Self-protection in P2P Networks: Choosing the Right Neighbourhood

Ivan Martinovic[1], Christof Leng[2], Frank A. Zdarsky[1], Andreas Mauthe[3], Ralf Steinmetz[4], and Jens B. Schmitt[1]

[1]Distributed Computer Systems Lab, University of Kaiserslautern, Germany
[2]Databases & Distributed Systems, University of Technology Darmstadt, Germany
[3]Infolab 21, University of Lancaster, UK
[4]Multimedia Communications Lab, University of Technology, Germany

Abstract. In unstructured peer-to-peer networks, as in real life, a good neighbourhood is not only crucial for a peaceful sleep, but also for an exchange of important gossips and for finding good service.
This work investigates self-protection mechanisms based on reputation in unstructured peer-to-peer networks. We use a simple approach where each peer rates the service provided by others and exchanges the collected knowledge with its direct neighbours. Based on reputation values peers manage their connections to direct neighbours and make service provisioning decisions.
To quantify the impact of our proposed scheme, we implement a simple protocol in a fully unstructured peer-to-peer network. We show that free riding and the impact of malicious peers trying to poison the network with bad files is minimised. Furthermore, we show that a good neighbourhood protects peers from selecting bad files, while free riders suffer in a bad neighbourhood of malicious peers.

Keywords:
Self-protection, Peer-to-Peer, Trustworthiness, Reputation, Free-Riding, Network Poisoning

1 Motivation

Network poisoning is a problem where malicious peers try to upload invalid files into a peer-to-peer network. Although standard cryptographic solutions like hash functions can help us to check the integrity of files, the impact of network poisoning has increased dramatically in the last few years. The reason is that in a decentralised, large-scale network where content is provided by the users themselves, one cannot easily transfer techniques from centralised content delivery environments. Many peer-to-peer networks have lost their popularity because the content provided within those networks is mostly malicious (infected by virus) or invalid, wasting the users' time and bandwidth. On the other hand, there is an increasing number of decentralized systems using reputation schemes as security measure to protect users and reward cooperative behaviour.

Recently, various research contributions studied the impact and strategies of network poisoning and analyzed its magnitude within P2P networks [14,5]. In this work, we analyse the concept of self-protection against network poisoning and free riding within unstructured peer-to-peer networks by using concepts from soft-security and

traditional security. We model our peer-to-peer network as a basic Gnutella network [12] without any structure and extend it only by a simple protocol we call Simple Trust Exchange Protocol (STEP), which enables peers to decide whom to connect to as direct neighbours and to whom to provide service. As the result, a trust-based construction of an overlay network is accomplished by local decisions.

To avoid the extreme case of individual "learning-by-doing" STEP provides a mechanism to exchange knowledge between its direct neighbours, which helps a neighbourhood to avoid selecting bad service or potentially malicious neighbours. Moreover, as the search in unstructured networks is mostly based on different flooding algorithms, having a good neighbourhood becomes essential for a peer's own search success.

Although scaling problems of the original Gnutella network are well-known, we take advantage of a fully unstructured network to avoid "deus ex machina" intervention and to support the evolutionary growth of topology. Furthermore, the concept introduced in this work does not depend on the flooding approach of Gnutella and could be adapted to different routing mechanisms (e.g. Ultrapeers).

2 Simple Trust Exchange Protocol (STEP)

To support service rating and exchange of knowledge among neighbouring peers, we introduce Simple Trust Exchange Protocol (STEP), the pseudo-anonymous token-based protocol. A token is a mutually signed transaction receipt which serves as a proof of service provision between a consumer and a provider (in all our scenarios we consider a service to be a file transfer between a provider and a consumer, but clearly, other scenarios are also conceivable). To participate in the system, every peer needs to create a public-private key-pair as its identity. Because the identity has no ties to the outside world and only becomes meaningful by a token signed with it, there is no need for any public key infrastructure.

To discourage the change of identity, in this work we follow the idea of "no profit to newcomers" [13], where every new peer gets the minimal possible reputation.

STEP relies on a peer-to-peer network to discover and deliver services and only extends systems by a token creation mechanism and knowledge exchange between peers. The structure of a token is shown in Table 1.

Name	Description
CID	Consumer's identity
PID	Provider's identity
TS	Timestamp of token creation
Length	Length/Duration of service (e.g. File size)
Rating	Consumer's rating ({*good*,*bad*})
CSig	Consumer's Signature
PSig	Provider's Signature

Table 1. Token structure

2.1 Token creation

A token is created upon a request of a service and is supplemented with rating after a transaction has finished. Upon the service request a consumer creates an initial token by filling the data fields except $Rating$ and signs it with his private key. The consumer sends the token and his public key to a provider. If the provider accepts the service request he also signs the initial token and sends the token and his public key back to the consumer. After the transaction the consumer rates the provider (filling the $Rating$ field) and finalizes the token by re-signing it. Finally, he sends a copy of the token to the provider.

The objective behind the consumer's double signing is to mitigate the incentives for not finalizing the token. For example, a free-rider could decide to reject finalization and publication of the token to avoid being detected as a consumer. In this case the initial token containing the free-rider's signature will be published by a provider. As a result, other peers will be able to collect tokens and if the provider is trusted the free-rider can still be detected.

2.2 Knowledge Exchange

Every peer can improve its own knowledge over another peer directly by requesting a service and then rating it, or indirectly by collecting other ratings about that peer and then using them to compute its own trust value.

Knowledge exchange between peers is realized through distribution of tokens within an overlay network and by storing the received tokens locally. We use the Gnutella-typical flooding approach for distributing tokens in a way similar to query for services. The $Knowledge$ message is wrapped in Gnutella $Query$ messages to provide maximum compatibility with legacy peers in mixed networks, analogous to the approach described in [8]. To enable the verification of a token, in case the local neighbours do not have the required keys, a peer forwarding the token appends the missing public keys of both, the consumer and provider, to the $Knowledge$ message.

2.3 Decision Making

The computation of a peer's reputation is a subjective matter and every peer can choose its own method for interpreting the collected tokens. In this work we focus on the impact of reputation and not on its calculation, which is why we use a simple approach of calculating the number of collected tokens where the provider was rated as good, and subtracting the tokens where the rating was bad. Although this mechanism does not deal with nodes that are actively trying to cheat the reputation system (such as attacks from malicious group of peers which can mutually sign tokens), STEP provides enough room for implementation of more sophisticated algorithms. For example, one peer can choose to follow the PGP's "Web-of-Trust" concept by considering the trustworthiness of both signatures, as well as by utilizing the transitivity of trust relationships (e.g. [4]) or more accurately computing reputation of token-based systems with a more complex attacker model (e.g. [10]). Furthermore, a very restrictive approach would be to base the trust computation only on a closed group od peers (similar concepts already exist

as “darknets” which represent a closed group of members within a public peer-to-peer network).

Search and Service Selection. After executing a Gnutella query and receiving search results a peer can immediately compute a reputation of available provider peers based on its local knowledge. The consumer peer then selects a file with a probability proportional to the sum of reputations of all providers offering the same file [13].

Due to assigning a reputation with positive minimum to every peer, every provider has the chance of being selected. This method decreases the overloading of a few top providers and increases the chance of selecting a newly joined peer if it provides frequently searched files.

Service Providing. After selecting a provider, a consumer sends a service request. If there is enough free bandwidth for an upload connection, the provider grants the service immediately to maximize the utilization of its bandwidth. Otherwise the provider computes a reputation of the new consumer and compares it with the lowest reputation of those consumers already connected. If the reputation of the new consumer is higher than that of any already connected consumers, the connection with the lowest value is cancelled and replaced with a new one. Although it would be better to use a non-preemptive queue to finish the already started file transfer instead of cancelling a transaction in progress, Gnutella does not support such concept.

Choosing the Neighbourhood. In Gnutella, every peer has a minimum number of desired neighbours (typically 4 for real-world Gnutella networks). If its current number of neighbours is less then desired the peer actively tries to connect to more neighbours. Instead of choosing the neighbours arbitrarily as in the classic Gnutella a STEP peer tries to connect to neighbours with a reputation at least as high as its current neighbours. If no such peers are available an arbitrary peer is chosen.

Most of the peers, depending on their bandwidth, support more than the minimum number of neighbours and thus are able to accept incoming connection requests by new peers. In the classic Gnutella every peer with available connection slots accepts all incoming connection requests. In STEP only requesting peers with a reputation that is in average not lower than those of the current neighbours, are accepted. If the STEP peer has reached the maximum number of Gnutella connections, then, for every further connection request the reputation of a requesting peer is compared with the local neighbourhood. If a neighbour with a lower reputation is already connected, it will be replaced.

3 Experiments

Service mentioned in all of our scenarios is a file transfer and we have modelled 4 different peer profiles:

– *Cooperative Peer*: a peer that offers services to the network by sharing valid files,

– *Freerider:* an opportunistic peer that does not share any files but only requests services,
– *Foul Dealer*: a malicious peer with a high bandwidth and maximum number of files (highest probability to answer with $QueryReply$), which tries to upload invalid files only (network poisoning),
– *Good Dealer*: opposite of the Foul Dealer, an altruistic peer with a high bandwidth and a large number of valid files.

The profile of a *Good Dealer* is only for measurement purposes in order to better compare the impact of good and bad dealers within a STEP-enhanced and legacy Gnutella network.

3.1 Configuration

One of the advantages of using Gnutella for our investigation is the availability of rich empirical data. To provide realistic assumptions of the peers' online times and available bandwidths, we have used empirical distributions based on the Gnutella analysis from [16]. The file popularity distribution was also based on empirical data taken from [17] which analysed the Kazaa peer-to-peer network. Other simulation parameters are listed in Table 2.

Network Size	2048, 4096
Number of Files	192.000
Token Validity Time	10 h
Query Interval	2-4 min
Simulation Time	80 hours

Table 2. Simulation Parameters

The number of Free Riders was set to 50% for both network sizes and number of Foul Dealers and Good Dealers was set to 64. The bandwidth was divided into upload- and download streams which can be asymmetric to conform to a real peer-to-peer network with heterogeneous peers (e.g. DSL users with 1024 Kbit/s downstream and 128 Kbit/s upstream). The minimum bandwidth represent modem users with both upstream and downstream of 32 KBit/s and the maximum bandwidth is a broadband user having 10 MBit/s for both directions. Both, Foul Dealer and Good Dealers are assumed to have maximum bandwidth, where other peers follow empirical distributions from the real-world statistics.

As the goal of this work was not to provide highly sophisticated mechanisms to detect malicious behaviour we assume all peers conform to Gnutella and/or the STEP protocol with the exception that every peer rated as bad will not publish tokens. The service is rated as good if the file was sound and bad if not. Every peer makes a wrong rating (unintentionally) with a probability of 5%. In the initialization phase, when a new peer joins the network it randomly tries to connect to peers from a global peer cache.

Because of a long transient phase of network initialization, we simulated 80 hours of network activity. This proved to be very important as the steady state was reached after approximately 15 hours. All simulations were conducted on an Athlon XP 2500+ with 1 GB RAM and one simulation run took approximately 24 hours to finish. The memory requirements of the simulation were 889 MB.

3.2 Results

Network Poisoning. In Figure 1 we analyse the impact of Foul Dealers and Good Dealers on network poisoning by comparing the number of successful uploads of bad files within both scenarios. As it can be seen, in the STEP-enhanced scenario the impact of Foul Dealers is decreased by 77% while the impact of Good Dealers is increased by 46%.

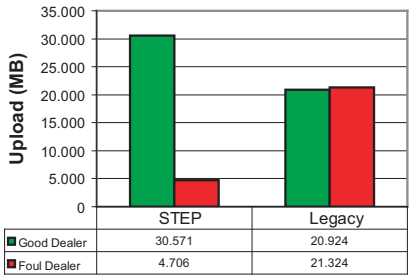

Fig. 1. Network Poisoning

Figure 2 analyses the downloads between cooperative peers and free-riders, again for the two scenarios. The total number of good downloads (cooperative and free-riders) remains almost equal in both networks, but the number of good downloads of cooperative peers is almost 20% higher than for free riders. In a legacy network there is no incentive to behave cooperatively since all peers equally download good and bad files. Even more interesting is the fact that there is only 13% bad downloads within the STEP-enhanced network out of which 83% are performed by free-riders.

Furthermore, we measured the number of good and bad query results (a bad result is service offered by a foul dealer) for both cooperative and free-rider peers when they search for a specific file. As previously mentioned, the search in the Gnutella peer-to-peer network is a simple flooding. Due to the limitation of flooding by a query's TTL counter, it is important for a peer to be in a good neighbourhood. As it can be seen in Figure 3, the number of bad results for cooperative peers decreases dramatically, while the free-riders are losing good results and gaining more bad results. The overall number of query results is decreased after 15 hours as a result of neighbourhood clustering.

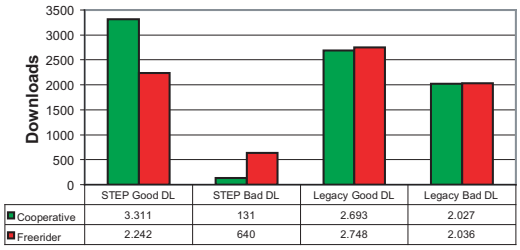

Fig. 2. Download Scenario

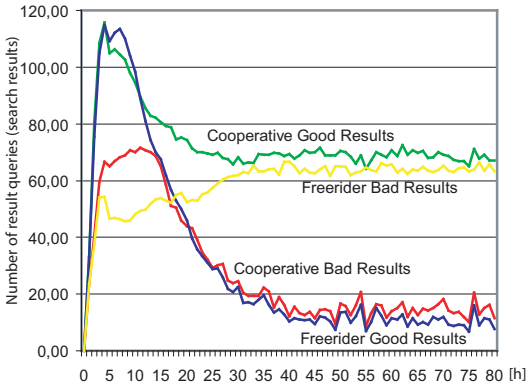

Fig. 3. Search results

Knowledge. Figure 4 (left) shows local knowledge of a peer over all tokens in the network. As it can be seen, after 15 hours the local knowledge covers over 40% of all tokens existing in the network. The reason for its decrease after the first 15 hours is the clustering of the network, when the good neighbourhood has been established and tokens from the bad neighbourhood cannot reach most of the peers within the good neighbourhood. The Figure 4 (right) shows the accuracy of a peer's local knowledge about every other peer. As a result, although every peer knows only 40% of all tokens in the network it has on average 50% of all tokens associated to the peer when calculating its reputation. This is due to the limited horizon of query and knowledge exchange messages, both affected by the location of a peer in the overlay network in the same way. Thus, the received tokens will be more related to providers' and consumers' reputations actually calculated than tokens beyond the horizon, which never reached the peer.

Topology. Figure 5 shows topologies captured during the simulation times of 10h, 20h, 30h and 40h. After 40h the topology has stabilised and no significant changes occured afterwards. A clear separation of neighbourhoods can be seen after 25 hours of

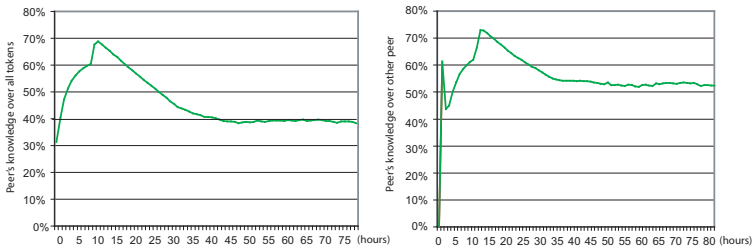

Fig. 4. Local knowledge

simulation where cooperative peers have distanced themselves from most of the Foul Dealers creating a neighbourhood with the Free Riders.

There are still many connections between the neighbourhoods, but due to the TTL scoping of $Query$ and $Knowledge$ messages most message exchanges remain inside neighbourhoods, resulting in decreased number of query results from remote neighbourhoods. This also results in a decreased number of queries sent from free-riders to good peers (and thus self-protecting) but also from other good neighbourhoods.

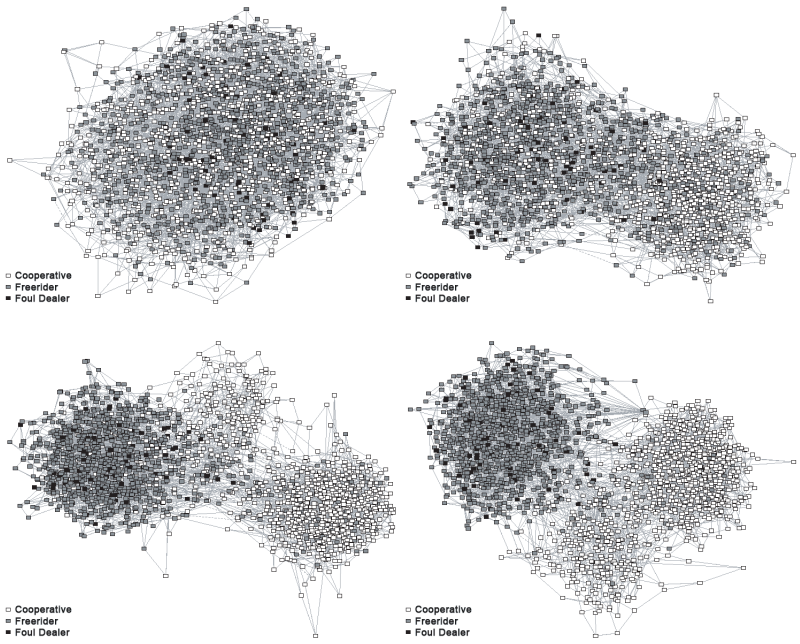

Fig. 5. Topologies after 10h, 20h, 30h and 40h

STEP Overhead. The Overhead of STEP can be mainly attributed (98%) to $Knowledge$ messages as they contain tokens and public keys. In the worst case every token has 2 public keys and if the key length is set to 512 Bit every token costs 128 Bytes. Nevertheless we implemented a simple caching strategy where only new keys are forwarded to direct neighbours decreasing the cost to only 0.6 keys per token and still provide key distribution within fully unstructured peer-to-peer network. Furthermore, the length of an average knowledge message was 750 Bytes and maximally 1500 Bytes.

As the result, we can state that the price to pay for STEP is a 28% increase in bandwidth usage over standard Gnutella control message overhead; yet, its advantage is a dramatical decrease in the number of bad downloads and enhanced incentives for cooperative peers as they are now able to increase the utilization of their bandwidth with valid downloads which are preferred over those of the free-riders.

4 Related Work

The problems caused by selfish and malicious participants in peer-to-peer and ad hoc networks have been examined in several studies (e.g. [9,11,3,2]).

Probably the first to propose a practical solution based on traditional cryptography and structured overlay networks were Aberer and Despotovic [1] using the P-Grid system to store and retrieve complaints about misbehaving peers. The principle of "no profit to newcomers" together with other design goals for peer-to-peer networks (self-policing, anonymity, no profit to newcomers, minimal overhead, robust to malicious collectives) has been described by Kamvar et al [13]. They created the EigenTrust peer-to-peer reputation system according to those design goals. EigenTrust uses a distributed way of approximating the global trust value of a peer. Damiani et al [8] proposed a distributed reputation system as an extension to Gnutella called P2PRep, with a seamless migration path for existing Gnutella networks. Those integration concepts can be applied to our system in a similar fashion. Instead of using signed transaction receipts they rely on a live poll of user opinions, which has its own advantages and disadvantages. The usage of reputation as a trust building concept within a decentralised network of WLAN providers was described by Efstathiou and Polyzos in [10]. They apply the concept of a token-based reputation to a system they call a Peer-to-Peer Wireless Network Confederation (P2PWNC). For reputation calculation they use a maximum flow based algorithm called NWAY which is very resilient to cooperating groups of malicious nodes. The tokens in their system are exchanged by interacting parties, instead of the active publication mechanism we use.

Very interesting work on using trust to build an unstructured overlay network (Adaptive P2P Topologies) was introduced by Condie et al. [6]. Instead of a reputation system, they count only the subjective experience of every node itself. This minimizes the overhead of the protocol but also limits the available information on other peers. As the result of their adaptive topologies, the malicious peers and free-riders are pushed to the fringe of the network, which is similar to our work. However, the major difference is the usage of tokens to avoid distribution of only subjective reports (gossips). We considered the token mechanism to provide further advantages such as, choosing indivdual reputation algorithms and support of the Web-of-Trust concept in a peer-to-peer network

where every peer can individually decide whose tokens can be more trusted based on the providers and consumers signatures.

Furthermore, tokens constitute the basis for coordination and control mechanisms as well as for pricing in commercial scenarios as described by Liebau et al. [15].

5 Conclusion and Future Work

This paper presents our initial work on trust in unstructured peer-to-peer networks. Our objective is to investigate into more depth both, the impact and efficiency of self-protecting mechanisms based on trust computation. We have chosen to use a completely unstructured peer-to-peer network as it allows us to investigate impacts on topology, as well as to support evolutionary growth of the network.

STEP defines a simple way to create transaction receipts and is adaptable to many different methods for knowledge exchange. The algorithm presented here adapts the Gnutella query method for maximum compatibility with legacy networks.

The Gnutella network is clearly not very scalable but many alternative routing mechanisms for message routing in peer-to-peer networks have been proposed in recent years. Alternatively, tokens can be accumulated at each intermediate peer and forwarded in regular intervals to reduce the number of messages needed. Also, as a part of our future work we intend to investigate how trust can be used for a better routing of search messages to mitigated overloading of trusted peers and to create incentives against free-riding. The idea of trust-based routing could help us to scale the efficiency of routing scheme based on trustworthiness of a peer or a neighbourhood.

Furthermore, if the service provided by the peers is the upload of shared content, the semantics of this content may also be used to achieve better routing of tokens, which leads to a more precise reputation calculation and consequently to a better reorganization of the overlay. It has been observed that content is normally clustered into categories with most peers being active only in very few categories [7]. Therefore, the interaction topology of such content distribution networks are typically highly clustered. This circumstance may be used to distribute tokens to peers which can make use of them more efficiently.

The STEP system is not only highly independent of the knowledge exchange method but also of the choice of reputation algorithms such that every peer can choose an algorithm independently. The choice of usable algorithms is very broad as every node can store a relevant proportion of the globally available knowledge about rated peers. To optimize the overall system performance the chosen algorithm should not only be resilient against attacks, but should also provide good results even with little aggregated data. To follow this idea, we consider not only token mechanisms, but also concepts of accumulated gossiping for building trust within peer-to-peer networks.

References

1. K. Aberer and Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In *CIKM '01: Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 310–317, 2001.
2. S. Buchegger and J.Y. Le Boudec. Performance analysis of the CONFIDANT protocol. In *MobiHoc '02: Proceedings of the 3rd ACM International Symposium on Mobile Ad hoc Networking & Computing*, pages 226–236, 2002.
3. S. Buchegger and J.Y. Le Boudec. A Robust Reputation System for Mobile Ad-hoc. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.
4. S. Capkun, L. Buttyán, and J-P. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.
5. N. Christin, A. S. Weigend, and J. Chuang. Content Availability, Pollution and Poisoning in File Sharing Peer-to-peer Networks. In *EC '05: Proceedings of the 6th ACM Conference on Electronic commerce*, pages 68–77, 2005.
6. T. Condie, S. D. Kamvar, and H. Garcia-Molina. Adaptive Peer-to-Peer Topologies. *IEEE Transactions On Systems, Man and Cybernetics, Part A*, 35(3):385–395, May 2005.
7. A. Crespo and H. Garcia-Molina. Semantic Overlay Networks for P2P Systems. Technical report, October 2002.
8. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Managing and Sharing Servents' Reputations in P2P Systems. *IEEE Transactions on Data and Knowledge Engineering*, 15(4):840–854, July 2003.
9. P. Dewan and P. Dasgupta. Pride: Peer-to-peer Reputation Infrastructure for Decentralized Environments. In *WWW Alt. '04: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pages 480–481, 2004.
10. E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos. Stimulating Participation in Wireless Community Networks. Technical report, June 2005.
11. M. Feldman and J. Chuang. Overcoming Free-riding Behavior in Peer-to-Peer Systems. *SIGcom Exch.*, 5(4):41–50, 2005.
12. Gnutella Developer Forum. The Annotated Gnutella Protocol Specification v0.4. http://rfc-gnutella.sourceforge.net/developer, 2006.
13. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 640–651, May 2003.
14. J. Liang, R. Kumar, Y. Xi, and K. Ross. Pollution in P2P File Sharing Systems. In *IEEE INFOCOM, Miami, FL, USA*, March 2005.
15. N. Liebau, V. Darlagiannis, A. Mauthe, and R. Steinmetz. Token-based Accounting for P2P-Systems. In *Proceeding of Kommunikation in Verteilten Systemen KiVS 2005*, pages 16–28, February 2005.
16. H. Luemkemann. Leistungsfaehige Verteilte Suche in Peer-to-Peer File-Sharing-Systemen. Master Thesis, CS Department, University of Dortmund, 2002.
17. S. Saroiu, P. Gummadi, and S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN'02)*, January 2002.