresponding resistor network

3 Framework Details and Essential Tools

Simple random walks have been used to route query messages in unstructured networks, in an effort to reduce the high communication costs of flooding [4,15]. However, simple random walks disregard completely the network's topology and link characteristics, e.g. the low latency of some high speed links to select destinations or the available bandwidth of the neighboring nodes. To address this handicap some authors use *biased* random walks that employ a locally stored metric to guide the random walker [2,3,8]. *Connectivity*, *bandwidth availability*, *latency times* or *socially-oriented information* are just some of the metrics that have been used and can be employed also in our framework. All these metrics are essentially information kept by nodes about their neighbors or statistics kept by each node and can be transmitted between them. These biased random walks have proven successful but their analysis lacks a unifying framework that will enable them to be studied comparatively in a specific overlay network or to study their performance in different networks.

The locally stored information at node i can be represented with a scalar metric m_i. This metric can be easily taken into account for the random walker's transition decisions by defining the transition probabilities as $p_{ij} = \frac{m_j}{\sum_{k \in \Gamma_i} m_k}$, where m_j is a metric stored in node i related to its neighbor j, and m_k are similar metrics of all its neighbors $k \in \Gamma_i$. On the other hand, it is known (cf. previous section) that there is a correspondence between weighted graphs and electric circuits. Can we use these probabilities to define a resistor network? That will enable the use of established tools and intuitive nature of that field. The lemma below shows that this holds for the generalized case of biased random walks; the interface and options for the study of problems in peer-to-peer applications that are enabled are given in more detail in the remaining parts of this section.

Lemma 1 (Equivalence lemma). *Suppose that the random walker makes a transition from node i to node j preferentially with probability $p_{ij} = \frac{m_j}{\sum_{k \in \Gamma_i} m_k}$, based on a locally stored metric m_j about node j, and stores the same information for all neighboring nodes $k \in \Gamma_i$. This biased random walk corresponds to a random walk on a weighted graph with weights $w_{ij} = m_i \cdot m_j$ for all edges (i, j).*

Proof. If p_{ij} and p'_{ij} are the transitional probabilities of the biased random walk and the random walk on the weighted graph respectively, we get the following equation, implying that the two random walks are identical:

$$p'_{ij} = \frac{w_{ij}}{\sum_{k \in \Gamma_i} w_{ik}} = \frac{m_i \cdot m_j}{\sum_{k \in \Gamma_i} m_i \cdot m_k} = \frac{m_j}{\sum_{k \in \Gamma_i} m_k} = p_{ij} \tag{1}$$

Using the previous lemma, we represent any biased random walk which uses local information as a resistor network and this enables us to form a basis upon which different random walks on different networks can be compared and analyzed. We can then directly transfer some known connections of electric and random walk concepts to the biased walks we are modeling (summarized in table 1).

3.1 Framework Interface

Laplacian Matrix. The Laplacian matrix of a graph plays a central part in proving analytical results on resistor networks, as well as evaluating experimentally key properties such as the mean first-passage and commute times. The most common usage of the Laplacian is through its inverse form as it is shown in the following paragraphs.

Effective Resistance. Another metric of interest is the effective resistance between any two nodes. On a local level, consider a node connected to his neighbors by regular resistances. The central node has a distance of one hop from each neighbor but since the random walker makes transitions based on the local resistances with different probabilities, some neighbors effectively come "closer" and some are pushed "further" away. On the other hand, Klein et al showed [11] that the effective resistance R_{xy} between two points is a *distance metric* on the corresponding graph, since it is non-negative, reflexive, symmetric and the triangle inequality holds. These two facts allow us to use the effective resistance to express the distance covered by a random walker. A recently proposed method by Bui et al [12] allows the computing of effective resistances between all node pairs with one matrix inversion, further allowing the computation of other metrics such as the mean first-passage times for all node pairs. Subsequently in this paper we will make use of the effective resistance in our experimental studies as well as illustrate its usage in an analytical setting.

Mean First-Passage and Commute Times. As Chandra et al showed [21], the effective resistance between two points expresses the mean commute time of the points in the weighted graph, multiplied by a network-dependent constant. This result further reinforces the position of the effective resistance as a key metric in our framework. As we will see shortly, it is also connected to the mean first-passage time and can therefore be used to measure the expected query answer times and other metrics related to query efficiency (see also table 1). However, computing the effective resistance between two nodes is no trivial task. Klein et al [11] showed that computing the effective resistance is possible using linear algebra calculations over the Laplacian matrix. The following lemma simplifies the notation used by Klein et al in their solution and aims to illustrate the usage of the Laplacian matrix in analytical computation of large scale properties of the resistor network. Note that to use this method both nodes must belong to the same connected component.

Lemma 2. *The effective resistance R_{st} between nodes s and t in a resistor network of size n, where nodes s and t belong to the same connected component, is given by $R_{st} = c^T \cdot L^+ \cdot c$, where L^+ is the Moore-Penrose generalized inverse of the Laplacian matrix of the network and c is the external current vector of size n with elements $c(x) = 1$, if $x = s$, $c(x) = -1$, if $x = t$ and $c(x) = 0$ otherwise.*

Proof. To compute the effective resistance R_{st} between nodes s and t we inject one unit of current through node s and take one unit of current from node t. We know that there is a path from node s to node t for this unit current to follow since they belong to the same connected component, and the external current vector c of size n is the one specified above. Since we injected one unit of current, the effective resistance is the difference in potential between node s and t. In order to find the potential vector v we have to solve the linear system $L \cdot v = c \ \Rightarrow v = L^+ \cdot c$, where L^+ denotes the Moore-Penrose generalized inverse of the Laplacian matrix [16]. We multiply with c^T to get the difference in potential between s and t, and so $R_{st} = c^T \cdot v = c^T \cdot L^+ \cdot c$.

The above result is complemented by the following formula of Fouss et al in [23] that enables us to use the same matrix L^+ to compute the *mean first-passage time* of a walker, where l^+_{ij} are elements of L^+ and d_{kk} are elements of D:

$$m(i,j) = \sum_{k=1}^{n} \left(l^+_{ik} - l^+_{ij} - l^+_{jk} + l^+_{jj} \right) \cdot d_{kk} \tag{2}$$

Algebraic Connectivity. The algebraic connectivity is a tool in spectral analysis techniques, widely used in applications on large networks such as the Web graph (i.e. the PageRank algorithm [24]). However, only recently have these techniques come under the spotlight of the scientific community regarding their potential towards rigorous analysis of connectivity and fault tolerance properties of networks [25]. Algebraic connectivity in particular can be used in a wide variety of applications, ranging from network health indicator and an early warning system in security studies to an analytical tool useful to compare the robustness achieved by different peer-to-peer configurations.

Hypernode Formation. Hypernodes have been used in previous resistor network studies [13] but play a more central part in a peer-to-peer context. Their strength comes from the observation that from a random walker's viewpoint there is no need to distinguish between different nodes if they all share a common property that places them to the endpoint of the walker's path. So, if some nodes are considered undistinguishable destinations for the random walker we can group them together in a hypernode, having the same set of neighbors as the entire collection of nodes, and study this walk as if they where inseparable. Note that the equivalent of a hypernode in a resistor network is short-circuiting nodes together, changing along the effective resistances of the network, but is a widely used technique in electric circuit studies. These observations can have a significant impact in both experimental and analytical studies, as they can help us address common and recurring problems in peer-to-peer networks such as content replication, community detection and load balancing issues.

3.2 Tackling Problems in Peer-to-Peer Applications

Content Replication. To study the effects of content replication on the query lookup time one can use the hypernode technique: Nodes that possess the replicated content cannot be distinguished in terms of the desired outcome, which is simply to find the queried information, so they can be considered as a single hypernode and destination of the walker. Using the effective resistance it is easy to compare different replication schemes by the amount of closeness between the query issuer and nodes possessing the queried information.

Overloading Issues. Since the biased random walker tends to move on least resistance paths (instead of geodesic paths), it is reasonable to use weighted shortest paths in the calculation of the network diameter. In that case, a node affects the network diameter only if it is situated on a maximum weighted shortest path and through the weights of the edges that participate in this path.

Consider the case of node overloading in a peer-to-peer network, due to excess network traffic: An effective way for a node v to declare that it is overloaded is to change its edge weights to some high values and avoid servicing its neighbors, since the high communication costs will force them to try routing messages through other nodes. We don't know if this node participates on a maximum weighted shortest path and through which edges (if any), but we can estimate the impact of its decision to change edge weights by considering the rest of the network as a hypernode h and node v connected with it in parallel by its edges. We can then compute the difference achieved by changing the edge weights to their high values, and make some observations about the new network diameter.

As an example consider the random walker that navigates through high degree nodes by setting weights $w_{xy} = k_x \cdot k_y$, where k_x and k_y are the degrees of nodes x and y respectively. It is easy to see that the effective resistance between hypernode h and node v is $R_{hv} = \frac{1}{\sum\limits_{i \in \Gamma_v} w_{vi}} = \frac{1}{k_v \sum\limits_{i \in \Gamma_v} k_i}$. If we use unit weights as high values, we get that the difference in effective resistance between overloaded and normal states is

$$\Delta R_{hv} = \frac{1}{k_v} - \frac{1}{k_v \sum\limits_{i \in \Gamma_v} k_i} = \frac{1}{k_v} \left(1 - \frac{1}{\sum\limits_{i \in \Gamma_v} k_i} \right) \tag{3}$$

and that is the effect on the network diameter inflicted by the overloaded node v. We can see that this difference increases for lower degree nodes and larger neighborhood sizes (due to their greater criticality as bridges). Note that if nodes can recover from their overloaded state after some time, this technique can be useful in showing *self-regulating* properties of peer-to-peer networks on the diameter size.

Fault Tolerance. Node failures should increase the expected query lookup times, so fault tolerance can be studied by looking at the comparative growth of both effective resistance and mean first-passage time. During node removal special consideration should be given to the connectivity of the resulting network, a prerequisite in order to compute the framework metrics. Checking for connectivity in our framework can be easily done through the algebraic connectivity, its computation being a by-product of the matrix manipulations necessary to compute the framework metrics (table 1).

Edge Selection/Dropping Policies. Our framework can be easily used to study how local decisions affect the network as a whole and correlate local and global phenomena in a quantifiable way [26]. Such an example can be the global effect of local edge dropping policies, i.e. selection of neighbors in the overlay. In most networks peers drop some connections (either periodically or on demand) in favor of others, in order to maximize some parameter such as the average latency, bandwidth utilization or an interest-based metric. A possible outcome of a study like this could be the specification of a local edge dropping policy with the least detrimental effect on the network.

Computational Demands. Inverting a large, dense matrix such as the Laplacian may educe a high computational cost. Also, the fact that the Laplacian needs global information about the network in order to produce metrics such as the mean commute time between even two nodes, coupled with its size (equal to the network size), may be considered as restrictive regarding its usage in an experimental setting. However, as we have seen in this section, this effect is mitigated by the fact that we can compute several metrics at each inversion, as well as the existence of useful by-products such as the algebraic connectivity. Furthermore, parallel algorithms for graph related studies relevant to these (e.g. shortest path problems) are the subject of revived and ongoing research in the scientific community [27]. The advances at this front, together with the fact that increasing degree of parallelism is continuously becoming available even in commonly used equipment, enable large sizes of networks to be analyzed. In the experimental study of this paper, which is given to exemplify the use of the framework, we used a SIMD multicore processor (GPU) with 32 cores in order to run the necessary elementary matrix operations in parallel.

4 Example Experimental Study

We illustrate the use of several of the methods outlined above in a concrete experimental study of example random walks and network types. For these studies we outline insights than can be gained and conclusions that can be drawn using our framework.

In particular, for these experiments we consider a random walker such as the one described in section 3, which uses node degrees as a navigational metric; i.e. the queries navigate strongly towards high degree nodes. Regarding the overlay topology, we use synthetic data of networks with power-law degree distribution, based on the Barabási-Albert preferential attachment procedure [28], that produce a clear, small set of high degree nodes (past studies have linked peer-to-peer networks, e.g. the Gnutella network, with heavy tail degree distributions such as the power-law distribution [8,29]).

The strategies we used for the experiments have common elements and are given in table 2. The experiments were conducted over 50 graph instances of size 1000. Much of the computational load was lifted by executing elementary matrix operations in parallel on a nVidia GPU having 32 cores, using the CUDA library [30].

Table 2. Strategies used in experimental studies

Content Replication/Fault Tolerance Strategies

Strategy	**Name**	**Description**
Highest degree node	HIGHREP/HIGHFAIL	Replication to (failure of) nodes in decreasing highest degree sequence
Lowest degree node	LOWREP/LOWFAIL	Replication to (failure of) nodes in increasing lowest degree sequence
Random node	RANDREP/RANDFAIL	Replication to (failure of) random nodes

Edge Dropping Policies

Strategy	**Name**	**Description**
Highest resistance edge drop	HIGHRESD	Highest resistance edge drop from node under consideration
Lowest resistance edge drop	LOWRESD	Lowest resistance edge drop from node under consideration
Random edge drop	RANDD	Random edge drop from node under consideration
Highest voltage neighbor drop	HIGHVOLTD	Drop of edge between node and neighbor of highest voltage

4.1 Content Replication Example

For the experiments we use a sample query, initiated in a fixed node, that tries to locate a piece of information that initially can be found only in one node. At each step we replicate this information to another node, using one of the strategies described in table 2, with all nodes in possess of the information forming a hypernode. As we described in section 3, we measure the effective resistance and the mean first-passage time between the query issuer and this hypernode of nodes possessing the requested information.

Observations/Conclusions. As can be seen in figure 2, the effective resistance and mean first-passage times tend to decrease as we replicate content into more nodes. This is only natural since the desired information can be found more easily as replication progresses, and distances in the network decrease. Furthermore, we expected the HIGHREP strategy to have greater impact since it targets the high degree nodes used preferentially by the random walker and indeed this is the case, followed by RANDREP and LOWREP. However, we also note a seemingly counter-intuitive behavior of HIGHREP: while mean first-passage time continues to drop, the effective resistance starts rising

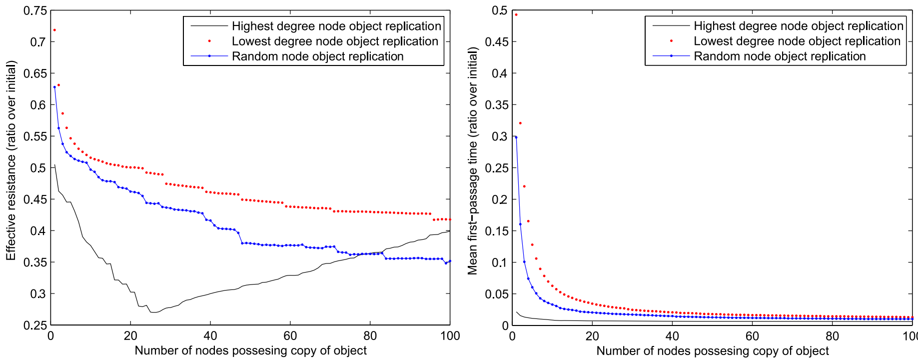

Fig. 2. Effective resistance and mean first-passage time between query source and destination, in the case of multiple object copies in the network

after some replications. This can be explained by recalling that the effective resistance is proportional to the mean commute time, so while the time needed to traverse the desired direction of the query (mean first-passage time of figure 2) is dropping, the time needed for the reverse direction can be increasing even more so. In a way, replication turns the growing hypernode (destination of the query) into a "gravity well".

4.2 Fault Tolerance Example

In this scenario we remove from each network 90 nodes in batches of 30, according to the strategies described at table 2, and we aim to study the detrimental effect of node failures. As we described in section 3, we expect distances in the network to grow so we compute the comparative growth of both the effective resistance and mean first-passage time at the end of each batch, all the while checking for connectivity through the algebraic connectivity. The means over all pairs, for all strategies and experiments, can be found in figure 3.

Observations/Conclusions. Observe that the HIGHFAIL strategy is the most destructive of all, raising the mean value of the effective resistance by a factor of 19. This was expected since it destroys high degree nodes and these are preferred by the walker. The LOWFAIL strategy has negligible effect on the network, since the low degree nodes are rarely used for routing purposes. The RANDFAIL strategy is proven equally powerless but slightly more destructive, since it may target a high degree node but low degree nodes are the majority.

However, the HIGHFAIL strategy behaves differently in various regions of the network. To investigate this phenomenon we studied more closely a subset of the previous results concerning two communication groups of interest, one of very high degree to very low degree nodes and its symmetric (figure 4). Although the qualitative effect of each strategy is the same in each communication group, the above mentioned unbalance is clearly shown. Again, this was expected since the walker navigates towards nodes of high degree and it appears that the highest degree nodes were connected to low degree

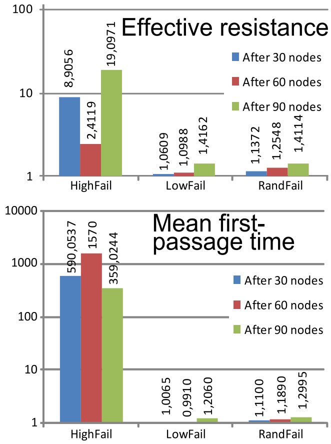

Fig. 3. Comparative growth of mean first-passage times and effective resistance for different failure strategies

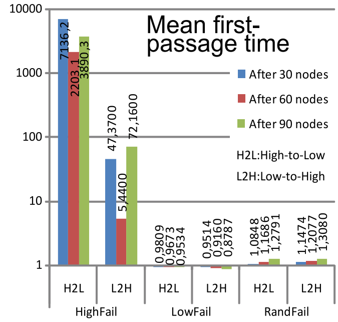

Fig. 4. Effect of failures on groups of interest

ones. Having lost these bridges it is becoming increasingly difficult for the walker to make the transition from high to low degree nodes. Another point of interest is the behavior of the HIGHFAIL strategy during the second batch of deletions, where the metric values drop. It appears that with the exception of a small number of initial deletions, subsequent deletions improve the mean first-passage and commute times, and only quite later these metrics begin to decline again. This is a direct consequence of the walker's memorylessness: he cannot track his previous steps on the query path and therefore falls into cycles, which can be quite large when involving high degree nodes since the latter constitute local "gravity centers" for the walker. So, by removing them with the HIGHFAIL strategy we remove these cycles and we actually help the walker to reach its destination faster. Of course this phenomenon is paired with the previously mentioned difficulty of high degree nodes reaching low degree ones, and the end result is constrained growth of mean first-passage and commute times. It appears that navigating through high degree nodes can be a double edged tool when dealing with a memoryless random walker.

4.3 Edge Dropping Policies Example

In a similar way to the study of fault tolerance, we expect the distances in the network to grow as we drop more edges, so we study the comparative growth of both the effective resistance and mean first-passage time. As we will see some strategies succeed in dropping slightly these metrics, compared to their original values. In order to study the global effect of the local edge dropping policies we employ the following experimental procedure: For each graph we are instantiating we select a set of 100 nodes that will perform the specified edge drop policy. Then we apply the policy to the set of nodes in batches of 25 and after each batch we calculate the mean values of the effective resistance and mean first-passage time between all node pairs in the network, divided with

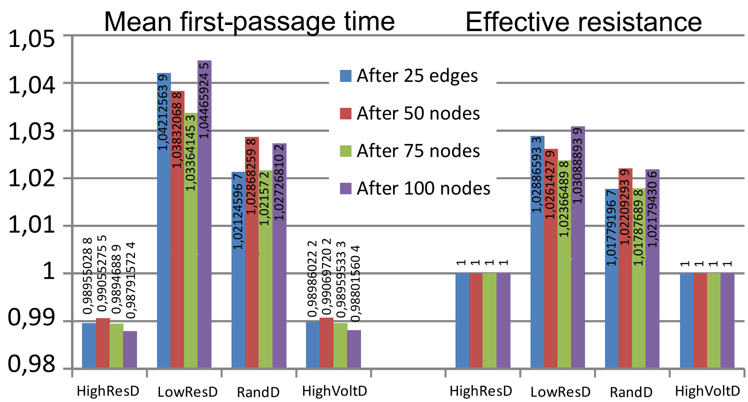

Fig. 5. Comparative changes of mean first-passage times and effective resistance for different edge dropping strategies

their original values to show the relative growth or shrinkage. We added one more strategy (HIGHVOLTD) to the ones used previously, in order to see whether better results can be achieved by using global information to decide on the edge we should drop. Note that here it is safe to look only at mean values over all pairs, since values of the metrics under study vary significantly less compared to the values of previous experiments.

Observations/Conclusions. Observe that the HIGHRESD strategy targets edges with high resistance, therefore connecting to lower degree nodes, in contrast to the LOWRESD strategy which targets edges with higher degree nodes. Consistent with the results regarding fault tolerance, we also observe that the LOWRESD strategy has the most detrimental effect, with HIGHRESD even having a slightly positive result and RANDD performing between these two. It may be worth to take a special look at the HIGHRESD and HIGHVOLTD strategies, which perform quite similarly and even succeed in maintaining the effective resistance to the same mean value as it was originally. However, HIGHVOLTD has a high computational cost (needing global information). Using this framework we are able to safely conclude that implementing a costly global policy gives us a negligible gain in performance, compared to the simple and local policy of HIGHRESD which also gives the best results over all strategies.

5 Conclusions

In this paper we introduced REPO, a framework for the analysis and systematic study of random walks on peer-to-peer networks. To illustrate the possible uses of REPO, we presented a basic set of analytical tools and a brief overview of recurring problems in peer-to-peer networks that can be studied experimentally. Our aim was not to enumerate exhaustively all potential application areas but rather to illuminate the multitude of connections between classic electric theory, random walks and peer-to-peer networks and present some examples of the framework's intuitiveness.

We present analytical tools that enable us to reason about the robustness achieved by different random walks on the same network and study the effects of overloading nodes to the peer-to-peer overlay. We also relate local policies with global phenomena observed on the network, and reason on the effectiveness of the policies under study. Finally, we draw conclusions on the interaction between random walks, their underlying peer-to-peer topologies and how they affect the query efficiency, especially using content replication techniques or under the presence of faulty nodes. We observe that the performance of the walker depends heavily on properties of the overlay. For example, the promising performance results reported by recent work [5,6] do not seem to transfer directly to other overlays, such as the power-law networks of our example study.

Other issues of peer-to-peer research that can benefit from our framework paradigm are load balancing techniques, as well as construction techniques that utilize specialized random walks. Algebraic connectivity and effective resistance can express structural properties quantitatively and be employed i.e. during the construction phase to steer it towards desirable values.

Acknowledgements

We are thankful to Daniel Cederman for his help with using the CUDA library in our experiments.

References

1. Balakrishnan, H., Kaashoek, F.M., Karger, D., Morris, R., Stoica, I.: Looking up data in p2p systems. Commun. ACM 46(2), 43–48 (2003)
2. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power-law networks. Phys. Rev. E 64, 46135 (2001)
3. Ata, S., Murata, M., Gotoh, Y.: Replication methods for enhancing search performance in peer-to-peer services on power-law logical networks. In: Performance and Control of Next-Generation Communications Networks. SPIE, vol. 5244, pp. 76–85 (2003)
4. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks. In: Proc. of 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2004), March 2004, vol. 1 (2004)
5. Sarshar, N., Boykin, P.O., Roychowdhury, V.P.: Percolation search in power law networks: making unstructured peer-to-peer networks scalable. In: Proc. of the 4th International Conf. on Peer-to-Peer Computing, pp. 2–9 (2004)
6. Zhaoqing, J., Jinyuan, Y., Ruonan, R., Minglu, L.: Random walk search in unstructured p2p. J. Syst. Eng. 17(3), 648–653 (2006)
7. Chockler, G., Melamed, R., Tock, Y., Vitenberg, R.: Spidercast: a scalable interest-aware overlay for topic-based pub/sub communication. In: Proc. of 2007 inaugural international conf. on Distributed event-based systems (DEBS 2007), pp. 14–25. ACM, New York (2007)
8. Fraigniaud, P., Gauron, P., Latapy, M.: Combining the use of clustering and scale-free nature of user exchanges into a simple and efficient p2p system. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 1163–1172. Springer, Heidelberg (2005)
9. Voulgaris, S., Kermarrec, A.M., Massoulie, L.: Exploiting semantic proximity in peer-to-peer content searching. In: Proc. of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004), pp. 238–243 (2004)
10. Hughes, B.: Random Walks and Random Environments: Random Walks, vol. 1. Clarendon Press, Oxford (1995)

11. Klein, D.J., Randić, M.: Resistance distance. J. Math. Chem. 12(1), 81–95 (1993)
12. Bui, A., Sohier, D.: How to compute times of random walks based distributed algorithms. Fundamenta Informaticae 80(4), 363–378 (2007)
13. Sohier, D., Bui, A.: Hitting times computation for theoretically studying peer-to-peer distributed systems. In: Proc. of the 18th International Parallel and Distributed Processing Symposium (2004)
14. Telcs, A.: The Art of Random Walks. Springer, Heidelberg (2006)
15. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proc. of the 16th International Conf. on Supercomputing (ICS 2002), pp. 84–95. ACM Press, New York (2002)
16. Barnett, S.: Matrices: Methods and Applications. Oxford University Press, Oxford (1990)
17. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Math. J. 23, 298–305 (1973)
18. de Abreu, N.M.: Old and new results on algebraic connectivity of graphs. Linear Algebra and its Applications 423(1), 53–73 (2007)
19. Chung, F.: Spectral Graph Theory. CBMS Regional Conf. Series in Mathematics, vol. 92. AMS (1997)
20. Doyle, P.G., Snell, L.J.: Random Walks and Electrical Networks. Mathematical Association of America (December 1984)
21. Chandra, A.K., Raghavan, P., Ruzzo, W.L., Smolensky, R.: The electrical resistance of a graph captures its commute and cover times. In: Proc. of the 21st Annual ACM Symposium on Theory of Computing (STOC 1989), pp. 574–586. ACM Press, New York (1989)
22. Alexander, C., Sadiku, M.: Fundamentals of Electric Circuits. McGraw-Hill, New York (2006)
23. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. 19(3), 355–369 (2007)
24. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. 30(1-7), 107–117 (1998)
25. Olfati-Saber, R., Murray, R.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Autom. Control 49(9), 1520–1533 (2004)
26. Georgiadis, G., Kirousis, L.: Lightweight centrality measures in networks under attack. In: Proc. of European Conf. on Complex Systems, ECCS (2005)
27. Madduri, K., Bader, D.A., Berry, J.W., Crobak, J.R., Hendrickson, B.A.: Multithreaded algorithms for processing massive graphs. In: Bader, D.A. (ed.) Petascale Computing: Algorithms and Applications. Chapman & Hall/CRC Computational Science Series, Boca Raton (2008)
28. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74(1), 47–98 (2002)
29. Ripeanu, M., Foster, I., Iamnitchi, A.: Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Comput. 6(1), 50–57 (2002)
30. NVIDIA Corporation nvidia.com/cuda: NVIDIA CUDA Programming Guide. 2.0b edn. (2008)