Firewalks: Discovery Mechanism for Non-replicable Reusable Resources

G.Exarchakos, N.Antonopoulos and K.Zhang

Department of Computing, University of Surrey, Guildford, Surrey, United Kingdom
e-mail: {g.exarchakos, n.antonopoulos, k.zhang}@surrey.ac.uk

Abstract

P2P Networks are traditionally used for file sharing, but have recently attracted attention for non-replicable reusable and mobile resource sharing. Existing blind search mechanisms for random directed graphs appear to have weak performance since they cannot cope with the dynamic behaviour of these resources. Firewalks is a blind search technique based on which queries hop from node to node using its most recently created link. This may ease the access to recently moved resources (from providers to requestors) and discovered providers. A number of experiments indicated that this technique has better and more stable performance without significant extra messages and manages to preserve the topology of the overlay in contrast to flooding and walkers.

Keywords

Peer-to-peer, adaptive search, non-replicable resources.

1. Introduction

Peer-to-Peer (P2P) Networks are a well used distributed resource discovery paradigm and appear to be suitable for large-scale deployment on dynamic environments. In practice they are application-level networks (overlays) decoupled from the underlying packet routing mechanisms. Even though traditionally used for file sharing, recent research interest focuses on applications beyond that, such as decentralized computational resource sharing. Each peer (node) of these networks executes an application-specific distributed algorithm that handles the resource discovery and affects the network's topology evolution.

The main focus of this paper is the discovery of non-replicable reusable mobile resources over P2P overlays. Upon discovery, these resources cannot be copied and can only be used by a single task at a time. After the task's termination it becomes available to other tasks. Computational resources (CPU, storage, services etc…), in principle, is a good example of these resources.

Based on Holt (Holt, 1976), reusable resources are those that allow exclusive only access (no more than one user at a time). 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, all these features, possible failures, uniqueness, exclusive usage and possible mobility, contribute to their intermittent behaviour and restrict their availability.

Many real world systems (Albert and Barabasi, 2002) (Faloutsos *et al.* 1999) (Jovanovic *et al.* 2001) (Newman, 2003) 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 (Albert and Barabasi, 2002) 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 (Chen *et al.* 2004).

Preferential attachment on networks with non-replicable reusable resources may severely deteriorate the efficiency of the deployed search mechanism, if 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 or moved. 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 form 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, reusability and mobility of resources.

Firewalks is the proposed blind search mechanism which is based on the idea that the requestors build up a knowledge about the network's status as long as they discover providers and resources. However, this knowledge is not hints or accurate information of the resources location as opposed to informed search techniques. It is information available to any node: its directed links to neighbours. The providers discovered by a neighbour's query may still have available resources and the discovered resources these neighbours currently use may soon be released. The more recent the requestor the more predictable his own and his neighbours' status is.

CPU, memory or storage P2P overlays could be three potential applications of firewalks. All of these resources are reusable and cannot be replicated. Representing these resources with a specification document that travels around the overlay, we can safely consider these resources mobile, too. CSOA (Exarchakos and Antonopoulos, 2007) is an existing example of such a system where the proposed algorithm is applicable.

The following sections present the work done on this or related topics, the description of the proposed algorithm and an evaluation of its performance and effect on the network's topology compared to flooding and random walkers.

2. Related Work

Popular discovery mechanisms that are used in various random networks are the flooding, breadth-first search (BFS) (Kalogeraki *et al.* 2002) and random walkers (LV *et al.* 2002). These blind techniques do not use any information to navigate the network but randomly choose the next target nodes. They are in general faultresilient 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 (Tsoumakos and Roussopoulos, 2006).

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 (Kalogeraki *et al.* 2002) (Tsoumakos and Roussopoulos, 2006). They require centralized or distributed repositories of hints on 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 and cannot efficiently cope with nodes' high failure rates (Tsoumakos and Roussopoulos, 2006). 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 (Albert and Barabasi, 2002) and allow logarithmic search efficiency (Hui *et al.* 2006). As argued in (Bollobás and Riordan, 2004), search techniques can be even more efficient (twice logarithmic) in scale-free networks. Though these networks have good resilience on node failures and attacks (Albert *et al.* 2000), a combined attack to selected high degree nodes may negatively affect the search efficiency or even fragment the network (Callaway et al. 2000).

LV *et al*. 2002, 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*. 2001, 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 *et al*. 2005, 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*. 2001 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 (Fraigniaud *et al*. 2005) but is not the case with the resources on which this work focuses.

Mihail *et al.* 2004 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.

3. Searching of Reusable Resources

High-degree nodes of a scale-free network are connected to a good number of other nodes thus making them easily discoverable. The main concept behind the proposed algorithm is that a query forwarded to recent requestors can quickly locate the good and famous providers. Using prior knowledge collected by other nodes, one can improve the success of his queries. This knowledge is recorded on each node by updating its neighbour list with the discovered providers. Every received answer triggers the updating process of requestor's neighbour list. Therefore, a recent requestor may have recently discovered resources which may soon release and know a set of other potential providers. The more distant in the past a requestor's last query is, the more uncertain its current status (requestor/provider) becomes.

Following the principles above, the proposed search mechanism is the firewalks. Initially, the requestor forwards a query to a number of its neighbours which in turn forward it to a single one of theirs. Its main difference from random walkers is that at each step the selected link is not random but the most recently created one. Each node may create a link to or accept from another one. While the former is an outgoing link and the node outbound neighbour, the latter is an incoming link from an inbound neighbour. By differentiating the outgoing from incoming links, the search mechanism is aware of recent providers and requestors.

3.1. Firewalks Algorithm Description

Firewalks 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. Practically, the INL's size cannot be indefinite but is much bigger than ONL's. The query originator node starts k-walkers selecting the ***k*** most recent neighbours from both its ONL and INL. Each walker travels from his originator via intermediate nodes (intermediaries). Firewalks are to be deployed on distributed non-replicable and reusable resource networks where a single response is adequate. Therefore, each walker terminates either upon discovering a response provider or after TTL steps away from his source. The resource is released after use, stays in the requestor node and then moves upon request.

If a node receives a query, is TTL-1 hops away from its originator and has not the requested resource, it forwards the same query to all the ***k*** most recent ones instead of a single one. A selected neighbour can be in any of the node's neighbour lists. That is, the node selects either one or k nodes from both INL and ONL. Intermediate nodes may use the same multicasting if their most recent link is not older than a predefined time window ***w***. After these multicasting actions, query forwarding stops even if the maximum TTL is not yet reached. The following diagram gives a visual representation of a query's propagation based on firewalks algorithm.

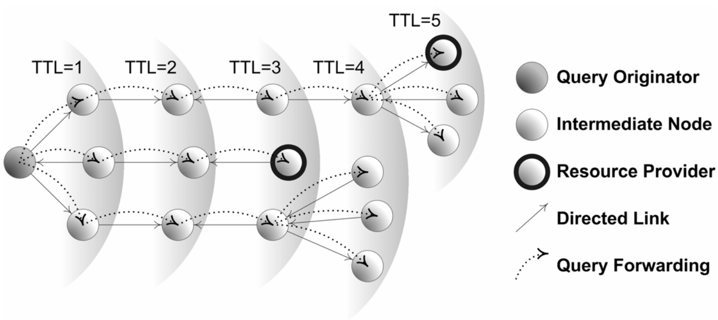

Figure 1: Example of Firewalks search mechanism

The figure above shows a snapshot of the firewalks messages as propagated using the most recent incoming and outgoing links of a network for a maximum TTL=5. The query originator chooses 3 links (3-Firewalks) to forward the query to and every intermediate just one. If the resource is found then the walker stops (middle walker). On TTL-1=4 the query is multicasted to 3 nodes and stops (upper walker). If the most recent link is created within the time window ***w*** then the query is multicasted and stops even if the TTL is not expired (lower walker). More details about the algorithm are presented in the following pseudocode.

```
  q={ttl, hops, resource};
  Firewalks(q) {
    q.hops = q.hops+1;
    if q.resource can be satisfied
      send response back to originator;
    else {
      if q.hops == q.ttl
        stop query forwarding;
      else if q.hops == q.ttl-1 {
        select k most recent links among the outgoing and incoming ones
        forward q to all these links
      } else {
        select the latest link among the outgoing and incoming ones
        if this link was created within the last w time units {
          q.hops = q.ttl-1;
          select k most recent links among the outgoing and incoming
ones
          forward q to all these links
        } else
          forward q to this link
  }  }  }
```

Figure 2: Pseudocode of Firewalks algorithm

All forwarding decisions are taken on a per-node basis following three ideas:

- Recent providers have some resources not yet used/moved/consumed by their neighbours.

- Recent requestors have discovered recent providers and may have released the resources they used. Outgoing only links drive the walkers to high-degree nodes and therefore their resources are quickly used. Firewalks, swapping between out- and inbound neighbours avoid the traps of famous but useless nodes.

- If the latest link's lifetime is shorter than a preset period, the node is very recently involved in a search activity and there is increased probability to discover the requested resource down that route. If that route is not successful, it is assumed that the walker can only further hop to older links. To avoid that latency, firewalks do one-hop broadcasting and stop. This may introduce a few more messages but are upper-bounded since only one such fixed-size broadcasting is allowed per walker.

Each node's INL may contain many neighbours, even disconnected ones to which the node may re-link depending the ***w***. Practically, INL has a maximum number of entries which are updated in a first-in-first-out mode.

4. Simulation and Results

Firewalks search technique aims to achieve a good performance despite the intermittent behaviour of the non-replicable reusable resources. A special-purpose simulator was built in Matlab to evaluate the technique on four different aspects:

- Success rate: the number of successful queries over the total number of queries. A query is successful when at least one appropriate resource is discovered. The success rate measures the goodness of the algorithm.

- Hops per response: the number of steps required to discover the first answer. This gives an estimation of the latency (time spent to answer a query).

- Number of messages generated by the search mechanism while crawling the network. This represents the cost of the mechanism.

- Incoming degree distribution changes: In non-static networks, the deployed discovery mechanism may change the network's topology via re-wiring. These changes become obvious on the nodes' incoming degree distribution and may influence the success rate, latency and messages of the deployed discovery mechanism.

4.1. Experiments Setup

All experiments are based on a fixed-size network of 1000 nodes. Every node stays always active throughout the experiments and is connected with other nodes via 5 outgoing and any number of incoming links. There is only one type of resource in chunks distributed among the nodes. Each chunk has on average 50 with a variance of 30 units. All chunks together add up to 1000 units and each one is initially assigned to a random node id. The probability of an ID's selection follows the normal distribution with (mean,variance)=(500,200).

Three search techniques were simulated: flooding, walkers and firewalks with TTL=4. Flooding uses three random out of five outgoing links at every step while walkers only at the beginning (3-walkers). Firewalks deploy three walkers (3-Firewalks) with a broadcasting time window ***w***=TTL and ***k***=5 broadcasting receivers.

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 replicated. 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. Assuming that the load of the network is the number of generated queries, then the network load increases and decreases linearly throughout the experiments simulating load fluctuations (from 200 to 2000 queries per iteration).

As in CSOA (Exarchakos and Antonopoulos, 2007), the discovered resources, before accepted and transferred to the requestor, are reserved so that they are not available to other queries. If the requestor rejects the resources, simply because are not necessary any more or the query was earlier resolved by another provider, they do not move and remain free for other queries on the same node. This prevents the situations two or more nodes have reserved the same resources.

Finally, the developed simulator simulates a re-wiring technique. Every requestor, upon a new answer, replaces the oldest link from its outbound neighbour list with the answer's provider. This replacement takes place even if the answer is not used. This technique is used to keep the links as recent as possible and ease the discovery of recent providers or requestors.

4.2. Results and Evaluation

All necessary data was generated by the simulator and a number of diagrams were produced depicting firewalks' benefits, costs as well as their effects on the network's topology.

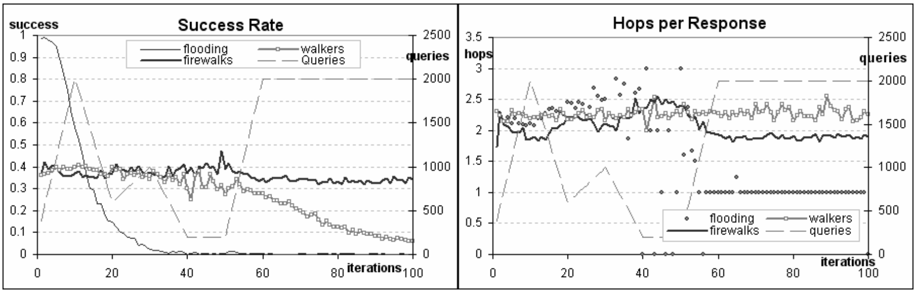

Figure 3: Achieved average success rate per query and hops per accepted response for flooding, walkers and firewalks.

The first set of graphs of Figure 3 present the average success rate per query and hops per response of flooding, walkers and firewalks. While the left axis is used for the evaluated metrics (i.e. success rate, hops), the right one is solely dedicated to the queries line which represents the workload fluctuations. Though, initially, flooding achieves 100% success, within the first 25 iterations its success drops below 10%. From 35th iteration onwards, its success rate is negligible and is not improved even when the network's load reaches its minimum. Walkers initially have about 40% success which gradually decreases below 10% more slowly than flooding. Firewalks manage to retain a success rate between 30-40% throughout the experiment.

There are two reasons why the success of flooding and walkers drops. The rewiring technique makes famous nodes even more famous as long as they have resources increasing their incoming links. Thus, blind techniques that use only the outgoing links to forward a query can easily locate those nodes. Moreover, nodes with enough resources have less probability to generate queries thus keeping a relatively outdated outbound neighbour list. Therefore, queries get trapped to paths that go quickly to famous nodes. Their resources move to requestors, away from these paths, as more and more queries reach them.

On the contrary, firewalks allow queries deviate from these paths using nodes' incoming links achieving a more stable success rate. Choosing the recent requestors, firewalks manage to follow the resources to where they move. Both walkers and firewalks have a bigger fluctuation when the load reaches its minimum. The low workload at this point generates few queries and the average success rate is calculated based on less data thus causing fluctuations.

Queries forwarded with flooding, for the first half of the experiment, have an increasing length as the first answer is always selected and closer resources are preferred. The movement of resources from high-degree providers to requestors, the trap of queries into the paths towards those nodes and the low workload cause fluctuations to the path length. The length rapidly reduces when no more resources can be found in those paths so that any response is discovered in immediate neighbours. Walkers seem to have a relatively stable length since with their randomness and slow resource consumption avoids the problems of flooding. Firewalks have, for the most of the experiment, smaller length than walkers because they target the nodes where a resource was recently moved to and not from.

From the right graph of the same figure, the path length of firewalks seems to get bigger when the workload drops. Rewiring takes place upon a received answer. The probability of a rewiring action decreases with the queries and therefore the existing links become older. Thus, the technique's capability to trace the resources' movement drops and travelling deeper in the network is necessary.

Workload	380	1860	640	920	200	380	2000	2000
Flooding	1921.947	922.2043	865.0938	870.913	877.1	874.3684	874.2	877.2
Walkers	11.76579	11.63548	11.625	11.67283	11.72	11.78684	11.729	11.937
Firewalks	12.15789	12.73925	13.85625	13.50326	14.57	14.49211	12.6665	12.6345

Table 1: Average Messages per Query for Flooding, Walkers and Firewalks

Table 1 presents the cost of the three simulated techniques. The first row is the number of queries from eight snapshots of the experiments. The remaining three rows are the average number of messages per query for each simulated technique on these snapshots.

While flooding has hundreds more messages than the other techniques, firewalks have from 0.39 to 2.7 more messages than walkers despite the local one-hop broadcasting effect (at worst case scenario the 3-Firewalks should have 15 more messages than walkers). This means that firewalks stop before TTL expires. The biggest difference in messages between firewalks and walkers is when the workload reaches its minimum. This is explained with the increase of hops per response on low workload situations as detailed above. Similarly, this difference (not a multiple of 5) is an indication that firewalks stop before TTL's expiry.

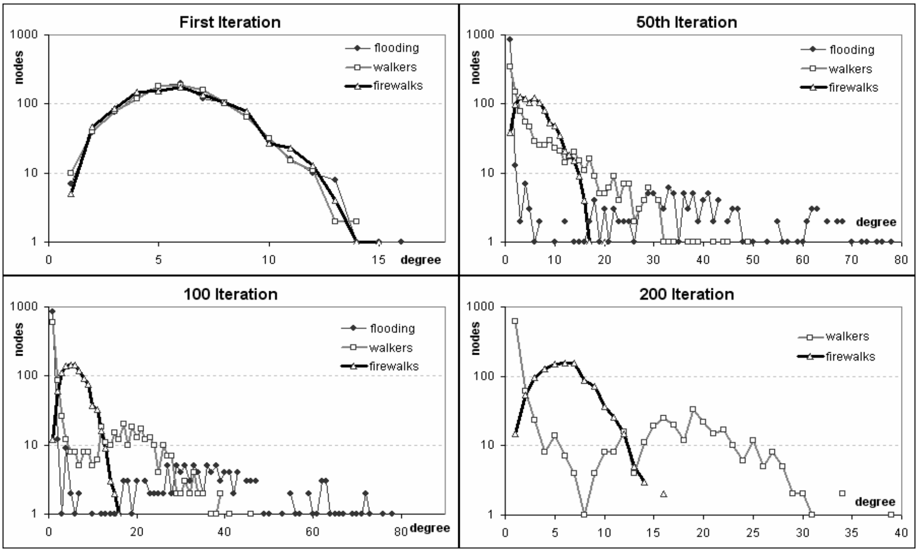

Figure 4: Incoming Degree Distribution on the 1st, 50th, 100th and 200th iteration for each search mechanism.

The previous figures present the incoming degree distribution of the network on four snapshots of the experiments (1st, 50th, 100th & 200th iteration). These experiments run for 200 iterations. Incoming degree is defined as the number of incoming links on a node. Its distribution shows the number of nodes with the same incoming degree for all possible degrees. The *x* axis of the graphs is the degree and the *y* axis is the nodes and has logarithmic scale. To read the graphs, answer the question: *'how many nodes have incoming degree i.e 5?'* Initially, first iteration, the network, no matter which search mechanism is used, has the same distribution (Poisson). On the following iterations, each technique has different implications on this distribution with the minimum caused by firewalks.

The initially Poisson distribution, on 50th iteration is transformed to a power-law one. The vast majority of nodes have low incoming degree for both walkers and flooding cases. However, firewalks preserves the initial distribution with a few changes. Its peak is a lower and its width bigger since some nodes with low incoming degree acquired more incoming links. An important observation is that, in flooding case, there are many nodes with very low degree (0-10), almost none for degrees 10-20 and then for degrees 20-50 few nodes.

On the 100th and 200th iterations the observations made above become clear. Firewalks still preserve the distribution while walkers and flooding create a small group of high-incoming-degree nodes. This group is present even for walkers on 100th iteration and quite clear on the 200th. The distribution widening in firewalks case seems to stop since the maximum degree does not exceed degree 20.

As stated above, flooding and walkers initially detect the providers, create links on them and queries are rapped on these paths. Due to random generation of queries, high degree nodes may generate queries and thus acquire new resources. This, in turn, may further increase the incoming degree of that node thus creating this small group of high-degree nodes.

5. Conclusions and Future Work

This paper identifies the key characteristics of non-replicable, reusable and mobile 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.

These features make existing blind search mechanisms inappropriate in such environments. Though the resources move as they are used, flooding and walkers cannot trace these movements and keep directing queries to high-degree nodes. The design of a new search algorithm, firewalks, was based on walkers to reduce the high cost of flooding and on the idea that recent requestors can provide either a recently discovered resource or information about potential providers. This idea got implemented by treating equally incoming and outgoing links on the query forwarding. A difference from walkers is that on the final hop the query is replicated and forwarded to more than one link. As soon as a response is discovered the firewalk stops.

Based on a number of experiments on a specific environment with the presence of a rewiring technique, firewalks mechanism has better success during most of the experiment compared to flooding and walkers. The average query path length is smaller than with the other two mechanisms in high workload situations and bigger in low ones. Firewalks' cost in messages is much less than flooding's and a bit higher than walkers'. This cost increases in low workload situations. Firewalks manage to retain the initial incoming degree distribution while flooding and walkers transform the network to a power-law one with few high-incoming-degree nodes. While the deployed rewiring technique tends to for a power-law network, firewalks, by using both incoming and outgoing links, manages to cancel this topology transformation and use it for its own advantage. The more queries on the network, more frequent the resources' movements are and smaller the average query path length while achieving a relatively high success rate.

Ongoing research focuses on deploying the algorithms on evolving 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.

6. References

Albert, R. and Barabasi, A.L., (2002), 'Statistical Mechanics of Complex Networks', *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47-97.

Albert, R., Jeong, H. and Barabasi, A.L., (2000), 'Error and attack tolerance of complex networks', *Nature*, vol. 406, pp. 378-382.

Adamic, L., Lukose, R., Puniyani, A. and Huberman, B., (2001), 'Search in power-law networks', Phys. Rev. E, vol. 64, no. 4.

Bollobás, B. and. Riordan, O., (2004), 'The Diameter of a Scale-Free Random Graph', Combinatorica, vol. 24, no. 1, pp. 5-34.

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

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

Faloutsos, M., Faloutsos, P. and Faloutsos, C., (1999), 'On Power-Law Relationships of the Internet Topology', *Proceedings of the ACM SIGCOMM*, pp. 251-262.

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

Exarchakos, G. and Antonopoulos, N. (2007), 'Resource Sharing Architecture for Cooperative Heterogeneous P2P Overlays', *Journal of Networks and Systems Management*, vol. 15, pp. 311-334.

Holt, R., (1972), 'Some deadlock properties of computer systems', *SIGOPS Operating Systems Review*, vol. 6, no. 1, pp. 64-71.

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

Jovanovic, M., Annexstein, F. and Berman, K. (2001), 'Modeling Peer-to-Peer Network Topologies through 'Small-World' Models and Power Laws', *IX Telecommunications Forum*.

Kalogeraki, V., Gunopoulos, D. and Zeinalipour-Yazti, D., (2002), ‘A local search mechanism for peer-to-peer networks’, *Proceedings of the 11th International Conference on Information and Knowledge management*, November, pp. 300-307.

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

Mihail, M., Saberi, A. and Tetali, P., (2004), ‘Random walks with lookahead in power law random graphs’, http://www.cc.gatech.edu/~mihail, (Accessed: 15 January2008)

Newman, M.E.J., (2003), ‘The Structure and Function of Complex Networks’, *SIAM Review*, vol. 45, no. 2, pp. 167-256.

Tsoumakos, D. and Roussopoulos, N., (2006), ‘Analysis and comparison of P2P search methods’, *Proceedings of the 1st International Conference on Scalable Information Systems*, May, vol. 152.