# Non-replicable reusable resources discovery on scale-free peer-to-peer networks

*Citation for published version (APA):*
Exarchakos, G., & Antonopoulos, N. (2008). Non-replicable reusable resources discovery on scale-free peer-to-peer networks. In *Proceedings of the 2nd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2008), February 26-29, 2008, Phitsanulok, Thailand* (pp. 28-33). IEEE Computer Society. https://doi.org/10.1109/DEST.2008.4635188

*DOI:*
10.1109/DEST.2008.4635188

*Document status and date:*
Published: 01/01/2008

*Document Version:*
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

*Please check the document version of this publication:*

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

*General rights*
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

*Take down policy*
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 04. Aug. 2024

2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

# Non-Replicable Reusable Resources Discovery on Scale-Free Peer-to-Peer Networks

Georgios Exarchakos[1], Nick Antonopoulos[2]

[1,2] Georgios Exarchakos and Nick Antonopoulos, Department of Computing, University of Surrey, Surrey, UK
e-mail : (g.exarchakos, n.antonopoulos)@surrey.ac.uk

*Abstract*— **Apart from files, P2P Networks can be used for sharing non-replicable and reusable (e.g. computational) resources. This paper explains why existing discovery mechanisms have low performance searching for those resources in random power-law degree distributions. While high-degree nodes become more popular, their resources are consumed, resulting into a power-law network topology with no more useful highly connected nodes. To address the problem, we introduce two blind search techniques, Scale-free FloodWalkers (SFFW) and Scale-free Walkers (SFW). The idea behind is that recent requestors have discovered recent providers and will soon release the recently acquired resources. We tested their performance through simulations and a number of experiments. They seem to have stable much higher success rate than flooding or k-walkers in scale-free networks while their cost in messages is one order of magnitude lower than flooding and equal to walkers, respectively.**

*Index Terms*—**peer-to-peer, power-law, search, computational resources.**

## I. INTRODUCTION

Recently, Peer-to-Peer networks have attracted a lot of research interest and support a variety of applications including distributed computational resource sharing. Self-configuration of these networks is a substantial prerequisite to ensure applicability.

This paper introduces two algorithms for searching non-replicable reusable resources over P2P overlays. Computational resources (CPU, storage, services etc…) in principle can only exist in a single copy and portions of them can be used by only a single task at a time. After the task's termination the used portion becomes available to other tasks.

According to Holt [1] reusable resources are those that allow exclusive only access (no more than one user at a time can access them). A critical feature of those resources is their intermittent behaviour since no replicas can be produced and their availability is affected by each task using them. In some systems, they may be represented as a set of non-replicable and reusable chunks that move from node to node as they are used by tasks. The more requests for a specific resource in a network and the shorter the lifetime of the tasks that are using it, the more mobile behaviour its chunks have. Therefore, these resources' availability is constrained by its failure, uniqueness, exclusive usage and possible mobility.

Many real world systems [2], [3], [4], [5] exhibit power-law degree distributions building scale-free networks. Their main characteristic is that very few nodes have many neighbours (high degree nodes) while the majority of the nodes have a small neighbourhood. As [2] proved, preferential attachment (the probability a node to link to another one is proportional to the latter's degree) can transform a random uniform network to a scale-free one. It is a technique explicitly or implicitly used by many networks [6].

Preferential attachment on networks with non-replicable reusable resources may severely deteriorate the efficiency of the deployed search mechanism, if it is not designed to cope with such resources. A good provider (node with plenty of resources) attracts many links. The more links are attracted the faster its resources are consumed. The problem emerges when the search technique forwards the queries primarily to high-degree nodes, assuming that they will always be good providers. This may create few high-degree nodes with few or no resources. Thus, the focus of this work is on the design of a searching mechanism that achieves good performance despite the non-replicability and reusability of resources and the network's power-law degree distribution.

The two introduced mechanisms, Scale-free FloodWalkers and Scale-free Walkers, are based on the idea that older queries from other requestors may have discovered providers and resources. These providers may still have available resources and the discovered resources these nodes currently use may soon be released. The older the requestor the more unpredictable his own and his neighbours' status is.

The following sections present the work done on this or related topics, the description of the two proposed algorithms and an evaluation of their performance compared to flooding and k-walkers.

## II. RELATED WORK

Existing P2P topologies and search strategies use different approaches to tackle discovery efficiency and latency. Most of them are based on the replication of the resources or on the well-defined overlay structure. These overlays can be either structured or unstructured depending on whether content is placed at specified locations or randomly (respectively) within the network [7].

Structured P2P networks (CAN, Pastry, Tapestry, Chord, Kademlia and Viceroy) [8] assign a key to every resource and map each tuple {key, resource} onto a specified peer. The organisation of the peers is flat; that is all peers are equally important and have the same amount of responsibilities within the network. Given a key, the resource discovery is efficient but a copy or pointer to the resource needs to be located at the peer responsible for that key. They are well-studied topologies and provide guarantees of discovery if the requested resource is available in the network. However, latency becomes a considerable problem in case of large networks (huge number of nodes) since each query is routed to the next intermediate peers as far as possible producing long-distance network traffic and delays. Low-capacity peers can introduce further delays since they can easily become overloaded even if they don't process the query but just forward it. Query routing is based on simple keys and thus structured P2P overlays do not support complicated queries.

Unstructured P2P overlays organise the peers in a random flat or hierarchical graph which is decoupled from the location of the resources. Replication methods may increase the availability of a resource within the network by copying the same resource to several places [9]. Popular discovery mechanisms that are used in various random networks are the flooding, breadth-first search (BFS) [10] and random walkers [11]. These blind techniques do not use any information to navigate the network but randomly choose the next target nodes. They are in general fault-resilient but do not guarantee discovery even though the requested resource may exist in the network. They usually produce large number of messages and/or experience high latencies in resource discovery [12].

Instead of randomly choosing the next destinations, a query may be forwarded to a node's neighbours that are likely to provide an answer either themselves or via them (informed search techniques [10], [12]). They require centralized or distributed repositories of hints or accurate network status that a query may use to access more promising nodes. These mechanisms drive a query faster to the target node, usually require frequent updates of the distributed information and cannot efficiently cope with nodes' high failure rates [12]. Therefore, they are practically unsuitable for searching so intermittent resources such as the ones this study focuses on.

Power-law networks have short average path length [2] and allow logarithmic search efficiency [13]. As argued in [14], search techniques can be even more efficient (twice logarithmic) in scale-free networks. Though these networks have good resilience on node failures and attacks [16], a combined attack to selected high degree nodes may negatively affect the search efficiency or even fragment the network [15].

LV et.al. in [11] claim that random walkers quickly reach high-degree nodes in power-law networks but have poor efficiency in rare resources and therefore the node degree should be ignored in query forwarding. Replication methods could alleviate the situation but in the current study this is not applicable. Adamic et.al. [17] proposed an algorithm based on random walkers. At each step of the walker, all the current node's neighbours are visited and if no success the walker goes to the one with the highest degree. This process stops after Time-to-Live (TTL) steps.

Fraigniaud, Gauron and Latapy [18] focused on developing an algorithm to exploit the power law degree distribution properties quickly locating the high-degree nodes. It is a modified DFS which first forwards the queries to neighbours with the highest degree. When a certain number of steps are completed, the query backtracks to visit the second highest degree neighbour. For every change of a node's degree all its neighbours have to be notified thus increasing the maintenance cost. Backtracking introduces high latencies in search mechanisms cancelling the advantage of short average path length of scale-free networks.

Adamic et.al. [17] also claimed that searching mechanisms in scale-free networks should give preference to high-degree nodes when forwarding a query. However, this assumes that the resources are replicable, not reusable and that always reside at the same node. This assumption is also used by QRE algorithm [18] but is not the case with the resources on which this work focuses.

Finally, Mihail et.al. in [19] formalised the random-walker with lookahead discovery mechanism on power-law networks. At every step of the walker all the neighbours of the current node are checked for the requested resource and their own degree. Only the one with the highest degree then forwards the query. This mechanism directs the queries to the highest degree nodes in the network but introduces high latency, too. Moreover, it assumes that the requested resource is in those nodes; that is not the case with reusable non-replicable resources which may move around the network.

## III. SCALABLE SEARCHING OF REUSABLE RESOURCES

It is common knowledge between a scale-free network's nodes, where the high-degree ones are located. The main concept behind the proposed algorithm is that a query forwarded to recent requestors can quickly locate the good providers. Using prior knowledge collected by other nodes, one can improve the success of his queries.

Moving a non-replicable resource from a provider to a recent requestor is an indication that it should be sought to the requestor rather than provider. Queries should be reluctant to visit old providers since they may have no available resources; high-degree nodes are not always a panacea.

Following the principles above, the proposed search mechanisms are the Scale-free (SF) FloodWalkers and Scale-free (SF) Walkers. While SF-FloodWalkers start as walkers, at each step copies of the query are forwarded to a portion of recent requestors which in turn broadcast it to their immediate outbound neighbours only. SF-Walkers instead of choosing one random outbound neighbour, they choose the most recent either outbound or inbound neighbour.

### *A. Scale-free FloodWalkers*

Scale-free FloodWalkers (SFFW) algorithm is a blind search algorithm. Every node in the network has a fixed-size ***M*** Outbound Neighbour List (ONL) of providers and a variable-size Inbound Neighbour List (INL) of requestors. The query originator node starts k-walkers selecting ***k*** random neighbours from its ONL. Each walker travels from his originator via intermediate nodes (intermediaries) and terminates when either discovering a response provider or

```
q={type, ttl, w, hops, resource};
SFFW(q) {
  q.hops = q.hops+1;
  if q.resource can be satisfied
    send response back to originator;
  else if q.hops < q.ttl {
    if q.type ≠ branch {
      select one $n \in ONL$ with equal probability;
      forward q to node n;
      if q.hops < q.ttl-1 {
        select most recent $m_i \in INL \quad i = 1,...,q.w$;
        bq = q;
        bq.type = 'branch';
        bq.ttl = 2;
        bq.hops = 0;
        forward bq to all $m_i$;
      }
    } else forward q to all $n \in ONL$;
} }
```

Fig. 1 Pseudocode of Scale-free FloodWalkers algoritm

after TTL steps away from his source. The aim of the algorithm is the efficiency and not the recall (number of responses over the total number of available resources in the network).

The pseudo code of Fig. 1 gives more details about the algorithm that operates in each node. The intermediaries use one random neighbour of their ONL to forward the walker to. If they are located at most TTL-2 steps away from query originator, they use random ***W*** of the most recent inbound neighbours from INL (branch-intermediaries) to forward the same query to. The branch-intermediaries either respond or again broadcast the query with unitary TTL to all their outbound neighbours. Therefore the query can be either of type *normal*, forwarded as walker, or *branch*, forwarded via branch-intermediaries.

The query's broadcasting from branch intermediaries to their whole ONL is the feature of the algorithm that uses exactly the knowledge collected from their own recent queries thus helping to quickly locate the recent providers. Given the fixed size of this list, the high-degree nodes are prevented from broadcasting to a big portion of the network. The walkers hop from node to node using their ONLs. This works as the mechanism for discovering providers and via them new requestors .

Each node's INL may contain many links, even inactive ones which can be used by a bigger *W*. Re-linking process happens as requestors discover new providers thus making incoming links to old providers inactive. Practically, INL has a maximum number of entries which are updated in a first-in-first-out mode.

### B. Scale-free Walkers

Following the same principle as in Scale-free FloodWalkers, k-Walkers can be modified to occasionally access intermediaries' recent requestors. Though SFFW uses one hop broadcasting, the traffic it produces may be comparable to flooding if the TTL is low. SFW tries to tackle this problem. Fig. 2 presents the pseudocode of this algorithm.

```
q={type, ttl, w, hops, resource};
SFW(q) {
  q.hops = q.hops+1;
  if q.resource can be satisfied
    send response back to originator;
  else if q.hops < q.ttl {
    select the most recent $n \in ONL$;
    select the most recent $m \in INL$;
    (both n and m cannot be the originator or the walker's
previous visited node)
    forward q to the most recent node between n, m;
} }
```

Fig. 2 Pseudocode of the Scale-free Walkers mechanism

Initially, a random subset from the query originator's ONL is selected to send the query to, but all intermediaries have to choose one neighbour, the most recently added to either ONL or INL. Each walker stops when a response is found or TTL nodes are already visited.

Assuming that the requestor has discovered more providers with its latest search, the query, targeting the most recent requestor, hopes to discover recent providers. If, in the meantime, the state of the requestor has changed to a provider, then it would also be beneficial for the query.

## IV. SIMULATIONS AND EVALUATION

Final phase of the proposed algorithms description is their performance evaluation. An appropriate simulator written in Matlab was developed and a carefully designed set of experiments gave useful results as presented in the following sections. The evaluation metrics used are the success rate and the number of messages produced based on which they are compared against flooding and k-walkers.

### C. Experiments Setup

As stated in the introduction and related work, important characteristic of scale-free networks is the preferential attachment. Though the two simulated networks have fixed size (1000 and 5000 nodes), a rewiring technique is introduced based on which a node inserts in its outbound neighbour list every received answer's originator rejecting a random entry, if it is not already present.

Initially, every node has five random outbound neighbours. The whole network has up to 1000 different resource keys whose frequency follows a Normal Distribution with (mean, standard deviation)=(500,200). That is, the keys around 500 are more frequent than the others. The number of keys per node also follows a Normal Distribution with (mean, standard deviation)=(50,30); most of the nodes have 50 resources.

Each resource is described by one key only and no node can have two resources with the same key. The resources once discovered are moved from node to node but not rep-

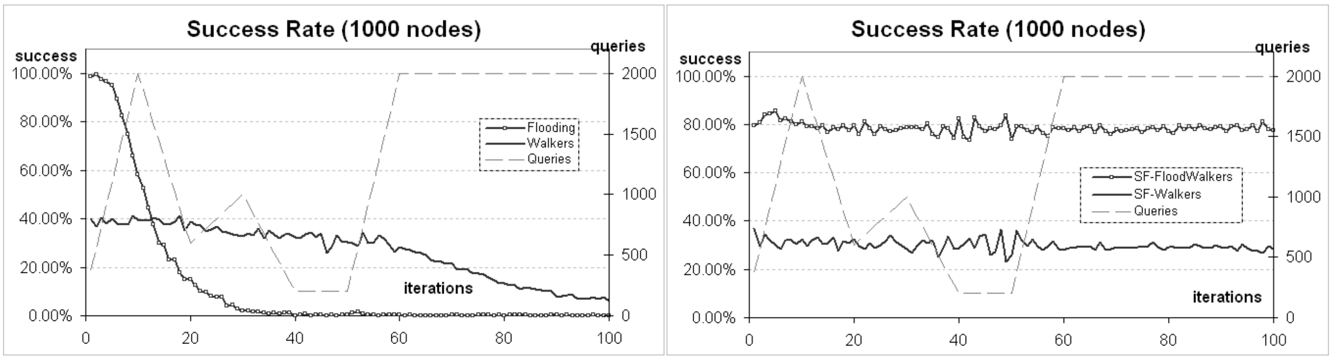

Fig. 4 rate diagrams of Flooding, 3 Walkers, 3 SF-FloodWalkers and 3 SF-Walkers for the 1000-node network.

located. The environments are simulated for 100 iterations each. Each query starts from a random node; thus, each node may produce from zero to more than one query.

The size of the ONL per node is equal to 5 and all the simulated walkers-based search mechanisms start with three walkers. SF-FloodWalkers have a fixed *W=2.*

Keeping the environment features the same and changing the network size, we simulate the low and high load environments. In small network, the queries can locate easier distant providers since the TTL is the same and thus more re-linking actions are taken speeding up the conversion of the network to a power-law one.

### *D. Results and Evaluation*

Applying the networks configuration above, running the simulations and measuring the metrics, we got the following results.

The first set of graphs, Fig. 3, refers to the success rate on the 5000-node network. While the left y-axis represents the number of successful queries per iteration the right one is the total new queries produced, forwarded and processed per iteration (gray dashed line). The left diagram is about flooding and walkers whereas the right one about the two proposed algorithms. The same graphs for the 1000-node network are shown in Fig. 4.

On both networks, flooding achieves a 100% success rate for the first few iterations and then goes rapidly down to almost 0%. A similar smoother phenomenon happens to the small network with the 3-walkers. This is due to the rewiring which forces many nodes to link with the big providers. The initial configuration of the networks had a Normal Distribution of resources over nodes. That means that very few had too many resources and therefore were high providers. As long as a node has resources it attracts more links while keeping its outbound neighbour list relatively outdated transforming high providers to high-degree nodes. Given the fixed size of outbound neighbour lists most of the nodes in the network start pointing to high providers gradually converting the network to a scale-free one.

Though this phenomenon shortens the average path length between two nodes, it also increases the number of low connectivity nodes and the probability these paths to pass through high providers. As the low-degree nodes become more they tend to produce the majority of the queries moving the resources from high providers to low-degree nodes which again point to high-degree ones. Further queries are forwarded to high-degree nodes which do not have any resources and have a small outbound neighbour list.

The topology is altered in a way that the resources are moved from few high-providers to many small ones between which the links are lost.

The more answers received the faster the rewiring is and thus quicker the drop of the success rate. The walkers in the small network manage to visit a bigger portion of the good providers compared to the big network speeding up the transformation of the network into a scale-free one. For the beginning of the experiments, 3-walkers have better performance since they exploit the initial randomness of the network whereas SF-Walkers force some queries to hop from requestor to requestor though they may have not yet acquired a good portion of resources from good providers.

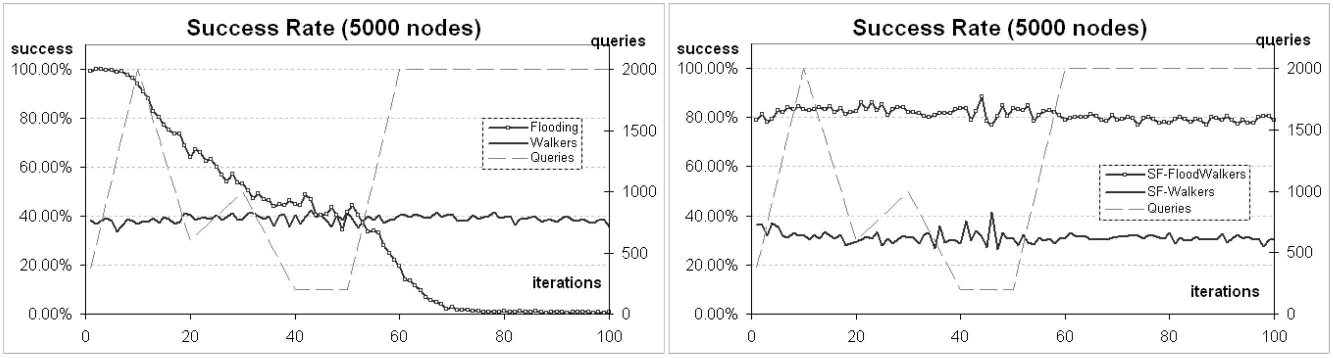

Fig. 3 Success rate diagrams of Flooding, 3 Walkers, 3 SF-FloodWalkers and 3 SF-Walkers for the 5000-node network.

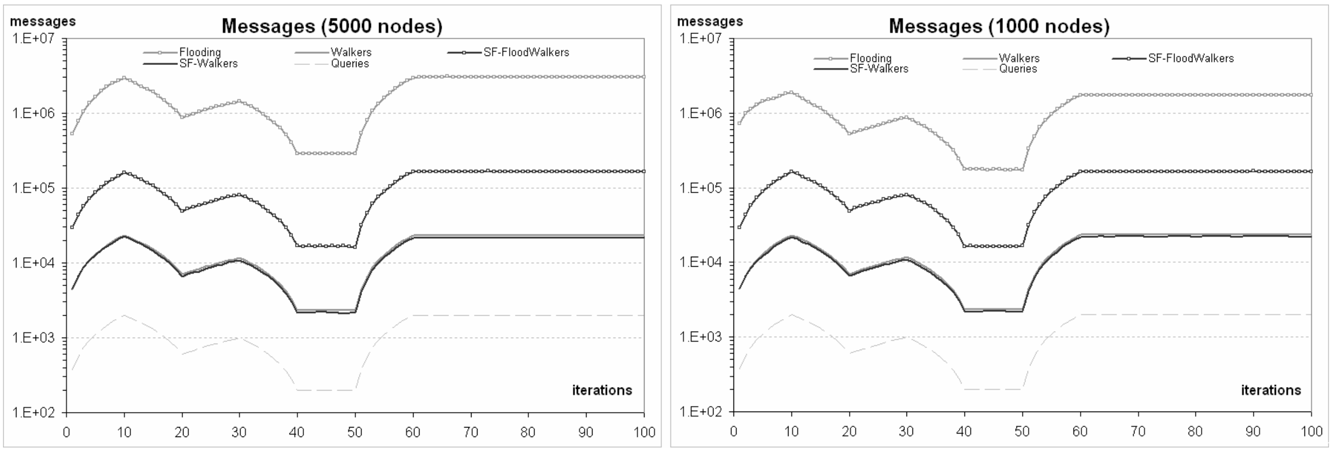

Fig. 5 Messages produced by Flooding, 3 Walkers, 3 SF-FloodWalkers and 3 SF-Walkers in the two network environments (5000 & 1000 nodes).

Both SF-FloodWalkers and SF-Walkers achieve a stable success rate throughout the experiments since they use a node's most recent requestors. Instead of preventing the formation of a scale-free network, they use the recent requestors of high-degree nodes to locate moved resources.

The diagrams of Fig. 5 show the number of messages spent in the two networks in logarithmic scale. Each node generates 3 Walkers or SF-Walkers for every query and with a TTL=4, the messages produced are close to one order of magnitude more than the number of queries. SF-FloodWalkers technique generates around 90 messages which, adding the number of query hits, are one order of magnitude more than Walkers and SF-Walkers. Given its walkers-based forwarding, the fixed *W* and the fixed branch intermediaries' ONL size the cost of SF-FloodWalkers increases linearly. Finally, Flooding produces 780 messages and adding the query hits, the cost becomes almost 10 times bigger than SF-FloodWalkers.

As a proof of concept, Fig. 6 presents the 5000-node network's incoming degree distribution. The more nodes have the same incoming degree the darker is the colour around this degree on that specific iteration. If the colour of a point in a contour graph determined by an iteration and a number of incoming links is e.g. very light gray, it means that less than 20 (not necessarily 0) nodes have that degree in that iteration.

While in all situations, initial network's degree follows a Poisson distribution, the simulated discovery mechanisms seem to gradually convert it to a Power-Law one. The more nodes a query accesses the faster this conversion is. Flooding quickly converts the network to a scale-free one since the vast majority of the nodes have less than 3 incoming links after 30th iteration. Note that the very light gray colour does not necessarily mean no incoming links. Walkers, producing less query hits than flooding, cause smoother conversion to power-law distribution. Flooding and Walkers seem to raise a number of scaling issues in so dynamic environments such as networks of non-replicable reusable resources. Though SF-FloodWalkers and SF-Walkers do not prevent this conversion, they achieve an almost stable success rate.

## V. CONCLUSIONS AND FUTURE WORK

This paper identifies the key characteristics of non-replicable and reusable resources that are useful to the design of an efficient searching algorithm on unstructured P2P power-law networks. Their intermittent behaviour is due not only to failures; there can be only a single instance of each resource in the network and only one task at a time can use either a portion or the whole of it. Though, physically, the resource may exists it may not be available.

Nodes in scale-free networks have the tendency to link to high-degree ones thus making the latter's shared resources more and more requested. If these resources are non-replicable and reusable, their availability falls quickly. To face this issue, two blind searching algorithms are proposed based on the idea that recent requestors have discov-

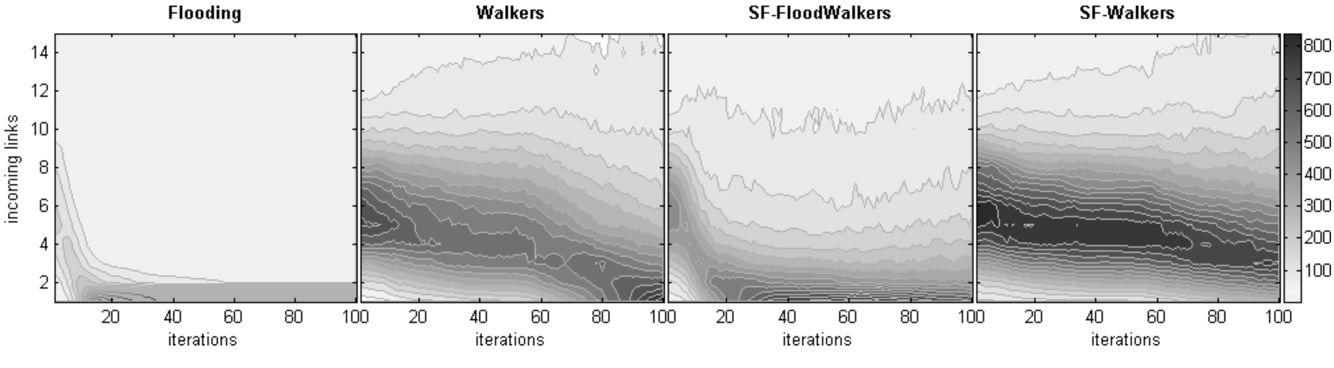

Fig. 6 : Impact of Flooding, Walkers, SF-FloodWalkers and SF-Walkkers on the 5000-node network degree distribution. Though initially it was a Poisson Distribution it gradually changed to a Power-Law one.

ered recent providers and resources. SF-FloodWalkers is a combination of flooding and walkers whereas the SF-Walkers is modified k-walkers. Instead of using the outbound neighbour list only, a query may also be forwarded to recent requestors accessing via them recent providers and recently released resources.

A number of experiments comparing these mechanisms with flooding and k-walkers have shown that they can achieve almost stable success rate despite the power-law degree distribution. Their performance becomes much better than both flooding and walkers when the resources in high-degree nodes are consumed. These achievements come with a linear cost for both with SF-FloodWalkers producing 10 times more messages than SF-Walkers or walkers.

Ongoing research focuses on deploying the algorithms on evolving scale-free networks. Their parameters (*k*, *W* and *TTL*) need dynamic modification and not preset values. All of them can be used to keep the number of messages and latency as small as possible but preserving its success rate. Formal analysis of their properties will help on the identification of the conditions under which their performance improves. Finally, further comparisons with other blind techniques are important to evaluate their applicability.

## VI. REFERENCES

[1] R. Holt, "Some deadlock properties of computer systems," SIGOPS Operating Systems Review, vol. 6, no. 1, 1972, pp. 64-71.

[2] R. Albert and A.L. Barabasi, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, vol. 74, no. 1, 2002, pp. 47-97.

[3] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," in *Proceedings of the ACM SIGCOMM*, 1999, pp. 251-262.

[4] M. Jovanovic, F. Annexstein and K. Berman, "Modeling Peer-to-Peer Network Topologies through 'Small-World' Models and Power Laws," in *IX Telecommunications Forum*, 2001.

[5] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, 2003, pp. 167-256.

[6] H. Chen, H. Jin, J. Sun, D. Deng and X. Liao, "Analysis of large-scale topological properties for peer-to-peer networks," in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, 2004, CCGrid 2004, pp. 27-34.

[7] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies", *ACM Comput. Surv.*, vol. 36, no. 4, December 2004, pp. 335-371.

[8] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes", *Communications Surveys & Tutorials, IEEE* (2005), pp. 72-93.

[9] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 32, no. 4, October 2002, pp. 177-190.

[10] V. Kalogeraki, D. Gunopoulos and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the 11th International Conference on Information and Knowledge management*, November 2002, pp. 300-307.

[11] Q. LV, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on Supercomputing*, June 2002, pp. 84-95.

[12] D. Tsoumakos and N. Roussopoulos, "Analysis and comparison of P2P search methods," in *Proceedings of the 1st International Conference on Scalable Information Systems*, May 2006, vol. 152.

[13] K. Y. K. Hui, J. C. S. Lui and D. K. Y. Yau, "Small-world overlay p2p networks: Construction, management and handling of dynamic flash crowds," *Computer Networks*, vol. 50, no. 15, October 2006, pp. 2727-2746.

[14] B. Bollobás and O. Riordan, "The Diameter of a Scale-Free Random Graph," Combinatorica, vol. 24, no. 1, January 2004, pp. 5-34.

[15] D. Callaway, M. E. J. Newman, S. H. Strogatz and D. J. Watts, "Network Robustness and Fragility: Percolation on Random Graphs," *Physical Review Letters*, vol. 85, no. 25, December 2000, pp. 5468 - 5471.

[16] R. Albert, H. Jeong and A. L. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, July 2000, pp. 378-382.

[17] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman, "Search in power-law networks," Phys. Rev. E, vol. 64, no. 4, 2001.

[18] P. Fraigniaud, P. Gauron and M. Latapy, "Combining the use of clustering and scale-free nature of user exchanges into a simple and efficient P2P system," in *Euro-Par 2005 Parallel Processing*, vol. 3648/2005, August 2005, pp. 1163-1172.

[19] M. Mihail, A. Saberi, P. Tetali, "Random walks with lookahead in power law random graphs", December 2007. http://www.cc.gatech.edu/~mihail