The Tell-Tale Neighbors: Trust and Reputation in Peer-to-Peer Systems *

Thomas Repantis
Department of Computer Science & Engineering
University of California, Riverside
trep@cs.ucr.edu

March 19, 2004

Abstract

Peer-to-peer systems are an attractive means of sharing data and services. However, the problem of how to efficiently decide which peers are to be trusted still remains unsolved. In this work we propose a decentralized trust management system based on reputation, for unstructured, self-organizing peer-to-peer networks. Our protocol takes advantage of the unstructured nature of the network to render malicious behavior like lying and colluding risky. The reputation information of each peer is stored in its neighbors and piggy-backed on its query-hits. By simulating the behavior of networks both using and not using a rating scheme we were able to show that just a few dishonest nodes can flood the network with false results, whereas this phenomenon is virtually eliminated when using a rating scheme.

1 Introduction

Peer-to-peer (P2P) systems have attracted a lot of interest, as a highly dynamic platform that enables autonomous computing nodes to share resources

*Course Project Report for CS255 - Computer Security, Spring 2004.

and services. The advantages of peer-to-peer environments, especially of unstructured ones, include their ability for self-organization, for adaptation to different loads, and for resiliency to node failures. Operation of all the peers as both clients and servers, and without a central coordinator eliminates possible bottlenecks in terms of scalability or reliability.

However, in an unstructured and decentralized topology several security issues arise. One of the most challenging problems, that is still being actively researched, is *how to create a trusted network of peers in the absence of a central trust managing authority.* Trust is important when sharing data or processing power, and crucial for e-commerce applications and auctioning.

By saying that peer A puts a level of trust into peer B, we mean that A estimates the probability of B acting in a way that will allow A to achieve a desired level of satisfaction.

One way a peer A can estimate the level of trust to put into another peer B, is by being based on the reputation of peer B. The reputation of peer B is measured from previous interactions of peer A with peer B, or also on previous interactions of other peers with peer B. As the level of trust a peer enjoys is based on its reputation, a peer is motivated to act according to the rules of the network, whether these are to share content, not to cheat, or anything else. Using the peers' opinion to establish a reputation is a process already very popular in the scientific community for example through peer review or citations.

One of the main difficulties in managing reputation-based trust in P2P networks is that information about peer interactions is spread across the network, and no single peer has a complete global view of the peers' reputations. Furthermore, malicious peers might tamper with reputation information while it is stored locally or transmitted, or even try to defame other peers.

In this work we propose a decentralized trust management system based on reputation, for unstructured, self-organizing peer-to-peer networks. Its novelty lies in the fact that it relies in the lack of network structure to provide a relatively safe trust management environment.

We discuss the background of our approach in section 2, and our protocol idea in section 3. The architecture of our proposed system is described in detail in section 4, whereas section 5 provides simulation results. Section 6

presents related work and section 7 concludes the paper and summarizes our contribution.

2 Background

We would set the following as major requirements for a trust-measuring system:

- To enable the peers to identify trustworthy and untrustworthy peers.
- To be simple enough, so that the protocol overhead is not hindering the interaction of peers.
- To make collusions impossible or very difficult.
- To make malicious actions (attempts to tamper with reputation information) identifiable and ideally impossible.

Storing the reputation information in a distributed manner and conducting polling to gather it, as proposed in [7] generates a large amount of network traffic and delay. Storing the reputation information in the peer this information refers to requires complicated operations to ensure that this peer will not tamper with his reputation [20]. On the other hand, storing the reputation information in just one peer is also risky, since that peer is controlling another peer's fate and may even try blackmailing or colluding with him/her. Anonymous storing of reputation information [21] is complicated and requires broadcasting, which is unacceptable for an unstructured peer-to-peer system. Storing the same reputation information in a group of peers, like [16] proposes for structured peer-to-peer networks, seems a reasonable approach, since it will allow the comparison and verification of the reputation information received by all or some of the peers of that group.

3 The Tell-Tale Neighbors

According to our opinion, a better still idea would be to *store reputation information in a group of peers that is not easily identifiable*, so that collusions

and blackmailing become cumbersome. That is the essence of our approach, which is targeting self-organizing, unstructured peer-to-peer systems. We propose that each peer's reputation information is stored in all its neighbors. In a simple approach that information is stored just in the immediate neighbors, but higher reliability can be achieved by also storing it in neighbors more than one hop away, paying the higher communication and processing cost.

The idea to store the reputation information of a peer in its neighbors provides several advantages. Specifically it guards the system:

Against lies: A neighbor sending back bogus reputation information might be revealed, since he/she may not be the only one answering and then the discrepancy will be noted.

Against blackmailing: Peers store their neighbors' reputation information and their neighbors store theirs'. This balance of power makes blackmailing infeasible.

Against collusions: Collusions on the other hand are difficult to achieve, because *the topology of the network is never known, and it also changes dynamically.* To change one's reputation, all its neighbors must cooperate, otherwise the deceit might be revealed. When the reputation information is stored in more than just the immediate peers, collusion becomes even more complicated to achieve and it would require a lot of message exchange.

The lack of structure and the dynamic nature of the network are usually regarded as major hindrances in managing trust information in unstructured peer-to-peer systems. Our approach is novel, in that it utilizes exactly those characteristics to create an environment that makes tampering with reputation information cumbersome and risky.

4 Architecture

4.1 System Components

Figure 1 presents the architecture of the proposed system. Each peer is comprised from several components:

Connection Manager Responsible for managing the connections to other peers.

Content Manager Responsible for managing the data objects stored in the peer. If services instead of data were provided, those would be managed instead.

Message Handler Responsible for handling all incoming messages. Queries are checked for local matches and propagated further. Query-Hits are created if local matches are found, or propagated to the direction the Queries were received. If a Query-Hit was originated in one of the direct neighbors of the peer, the Reputation Information (the collection of all the ratings) for that neighbor is piggy-backed to the Query-Hit. If a Rating for a neighbor is received, the rate is added to the reputation information for that neighbor.

Reputation Information Manager Responsible for managing (storing and providing) the Reputation Information of the neighbors of the peer, as well as for managing (collecting and storing) the Reputation Information of the peers this peer has interacted with.

Rater Responsible for assigning a rating to a peer that has interacted with this peer, and for propagating this rating to the other peer's neighbours

Rating Verifier Responsible for determining that the Reputation Information (collection of ratings) of a peer, that is contained in several Query-Hits, is the same.

Peer Selector Responsible for selecting the most appropriate among the peers that offer an object, according to their Reputation Information.

4.2 Queries and Query-Hits

We assume a logical network of peers that provide resources or services to each other. Each peer maintains connections with other peers. The network is decentralized and self-organizing, meaning that peers make their own decisions on which peers to connect to or to query for resources or services. Peers search for resources/services by sending query messages to their immediate

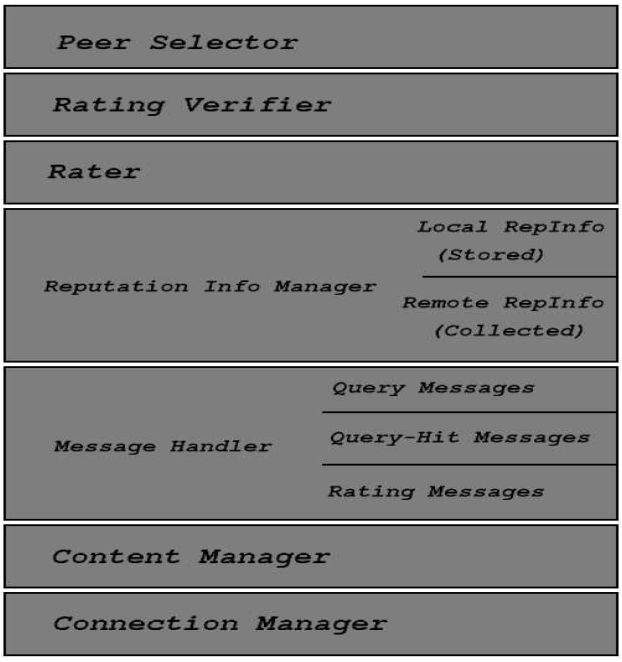

Figure 1: System architecture.

neighbors. Those queries are evaluated locally in each peer and in case there are matching resources/services, results are returned to the searching peer. The queries are propagated further, until their TTL expires. The query-hits follow the same path as the queries to reach the searching peer. The neighbor of a peer that answers with a query-hit is responsible for adding the reputation information of the answering peer to the query-hit message, and he/she is possibly not the only one with that duty, depending on the topology of the network.

Figure 2 shows an example of a query and query-hit exchange. Let us assume that A creates a query with $TTL = 3$ that is propagated and eventually reaches F, who –having the resource/service– creates a query-hit. That reply is following the same path as the query to reach A. The neighbors of F, namely C and D add the reputation information to the query-hit, before propagating it further. In this topology two query-hits will be generated, so that peer B will be able to verify that the reputation information on both of them is the same. This redundancy is newly introduced, since normally F would have just replied once. Since F's neighbors do not know if they

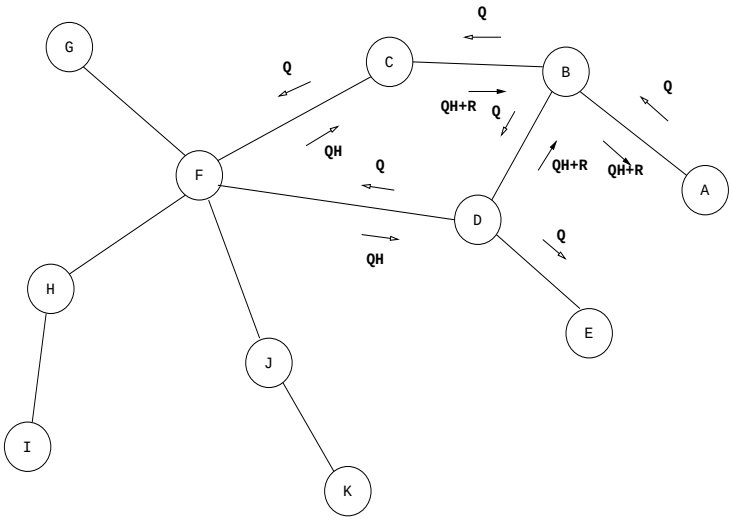

Figure 2: Query and query-hit example.

will be the only ones propagating the current query-hit, they may not risk tampering with the reputation information.

4.3 Ratings

After an interaction, a peer may rate the resource/service he was provided. The rating message is propagated using the same flooding-based mechanism as the query message. However, the TTL of the rating message is bigger than the TTL of the query message, so that the rating can reach all the neighbors of the peer that is being rated. For example, to reach just the immediate neighbors, the TTL of the rating message would be bigger than the TTL of the query message by one.

Figure 3 shows an example of rating. After A uses the resources/services provided by F, he/she creates a rating message, with a $TTL = 3+1 = 4$, that is propagated and reaches all of F's immediate neighbors (C, D, G, H, J), who update the reputation information they store for F. The rating message also reaches peers like B and E, that do not need to store rating information for F, since they will not be asked to provide it.

An important observation is that F is asked to propagate his own rating. To avoid tampering, ratings might be digitally signed, or transferred in a

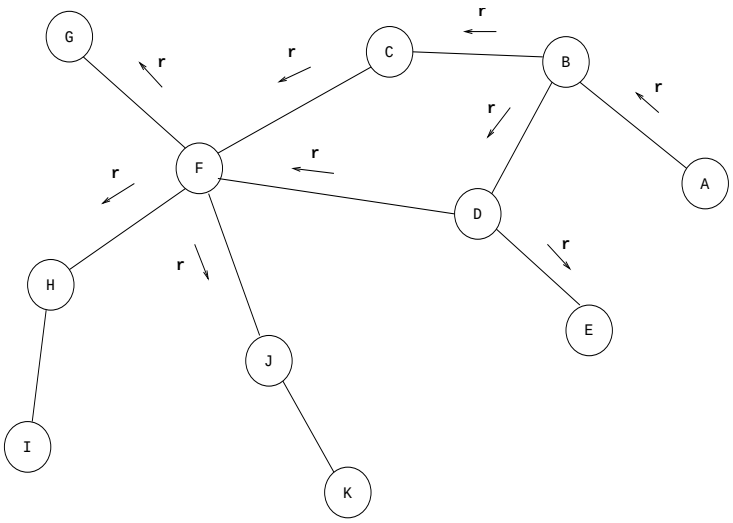

Figure 3: Rating example.

way that makes it impossible for peers to discern to whom the ratings refer, or maybe even the fact that the messages contain ratings. Only the final recipients should be able to extract this information. But even if F identifies the rating messages and changes or discards them, the ratings will still be stored intact in some of its immediate neighbors, namely C and D, and therefore the discrepancy might be identified.

4.4 System Algorithms

The formulas for choosing and for rating a peer are of particular interest:

- The formula for choosing the peer with the best reputation might weigh the ratings, according to the personal opinion of the peer for the raters, or according to the raters reputation as peers or even as raters.
- The formula for rating might use a scale that allows comparison with the current rating average. In that way, ratings far away from the average might be noted, and the responsibility of the rater might also be rated. Moreover, both the data/service provider and consumer may rate each other.

Moreover special care should be taken for nodes entering the system, so that they can receive the reputation information of the nodes they connect to. As for their own reputation, it will be built as they engage in transactions. By not giving any initial reputation to newcomers, we discourage peers with bad reputation to leave the system and reenter under a new identity, since building a reputation is a tedious process.

5 Experimental Evaluation

5.1 Simulation Infrastructure

In order to evaluate the efficiency of a protocol like ours, that enables peers to identify and isolate the untrustworthy among them, we conducted a set of experiments. In order to be able to evaluate networks of thousands of peers, we implemented our system on top of the Gnutella [10] unstructured peer-to-peer network, using the NeuroGrid simulator [14]. Neurogrid is scalable, since it simulates the protocols at message- and not at packet-level.

Apart from the honest peers, that provide the data objects they claim they have, we included a number of dishonest peers in the network. These malicious peers claim that they have every object they are asked for, in other words reply with a Query-Hit to every Query they receive, without of course being able to provide the real requested data object. We observed the effect of that behavior on the operation of the network, with and without using a rating scheme. When the rating scheme is used, we assume that the malicious peers can only cheat once, since then they are discovered and receive a bad rating that discourages other peers to interact with them.

5.2 Variable percentage of dishonest peers

For the first experiment, we kept the total number of nodes to 1000, and we varied the number of honest nodes in the network. The simulation details are presented on table 1. Our goal was to determine to what extent the percentage of dishonest nodes affects the operation of the network.

Node Parameters	Number of nodes Number of honest nodes	1000 Varying
Content Parameters	Size of pool of available objects Number of objects per node Distribution of objects over nodes	3000 30 Uniform
Network Parameters	TimeToLive of messages Number of connections per node Network topology Searches Forwarding	7 3 Random Random Random
Simulation Parameter	Number of averaged measurements	5

Table 1: Simulation settings for the varying number of honest nodes experiment.

Figure 4 shows the average number of untrue Query-Hits. Without utilizing the rating scheme, this is quite high, even for relatively small percentages of dishonest nodes. By using the rating scheme the number of false matches is virtually eliminated, even for networks with many malicious peers.

Figure 5 shows the average proportion of Query-Hits that were true. When using the rating scheme, that proportion remains very high. What surprised us are the results for when not using any rating scheme. The precision (the proportion of true Query-Hits) remains practically close to zero, even when 80% of the nodes are honest. *If 1 out of 10 nodes is dishonest, 9 out of 10 Query-Hits are bogus. This means that a few dishonest nodes have the ability to flood the network with false matches, representing a real threat to its operation.*

5.3 Variable number of peers

It was interesting to see if dishonest behavior is equally threatening in larger-scale networks. Therefore in the second experiment, we kept the percentage

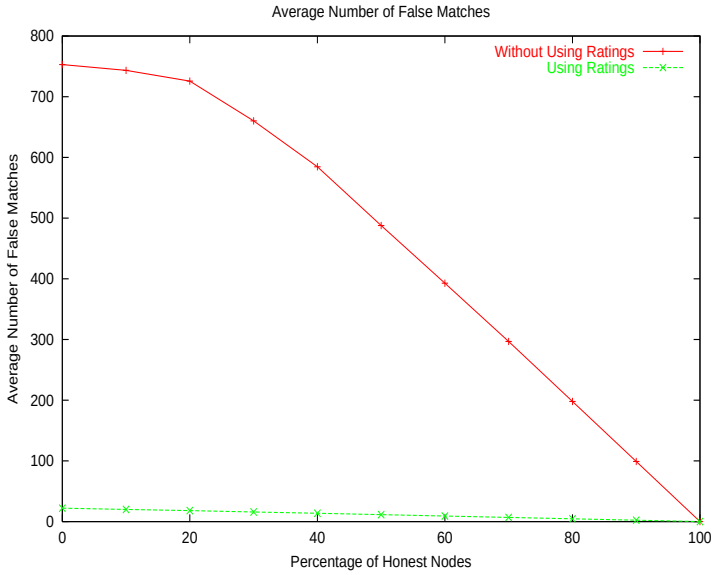

Figure 4: Average number of untrue (falsely reported) matches to a search, for a variable number of honest nodes.

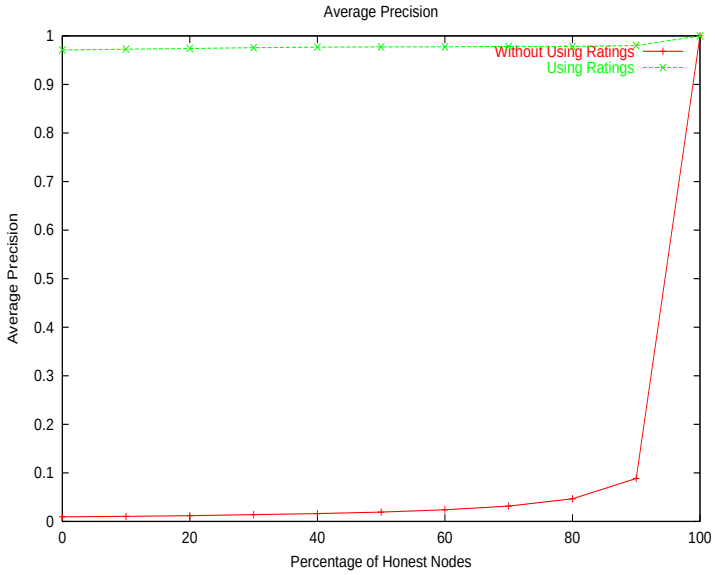

Figure 5: The average proportion of the returned results that were not false matches, for a variable number of honest nodes.

Node Parameters	Number of nodes Number of honest nodes	Varying 75%
Content Parameters	Size of pool of available objects Number of objects per node Distribution of objects over nodes	3000 30 Uniform
Network Parameters	TimeToLive of messages Number of connections per node Network topology Searches Forwarding	7 3 Random Random Random
Simulation Parameter	Number of averaged measurements	5

Table 2: Simulation settings for the varying number of nodes experiment.

of honest nodes to 75%, and we varied the total number of nodes in the network. The simulation details are presented on table 2.

Figure 6 shows the average number of untrue Query-Hits. Without using the rating scheme, the number of false matches grows very fast, for large networks. Again, the rating scheme prevents that behavior.

Figure 7 shows the average proportion of honest Query-Hits. Again by using the rating scheme this proportion remains high, even for large networks. However, without a rating scheme, the dishonest nodes present a threat even to large-scale networks. Even though 3 out of 4 nodes are honest, the percentage of honest Query-Hits remains close to zero. *We observe that the dishonest nodes are able to flood even large networks.*

6 Related Work

Several P2P reputation systems have already been proposed, including EigenTrust [16] [15], RMS [11], P2PRep [8] [7] [9], RCertPX [20], TrustMe [21], Poblano [6], OpenPrivacy [5], a scheme for trust inference in NICE [18], and

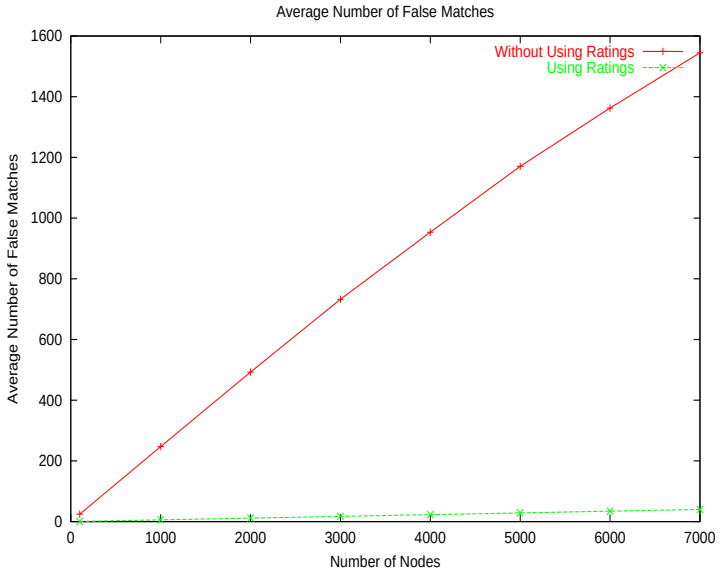

Figure 6: Average number of untrue (falsely reported) matches to a search, for a variable number of nodes.

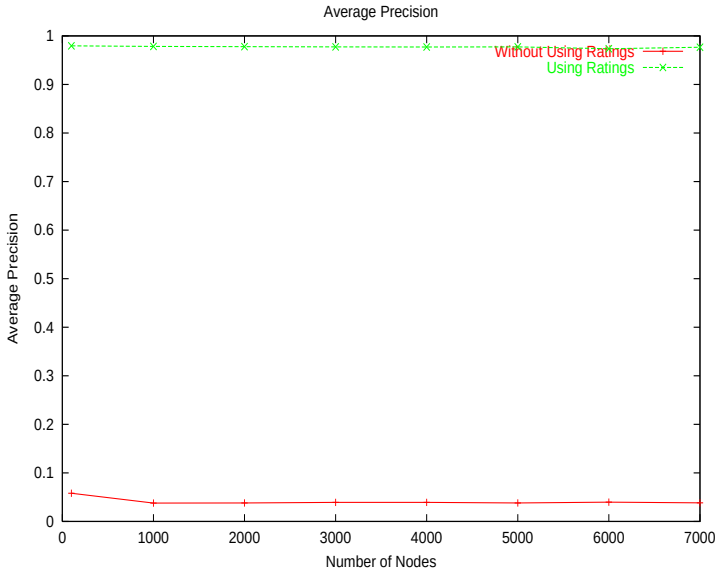

Figure 7: The average proportion of the returned results that were not false matches, for a variable number of nodes.

a trust management system on top of P-Grid [2].

In RCertPX [20] a reputation certificate is created and is stored in the peer that it refers to. After each transaction the reputation certificate is updated. In order to eliminate the possibility of tampering with the reputation certificate, the last rater always digitally signs the whole certificate. When a peer wants to interact with another peer, it requests its reputation certificate, and it contacts the last rater, in order to verify the correctness of the certificate. If the last rater is unavailable, the one before it is contacted. Since every rater revokes the previous reputation certificate, a peer can only use the latest reputation certificate, that includes all the ratings. When a peer wants to decide with which peer it should do a transaction, it contacts all peers that offer the service/data and they provide it with their reputation certificates. The peer then contacts the latest rater to determine if the reputation certificate is still valid. The network traffic produced is relatively low, as well as the decision delay. Furthermore, even if some peers have left, their ratings are still available. Also important is the fact that the ratings cannot be changed once they are determined. In that way, a peer will not change its rating of another peer if it becomes a competitor. However, the protocol is complicated, especially when the last rater is offline, which is often the case in P2P systems. It also uses ratings even from untrusted peers, which might be even more critical in the case a rater and a ratee collude to change the ratings. A peer can collude with another peer so that previous certificates will not be revoked and therefore bad ratings may be lost.

In P2PRep [8] [7] [9] a polling based protocol is proposed and implemented (on top of Gnutella), that uses public key cryptography for authentication. Any peer A that wants to query the trust value of another peer B broadcasts a query to the network. The peers that have had transactions with B reply with their IP and port in a message, encrypted with the public key of A. A then individually contacts the voters and asks them to confirm their votes, so that fake messages are filtered out, and combines the valid votes to make a decision. Apart from the network traffic generated and the delay of the process, this approach counts only the reviews of those present peers that can be reached.

TrustMe [21] identifies anonymity as an important feature of trust-managing systems. The trust rating of each peer is placed at another random peer, which replies to all queries for the trust values it holds. A peer can anony-

mously issue a query and get the true value without needing to know where that value is stored. Public-private key pairs are used to preserve anonymity. One drawback of this protocol is that it relies on broadcasting, making it unacceptable for large-scale, unstructured networks.

EigenTrust [16], [15], a global variable regarding a peer's reputation is stored in a peer's mother peers. The global variable is generated by aggregating local variables in all peers, in an iterative process. The algorithm does not prevent mother peers from blackmailing a peer, nor from colluding against a peer.

In NICE [18] cooperating peers form a graph, that later helps one of them to identify others that store reputation information relevant to him. Specifically, a peer providing a service is responsible to prove his reliability to a peer that would like to use that service, by finding a path in the graph to that peer. A path identifies previous interaction with common acquaintances. However during this discovery process flooding is used and many irrelevant peers may be contacted. Moreover since the peer providing a service is gathering his reputation information and sends it to a peer to use it, he may omit bad ratings.

In [2] a trust managing system on top of the P2P system P-Grid is described. The reputation of a peer in this system is expressed as the number of complaints he has (not) received. The complaints of each peer are stored in a virtual binary search tree. Replication in storage satisfies the integrity of the stored complaints in a probabilistic manner.

In OpenPrivacy [5] a web of trust is formed by identities and evaluation certificates. Certificates are digitally signed to ensure integrity and are stored at their creator and at the peer they evaluate.

In RMS [11] another distributed reputation management system is described. Reputation information is kept by its owner, and public key cryptography is used to ensure its integrity. A trusted third party (one or multiple servers) signs the reputation certificates and records the transaction history of the peers, providing that information to peers that want to check the correctness of a certificate.

Poblano [6] is the decentralized trust model on the Project JXTA platform. Some of the peers communicate using CA signed certificates, while other use self-signed certificates. The members of a peergroup assign a level

of trust to one another. Three main components form a peer's trust level, namely codat (data) confidence, peer confidence, and risk.

In [23] a Bayesian network-based trust model is proposed, together with a method for building reputation based on recommendations of other peers. Bayesian networks help in representing and combining trust in different specific areas of a peer's capability.

7 Conclusions and our contribution

We have proposed a decentralized trust management system based on reputation, for unstructured, self-organizing peer-to-peer networks. We have shown that –when using our protocol– random topologies that may be created make malicious behavior like lying and colluding risky. Moreover, all peers are equally powerful, controlling the fates of their neighbors, while their fates are controlled by their neighbors. Two main characteristics of peer-to-peer systems, namely the absence of a governing authority and the unstructured, and dynamic nature of the network are usually regarded as obstacles in ensuring trust. With our protocol, we are using exactly those two characteristics for achieving that goal.

We have tried to keep our protocol simple and easy to build on top of the existing infrastructure available for the exchange of messages, to minimize its overhead. An advantage of the protocol is the fact that the ratings of peers that have left the system are still present. Moreover the communication overhead of polling-based protocols is avoided and the only extra messages introduced are those carrying a new rating. Our future work includes elaborating on the peer choosing and rating algorithms.

By simulating the behavior of networks both using and not using a rating scheme we were able to make some interesting observations: First of all, the rating scheme practically eliminates the effect of the dishonest peers on the network. Moreover, without a rating scheme, we saw that just a few dishonest nodes are able to flood the whole network with false results, even for large networks, and thus pose a real threat to its operation. The need for a trust management system for peer-to-peer networks becomes therefore even clearer.

A Appendix

neighbor, n:
One whom we are commanded to love as ourselves, and who does all he knows how to make us disobedient.
– Ambrose Bierce, "The Devil's Dictionary"

References

[1] Alfarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *Proceedings of the New Security Paradigms Workshop*, 1997.

[2] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the International Conference on Information and Knowledge Management, CIKM*, 2001.

[3] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In *Proceedings of the Symposium on Operating Systems Design and Implementation, OSDI*, 2002.

[4] D. S. Bernstein, Z. Feng, B. N. Levine, and S. Zilberstein. Adaptive peer selection. In *Proceedings of the International Workshop on Peer-to-Peer Systems, IPTPS*, 2003.

[5] Kevin A. Burton. Design of the openprivacy distributed reputation system.

[6] R. Chen and W. Yeager. Poblano: A distributed trust model for peer-to-peer networks.

[7] F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a p2p network. In *Proceedings of the International World Wide Web Conference, WWW*, 2002.

[8] F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Implementing a reputation-aware gnutella servent. In *Proceedings of the International Workshop on Peer-to-Peer Computing, WP2P*, 2002.

[9] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks, 2002.

[10] Gnutella Protocol Development. http://rfc-gnutella.sourceforge.net/, 2003.

[11] D. Fahrenholtz and W. Lamersdorf. Transactional security for a distributed reputation management system. In *Proceedings of the International Conference on Electronic Commerce and Web Technologies, Ec-Web*, 2002.

[12] K. Fujimura and T. Nishihara. Reputation rating system based on past behavior of evaluators. In *Proceedings of Conference on Electronic Commerce, EC*, 2003.

[13] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2003.

[14] S. Joseph. An extendible open source P2P simulator. *P2P Journal*, pages 1–15, November 2003.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. Eigenrep: Reputation management in p2p networks, 2003.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the International World Wide Web Conference, WWW*, 2003.

[17] S. Ketchpel and H. Garcia-Molina. Making trust explicit in distributed commerce transactions. In *Proceedings of the International Conference on Distributed Computing Systems, ICDCS*, 1996.

[18] S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in NICE. In *Proceedings of Infocom*, 2003.

[19] S. Marti and H. Garcia-Molina. Identity crisis: Anonymity vs. reputation in p2p systems. In *Proceedings of the International Conference on Peer-to-Peer Computing, P2P*, 2003.

[20] B. C. Ooi, C. Y. Liau, and K. L. Tau. Managing trust in peer-to-peer systems using reputation-based techniques. In *Proceedings of the International Conference on Web Age Information Management, WAIM*, 2003.

[21] A. Singh and L. Liu. Trustme: Anonymous management of trust relationships in decentralized p2p systems. In *Proceedings of the International Conference on Peer-to-Peer Computing, P2P*, 2003.

[22] Dan S. Wallach. A survey of peer-to-peer security issues. In *Proceedings of the International Symposium on Software Security, ISSS*, 2002.

[23] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proceedings of the International Conference on Peer-to-Peer Computing, P2P*, 2003.

[24] L. Xiong and L. Liu. Building trust in decentralized peer-to-peer electronic communities. In *Proceedings of the International Conference on Electronic Commerce Research, ICECR*, 2002.

[25] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *Proceedings of Conference on Electronic Commerce, EC*, 2003.

[26] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18:535–549, 2002.