EPiC Series in Computing
Volume 64, 2019, Pages 11–20
Proceedings of 28th International Conference
on Software Engineering and Data Engineering
EPiC
Computing

A Peer-to-Peer Reputation Evaluation System

Ming-Chang Huang
Department of Business Information System/Operations Management
University of North Carolina at Charlotte
mhuang5@uncc.edu

Abstract

Lack of incentives makes most P2P users unwilling to cooperate and lead to free-riding behavior. One way to encourage cooperation is through service differentiation based on each peer's contributions. This paper presents FuzRep, a reputation system for P2P networks. FuzRep uses fuzzy logic method which uses requester's reputation and provider's inbound bandwidth as input information to create incentives for sharing and to avoid overloading problems for primary file providers. Reputation sharing in FuzRep is implemented by interest-based selective polling, which can significantly decrease overheads for reputation communication.

Keywords: P2P network, free-riding, reputation system, incentive, fuzzy logic

1. Introduction

File sharing system is one of the most popular P2P applications today. One main issue, however, is widely noted in current P2P systems—free-riding. Measurement study of free riding on Gnutella exhibited by Adar in 2000 [1] indicated that approximately 70% of Gnutella users did not share any files, and nearly 50% queries were responded by top 1% peers. In contrast, the percentage of free riders in Gnutella had risen to 85% in 2005 [7]. This reveals how serious the free-riding problem has become. In fact, free-riding roots in the nature of P2P networks – anonymity, autonomy, but lack of incentive. Most users have chosen to freeride since they do not get any benefit by providing resources. In addition, issues such as "hotspot" and "the tragedy of the digital commons" are coming along with serious free-riding problem.

The proposed solution in this paper, FuzRep, is a reputation system for P2P networks. FuzRep is designed for two purposes—motivating information sharing through service differentiation and avoiding overloading for providers. To these ends, FuzRep uses fuzzy logic as a tool to manage reputation and bandwidth. Fuzzy logic fills the gap between engineering mechanical design and human linguistic understanding. While reputation communication is always an important issue in a fully decentralized environment, FuzRep applies selective polling method, which is inspired from social network researches [8, 9], to discover a peer's global contribution score by selectively polling peers in the same community. Figure 1 shows the architecture and processes of FuzRep.

The rest of paper is organized as follows. In Section 2, we present related work. In Section 3, we discuss the premises of this paper. In Section 4, we address the detail design of FuzRep. Experiments are presented in Section 5. Section 6 is our conclusion and areas for future work.

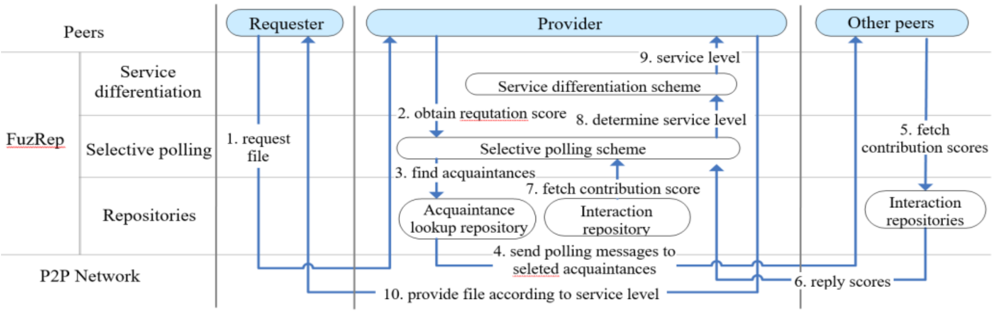

Peers
Requester
Provider
Other peers
FuzRep
Service differentiation
Selective polling
Repositories
P2P Network
1. request file
2. obtain requtation score
3. find acquaintances
4. send polling messages to seleted acquaintances
5. fetch contribution scores
6. reply scores
7. fetch contribution score
8. determine service level
9. service level
10. provide file according to service level
Service differentiation scheme
Selective polling scheme
Acquaintance lookup repository
Interaction repository
Interaction repositories

Figure 1. FuzRep architecture and operational processes.

2. Related Work

Most research on P2P free-riding aim at providing incentives to encourage corporation. While some recent studies are trying game theoretical approaches [2, 5], two major approaches are through monetary payment and reputation judging,

Payment schemes inherit the philosophy of economics, which treat P2P file exchanges as transaction activities and thus requesters should pay for their received files. Since tokens are required for each download, free riders are forced to share files to earn tokens. In [17, 18], incentives are driven by payment schemes. One advantage of payment scheme is its fast speed to converge; that is, peers tend to stop free-riding in a short time under payment schemes. However, one critical drawback of payment schemes is the high mental transaction cost indicated in [11, 14].

Instead, reputation systems learn from sociology concept, which treats each file exchange as a relationship building process. Sharing files to others brings higher reputation, and, on the contrary, free-riding leads to a bad social image. Reputation systems are used to map scores of reputation to adequate service levels. [3, 10] show how to manage reputation in a fully decentralized manner. However, most P2P reputation systems emphasize on the designs of security related issues rather than clearly provide incentives to encourage sharing.

3. Premises

We will discuss the basic design considerations and assumptions in this section.

3.1 Design Considerations

There are several considerations while we design the reputation system for P2P networks to make it more scalable and deployable.

1. The reputation management should not rely on any auditing authority, which means there is no centralized agent to store and deliver reputation information.
2. Overhead of reputation management should be minimized, especially the bandwidth consumption on communication messages.

The design of our reputation system is mainly based on Gnutella network; however, the reputation system should be easy to be deployed to other P2P systems.

3.2 Basic Assumptions

To realize FuzRep, there are two basic assumptions. First, we assume each peer keeps two repositories locally—*interaction repository* and *acquaintance lookup repository*. Details of the two repositories are described as below.

- The *interaction repository* contains two attributes (*servent_id*, *contribution_score*) where *servent_id* is the unique identifiers of transacted peers, and *contribution_score* is calculated from old interactions based on file sizes.
- The *acquaintance lookup repository* contains two attributes (*servent_id*, *IP_address*) where *servent_id* is inherited from interaction repository, and *IP_address* is the online acquainted peers' IP addresses which can be gathered and updated from Gnutella ping and pong descriptors [15].

To manage reputation in a decentralized way, interaction records must be stored locally. However, peers have motives to tamper their interaction records for their personal benefits, no matter to hide downloading activities or to exaggerate the shared amount. To avoid this, a peer's contribution scores are stored in other peers' interaction repositories that the peer has been interacted with. Therefore, as you can imagine, a peer's contribution scores are distributed among other interacted peers. To obtain a peer's complete contribution score, we have to collect it from those peers.

The idea of interaction repository is quite similar to XRep's servent repository [3]. We additionally consider file size as a rating criterion because the efforts to share a big-size file should be more than smaller one in terms of storage space, consumed bandwidth and required transmission time. Therefore, we separate files into 3 categories—greater than 10MB, between 1MB and 10MB, less than 1MB. Based on the file size distribution shown in [12], each category contains about one-third of total sharing files. Providers will earn 5 contribution scores for sharing a large file, and it will cost 5 contribution scores for the requesters to request that, while 3 for medium file and 1 for small file respectively. On the other hand, the acquaintance lookup repository is designed to accomplish reputation communication process. Details about it will be discussed in the following section.

Secondly, to accomplish selective polling, we also assume peers with the same interests form virtual communities automatically, which have been observed and proven in [9]. Therefore, peers in the same community will frequently interact with each other. It would be much more efficient to discover a peer's global reputation by polling other members in the same communities instead of fooding polling messages to unrelated peers.

4. FUZREP

FuzRep is a design of fuzzy-based reputation system for P2P networks. It includes three techniques—reputation determination, selective polling, and service differentiation. In this section, we are going to describe how FuzRep works by revealing answers of the following questions.

- How to determine a peer's reputation level? What are the criteria? How to maintain it?
- How and when to share the contribution information?

How to encourage sharing and discourage free riding? How to differentiate the service level?

4.1 Reputation Determination

In FuzRep, one's reputation is determined by his contributions to communities. To this end, a peer should save interaction information into local interaction repository, including the unique IDs of interacted peers and subjective, accumulated contribution scores of them. The interaction repository is updated after every successful interaction. Note that the initial local contribution score is set to 0 originally for any pre-unknown peers at their first interactions.

A global aggregated contribution score is used to determine a peer's reputation. It is carried out in two phase computes—personal reputation inference and global reputation deduction. In personal reputation inference, peer simply fetches targeted peer's contribution score from local interaction repository. Then, in global reputation deduction, he should run a reputation aggregation process, namely selective polling, in FuzRep.

4.2 Selective Polling

Reputation information sharing in the P2P environment is a big challenge. We classify different solutions into three categories: centralized authorizing, distributed polling, and transitive computing. Examples of the three approaches are illustrated in Table 1.

Table 1. Three reputation sharing approaches

	Centralized authorizing	**Distributed polling**	**Transitive computing**
Examples	CORC, DCRC [6]	XRep [3]	EigenTrust [10]

In [8], authors present an efficient interest-based content location scheme, which gives us an idea on reputation discovering process. We propose a novel approach called selective polling to facilitate our reputation information sharing process in P2P networks. Selective polling is conceptually based on social networks. In self-organizing P2P communities, if a peer *Q* requests a file from another peer *M*, this implies that they have similar interests. It is reasonable to suppose that *Q* had downloaded files from other peers in the same community which have interacted with *M*. Therefore, from *M*'s point of view, a nature and efficient way to obtain *Q*'s global contribution score to decide a proper service level for *Q* is polling other members in his community.

Algorithm 1 demonstrates the recursive discovery algorithm of selective polling. There are two key parameters, illustrated in Figure 2, which should be decided first. There are two key parameters, illustrated in Figure 2, which need to be further decided here.

- *H* denotes how many hops should be taken to aggregate a satisfied global contribution score, which has similar function as TTL (time to live).
- *K* denotes how many peer should be selected in each iteration.

One should note that if *M* is a new peer and never shows up in the system before, then its acquaintance lookup repository is empty. Similarly, even though *M* interacted with other peers before, if all of his acquaintances are not online, *M* faces the same predicament as a new peer. EigenTrust introduces pre-trust peers to fix this problem. Our solution is to turn selective polling into distributed polling which is refered as "restricted flooding" in our selective polling algorithm. Distributed polling is to spread polling messages to all direct connected peers and, then, keeping messages flooding until TTL decreased to zero. In our case, the value of TTL for flooding in our case is set to a very small number (about 2) since it is used to be an auxiliary of the first hop of *H*. After collecting the responses, peer M randomly selects K contribution scores out of the replies for the use of reputation deduction. The selective polling brings a new way to discover a peer's global reputation. After gathering global reputation scores, we are able to discriminate service level.

```
Selective_Polling ($H'$, $K$) {
  Let $C$ represent # of lacked acquaintance;
  Let $TTL$ represent the time to live for flooding packets;
    if $H' == 0$ then
      if ever interacted with $Q$ before then
        $V_Q \leftarrow Q$'s contribution score in trans. repository;
        Return $V_Q$;
      else
        Ignore query;
    else if ($H' == H$) and ($C > 0$) then
       Select IP addresses of all peers from acquaintance lookup repository;
      if ever interacted with $Q$ before then
        $V_Q \leftarrow Q$'s contribution score in trans. repository;
        Return $V_Q$ + Selective_Polling($H'$-1, $K$-$C$) +
          Restricted_Flooding($C$, $TTL$);
      else
        Return Selective_Polling($H'$-1, $K$-$C$) +
        Restricted_Flooding($C$, $TTL$));
    else
      $K' \leftarrow (C > 0 \ ?\ K\text{-}C, K)$;
      Select IP addresses of $K'$ peers from acquaintance
        lookup repository;
      Send query to the $K'$ peers;
      if ever interacted with $Q$ before then
        $V_Q \leftarrow Q$'s contribution score in trans. repository;
        Return $V_Q$ + Selectvie_Polling($H'$-1, $K'$) ;
      else
        Return Selective_Polling($H'$-1, $K'$);
```

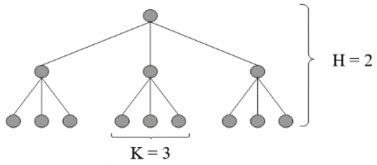


Figure 2. Selective polling with parameters H=2, K=3

Algorithm 1. Selective polling algorithm

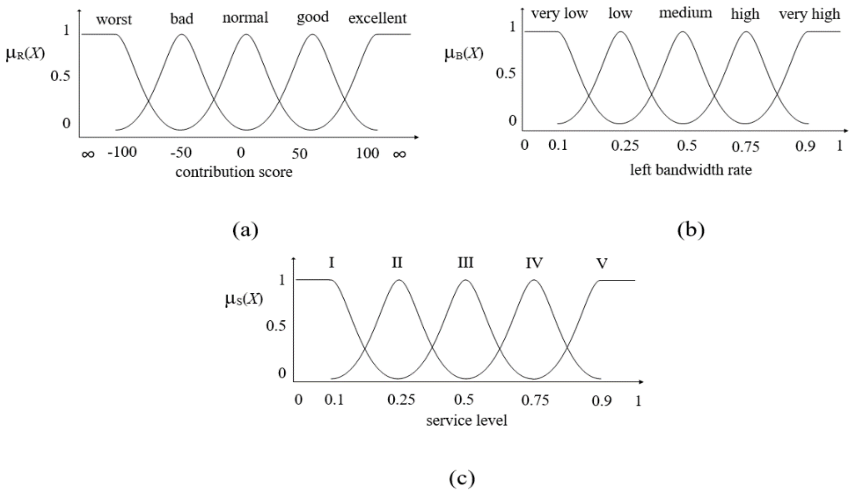


Figure 3. Membership function of (a) reputation, (b) bandwidth, and (c) service level

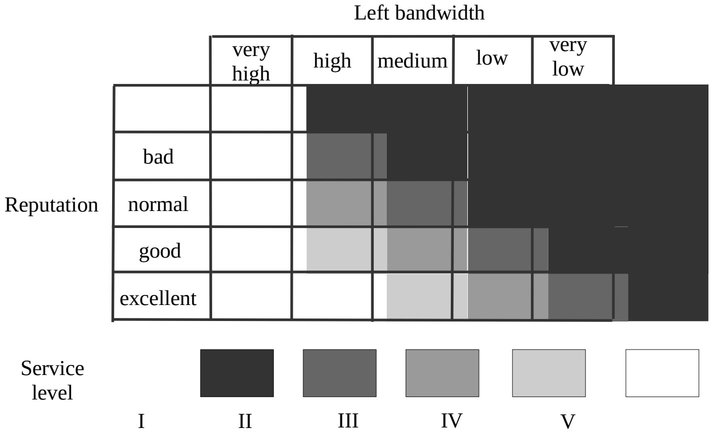


Figure 4. Reference matrix

4.3 Service Differentiation

The service differentiation is determined by a reference matrix comprised by two metrics including the deducted global reputation of file requester and the available inbound bandwidth of file provider. The goal of the first metric, deducted global reputation, is meant to encourage file sharing; the second metric, left bandwidth, is to prevent overloading and to alleviate the hotspot issue. Figure 4 shows the reference matrix. It is a two input, one output system. In fuzzy logic, membership function is used to shown the degree of membership of variables. We choose to use Gaussian membership function because its smoothness notation is capable to deal with nonlinear variables. Figure 3 shows the membership functions of reputation, bandwidth, and service level.

The output value of service level is between 0 and 1. Then the allowing sharing speed is the output value multiplied by the maximum connection speed that the provider is willing to offer. If a requester's reputation is worst or a provider is about to overload, the output value would be close to zero which will reject file transfer.

5. Experiments

In this section, we discuss the design, implementation and results of our experiments to disclosure the efficiency and effectiveness of FuzRep. They are conducted in two phases. Phase one takes a macro perspective to examine selective polling by simulating a 100 node P2P network which will be discussed in Section 5.1. Phase two, as described in Section 5.2, takes a micro perspective to evaluate service differentiation based on a designed scenario happened on an individual peer. FuzRep is determined successful if and only if it can collect reasonable global contribution scores of requesters by selective polling and can reflect the scores to corresponding service levels.

5.1 Experiment on Selective Polling

The simulation recorded every interaction, thus a list of true global contribution scores of each peer is available for the use of evaluation. After 300 rounds of bootstrapping interaction, one randomly choosed peer *S* started to discover the other 99 peers' contribution scores by selective polling. We separate the results of selective polling into two sets, *G1* and *G2*. *G1* is the set of peers who have at least one interest file category the same with *S*. Peers in *G1* are highly possible to interact with *S*. *G2* is the set of peers who do not have any interest overlapped with *S*. They are unlikely to interact with *S*.

Table 2. Simulation settings

Peer	# of peers	100
	% of free rider	85%
	# of direct connected neighbor	*D-U*(*1*,*10*)
	Online possibility	*U*(*0*,*1*)
	Possibility to send a query	*U*(*0*,*0.5*)
	# of interest file category	*D-U*(*1*,*3*)
File	# of file categories	5
	# of files in each category	50
	file popularity	Zipf's distribution [7, 12]
	file size distribution	Distribution in [12]
Selective polling	Parameter H	2
	Parameter K	10
	TTL	2
Simulation	# of cycles in a experiment	300
	# of experiments	5

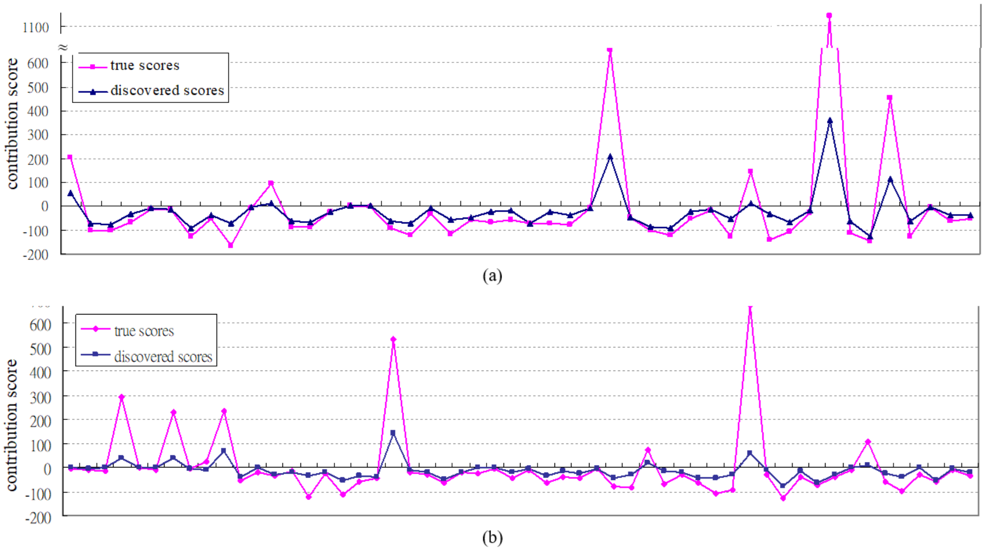
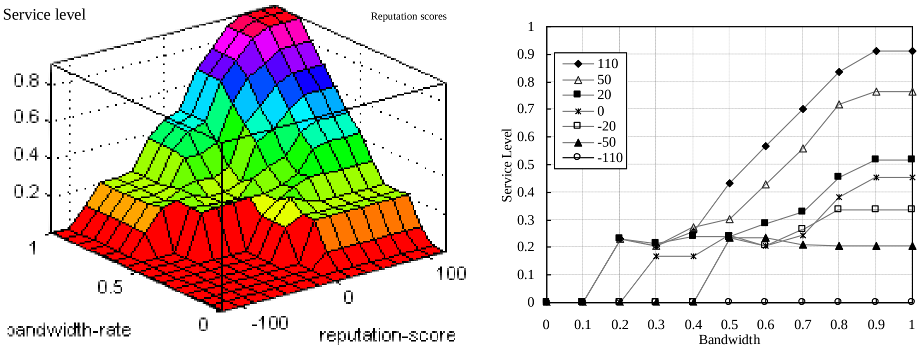


Figure 5. Selective polling evaluation: comparing the true scores with discovered scores. (a) shows the scores differences of peers who have interest overlapped with *S*; (b) shows those of no interest overlapped with *S*.

Simulation settings are shown in Table 2. We evaluate the selective polling scheme by comparing a peer's real global contribution score with the score gathered from the selective polling Figure 5 shows the result of our simulation. There are 46 scores in *G1* and 53 scores in *G2*. The distances between true scores and discovered scores are errors of selective polling. Since selective polling does not poll every peer in the network, existence of errors is unavoidable. However, as we observed from Figure 5, selective polling is capable of discovering a reasonable score to represent a peer's global contribution. If we exclude outliers, whose true scores are greater than 100 or less than -100, there are 72 scores left, and the mean error between the 72 true scores and the corresponding 72 discovered scores is 18.25. That would not cause significant difference on reputation determination under fuzzy-based service discrimination.



(a) Left bandwidth ratio

(b) Contribution score

Figure 6. Service differentiation; (a) surface of service level from Mtalab, (b) illustration of actual values

5.2 Experiment on Service Differentiation

MATLAB Fuzzy Logic Toolbox [16] provides a convenient way to design and simulate fuzzy logic systems. We make use of it to develop and examine the service differentiation model of FuzRep. Figure 6 shows how service level, reputation, and bandwidth interact each other.

Under service differentiation schemes like ours, it is still difficult to say whether a free-riding peer will change his behavior or not because of the complexity of human decision. Heterogeneous peers have different tolerances to transmission speeds. It is difficult and obviously absurd to define a threshold, say 5 Kbps, and assumes that free riders would definitely stop free riding if his perceived service level is lower than that threshold. However, as shown in Figure 6(b), the discriminated service level has been demonstrated as an incentive for peers to cooperate.

5.3 Discussion

As what Figure 5 shows, after excluding outliers, the mean error of scores of *G1*, the set of peers who have interest overlapped with *S*, is 18.21; meanwhile, the mean error of scores of *G2*, the set of peers who do not have any interest overlapped with *S*, is 18.39. It shows no significant difference on the effectiveness of selective polling when it is applied to different interest groups. The reasons, what we believe, may be due to the consulted peers, which have interests partially the same as *S*, were also possibly interested in categories which *S* is not interested The diversity of interest may increase as the *H* of selective polling growing up. It could relax concerns that selective polling was only capable to discover reasonable scores of peers of the same interests. The finding is especially useful as the number of peers or interest categories increased in a real P2P network.

Also, the characteristic of zero cost identities [4] in P2P networks might be a threat to FuzRep. It is true that a free-riding peers can easily change their identity which turns his negative contribution score into zero. Although recounting contribution score may make a difference in service level, that will always keep free riders swinging their service level from I to III. That is, sharing files is the only way for peers to raise their service levels.

Another issue is the adaptability of FuzRep. In our experiment, *H* is set to 2, and *K* is set to 10. However, these parameters for selective polling should not be fixed. Instead, *H* and *K* should be dynamically decided based on dynamic conditions of P2P environment such as total number of peers, churn rate of peers, frequency of interactions between peers, strength of interest based P2P communities, etc. Nevertheless, applying fuzzy logic to determine peers' reputation can compensate unavoidable errors of the selective polling scheme in FuzRep.

6. Conclusion

In this paper, we present a reputation system, FuzRep, in P2P networks to create incentives for file sharing and to avoid overloading for hotspot peers. The experiments and data presented above indicate that FuzRep is able to create incentives in P2P networks. It is a simple, cost efficient and effective mechanism. By well incorporated with selective polling and service differentiation, FuzRep brings a new idea to alleviate free-riding behaviors.

Our future work would aim at two directions. First, to provide higher adaptability, we still want to optimize settings of the reputation system under dynamic P2P network environments. Second, the reputation system should safeguard reputation values from cheating behaviors which are not addressed in the paper.

References

[1] E. Adar, and B. A. Huberman, “Free riding on Gnutella,” Proc. *First Monday* 5(10), 2000.

[2] C. Buragohain, D. Agrawal, and S. Suri, “A game theoretic framework for incentives in P2P systems,” Proc. of the *International Conference on Peer-to-Peer Computing*, 2003.

[3] E. Damiani, DCD. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, “A reputation-based approach for choosing reliable resources in peer-to-peer networks,” Proc. *of the ACM Conference on Computer and Communications Security*, 2002.

[4] E. Friedman, and P. Resnick, “The social cost of cheap pseudonyms,” Proc. *Journal of Economics and Management Strategy* 10(2), pp. 173–199, 1998.

[5] M. Feldman, K. Lai, I. Stoica, and J. Chuang, “Robust incentive techniques for peer-to-peer networks,” Proc. *of the ACM Conference on Electronic Commerce, 2004*.

[6] M. Gupta, P. Judge, and M. Ammar, “A Reputation System for Peer-to-Peer Networks,” Proc. of 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2003.

[7] D. Hughes, G. Coulson, and J. Walkerdine, “Free riding on Gnutella revisited: the bell tolls?” Proc. *IEEE Distributed Systems Online* 6(6), 2005.

[8] A. Iamnitchi, and I. Foster, “Interest-aware information dissemination in small-world communities,” Proc. *of the 14th IEEE International Symposium on High Performance Distributed Computing*, 2005.

[9] A. Iamnitchi, M. Ripeanu, and I. Foster, “Small-world file-sharing communities,” Proc. INFOCOM’04, 2004.

[10] S. D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, “The EigenTrust algorithm for reputation management in P2P networks,” Proc. *of the International World Wide Web Conference*, 2003.

[11] A. Odlyzko, “The case against micropayments,” In *Financial Cryptography 2003, J. Camp and R. Wright, eds, Lecture Notes in Computer Science, Springer*, 2003.

[12] K. Ooi, S. Kamei, and T. Mori, “Survey of the state of P2P file sharing applications,” *NTT Technical Review*, 2004

[13] K. Sripanidkulchai, B. Maggs, H. Zhang, “Efficient content location using interest-based locality in peer to peer systems,” Proc. of the *INFOCOM’03*, 2003.

[14] Szabo, N, “Micropayments and mental transaction costs,” Proc. of the 2nd Berlin Internet economics Workshop, 1999.

[15] The Gnutella protocol specification v0.4, 2001. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[16] The Mathworks, inc. MATLAB 6.5, 2001.

[17] V. Vishnumurthy, S. Chandrakumar, and EG. Sirer, “KARMA: A secure economic framework for P2P resource sharing,” Proc. *of the Workshop on Economics of Peer-to-Peer Systems*, 2003.