The MoR-Trust distributed trust management system: Design and simulation results

Stephanos Androutsellis-Theotokis[1] Diomidis Spinellis[2]
Vasileios Vlachos[3]

Department of Management Science and Technology
Athens University of Economics and Business
Greece

Abstract

MoR-Trust is a purely decentralized peer-to-peer trust management system, targeted towards networks and applications supporting transactions or collaborations of a quantitative nature. MoR-Trust is based on the notion of expressing trust in terms of monetary units, thus directly coupling the trust estimates circulated in the network with the values of the transactions taking place and their outcomes. We have validated our design decisions and algorithms through simulation. The results indicate that our system converges towards a small error in the trust estimates distributed throughout the network.

Keywords: peer-to-peer networks, trust management, reputation management.

1 Introduction

It is being progressively recognised that information systems and applications supporting collaborative tasks and/or transactions, that are traditionally designed based on centralized or client-server models, can also be based on the new, maturing wave of "peer-to-peer" architectures (the motivation for this is discussed in [6,5]). In this new domain, trust plays an even more important role as a foundation for effective collaboration and fair transactions. However studies of the behavior patterns in these on-line communities reveal a high degree of selfish and uncooperative behavior, and make apparent the need for incentive mechanisms to be applied to stimulate cooperation and fairness.

A variety of incentive mechanisms have been proposed, with reputation or trust-based mechanisms being identified as the most appropriate choice [4].

[1] Email: stheotok@aueb.gr

[2] Email: dds@aueb.gr

[3] Email: vbill@aueb.gr

This paper is electronically published in
Electronic Notes in Theoretical Computer Science

1.1 Distributed trust management

Trust is critical for any society to exist [1], as it influences many everyday interactions; We ask people that we trust for information, and we collaborate with people that we trust. According to [1], trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, before such action can be monitored. Based on this definition of trust, an entity's *reputation* is some notion of its propensity to fulfil the trust placed in it; an expectation about an entity's behavior based on information about or observations of its past behavior. Reputation is thus created through feedback from individuals who have previously interacted with the entity.

The main goal of a trust management system, such as MoR-Trust, is to maintain and distribute trust information about the parties (peers) engaged in collaboration or transaction processes. This information is used to provide a reputation measure, i.e. an expectation about another node's behaviour in a transaction.

Online trust management systems can thus be described as large-scale "online word-of-mouth communities" in which individuals share opinions about other individuals. Maintaining a high trust rating can be used as an incentive to reduce the degree of selfish or lavish behavior of peers, that is often observed.

Centralized trust or reputation management systems [21,14,13], such as the one behind the popular eBay site, are successful to a large extent because people trust the reputation information presented by them. In a completely decentralized environment, such as a peer-to-peer network, however, there is no single, recognizable organization or entity to maintain and distribute trust. As a result, trust information must be itself securely distributed throughout the network, and hosted on many different nodes. Distributed trust management systems offer mechanisms for achieving this, by extending the architecture and functionality of the transacting nodes.

1.2 Motivation and aims

MoR-Trust (stands for *monetary-ratcheted trust*) is is based on purely decentralized peer-to-peer architectures and algorithms, and is targeted towards systems focusing on collaborative tasks or transactions, and is based on the notion of modelling and expressing trust in terms of a quantitative *monetary* units, thus coupling trust estimates with transaction values. This is the main characteristic of MoR-Trust as compared to other systems (see Section 2). This approach allows the design of algorithms for the estimation, usage and propagation of trust estimates throughout the network, reflecting the way in which trust and reputation are utilized in real life communities. Central to our approach is the notion of *ratcheting* trust estimates, i.e. allowing the build-up of trust as a result or repeated successful transactions, potentially beyond the actual transaction value.

2 Related Work

We concisely present below the main solutions that have been proposed for distributed trust and reputation management. They address either of both of the following two problems (see also [3]):

(i) The data modelling (or semantic) problem: How to generate, interpret and process the trust / reputation data; and

(ii) The data management (or system) problem: How to store, retrieve, distribute and secure the trust / reputation data in a scalable and efficient manner.

The EigenTrust system [18] performs a distributed computation of a single network-wide reputation value for each peer, based the outcomes of past interactions. Local trust values that result from interactions with other peers are aggregated in such a way that the global trust values correspond to an eigenvector of a matrix containing local (normalized) trust values. Security is provided by the local trust values not being kept by the interested peers, but by other peers that are selected in the network based on the properties of a structured underlying routing mechanism. An extension of this approach to provide protection against collusion is presented in [30].

The PeerTrust system [29] focuses heavily on the data modelling aspect, with less emphasis on the data management side. A complex model for describing, interpreting and combining a variety of different trust metrics, based on feedback, transaction frequency, credibility, and different context factors is described. At the data management level, each peer stores a small portion of the global trust data, while *trust manager peers* are assigned to monitor and evaluate the trustworthiness of other peers. A structured routing infrastructure provides the necessary means of organizing the peers and their trust information distribution. The problem of misbehaving peers is addressed by means of majority voting, data replication and encryption.

The Credence system [28] is based on the notion of the reputation of *data objects*, instead of nodes. It allows nodes to contribute evaluations of data objects, and it also supports a network wide statistical correlation scheme between nodes, based on whether the votes of nodes for the same objects generally agree or disagree, thus forming a correlation matrix.

In [8], a distributed reputation system is designed based on a Bayesian approach. Nodes maintain first-hand reputation information which they regularly publish to other peers. More global reputation values are thus built by the peers receiving the local reputation scores. This work describes the mathematical model, however does not elaborate particularly on the data management approach or the security considerations.

The work in [3] is based on the P-Grid structured peer-to-peer routing algorithm [2]. It adequately addresses both the semantic level (trust model) and the data management level. Essentially the data management is taken care of by allowing a set of agent peers to monitor and asses the behaviour of other peers, through the use of P-grid and the associations it creates between groups of peers. Each transaction is monitored and the result registered. At the data modelling level, the trust reported by a monitoring peer is weighted by the trust placed on the monitoring peer itself. Trust and mistrust are represented in a rather limiting, binary fashion.

One of the main characteristics of the TrustMe system [22] is that it provides anonymity. It uses a random assignment of *Trust-holding Agent Peers* and uses public key mechanisms to prevent any loss of anonymity. The assignment is carried out by a bootstrap server (note: an element of centralization). This work focuses on the data management aspect of the problem, and does not elaborate much on the trust model. Essentially any peer interacting with another one can file a report with its Trust-holding agent regarding the interaction. The trust value for any peer is a result of the cumulative value of all these reports.

The XRep protocol [11,10] essentially utilises a simple collection of votes based on random polling of other peers that may or may not maintain local reputation information for specific peers.

In [16] a partially centralized mechanism using *reputation computation agents* and data encryption is described, in which the reputation values are calculated, encrypted and stored locally using a reputation computation agent. They propose two different schemes for calculating reputation values: a credit / debit scheme and a credit only scheme.

The Poblano system [9] is based on each peer maintaining a table with a confidence value placed on other peers. The confidence value that peer A places for peer B results from forming one or more paths from A to B, by following the table entries of A and other peers between A and B. The results are combined according to formulas describing the data model, and, in case of multiple paths, weighted averages of the results of each path are used.

The authors of [20] present two alternative designs, one focusing on storage and the other on network bandwidth as the resource of interest. The first requires the existence of a public key infrastructure and strong node identities, and is based on digitally signed usage records and a series of auditing procedures. The latter does not require such infrastructure, and uses the number of objects sent or received by each node to produce figures for the debt or credit of the node, and the confidence that other nodes can have on this peer.

Finally an architectural approach to decentralized trust management, including a brief threat analysis can be found in [25].

3 MoR-Trust System Design

Although our design is not limited within the domain of business or commercial transactions, for the purposes of this work we adopt (and slightly adapt) the concepts of *Collaboration* and *Transaction*, as defined by the ebXML Business Process Specification Schema [26], to define the scope of our system:

Transaction: An atomic unit of work that can only involve two parties and can result in either a success or a failure.

Collaboration: A combination of choreographed Transactions that can involve any number of parties and defines the ordering and transition between them.

This definition is wide and generic enough to encompass a broad range of applications, as individual transactions constitute the building blocks of more complex collaborations. In the following sections we will be describing a trust management system that focuses on the *transaction level*; any choreography, workflow or more complex collaboration level built above it is considered transparent.

3.1 Trust model

Central to the system's design is the representation of trust information. Mor-Trust is designed to support transactions of a clear quantitative nature. Without any loss of generality, the reader can imagine our system as supporting financial transactions, in which specific amounts of money are exchanged for goods or services.

Within this context we chose to use an orthogonal and practical way of expressing trust information; namely in terms of the monetary value used for the transactions themselves, as a continuous scalar value. In other words, if node n_a estimates a trust value V for node n_b, it means that node n_a should generally trust node n_b only for transactions whose value does not exceed V monetary units.

Based on this coupling between transaction value and trust estimate, we devise a series of practical algorithms, described in the following sections, for combining and estimating trust measures, deciding on whether to proceed with transactions or not, updating trust estimates based on transaction values and outcomes, and propagating trust information to other nodes.

The transaction outcome itself, for a transaction with monetary value V, can be a number between 0 and V, denoting the degree to which the transaction was successful in the view of the transacting nodes. Typically 0 and V will be the values most often encountered, corresponding to either failed or completely successful transactions; however intermediate values are also possible, in cases of partially successful transactions (e.g. goods delivered but with delay, specifications not entirely met etc. etc.)

Our model further allows the build up of perceived trust values as a result of repeated successful transactions. We describe our mechanism as ratcheted, since it carries inherently a way of incrementing the trust value further than the actual transaction value, based on the transaction outcome, as will be described in detail in the following sections.

3.2 System and functional description

To incorporate Mor-Trust, the architecture of the network nodes is extended to maintain a local trust store, in the form of a table associating node identifiers N with estimated trust values t_N for those nodes. This table will only maintain trust information for a subset of the network (the node's "trust neighbours"), and will be dynamically updated as trust information is propagated from other nodes, or as a result of transactions carried out. This trust store generates a separate overlay network, independent of the underlying structured or unstructured peer to peer network.

The node's functionality is also extended to perform five tasks with each transaction cycle:

1. **Trust path generation.** Due to the distributed nature of the system, it is likely that the transaction initiating node will have no trust information for the target node in its local store. There is therefore need for a mechanism to consult other "intermediate" nodes, until some trust estimate is obtained for the target node. Trust links, or paths, are in this way built between the nodes.

2. **Trust estimation.** Evaluating and combining trust data collected from other nodes along paths to generate a local a-priori trust estimate for the target node.
3. **Decision.** According to the a-priori trust estimate, decide whether to engage in the transaction or not.
4. **Processing transaction outcome.** Evaluating transaction outcome and producing a new trust estimate for the target node.
5. **Trust Propagation.** Updating and propagating the trust estimates throughout the network.

The following sections describe the above tasks in more detail.

3.3 Trust path generation

The general approaches to forming trust links between peers are the following [12] (see also Figure 1):

“Web of trust” approach. Trust information is obtained by finding a path leading from the initiating node to the target node, following links through the nodes’ local trust stores (an example is the Poblano system [9]).

“Statistical” approach. Involves obtaining trust information from many peers and then forming a quorum (an example is the P-Grid system [2]). Such an approach relies on an efficient, decentralized storage infrastructure.

Hybrid approach. Consists of obtaining trust information through different independent paths, and then forming a weighted quorum dependent on the relative confidence placed on these paths.

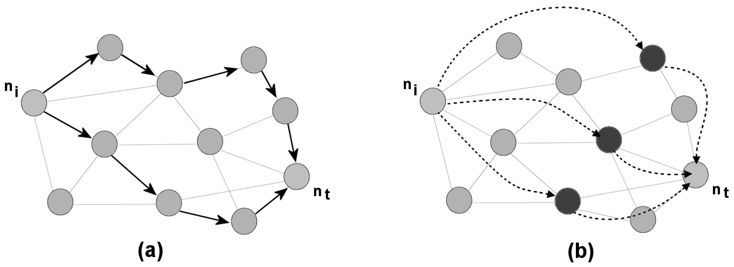

Fig. 1. On the left two separate reputation paths are formed to join the initiating and target node. On the right, three independent nodes hold trust information about the target node, and are interrogated by the initiating node.

In our design we adopted the hybrid approach. The network is scanned based on the local trust stores in a way that ensures that the paths created are independent and non-intersecting. We experimented with different approaches in the path creation, including biased random walks in which the direction selected is biased by the trust estimate placed for each intermediate node.

3.4 Trust estimation.

Assuming the nodes in a path P of length l are labelled $n_1, n_2, ..., n_l$ (where n_1 is the initiating node and n_l is the target node), an overall path trust estimate $T_P(n_1, n_l)$ for the target node is produced through the path, based on the following recursive definition, where $i < j$ and $T_l(n_i, n_j)$ is the trust estimate locally maintained in the trust store of node n_i for node n_j:

$$(1) \quad T_P(n_i, n_j) \quad = \quad \begin{cases} T_l(n_i, n_j), & \text{if} \quad n_j > n_i + 1 \\ T_l(n_i, n_{i+1}) \;\otimes\; T_P(n_{i+1}, n_j), & \text{otherwise.} \end{cases}$$

In the above formula, the path is scanned from n_1 to n_l, and in each step the local trust estimate for each successive node is recursively applied as a weight on the estimate produced for the rest of the path, using function $t_w \otimes t_e$ (denoting the weighing of a trust estimate t_e proposed by a node N, by another trust estimate t_w for N), which is defined as:

$$(2) \quad t_w \otimes t_e \quad = \quad \begin{cases} t_e, & \text{if} \quad t_w \geq t_e \\ t_w \cdot (2 - \frac{t_w}{t_e}), & \text{otherwise.} \end{cases}$$

This function, plotted in Figure 2.a, states that if the weight t_w placed on the trust estimate t_e is greater than the trust estimate itself, then the trust estimate is accepted as is. Otherwise, the trust estimate is reduced according to the trust weight. The reduction follows a quadratic form, increasingly penalising estimates for which the trust weight is comparatively lower.

A confidence value C_P is also created for each path P that considers the entire path length, penalizing longer paths, as well as paths including nodes with trust estimates particularly lower than the overall trust estimate it produces:

$$C_P = \frac{\sum_j (\max(T_P(n_1, n_l) - T_l(n_j, n_{j+1}), 0))^2}{\text{length}(P)}$$

The estimates provided by the different trust paths P_j are then combined, weighted by their respective confidence values C_j, to provide the initiating node with an overall a-priori trust estimate $T(n_i, n_t)$ for the target node:

$$T(n_1, n_l) = \sum_j \left(\frac{C_j * T_{P_j}(n_1, n_l)}{\sum_j C_j} \right)$$

3.5 Decision

Based on the a-priori trust estimate described above, the initiating node n_i needs to decide whether to proceed with the transaction with target node n_t or abort.

We allow some flexibility by incorporating in each node n_i a notion of a “risk factor” rm_i, expressed as a percentage where $rm_i > 1$, so that the decision is:

$$\text{Proceed if } T(n_i, n_t) \geq \text{transaction value} \cdot \; rm_i.$$

The risk factor can, for example, be increased as successful transactions are accomplished and more confidence is gained, however in the current implementation it is maintained constant.

3.6 Processing transaction outcome.

If a node decides not to proceed with a transaction, this step is skipped, and the node moves straight to the trust propagation stage.

The actual transaction process is beyond the scope of the trust algorithm, except that a value denoting the degree of success of the transaction must be reported back from the application to MoR-Trust. As described above, usually the value will either be 0 for transaction failure, or equal to the transaction value V for success, however all intermediate values are allowed.

The a-priori trust estimate t_e obtained before the transaction, is combined with the transaction result r to produce a new trust estimate n_t for the target node. In our current implementation we examine the following options:

- Application of a formula similar to equation 2:

  $$(3)\quad t_n = \begin{cases} t_e, & \text{if } r \geq t_e \\ r \cdot (2 - \frac{r}{t_e}), & \text{otherwise.} \end{cases}$$

  In this way no increase in the perceived trust for the target peer is achieved by the initiating peer in case of a successful transaction, no matter what the transaction value was. In case of a not completely successful transaction the reputation estimate is decreased accordingly.

- Application of the formula:

  $$(4)\quad t_n = \begin{cases} \left(\frac{V_a \cdot r}{t_e}\right)^2 + t_e, & \text{if transaction successful} \\ V_b \cdot \left(t_e - \frac{t_e^2}{r + t_e}\right), & \text{otherwise,} \end{cases}$$

  where V_a and V_b are parameters used to fine tune the algorithm performance (see also Section 5).

  This formula is plotted in Figure 2.b for different values of V_a and V_b. The three lower curves correspond to unsuccessful transactions, while the three upper curves to successful ones. It penalizes an unsuccessful transaction, however for successful transactions whose result exceeds the a-priori trust estimate t_e it will produce an increased new perceived trust. The suggested increase might seem excessive for transaction results significantly exceeding t_e, however it will only occur in the event the initiating peer decides to engages in such a transaction, which already suggests a perceived trust on its behalf that exceeds the generated trust estimate.

3.7 Trust propagation

The way in which the new trust information is propagated to other nodes influences the overall trust network convergence rate, but also the overall traffic, scalability and node trust store size.

We experimented with the following options, based on formula 2, for weighing a new trust estimate with the perceived trust of the node suggesting it:

(i) Only update the trust estimate of the initiating node.

(ii) Update trust estimates for all the nodes in the trust paths.

(iii) Update trust estimates for all the nodes the trust table of the initiating node.

It became apparent that propagating trust further than that would impose too heavy a burden on the network traffic to be deemed acceptable. In Section 5 we present results for option *(i)*.

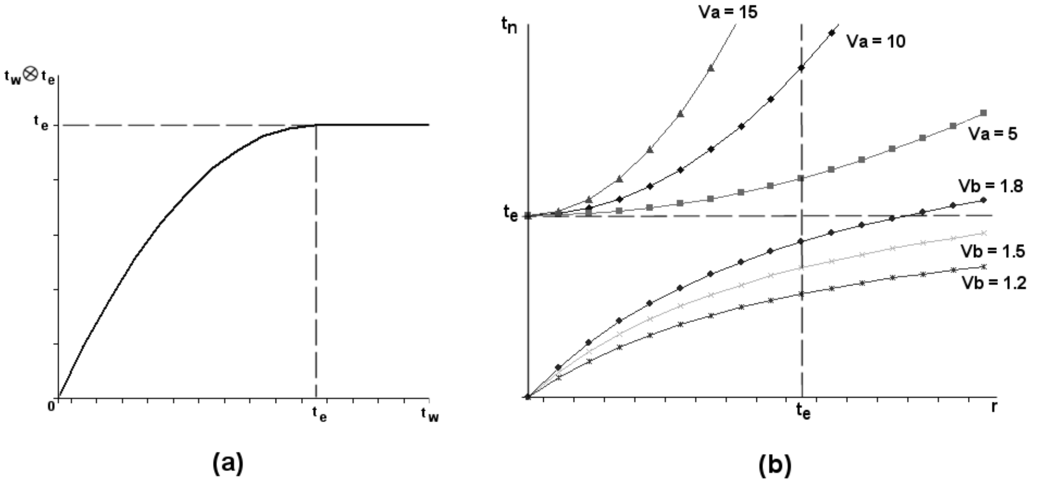

Fig. 2. (a) Function $t_w \otimes t_e$, for weighing a trust estimate t_e proposed by a node N, by another trust estimate t_w for N; (b)The function combining an a-priori estimated trust value with a transaction result to produce a new estimated trust value for the target node. The top three curves are used in case of completely successful transactions, the three bottom curves if not. We show the curves for different values of the parameters V_a and V_b.

4 Simulation setting

The main elements of our simulation setting for experimenting with the various options and parameters of our design are the following.

Network initialisation

Initialisation of the underlying peer-to-peer network structure, the node parameters, and the trust overlay network, based on acceptable statistical distributions.

Both the underlying peer-to-peer network and the trust network, as described through the local trust stores maintained in the various nodes, were generated based on the NGCE [27] application, a tool for generating graphs based on various parameterizable graph topologies including homogeneous, random, and scale-free.

Our networks were set up according to power-law distributions forming scale-free graphs, which are known to describe both the connectivity of the internet and many other technical and social networks [7,15,17,19]. In [23] it was also found, based on eBay transaction traces, that peer ranks also followed a power-law distribution.

Additional node parameters, such as the node “honesty”, a scalar value that drives its behaviour in transactions, and the risk factor, were based on standard statistical distributions.

Network operation

Simulation of transactions is carried out based on average transaction rate per node, and transaction parameters (transacting nodes, value etc.) selected based on standard distributions. For each transaction event the process is carried out as described in Section 3.2.

Trust management system and network status evaluation

The network trust status, and consequently the effectiveness of the trust management system, can be evaluated based on different metrics, such as:

(i) Average transaction satisfaction.

(ii) Correlation between average estimated trust values across network and actual node fairness.

(iii) Correlation between initiated transaction values and target node fairness.

5 Simulation results

The MoR-Trust system was implemented in Java (the code will be made available as an open source project).

The results presented here are preliminary, and further experimentations are currently carried out to verify the system effectiveness.

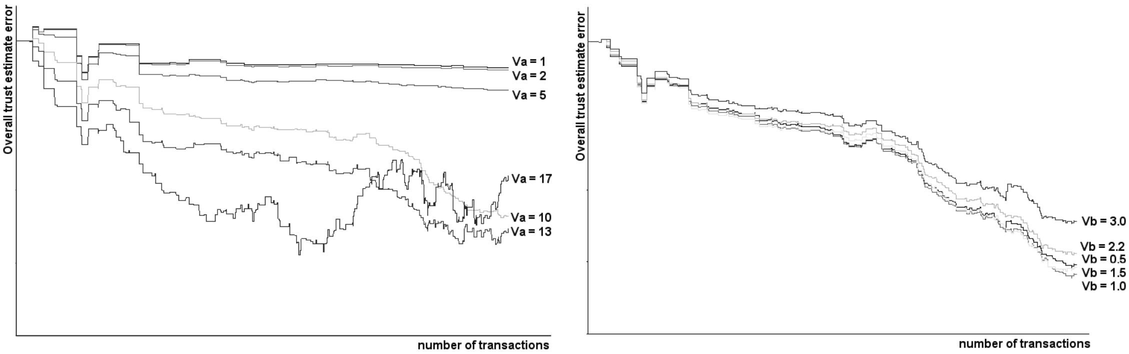

Fig. 3. Convergence of overall network correlation between trust estimates and real node honesty. The left curve shows how parameter V_a affects correlation, the right curve parameter V_b.

Figure 3 shows the convergence of the overall network correlation between global trust estimate and real node honesty values that guide the node behaviour in the transactions. The two graphs show how the convergence is affected by the parameters V_a and V_b in formula 4. We note that an excessive increase in the value of V_a, which dictates the confidence with which trust estimates will be increased beyond the transaction value as a result of a successful transaction, leads to poor convergence; too small a value with stall convergence.

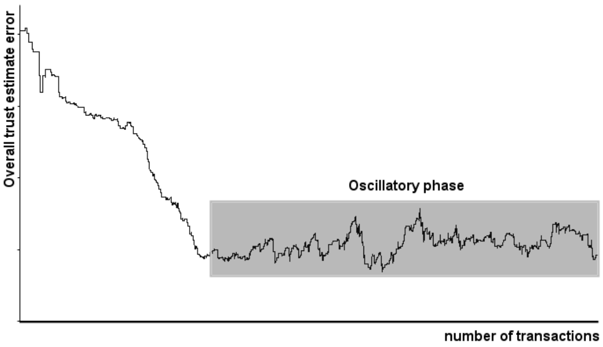

Fig. 4. The oscillatory phase following initial convergence, for large numbers of transactions.

An interesting observation is also shown in figure 4, that shows the trust correlation for a very large number of transactions. We observe that after an initial period of convergence, the average trust estimate correlation enters an “oscillatory” phase, in which the trust estimates move over and under the actual node honesty. We believe that this is introduced by the ratcheting mechanism, which allows the trust estimates to rise above successful transaction values. Though seemingly undesirable, a similar behaviour is also observed in real data from on-line transactions, where reputation that is built in one period of "honest" operation is then "milked" in a subsequent period, thus leading to a similar oscillatory phase [13].

6 Discussion, conclusions and future work

MoR-Trust is based on the notion of expressing trust in terms of monetary units, thus directly coupling the trust estimates circulated in the network with the values of the transactions taking place and their outcomes.

Our design is flexible in that it subdivides the trust management process in separate modules and tasks, allowing the implementation of different approaches for each one.

We are currently in the process of enhancing our implementation, collecting and analyzing more simulation results, and exploring the following research directions:

- Incorporation of security measures (confidentiality, integrity, authentication). Approaches to this are already proposed in the literature (see [6] and references therein) but in our initial implementation we have omitted them for reasons of simplicity.
- Study of resource utilization (bandwidth, node storage, computational needs) as a result of the different variations of the proposed algorithms.
- Implementation of the trust management system on top of more robust, DHT-based peer-to-peer routing systems, such as CHORD [24] or PeerTrust [29].
- Evaluation of alternative algorithms based on disciplines related to sociology game theory, etc.
- Performing an in-depth risk analysis to determine the system's viability and stability in the face of security attacks by a number of malicious nodes.

7 Acknowledgements

This work was supported by the PENED2003 programme and the Heraclitus programme of the General Secretarial for Research and Technology of the Greek Ministry of Development.

References

[1] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *HICSS*, 2000.

[2] Karl Aberer. P-Grid: A self-organizing access structure for P2P information systems. *Lecture Notes in Computer Science*, 2172:179–194, 2001.

[3] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 310–317. ACM Press, 2001.

[4] S. Androutsellis-Theotokis. Social behaviour, incentives and technology in peer-to-peer content distribution networks. *The Ethicomp Journal*, 1(3), August 2004.

[5] S. Androutsellis-Theotokis and D. Spinellis. Performing peer-to-peer e-business transactions: A requirements analysis and preliminary design proposal. In *Proceedings of the IADIS eCommerce 2004 conference*, Lisbon, Portugal, December 2004.

[6] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371, December 2004.

[7] A. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, pages 60–69, May 2003.

[8] Sonja Buchegger and Jean-Yves Le Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *In Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[9] R. Chen and W. Yeager. Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems., 2001.

[10] 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-August 2003.

[11] 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. In *Proceedings of the In 9th ACM Conf. on Computer and Communications Security*, Washington DC, November 2002.

[12] A Datta, M Hauswirth, and K Aberer. Beyond 'web of trust': Enabling p2p e-commerce. In *Proceedings of the IEEE International Conference on E-Commerce Technology (CEC'03)*, 2003.

[13] C Dellarocas. Analyzing the economic efficiency of ebay-like online reputation mechanisms. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, Florida, October 2001.

[14] C. Dellarocas. Building trust on-line: The design of robust reputation mechanisms for online trading communities. information society or information economy? a combined perspective on the digital era. IDEA Book Publishing, Hershey, PA, 2004.

[15] H. Ebel, L. Mielsch, and S. Bornloldt. Scale-free topology of e-mail networks. *Physical Review*, E 66(035103(R)), September 2002.

[16] M Gupta, P Judge, and M Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the NOSSDAV'03 Conference*, Monterey, CA, June 1-3 2003.

[17] M. Jovanovic, F. Annexstein, and K. Berman. Modelling peer-to-peer network topologies through ?small-world? models and power laws. In *Proceedings of the 9th Telecommunications Forum Telefor*, Belgrade, 2001.

[18] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust algorithm for reputation management in p2p networks. In *Proceedings of the twelfth international conference on World Wide Web*, pages 640–651, New York, 2003. ACM Press.

[19] A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. *ACM Computer Communication Review*, 30(2):160–163, April 2000.

[20] T.W. Ngan, A. Nandi, A. Singh, D.S. Wallach, and P. Druschel. Designing incentives-compatible peer-to-peer systems. In *Proceedings of the Second Bertinoro Workshop on Future Directions in Distributed Computing (FuDiCo2004)*, Bertinoro, Italy, June 2004.

[21] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[22] Aameek Singh and Ling Liu. Trustme: Anonymous management of trust relationships in decentralized p2p systems. In *Proceedings of the IEEE Intl. Conf. on Peer-to-Peer Computing*, September 2003.

[23] Shanshan Song, Kai Hwang, Runfang Zhou, and Yu-Kwong Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 9(6):24–34, 2005.

[24] I Stoica, R Morris, D Karger, MF Kaashoek, and H Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of SIGCOMM 2001*, August 2001.

[25] Girish Suryanarayana, Justin R. Erenkrantz, and Richard N. Taylor. An architectural approach for decentralized trust management. *IEEE Internet Computing*, 9(6):16–23, 2005.

[26] UN/CEFACT and OASIS. ebxml business process specification schema v1.01. *Online at: Online at http://www.ebxml.org/specs/ebBPSS.pdf*, 2001.

[27] Vasileios Vlachos, Vassiliki Vouzi, Damianos Chatziantoniou, and Diomidis Spinellis. NGCE — network graphs for computer epidemiologists. In Panagiotis Bozanis and Elias N. Houstis, editors, *Advances in Informatics: 10th Panhellenic Conference on Informatics, PCI 2005*, pages 672–683, Berlin, November 2005. Springer-Verlag. Lecture Notes in Computer Science 3746.

[28] K Walsh and E Gun Sirer. Fighting peer-to-peer spam and decoys with object reputation. In *Proceedings of the SIGCOMM'05 Conference Workshops*, Philadelphia, PA, August 2005.

[29] L Xiong and L Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(16), July 2004.

[30] H Zhang, A Goel, et al. Improving eigenvector-based reputation systems against collusion. Technical report, Stanford University, Workshop on Algorithms and Models for the Web Graph (WAW), October 2004.