Possibilities for Managing Trust in P2P Networks

(EPFL Technical Report IC/2004/84)

Zoran Despotovic, Karl Aberer
Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland

November 2, 2004

Abstract

Reputation systems offer a viable solution to the old problem of encouraging trustworthy behavior in online communities. Their key presumptions are that the participants of an online community engage in repeated interactions and that the information about their past doings is informative of their future performance and as such will influence it. Thus, collecting, processing, and disseminating the feedback about the participants' past behavior is expected to boost their trustworthiness. We investigate and classify the possibilities appeared so far in the literature to do this in the context of P2P networks. We identify three broad classes of approaches: social networks formation, probabilistic estimation techniques and game-theoretic reputation models. They differ greatly in the accompanying trust semantics, mainly reflected in the possibilities offered to the decision makers, and the implementation overhead they incur. The paper bridges the gap between the existing works on trust and reputation management in decentralized networks, driven by the characteristics of the target environment and the formal game-theoretic treatment of reputation, aiming at a clear and analytical decision making. This view leads us to identify the open research issues, oriented towards both efficient and analytical usage of reputation to build trust in P2P networks.

Keywords: P2P Systems, Trust, Reputation

1 Introduction

Recent empirical studies have shown that a great deal of the commercial success of eBay, the largest online auctioning site, can be attributed to its reputation mechanism (Feedback Forum) as a means of deterring dishonest behavior. The analysis of eBay data carried out by Resnick and Zeckhauser (2002) has shown that "reputation profiles were predictive of future performance", while more specific analyses of Houser and Wooders (2001) and Melnik and Alm (2002) brought the conclusion that Feedback Forum fulfilled its promises: the positive feedback of the sellers was found to increase their prices, while the negative one reduced them.

eBay's Feedback Forum is just a well known example of what Resnick, Zeckhauser, Friedman, and Kuwabara (2000) calls *reputation systems* and define as "systems that help people decide whom to trust, encourage trustworthy behavior, and deter participation by those who are unskilled or dishonest through collecting, distributing, and aggregating feedback about the participants past behavior."

Reputation systems appear to be the only way to achieve these goals in P2P networks as *open* and *decentralized* electronic communities, where no conditions on who can join and when one can join and leave the community are imposed. The classical assurance mechanisms, such as contractual agreements and litigation, are practically ineffective and no help of central authorities, trusted third parties in particular, can be assumed. However, mostly due to the mentioned characteristics, designing reputation systems for this class of online environments is not at all an easy task and must be done with great care. In this paper we will address this question by classifying and discussing different alternatives appeared so far in the literature. We will identify their stong and weak points with respect to boosting trust and implementation overhead of the involved algorithms.

The paper is structured as follows. We start in Section 2 by introducing the problem that P2P reputation systems address and giving their general definition. We also outline here the main dimensions around which we center our classification of the existing works. In particular, the definition of trust we will be using in the paper along with a categorization of the underlying peer behavior is given. Our view on trust is simple but powerful in the sense that it enables a clear classification of various reputation models with respect to how and to what degree they promote trust in a community in which they are deployed. Section 2.3 introduces the P2P aspect of the problem, with a particular emphasis on the possibilities to manage the reputation data. Sections 3, 4, and 5 describe the main classes of approaches we identified: Section 3 offers an overview of the concept of social networks and their (in)appropriateness for P2P environments. Section 4 deals with probabilistic approaches for making estimation of the likely future behavior of the peers given their past actions, while Section 5 discusses game-theoretic reputational models. In Section 6 we provide a comparative analysis of the identified solution classes, while in Section 7 we conclude by pointing out open research issues. For completeness, we provide a more detailed overview of the P2P paradigm and main concepts of game theory. They are given in Appendices A and B.

2 P2P Reputation Systems

In the following we introduce a general view on P2P reputation systems design, broad enough to cover all specific works we are aware of. The purpose of the section is to describe in general the problem that trust and reputation management considers. Sections 3, 4 and 5 then describe particular classes of solutions of the problem in the P2P context.

Our starting assumption is that the peers engage in bilateral interactions and that this process results in forming a directed weighted (trust) multigraph. Its node set coincides with the set of peers and the set of edges with the set of interactions between the peers. Because any pair of peers may have multiple interactions the formed graph is actually a multigraph. The requirement that the multigraph is directed is an implication of the assumed semantics of the underlying interactions. Namely, we suppose that in any interaction the service provider and the service consumer can be identified and that the service consumer is the source of the correspoding edge. Generally, the weight assigned to an edge represents the service consumer's feedback of the service provider's trustworthiness in the corresponding interaction. Examples of the set of all weights (we will also call it feedback set and denote it W from now on) include: the interval $[0, 1]$, the two element set $\{0, 1\}$ or any discrete grading such as the four element set $\{very\ good, good, bad, very\ bad\}$. In any case, the set W as well as the semantics associated with its individual elements are assumed to be universally known and agreed upon. In particular, we are assuming that there is a binary partial ordering relation (call it "greater than") defined on W with the interpretation that "greater" elements mean better feedback. Worth mentioning is also that the contexts of the interactions may be separated. In this case the set of weights has the form $C \times R$, where C denotes the set of all contexts and R the set of all ratings. For instance, let $W = \{r, d\} \times [0, 1]$. Here, we assume two possible interaction contexts: (1) recommendations, when the destination node acts as a recommender of other nodes capable of performing a specific task or other recommenders (context r), and (2) the task performances themselves (context d).

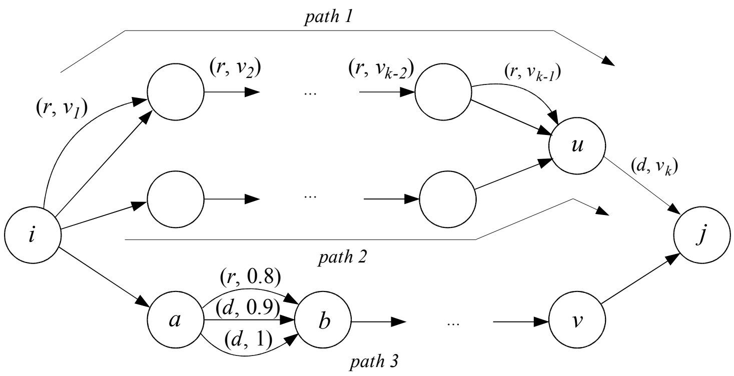

Figure 1: A P2P Trust Multigraph

Figure 1 presents an example. The way we should understand this figure is as follows. Node a had three interactions with node b: once node b acted as a recommender of other entities (flag r) and node a's contention with the recommendation was evaluated 0.8 and twice node b provided the service in question to node a (flag d) and a's evaluations of the service provisions were 1 and 0.9 respectively.

To better understand the multigraph formation process consider a group of people, some of which can be close friends, some acquaintances and some even complete strangers. Assume that they meet and provide a specific service to one another. Any service provision is followed by the service consumer's making opinion about the service provided (or the service provider himself). Sticking to the above example of two contexts, this results in adding a new edge with the flag d to the multigraph. These opinions can be spread to the service consumers's friends, acquaintances or even unknown parties, who, after possibly processing or reinterpreting them, spread them further to their friends and so on. Any time such an opinion is spread, its receiver may use his own knowledge about the target of the opinion to form a new opinion about the sender's ability to recommend other service providers or other recommenders. This corresponds to adding a new edge with the flag r in the multigraph. In this way social networks are formed and, as social sciences argue, people find their ways to judge and make informed decisions about people they never met before. The challenging part of the task of "digitizing" these human networks is what information to transfer among the neighbors, how to aggregate the gossip along a chain of an arbitrary length and how to combine those from different chains.

More generally, the core of any P2P reputation system is in the answer to the following question: how can a given peer use the information on experiences between the peers, that it can retrieve from the network, to evaluate the trustworthiness of any other peer?[1]

In the example from Figure 1 a rough answer might read: to assess the trustworthiness of a node (say, peer j), propagate direct experiences of the nodes which interacted with that node (peers u and v) through the graph down to the node doing the assessment (peer i), filtering them out by the recommendation experiences among the neighboring nodes along the paths to decide on their credibility. Different works propose different strategies for doing this. We stress that most of the existing works do not model explicitly the context of recommendations but rather use direct experiences as credibility filters. This can be thought of as weighting one's reports by his trustworthiness to perform a sevice rather than his ability to recommend.

In a word, to assess peers' trustworthiness we need an algorithm, denote it A, that operates on the formed multigraph, aggregates the feedback available in it, and for any peer given as its input it outputs a value $t \in T$ denoting the estimate of that peer's trustworthiness. Just as with the feedback set W, the trustworthiness levels, represented by the elements of the set T, have globally agreed upon semantics. Note that the feedback set W and the trustworthiness levels set T may coincide, but we do not impose this constraint.

[1] A bit of caution is needed here: we do not say that every peer knows the whole trust multigraph. Instead, we assume that it is reconstructed (the whole graph or a part of it) by querying the other peers, which may not be available or may simply misreport.

All this reasoning leads us to define P2P reputation system as follows:

Definition 2.1 (Reputation System). A P2P reputation system is a quadruple (G, W, A, T), where G is a directed weighted multigraph (P, V) with P being the set of peers and V the set of edges which are assigned weights drawn from the set W. A is an algorithm that operates on the graph and outputs a specific value $t \in T$ for any peer given as its input.

The problem of trust management based on the peers' reputations can be now stated simply as follows: *define the type of feedback to be taken from interacting peers about their partners' trustworthiness (set W) and define a strategy to aggregate the available feedback (algorithm A) and output an estimate of the trustworthiness of any given peer (set T) so that the trustworthy behavior of the peers is encouraged.*

2.1 Classification Criteria

A clear categorization of P2P trust management solutions based on managing peers' reputations must consider in the first place their properties with respect to the following two dimensions:

- Trust related model semantics and
- Incurred implementation costs.

In the context of Definition 2.1, these dimensions are determined by the choices of the triples (W, A, T). Let us describe each of these dimensions in greater detail.

Trust related model semantics. The notion of trust is accompanied by quite some disagreement in the literature. In this paper we will adopt a simple definition of trust that will lead to a clear classification of various trust models with respect to how and to what degree they promote trust in a community in which they are deployed. Namely, deriving from Usunier (2001), we view trust as being inseparable from vulnerability and opportunism associated with the interacting parties. Consequently, we say that

> *agent A (as trustor, or source of trust) trusts agent B (as trustee or target of trust) if the interaction:*
>
> - *generates a gain to be shared with and by agent B;*
> - *exposes agent A to a risk of loss, if agent B takes a too large portion from the joint gain.*

As further pointed out by Usunier (2001), "a general issue of trust management is to assess vulnerability and risks in interactions assuming that the interactants are self-interested." These assessments must be made in such a way that they enable:

- *reducing the opportunism of the trustee (before it occurs),*

- *reducing vulnerability of the trustor and*
- *deciding if and when to enter an interaction, after everything has been done to properly address issues related to the above two points.*

The main goal of any trust and reputation management mechanism is, if possible, completely reducing the opportunism and/or vulnerabilities of the interacting parties. However, if this is not possible then the mechanism should assess the risks and enable the participants to unambiguously decide whether to interact or not.

Inseparable from the trust semantics are the assumptions on the peers' behavior that any given model targets. Broadly, the following two classes of peer behavior have been treated in the literature: rational and probabilistic.

Rational behavior normally implies that there is an underlying economic model in which utilities are associated with various choices of the peers and that the peers act as to maximize their utilities. This behavioral assumption makes it possible to analytically prove that trustworthy behavior is in the best interest of the peers under given feedback aggregation strategies. We will not further elaborate on this type of peer behavior as it has been broadly studied in the literature and normally the reader's intuition is quite sufficient to understand it.

On the other hand, the mentioned analyticity is not present in the models dealing with the probabilistic behavior of the peers. In principle, this class implies that a joint probability distribution is associated with the set of all peers, describing their innate characteristics and determining how they behave in the context of trust and when reporting on others' performance. Thus any event in the form

$$(\ldots, p_k \textsf{ performes } w_k \ \ldots, p_l \textsf{ reports } w_l \textsf{ when } p_m \textsf{ performs } w_m \ \ldots),$$

where $w_k, w_l, w_m \in W$, should have an assigned probability. For instance, a given distribution may specify

$$P[p_1 \textsf{ misreports on } p_3 | p_2 \textsf{ misreports on } p_3] \neq P[p_1 \textsf{ misreports on } p_3],$$

meaning that peers p_1 and p_2 misreport on peer p_3's performance in a coordinated manner, forming thus a collusive group. Needless to say, the exact distribution in not known in advance. The goal of a trust management solution is to devise a method for any trust computation source peer p_j to assess the marginal probability of any trust computation target peer p_i performing in specific ways $P[p_i \textsf{ performs } w_m]$. From this we can easily formulate measures for evaluating performances of the solutions in this class as being dependent on two important parameters: estimation accuracy of the above marginal distribution and range of possible distributions for which this accuracy is achieved.

Incurred implementation costs. The second of the above mentioned classification dimensions is important because P2P networks normally involve large numbers, e.g. millions, of nodes. Thus, particular attention should be paid to cutting down the total implementation overhead introduced by the employed reputation management solution. Generally, it consists of:

- the communication costs associated with the process of retrieving the necessary feedback,
- the involved storage costs,
- and the computation overhead related to the feedback aggregation.

The communication costs are mainly determined by the amount of the necessary feedback on which a given algorithm operates, while the storage costs become important primarily in the case a caching scheme is used to cut the communication costs. The employed feedback aggregation strategy determines the computation overhead.

2.2 Solution Classes

In the above definition of trust and trust management we said that any trust management solution should first try to completely reduce the opportunism and/or vulnerabilities of the interacting parties, or when this is impossible, to provide the participants clear decision making procedure on whether to interact or not. The degrees at which different mechanisms achieve these goals vary. We see the following three broad classes:

- *social networks formation,*
- *probabilistic estimation techniques and*
- *game-theoretic models.*

This division is followed in the rest of the paper so that each of these three classes will be discussed in a separate section. In short, both social networks and probabilistic estimation approaches target probabilistic behavior we just described. Another commonality of these two classes is that neither of them does completely remove possibilities to misbehave. However, they differ with respect to how they enable decision making. As a consequence of ad hoc feedback aggregation which does not make the probabilistic behavioral assumption explicit, it is pretty vague how the outputs of social networks can be used to decide when to enter an interaction. This is much clearer with probabilistic estimation because their outputs are probabilities of specific behaviors of the peer in question. Performance quality makes another distinction between social networks and probabilistic estimation. Generally, social networks tend to cover broader ranges of unknown behavioral probability distributions, while probabilistic techniques require more knowledge on them.

On the other hand, the game theoretic reputational models, targeting rational behavior, make a step even further towards a clean decision making process by prescribing the exact behavior of the interacting parties in the form of their equilibrium strategies. In some cases trustworthy behavior (for instance a seller always delivering goods even after payment reception) can be enforced as the only equilibrium of the game being played. Thus, game theoretic models may in principle completely remove any possibility to misbehave.

2.3 P2P Systems Perspective

From a systems perspective P2P computing can be seen as sharing of computer resources (disk storage, processing power, exchange of information, etc.) by direct communication between the participating computing systems avoiding central control. Thus, it takes advantage of already existing computing resources allowing a network of computers to more effectively make use of their collective power. A more comprehensive technological overview of the existing possibilities to achieve this in the context of data sharing is given in Appendix A. In this section we will give a brief high level overview of the existing solutions, with a particular emphasis on the possibilities to manage the reputation data in a trust multigraph.

Managing reputation data can be seen as the problem of managing a simple database consisting of a binary relation storing $(key, value)$ pairs, where key is the identifier of a peer and $data$ holds the associated reputation information. In a P2P architecture this data is distributed among a dynamically evolving set of peers so that basic data access operations, such as search and update, are efficient and the storage space required at each peer is small in comparison with the size of the database. In addition, no central control is used and the system should tolerate dynamically leaving and joining peers. In order to achieve these tasks peers organize themselves in so-called *P2P overlay networks*.

There exist two fundamental approaches to construct P2P overlay networks: (1) unstructured P2P networks, e.g. Gnutella (2001), and (2) structured P2P networks, e.g. Aberer (2001), Ratnasamy, Francis, Handley, Karp, and Shenker (2001), Stoica, Morris, Karger, Kaashoek, and Balakrishnan (2001). In unstructured P2P networks peers are connected in a randomized fashion with a small number of neighbors. $(key, value)$ pairs are randomly associated with peers and broadcasting mechanisms are used for searching. In structured P2P networks peers are associated with keys from the key space and consequently become responsible to store $(key, value)$ pairs that correspond to their chosen key, typically close-by keys. They maintain routing tables with references to neighboring peers that are constructed such that search requests to a responsible peer can be routed with a low number of hops. Normally, unstructured P2P networks exhibit high lookup costs of $O(E)$ generated messages, E being the number of the edges, while in most of the structured networks this cost is logarithmic in the number of the nodes. On the other hand structured P2P networks incur higher maintenance cost, be it for data insertion and update, or in the presence of node joins and failures.

Having clarified this, it should be clear that the trust graph can be actually stored in an underlying P2P system. In the case of an unstructured P2P network every peer can store its outgoing edges from the trust graph (the identifier of the destination node and possibly time stamp may act as the key), while in the case of a structured P2P network the triples $(destination, source, timestamp)$ may act as the keys for the trust graph edges and be stored at peers just as dictated by the P2P network, as described by Aberer and Despotovic (2001), not necessarily at the peers that made the corresponding feedback. This imposes a new problem for structured P2P networks. Namely, the peers storing the feedback may find it profitable to misreport. To this end we are assuming that the underlying structured overlay network is configured in such a way that the feedback is replicated (the same edge from the trust graph is stored at multiple peers) and that an appropriate voting scheme to eliminate possible misreports of the feedback stores is employed, e.g. Aberer and Despotovic (2001).

In the cases of both structured and unstructured P2P networks the weights of the edges may act as the values. Thus, exploring the trust graph reduces actually to searching the underlying P2P network. More specifically, retrieving feedback about any specific peer is subdued to searching for the data items with the keys starting with that peer's identifier.

It is also possible to use the trust graph directly as a new overlay network on top of the existing P2P overlay network to retrieve the necessary reputation data. However, we do not believe that this is a good idea. In the case of an underlying structured P2P network the reputation data about any specific peer can be retrieved with $O(\log N)$ overhead ($O(N)$ to retrieve the entire trust network). The case of the underlying unstructured networks is somewhat different but the conclusions are the same. Namely, the trust network should normally have a lot more edges than the P2P overlay network. Assuming that both the networks are explored in a flooding or breath-first search like fashion, in which the number of the edges matters, we reach the conclusion that the P2P overlay should be used.

3 Social Networks

Social networks target probabilistic peer behavior and are mainly characterized by ad hoc feedback aggregation strategies. Normally they imply the aggregation of all reputation information available in the formed trust network, as illustrated in Figure 1. A natural interpretation of this process involves the following steps: 1) enumerating all paths from the trust computation source to the target node, 2) aggregating the trust values along the paths to give a path wide gossip and 3) merging these gossips into a final value.

Beth, Borcherding, and Klein (1994) present one of early examples characterized by the distinction between the contexts of recommendation and direct trust. The feedback is binary in the both contexts so that $W = \{r, d\} \times \{0, 1\}$. The feedback aggregation algorithm starts by aggregating all direct and recommendation interactions between neighboring nodes. Given its p "positive" direct experiences with any peer j, any peer i computes the direct trust $v_d^{ij}(p) = 1-\alpha^p$, $0 < \alpha < 1$; having at least one negative experience with peer j, peer i should put $v_d^{ij} = 0$. On the other hand, given p positive and n negative experiences with a recommender the recommendation trust becomes $v_r^{ij}(p, n) = 1 - \alpha^{p-n}$ if $p > n$

and 0 otherwise. The exact value of parameter α was left unspecified. Thus the initial multigraph gets transformed so that for any ordered pair of nodes there are at most two edges between them, one per context, carrying these aggregated values. Further, the source node enumerates all paths to the destination node, selects only those such that the last branch carries direct trust (let us denote it v_k for a generic path *path i*) while all intermediate ones carry recommendation trust (denote them $v_1, \ldots, v_{k-1}$) and propagates the trust of the destination node through them according to the formula $v_{path\ i} = 1 - (1 - v_k)^{v_1 v_2 \cdots v_{k-1}}$. The path *path* 1 from Figure 1 presents an example. Then, it groups together the paths with a common penultimate node (paths *path* 1 and *path* 2 from Figure 1 should be grouped, for instance) and merges their trust values acoording to formula $v_{group\ j} = \sqrt[m]{\prod_{i=1}^{m}(1 - v_{path\ i})}$, where m is their number and $v_{path\ i}$ their values. Finally, it merges computed trust values of all groups $v = 1 - \prod_{l=1}^{s} v_{group\ l}$. As can be easily seen, the output value $v \in [0, 1]$ and thus $T = [0, 1]$.

A simple analysis of the presented algorithm shows that, because all paths between the two concerned nodes must be explicitly taken into account, the algorithm complexity may be exponential in the number of the nodes in the network. This is clearly not acceptable in P2P networks in which this number can easily reach the order of magnitude of millions. Note that we are not strict here in the sense that this explosion must happen if we consider *all* the paths. As we will see shortly there is a way around this problem through a synchronous computation if the feedback is conveniently propagated and aggregated. However, the above algorithm does not fulfill this requirement and its computation complexity is in fact exponential, as the authors show.

Yu and Singh (2000) offer a polynomial time feedback aggregation algorithm. Unlike Beth, Borcherding, and Klein (1994) this approach does not consider the context of recommendations separately. Again, the direct experiences are rated binary (good and bad, $W = \{0, 1\}$). The feedback aggregation algorithm starts, just as with Beth, Borcherding, and Klein (1994), with aggregating the interaction outcomes between the neighbors, which transforms the initial multigraph into a graph such that only one edge remains for any ordered pair of nodes. To this end, the authors propose a well known machine learning technique, delta learning namely, resulting in values between -1 and 1. To propagate the values over chains the authors use multiplication, with the constraint that trust over a chain with at least one negative link must be negative. When merging the values of multiple chains they choose only those chains carrying maximal values from the source to all the neighbors of the destination node. In the example from Figure 1 only one of the two chains from peer i to peer u would be selected, the one having the larger associated value. The mean value of all these chains is selected as the algorithm output. Thus, the output values remain in the interval $T = [-1, 1]$. This computation is polynomial in the number of nodes (both finding all j's neighbors and selecting the maximum weighted chains toward those neighbors are polynomial operations and so is their union) and thus presents a considerable improvement as compared to Beth, Borcherding, and Klein (1994).

Richardson, Agrawal, and Domingos (2003) discuss another possibility for an efficient aggregation of the feedback in a trust multigraph with the above interpretation of exploring all the paths between two given nodes and then merging their aggregated trust values.[2] It is characterized by the synchronous feedback aggregation in which trustworthiness values of all peers are computed. On the other hand, it is important because it provides a feedback merging method with more efficient computation.

Consider a trust multigraph with the feedback set $W \subset \mathbb{R}$ (thus W is one-dimensional, the recommendation context is not modeled) and assume that the multigraph has been transformed into a graph by merging individual interactions between neighbors. At the moment it is not important how exactly this merging is done, any of the previously discussed strategies will do. Consider now the matrix $M \equiv [M_{ij}]_{i,j=1}^{N}$ (N is the number of the peers) corresponding to the obtained graph and assume that it has been normalized so that for any $1 \leq i, j \leq N$:

$$0 \leq M_{ij} \leq 1 \text{ and } \sum_{k=1}^{N} M_{ik} = 1.$$

Central to the approach are two binary operators: trust concatenation (propagation), the symbol $\circ$ will be used to denote it, and trust aggregation, we will use the symbol $\diamond$. Normally, the former is applied on two consecutive edges in a chain of trust, while the latter applies to two chains. Simple multiplication and addition are good examples of these operators. The authors then introduce a matrix "multiplication" operation $\bullet$ defined as $C = A \bullet B$ such that $C_{ij} = \diamond(\forall k : \ A_{ik} \circ B_{kj})$. If $A = B \equiv M$, where M is the matrix representation of a given trust graph then the interpretation of C_{ij} is aggregated trust that i puts on j over all chains of length 2. Again, if $\circ$ and $\diamond$ are ordinary multiplication and addition then $\bullet$ becomes the ordinary matrix multiplication.

Now, the most interesting result is that if $\diamond$ is commutative and associative and $\circ$ is associative and distributes over $\diamond$ then the aggregated value of all paths (of any length) between any pair of users can be obtained by the following simple algorithm:

$$Q^{(0)} = M, \quad Q^{(k)} = M \bullet Q^{(k-1)} \quad \text{until } Q^{(k)} = Q^{(k-1)}. \tag{3.1}$$

The computation will converge after a finite number of steps if the matrix M (or the trust graph) is irreducible and aperiodic. It is important to see that the computation can be performed locally, after each iteration k all the peers can retrieve from their neighbors the currently computed opinions of those neighbors about all other peers and then do the computation of the step $k+1$. It turns out that this algorithm requires at most $O(N^3)$ computations per peer.[3] However, it is not clear what should be the overall latency the algorithm introduces because

[2]Semantic Web is the exact setting considered by Richardson, Agrawal, and Domingos (2003) but most of the results can be without any change transferred to P2P networks. Also, the terminology used in the paper is slightly different from the one we used so far. However, to be consistent with the previous discussion we will use continue to use our terminology.

[3]The complexity depends on the underlying graph connectivity. It is slightly less for low connected graphs.

it is determined by the number of iterations and this number is in turn affected by the network graph connectivity.

For completeness we have to say that a number of other works can be considered as special cases of Richardson, Agrawal, and Domingos (2003). Page, Brin, Motwani, and Winograd (1998), used as Google's method for Web pages ranking, and Kamvar, Schlosser, and Garcia-Molina (2003), targeting P2P networks, present two examples. They both use the ordinary matrix multiplication as the matrix operation from (3.1).

Xiong and Liu (2004) presents the only work we know about that avoids aggregation of the individual interactions (transforming the interaction multigraph into a graph) but rather operates on the multigraph directly. It does not consider the recommendation context and uses ratings from the interval $[0, 1]$, so again $W = [0, 1]$. The main idea of this work is to compute the trustworthiness of a given peer as the average feedback about it weighted by the trustworthiness of the feedback originators themselves. This can be expressed by the formula:

$$t_j = \sum_{e \in incoming(j)} w_e \frac{t_{source(e)}}{\sum_{f \in incoming(j)} t_{source(f)}}, \tag{3.2}$$

where $incoming(j)$ is the set of all edges ending at node j, w_e is the feedback belonging to the edge e and $t_{source(e)}$ the trustworthiness of the originator of this feedback. As the authors claim, this formula can be computed by using an iterative computation, similar (but still different) to (3.1). As such, it suffers from more or less the same problems: the computation is inefficient and the whole network must be retrieved. But, the authors also develop a simple caching scheme in which the trust values of the feedback originators are taken from a cache (default values are used in the case of cache miss) and the computed trust of a peer replaces its corresponding cache value.

Building on Dellarocas (2000) and Zacharia, Moukas, and Maes (1999), the same work (Xiong and Liu (2004)) proposes another approach based on so called collaborative filtering technique. The idea is very similar to the previous one, the only change is that the weights in (3.2) are replaced with "similarity coefficients" between the raters and the trust computing peer. They are computed by finding a common set of peers that interacted with the computation source and a given rater and calculating the standard deviation between the two corresponding vectors.

3.1 Trust Semantics

Let us assume now that we used one of the above approaches to compute the trustworthiness of a peer and that we got a value, say 0.3. The question we ask now is: how can we use this value to decide whether to interact with this peer or not? Higher values should imply more trust but is this particular value high enough for us? From the way how the value has been computed in any of the above methods it is clear that it cannot be interpreted as the (estimated) probability of the trustworthy behavior of the target peer. So, what does the value actually represent? The lack of a plausible answer to this question is what all the discussed approaches have in common. Precisely, the computed values lack a plausible interpretation on an absolute scale and therefore only scenarios in which they can be used should involve ranking the trust values of many peers and selection of the most trustworthy one(s) among them. Such scenarios are certainly relevant for P2P networks - selecting the most reliable source for file download being a well known example.

There is another interesting point we would like to mention. Consider algorithm (3.1) again and assume that the operations $\diamond$ and $\circ$ are addition and multiplication so that $\bullet$ becomes the simple matrix multiplication. Now, if the trust graph is irreducible and aperiodic then the k-th power of the corresponding trust matrix M from (3.1) converges for sufficiently large k to a matrix in which all the rows are the same and sum up to 1. (In parlance of matrix calculus, this is the primary Eigenvector of matrix M.) Thus the trust values of the peers, as computed in (3.1), have global meaning - they are independent of the computation source. On the other hand, because all the values sum up to 1, it seems as if the trust was distributed among the peers. Some authors argue that this is a desirable property. But, if we have the value for only one peer (suppose that we used some approximate method to compute it or simply evaluated all the paths to the target) does it mean that the value is low because the concerned peer is malicious or because it is trustworthy but had to "share the trust" with other trustworthy peers. Further, even if we have the values for all the peers, but they are approximately close we are in doubt whether the whole network is trustworthy or it is malicious.

3.2 Performance Analysis

All the mentioned works deal with the probabilistic behavior of the peers, as we explained in Section 2.1. Thus, elaborate analyses of the performances of the proposed methods require numerous tests under many different settings. More specifically, the proposed feedback aggregation strategies must be evaluated against various types of malicious behavior represented by different probability distributions, including uncoordinated (i.e. independent) misbehavior of the peers as well as forming collusive groups of different sizes and varying collusion patterns. However, only Kamvar, Schlosser, and Garcia-Molina (2003) and Xiong and Liu (2004) offer informative simulations, whose condensed view we now present.

Kamvar, Schlosser, and Garcia-Molina (2003) and Xiong and Liu (2004) report good performance when the fraction of malicious peers is small (below 45% approximately) and their maliciousness is uncoordinated - they cheat independently in the interactions and distort their ratings of other peers. To a large extent this is true irrespective of whether they cheat always or also cooperate with some non-zero probabilities. The robustness of the methods in this case lies primarily in the fact that, as long as the cheating population size is below the mentioned threshold, the fraction of misclassified peers remains almost constant and very low (within 5%). Interestingly, the approximate computation of

Xiong and Liu (2004) exhibits very similar behavior, the only difference is that the rise of the misclassification rate falls into a wider region starting at around 35%.

Xiong and Liu (2004) further report the complete breakdown of the original mechanism, given by (3.2), and the approximate one when the cheaters take more than a half of the overall population or when they collude. Making numerous fake interactions and giving good ratings to the partners was identified as the right collusion scheme to defeat the mechanism. With respect to this we believe that certain improvement can be achieved by not taking into account all interactions reported by peers in (3.2) but only a (random) fraction of them.

On the contrary, Kamvar, Schlosser, and Garcia-Molina (2003) claim almost full effectiveness of the presented mechanism even when the malicious peers make the larger fraction of the population and collude in various ways. This results from the assumption that a number of pretrusted peers exist each of whom is assigned some non-zero trust by the rest of the community, including the malicious peers. Thus, there must be an edge with a non-zero weight leading from any node in the trust graph to any of the pretrusted nodes. (This is actually a well known technique studied in so called "random walker" models for Web pages ranking.) However, we do not see a clear way to enforce this behavior in a distributed computation. On the other hand, it can be simulated if the computation is performed centrally at one peer after all feedback in the network has been retrieved. But in this case the computation overhead grows by an order of magnitude. It would be interesting to study various approximations of the method that could be done locally and at the same time retain good performances of the original approach. The authors also analyze various collusion scenarios and identify the following as the most effective one. Malicious peers split into two groups. The peers from the first group cheat always, while the peers from the other group never cheat in direct interactions and give high ratings to the peers from the first group. We add that such collusion scenarios can be defeated only by separating the contexts of recommendations and direct service provisions.

Xiong and Liu (2004), the similarity based technique, may offer a solution to this problem. According to the authors, this scheme stays effective even when the majority of the peers are malicious and they form collusive groups that make fake interactions. It would be interesting to check its performance in presence of the collusive behavior explained in the previous paragraph.

Needless to say, both Kamvar, Schlosser, and Garcia-Molina (2003) and Xiong and Liu (2004) report high benefits in the system performance from the employment of a trust management scheme.

3.3 Implementation Overhead

It should be obvious from the above discussion that in the most of the mentioned approaches all available information in the network is used when assessing the trustworthiness of a single peer. This immediately implies that all the peers in the network are affected and that the associated communication costs are high.

Further, some of the discussed methods require exploring all the paths between two given nodes incurring thus a considerable computation overhead. We see this as the most serious obstacle to their usability in practice in the context of P2P computing. Polynomial complexity of Yu and Singh (2000) offers a certain improvement, though a considerable fraction of the peers may be still affected by a single trust computation. The approximate method of Xiong and Liu (2004) imposes the least overhead as only the direct witnesses of the target peer's performance are involved in the computation.

Kamvar, Schlosser, and Garcia-Molina (2003) and Richardson, Agrawal, and Domingos (2003) propose a synchronous polynomial complexity. However, even if we see its total overhead as acceptable, we do not believe that the algorithm as just specified is feasible in a P2P network. Simply, P2P networks are highly dynamic systems - peers go offline and come online at unpredictable time instants, the trust values between neighbors keep changing constantly and the recomputation of the algorithm triggered by any such event would be highly impractical, if not impossible. Instead, we believe that an incremental computation is something worth further investigation. Caching schemes proposed by Xiong and Liu (2004) offer important insights with respect to this.

4 Probabilistic Estimation

As we have just seen, an important problem with social networks formation based approaches is their implementation overhead. A natural way to eliminate this problem is to consider only own experiences with a peer whose trustworthiness is being estimated or, in the lack of (a sufficient number of) these, to take into account the second level judgments, i.e. reports of the other peers about its past behavior. Under these circumstances it becomes easy to construct probabilistic models, i.e. models whose outputs can be interpreted as probability distributions over the possible behaviors. The importance of such models becomes clear if we recall the view on trust we presented in Section 2.1. If the opportunism of the trustee and the vulnerability of the trustor cannot be reduced completely then it becomes important for the trustor to be able to estimate the risks of the interaction and decide whether to enter it or not. If the individual outcomes of the interaction are assigned the probabilities and the trustor can assign them utilities as well then this task becomes easy - the trustor just needs to compute whether entering the interaction has a higher utility than staying out.

A typical probabilistic model would do the following. First, it would explicitly introduce the assumptions about the probabilistic behavior of the peers, just as we explained in Section 2.1. For instance, such an assumption might be that any peer is trustworthy with a certain, but unknown probability. Or, when reporting its own experiences with others, each peer may lie with some, again unknown probability. Second, it would use well known probabilistic estimation techniques to learn all unknown parameters, each time it obtains new information, either a direct experience from an interaction or a new report from a peer.

Bayesian estimation is a typical representative of these techniques and can be used for this purpose. Its main idea is to assign a prior probability distribution to an unknown parameter and, given an observed set of samples, calculate its posterior. We will here describe in short how it can be used for estimating the unknown parameter of a Bernoulli distributed random variable. Such a task can be encountered for example when the trustworthiness of the peers is treated as a Bernoulli random variable and the probabilities of their trustworthy behavior have to be estimated. According to the notation we established in Section 2 this would mean $W = \{0, 1\}$ and $T = [0, 1]$. Any prior knowledge on the distribution of the unknown parameter in this case can be easily modeled as a beta distribution with appropriately chosen parameters. Interestingly, posterior realizations of the variable lead only to a change in the beta distribution parameters, they do not change the distribution type. Thus, the unknown parameter remains beta distributed, but now with different parameters. Formally, let us assign the unknown probability θ of the trustworthy behavior of a specific peer the prior distribution $Beta(a, b)$.[4] If we now observe n realizations of the peer's behavior k of which were trustworthy then the posterior distribution of θ becomes $Beta(a+k, b+n-k)$. The Bayes' estimator of θ is the expected value of $Beta(a + k, b + n - k)$. It can be shown that this value equals $\frac{a+k}{a+b+n}$ and that the estimator is asymptotically unbiased and consistent.

All this applies directly to the case of allowing only for own experiences. Mui, Mohtashemi, and Halberstadt (2002) present an example in which this approach is used. Apart from this the authors also provide the minimum bound on the number of encounters one has to have with another peer in order to retain the probability of a specific error of the estimation within given bounds. It is given by the following inequality:

$$m \geq \frac{1}{2\epsilon^2} \ln\left(\frac{\delta}{2}\right),$$

where ϵ and δ are the estimation error and confidence level respectively. However, it was left unspecified how to apply the method to integrate reports from direct witnesses when no meaningful decision can be made based on own experiences only. Buchegger and Le Boudec (2003) make a step towards this. However, even though the authors discuss a number of possibilities to deal with the "second hand" opinions they use an intuitive approach in which all second level information sources are given equal weights. To the best of our knowledge, there is no P2P reputation model extending the Bayesian estimation technique in the most natural way (so called "super Bayesian estimation") to take these or higher level beliefs into account as well. In comparison with the "one source" Bayesian models the only difference would be that the samples do not come from the same distribution representing the service provider's trustworthiness

[4] A random variable Θ is distributed $Beta(a, b)$ if its probability density function is $f(\theta) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$, $0 < \theta < 1$, $a > 0$, $b > 0$, where $B(a, b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx$. a and b are the parameters of the distribution, the case $a = 1$ and $b = 1$ corresponds to the uniform distribution.

but from different ones as the original samples now pass through the second level sources who may misreport. But, the probability distribution of these misreports can be also estimated and updated with new experiences so that calculating the posterior distribution of the unknown parameters is not harder at all. This also enables an easy separation of the contexts of recommendations and direct service provisions, that we found in Section 3.2 to be quite robust. Thus the approach has some common points with the previously mentioned similarity based computation of Xiong and Liu (2004), the key difference is that it has a stronger theoretical foundation. In a similar way it can be extended to cover third and even higher level experiences.

Another way to make statistical inferences is the method of *maximum likelihood estimation*. As compared to the Bayesian estimation technique it does not provide a method of assessing unknown parameters probability distributions but gives instead their most likely values given a set of samples. Assume that we know the form of the probability distribution of a random variable but do not know the exact values of involved parameters. The gist of the approach is to compute the likelihood of the observed sample for general values of the unknown parameters and fit the values that maximize it. For instance, let us suppose that we get reports from a number of peers about the trustworthiness of another peer. Now, assuming that the witnesses lie and the target peer is trustworthy with some unknown, possibly different, probabilities it becomes easy to compute the probability of the obtained sample. Then a set of values that maximize this probability is the maximum likelihood estimate of the unknown parameters. Despotovic and Aberer (2004) apply this method in a setting in which peers are characterized by two Bernoulli distributed random variables: one characterizes their trustworthiness and the other one their behavior when reporting the other peers' performances. Thus there are two parameters associated with each peer: the probabilities of trustworthy behavior and reporting honestly about others. Consequently, $W = \{0, 1\}$ and $T = [0, 1]$. The main idea of the approach is to use maximum likelihood estimation to assess the former one of these two parameters, while the latter one is approximated by checking reports about own performances. As the authors show, the computation is highly efficient. However, a good performance of the method has been reported only in non-collusive setting, i.e. when peers act independently.

Aberer and Despotovic (2001) present another approach to managing trust in P2P networks built explicitly on probabilistic assumptions. It does not provide any prediction of a likely future performance of the peers in terms of a probability distribution over the possible performances. Instead, it just tries to assess whether a given peer has ever cheated in the past. Precisely, the interactions considered in this work are binary (the peers either cheat or cooperate) and so is the feedback. After any interaction its participants can file complaints against each other. It is assumed that after cheating in an interaction any peer will file a complaint against its partner, trying to hide in this way its own misbehavior. Then, the mentioned decision is made by analyzing and separating the probability distributions of the filed and received complaints of any peer. Though it is very questionable how precisely the information on whether a peer ever cheated or not can be used to make decisions on entering an interaction with it, it is clear that it provides a certain insight to its trustworthiness.

4.1 Trust Semantics

As we pointed out at the beginning of this section, the main reason for separating the discussion on the probabilistic estimation techniques from the social network approaches is the clear interpretation of the resulting values. They are probabilities of specific behaviors of the computation target. Thus, they have a meaningful and theoretically founded interpretation on the absolute scale $[0, 1]$ and no rescaling or comparing with the values of other peers is necessary.

Despotovic and Aberer (2002) present a scenario in which this can be of interest. Safe exchange (Sandholm (1996)) offers an approach to gradual exchanges of goods and money in which both payments and goods are chunked with their deliveries scheduled in such a way that both exchange partners are better off by continuing the exchange till its end than by breaking it at any step before. Despotovic and Aberer (2002) provide a trust aware extension of the original approach; they take expected utility inputs from the exchange partners (where expectations are computed with respect to their predicted trustworthiness) and provide an efficient algorithm to find a safe schedule of deliveries of goods and payments that satisfies the partners' expectations.

4.2 Performance Analysis

It is hard to give precise judgment on the performance of the probabilistic estimation approaches because of the lack of informative simulation results in the papers we reviewed. Normally, they are more focused in the sense that they consider a narrower range of possible peer behaviors by introducing constraining assumptions on the unknown behavioral probability distribution. For instance, collusive misreporting is often excluded. Therefore, one can intuitively expect their better performance as compared to the social networks if the introduced behavioral assumptions are satisfied. Despotovic and Aberer (2004) confirm this intuition. The authors report the average estimation error within several percents even when misreporters take up to 20-30% of the peer population and relatively small numbers of interactions per peer are considered (up to 30). On the other hand, the assessment quality is highly dependent on the amount of information used. This explains why the mentioned performance is just slightly better than that of the social networks, why the social network approaches perform well even under varying assumptions on the peer behavior and why it makes sense to consider probabilistic techniques involving higher level reports.

4.3 Implementation Overhead

The reputation information on which all previously mentioned probabilistic approaches operate is localized around the trust computation target peer and thus one can expect that the communication costs associated with the computation are low. Despotovic and Aberer (2004) make this precise by specifying that even a total of 30-40 reports on the performances of the target peer are enough for making good estimates. The computation overhead and storage costs are virtually negligible.

However, probabilistic estimation techniques may span a larger fraction of the network if many levels of reports are considered. An interesting question to investigate with respect to this is evaluation of dependency of the estimation quality on the used information scope.

5 Modeling Reputation in Game Theory

The phenomenon of reputation has been extensively studied in economics, game theory in particular. The game theoretic framework for analyzing reputation is that of repeated games in which some players are uncertain about the payoff structures of their opponents. Needless to say, the main underlying assumption on the behavior of the involved economic agents is that they act as to maximize their utilities. As will be seen, it is exactly this assumption that brings the advantages to the game-theoretic reputation systems in terms of a clean decision making process when considering past doings as a predictor of one's future behavior.

The right game-theoretic tool for modeling the mentioned uncertainties is that of games with incomplete information (Bayesian games) in which different players may have different information about some important aspects of the game, including a special case in which some players know more than others and may use this informational asymmetry to get better payoffs. Central to Bayesian games are *types* of the players by which the players' private information is modeled. In most of the models the types simply correspond to different payoff structures of the players.

On the other hand, the repeated interactions are modeled by well studied models of repeated games, where a given stage game is played many (finitely or infinitely) times and the players maximize their long-run payoffs.[5] Generally, the set of players can vary among the stages, but the current literature focuses on the following two extreme cases: (1) the models in which all the players stay active in each stage (long-run players) and (2) the models in which one or more players stay active in each stage (long-run players) while the others play at one stage only (short-run players). We will concentrate here on the latter, because the large number of participants in today's online groups makes repeated interactions between the same players highly improbable. Our main concern in the rest of the section will be to show how trustworthy behavior of the long-run player, perceived as the trustee, can be induced by making the feedback on his past play available to the short-run players, seen as the trustors.

Let us first show how game theory formalizes the concept of reputation. Figure 2 (adapted from Kreps and Wilson (1982), a seminal work in the area)

[5] We consider here only δ-discounted ($0 < \delta < 1$) repeated games in which any k-stage payoff is δ discounted as compared to that of the stage $k-1$.

presents an example. Assume that player 1 is a firm already established in a market (monopolist hereafter) and that player 2 is another firm that decides on entering the market or staying out (entrant hereafter). The monopolist can be weak or strong but the entrant is uncertain about which of these two types it is actually facing. In both cases the monopolist can choose to fight the entry (say, it may opt to a sharp price cut) or to acquiesce (to share the market peacefully), while the entrant chooses whether to enter or stay out. The payoffs of the two firms are as shown in Figure 2. It is important to notice that a strong monopolist prefers fighting the entry ($1 > 0$) and the entrant prefers staying out if he believes that the fight will occur ($0 > -1$). If the entrant stays out the monopolist gets the best payoff he can get in the game.

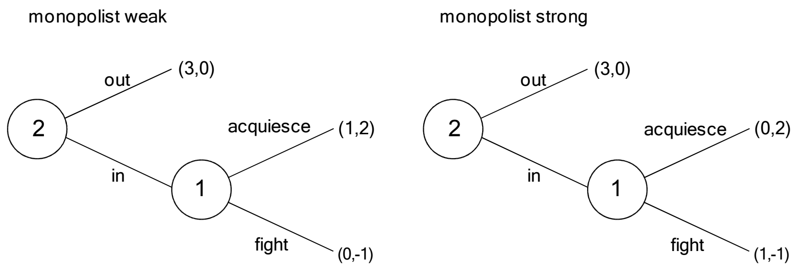

Figure 2: Bayesian Chain Store Game

Assume now that a *weak* monopolist plays this stage game against a finite number of entrants each of whom plays only once but is informed about the whole previous play. Assume also that all the entrants assign a common prior probability distribution to the set of the monopolist's types (weak and strong) and update at each stage their beliefs on the types according to the monopolist's previous move as well as that both the distribution and the entrants' beliefs update function are common knowledge in the game (thus the monopolist knows them too). The question we ask here is: does it make sense for the monopolist to fight eventual early entries in order to deter later ones (to develop reputation for "toughness")? Intuitively, this might make sense because not sharing the market in later stages may offset losses from fighting early ones (recall that the monopolist is actually weak).

Kreps and Wilson (1982) offer the answer. Namely, if all the entrants know that the monopolist is weak (so the prior probability of being strong equals 0) then there is only one equilibrium of the repeated game in which all entrants enter and the monopolist acquiesces at all stages. So, a monopolist that is known to be weak will never fight an entry under these conditions. Put differently, the monopolist cannot develop reputation for being strong when its opponents cannot believe that it can be strong. But, as soon as this probability is non-zero there are equilibria in which the monopolist would fight some early entries and the affected entrants would consequently stay out. This is exactly the effect of the monopolist's reputation, or more precisely the possibility to develop it. The point is that fighting these early entries would convince future entrants that the monopolist is strong so that the losses incurred by these moves would be offset by not sharing the market at later stages. Therefore, the entries might occur only in a number of last stages. Interestingly, this number depends only on the initial probability that the monopolist is strong, it is independent of the total number of stages played. Thus, the payoff the monopolist receives when the total number of stages grows approaches 3, the payoff corresponding to the monopolist's most preferred stage game outcome.

The idea behind this reasoning was generalized by Fudenberg and Levine (1989). It considers the setting with one long-run player facing an infinite sequence of short-run opponents and define so called Stackelberg payoff as the best payoff the long-run player could get by committing himself to a specific strategy (which is called Stackelberg strategy). The authors introduce then a new type of the long-run player for which playing Stackelberg strategy is the dominant strategy in the repeated game. Finally, the authors prove that under these conditions the limit of the long-run player's payoff in the repeated game when his discount factor δ approaches 1 (the long run player is sufficiently patient) is exactly his Stackelberg payoff. Applied to the above example, this means that a monopolist with a sufficiently high discount factor can always get his Stackelberg payoff (which equals 3) when playing against an infinite sequence of short-run entrants.

An underlying assumption in these models is that the short-run players are perfectly informed about the history of play. Or, put differently, they perfectly observe the long-run player's moves and report them honestly. In real-world online settings this assumption is not always satisfied. Much closer to the reality are the models with so called imperfect monitoring in which the players do not observe the past actions but only their imperfect signals. If the signals are common for all the players (public signals) then the monitoring is said to be public, otherwise it is private. Surprisingly (or not), this seemingly small change in the information structure of the repeated games makes an important difference in the tractability of these two classes of the models. While some convenient mathematical tools for analyzing the imperfect public monitoring games have been found (Abreu, Pearce, and Stachetti (1990)) and the long-run player's payoff characterization results have been obtained (Fudenberg and Levine (1992)), there is a lack of similar results of the imperfect private monitoring class of games. Instead, only specific games (mostly the Prisoners' Dilemma) have been analyzed and their equilibria derived. Kandori (2002) offers an introduction to the imperfect private monitoring games and explains why this is the case.

5.1 Why and why not game-theoretic modeling?

The example we just saw demonstrates how game theory formalizes the concept of reputation. However, it does not show how reputation helps promoting trustworthy behavior. We see two reasons for this.

First, the setting of the game was such that the socially desirable outcome (sharing the market) was impossible in the long run. Or, put another way, the Stackelberg outcome was socially undesirable. Quite often this is not the case. To see this, consider another example (adapted from Dellarocas (2003a)). Assume that a sequence of short-run player 2's, call them buyers, plays the stage game from Figure 3 against a long-run player 1, a seller. At each stage the buyer can bid high or low to buy an item the seller is about to sell.[6] Upon receiving the payment the seller can cooperate (deliver a high quality item) or cheat (deliver a low quality one).

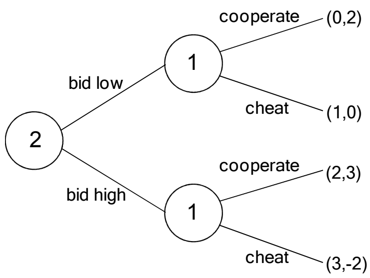

Figure 3: Auction Game

The problem with this game is precisely that the inefficient outcome (bid low, cheat) will happen at all stages. On the other hand, the efficient outcome (bid high, cooperate) is the Stackelberg outcome of the seller. Now, the reasoning from the previous section, applied to this example, reads: if the buyers assign a non-zero probability to the event that the seller is honest (will deliver high quality after receiving the payment) then the seller can develop reputation for being honest and the preferred outcome (bid high, cooperate) will be observed in most of the stages.

The second reason is the key one for understanding game-theoretic reputation system design as a systematic and analytic discipline. It happens, just as in the case of the game from Figure 2, that given a stage game representing the identified interaction in the considered community and the way the information about the past play is shared among the participants, no "good" equilibrium (say, one promoting trust or any other socially desirable outcome) actually exists. But there is a solution to this problem: by processing the raw feedback properly (choosing a proper feedback aggregation strategy) game-theoretic reputation systems designers can tune the equilibria set according to their wishes and, in the ideal case, narrow down this set to only one equilibrium, presumably the socially most desirable one. Needless to say, this is exactly the main purpose of the reputation mechanisms. The point is here that there is a clear link between the feedback aggregation strategy and the actual behavior of the participants, reflected in the chosen equilibrium. It is exactly this link that enables the mentioned systematics and analyticity.

With all this reasoning in mind we can derive quite precise steps in game-theoretically founded reputation mechanisms design:

[6] Think about an auction in which only the winning buyer is considered.

- Analyzing interaction patterns in the considered community and identification of the stage game.
- Defining of the feedback (past play) type along with a solution for incentivizing short-run players (if any) to provide it.
- Defining the way the feedback is aggregated.

In the rest of this section we will briefly describe how the current literature addresses the last two tasks. (We believe that the first task is clear in itself.)

Let us start with outlining how the game-theoretic models fit the general picture we introduced in Section 2 and what benefits they bring as compared to the other two classes of approaches, social networks and probabilistic estimation techniques. First, after the stage game has been identified, the feedback set W on the long-run player's behavior is easy to determine. It coincides with the set of possible stage game outcomes, or the signals if the stage game is with imperfect monitoring. The output set T can be understood as the action set of the considered long-run player's opponents. The algorithm A is just mapping from the feedback on the long-run player's past play to the set T representing the actually implemented equilibrium. It should be clear from this that the implementation overhead does not impose a problem with game-theoretic reputation models. Only first level reports are considered and the associated implementation cost are not considerable. Regarding the benefits, we know that with both social networks and probabilistic approaches the trustee's opportunism and the trustor's vulnerability are reduced but not completely removed. Thus, the decision making process based on the feedback is error-prone. The power of game-theoretic reputation systems, as based on an analytic discipline, is precisely that they eliminate this problem: given that the players are rational utility maximizers then they cannot do anything better than playing exactly what is specified by an equilibrium, whichever is actually implemented.

All the models discussed so far involved short-run players sharing information about the long-run player's actions among themselves. In economic models of online settings, where there is a cost associated with communicating this information (leaving feedback), this cannot be taken for granted. Another difficulty is that even if the short-run players are incentivized to leave feedback how can we make sure that this is done truthfully. Miller, Resnick, and Zeckhauser (2002) and Jurca and Faltings (2002) analyze exactly this problem and propose a payment-based system based on the application of proper scoring rules to the reports. They prove that honest reporting is a Nash equilibrium and that, on the other hand, the budget is balanced by charging involved players for payments that are based on the reports of others.

The last task in the above list is certainly the core one. Unfortunately, it turns out that in general it is not at all easy. Deriving reputational equilibria can be quite complicated even when the stage game has a simple structure, the reverse task of devising a game with specific equilibria is in principle even more complicated. This is particularly true for private imperfect monitoring games and, unfortunately, exactly this class of games is the most frequent in online settings because any aggregation of the feedback makes the monitoring imperfect, while in the models with one long-run player playing against a sequence one-shot opponents it becomes necessarily private. This is exactly why the game-theoretic reputational models of online environments are very rare.

Dellarocas (2003b) presents one such model in which an eBay-like setting with a long-run seller facing an infinite sequence of one-shot buyers was considered. In each round the seller sells an item that can be perceived on the buyers' side as being of high and low quality and the buyers compete to buy the item in an auction. The seller can exert high or low effort when delivering the item influencing thus the quality observations on the buyers' side. Both low and high quality perceptions of the buyers are possible under both high and low effort. The setting is clearly with private imperfect monitoring because the seller's actions (high and low effort) are not observed by the buyers. Instead, their imperfect signals (high and low quality) are. The feedback on the sellers past play then consists of the observed signals. As the feedback aggregation strategy the author proposes maintaining a fixed size window of the reports on the observed past qualities in which any new report replaces an old, randomly chosen, one. The sellers optimal strategy in this setting is to always exert high effort if he has zero low quality reports, otherwise he should follow a mixed strategy in which the probability of exerting high effort is a linearly decreasing function of the current number of low quality reports. However, it turns out that there is an efficiency loss in comparison with the case where seller can commit to exerting high effort. But there is no efficiency loss in comparison with the case where the entire feedback history of the buyer reports is publicly known. The author also shows that the whole mechanism is pretty robust in the presence of buyers' misreporting the seller's actions.

Apart from the mentioned complications that the game-theoretic modeling of reputation normally involves, there are also some very practical arguments against it. There is a concern about how well the described models approximate the real-world settings. For instance, is the model with one long-run player and many one shot opponents a good approximation of a typical eBay scenario in which one seller sells items to many buyers or this interaction cannot be considered in isolation from other interactions of the seller and the buyers? Some other concerns deal with the discounting criterion and the exact values of the players' discount factors. In the case we need them, how can we discover the discount factors or what happens if two or more (long-run) players have different factors. These practical issues are not usually studied in the literature.

Finally, there is a huge body of work proving that humans do not normally act as rational economic agents (Fehr and Gächter (2002), for example) raising the question of the practical usability of the game-theoretic modeling in general. More specifically, Bolton, Katok, and Ockenfels (2002) report on some peculiarities in how people treat reputation systems which can be again explained by their irrationality.

5.2 Game-Theoretic Reputation Models and P2P

In P2P networks the hard problem we just described becomes even considerably harder. An implicit assumption of the above discussion was that the feedback aggregation was done by a central authority (eBay, for example) that did not have any incentive to distort it in any other way. Unfortunately, this assumption is not valid in P2P networks where the aggregation can be performed only by the peers themselves. But then the peers may find it profitable to distort the feedback or simply to reenter the network under a different identifier and abuse their knowledge of the complete feedback in the interactions it covers. These possibilities must be considered a part of the peers' strategic choices. To the best of our knowledge, there is no game-theoretic model dealing with this problem.

Nevertheless, we believe that these models are worth investigation. The reason is, as explained above, that the question of whom to trust is clearly solved. Implementation overheads are also minimized because the feedback used in the computation is localized around long-run players.

6 Comparative Analysis

	Assumed peer behavior	Trust related model semantics	Trust related performance	Implementation overhead
Social Networks	Probabilistic	Unclear decision making	Robust to a wide range of misbehaviors	High; Low for approximate computations
Probabilistic Estimation	Probabilistic	Clear	Robust to a narrow range of misbehaviors	Low
Game-theoretic Models	Rational	Clear	No misbehavior possible	Low

Table 1: Main Properties of various Trust Management Approaches

Table 1 summarizes our findings about the approaches we discussed throughout the paper. Let us briefly comment on them.

Social networks target probabilistic behavior. One of the problems we identified with them is unclear decision making. Namely, they output values that are hard to interpret and it is not clear how precisely they contribute to building trust. On the other hand, social networks appear to be robust to a wide range of misbehaviors. Put differently, they do not require particular knowledge on the underlying peer behavior. This is mainly an implication of their aggregating of a large amount of reputation data that is available in the network. However, this also has an undesirable effect: high implementation overhead.

Probabilistic estimation methods start with the assumption that the underlying behavior is probabilistic and try to estimate the parameters of the peer behavior characterizing distributions. As such, they enable clean decision making. Normally, probabilistic models explore only a small fraction of the available feedback. Thus, they incur small implementation overhead, that is acceptable for P2P networks. However, they require more knowledge about the assumed peer behavior than social networks and are less robust.

Game-theoretic models are based on the assumption that the peers are rational utility maximizers. Thus, their behavior is fully determined by the equilibria of the underlying game they are playing. This is why we say that decision making is clean: equilibria precisely specify a move after any history of play. Due to the mentioned schemes that incentivize players to leave truthful feedback we say that no misbehavior is possible. Implementation overhead is acceptable because, just as in the case of probabilistic estimation, only a small portion of the available feedback is aggregated.

As for the application of the approaches, the exact characteristics of the target environment and the expected behavior of the peers determine the solution that should be chosen. Normally, if peers behave as rational, utility maximizing economic agents one would opt for a game-theoretic solution. However, where such an assumption is not valid, probabilistic or social network solutions should be used. Probabilistic techniques should be favored if no formation of large collusive groups is expected, otherwise a social network approach with the recommendation context included or, if the network is small, with centralized "pretrusted peers" based computation should be employed.

7 Conclusions and Research Issues

In this paper we reviewed the literature on trust and reputation management and established its classification that is primarily based on two dimensions we find important for the context of P2P systems: trust semantics and implementation overhead. There are a number of conclusions we can draw from the presented analysis.

First, reputation management does matter. Many empirical studies confirm that managing participants' reputations helps building trust in the concerned communities.

Second, there is a need for reputation management in many areas. We saw that besides P2P networks, reputation management was studied in such diverse contexts like the semantic web, centralized e-commerce settings, resource allocation in mobile networks etc.

Further, we identified many possibilities that differ in what type of the underlying behavior they target, what trust semantics with respect to decision making they offer and what implementation costs they impose. However, we believe that the research in this area is still in its infancy. This statement can be justified as follows.

Rational peer behavior requires game-theoretic modeling. But, the reputational game-theoretic models are extremely hard to analyze and so far no effective mathematical tool to deal with imperfect private monitoring has been devised. We believe that research in this area should go in this direction. As well, models considering feedback aggregation possibilities as an explicit strategy space constituent must get in the focus of the researches targeting P2P networks specifically.

Probabilistic behavior raises certain concerns as well. First of all, we are not aware of any confirmation that this type of behavior characterizes typical online communities and P2P networks in particular. This is where a tighter collaboration with social sciences is necessary. Even if this is the case there are still problems such as: unclear decision making and implementation overhead of the social networks and too strong limiting assumptions of the current work based on probabilistic estimation. We believe that devising new methods that will merge good properties of the two classes of approaches is what the future research should focus on.

APPENDIX

A P2P - A Technology Overview

The limitations of client-server based systems become evident in an Internet-scale distributed environment. Resources are concentrated on a small number of nodes, which must apply sophisticated load-balancing and fault-tolerance algorithms to provide continuous and reliable access. Additionally, network bandwidth must be increased steadily to handle requests to and from successful Internet servers. Caching and replication were introduced a posteriori to remedy these problems in a client-server setting when the World Wide Web, as the most successful Internet service, developed into a network bandwidth nightmare.

Peer-to-peer systems offer an alternative to traditional client-server systems for some application domains. In P2P systems, every node (peer) of the system acts as both client and server (servent) and contributes a part of the resources necessary to provide a service. The P2P approach circumvents many problems of client-server systems but requires considerably more complex mechanisms for node organization, resource location and security. A central problem for P2P systems is resource location without centralized control. Resources R required to provide a service, in our case, for example, providing reputation data on a specific peer, are distributed over peers and identified using application-specific identifiers chosen from a key space K. The problem to be solved is then the following: any peer participating in a P2P system should be enabled to locate a peer with (Internet) address $p \in P$, that holds a specific resource $r \in R$ that is identified by a key $k \in K$. In order to perform this task the peers maintain knowledge about some other peers (typically a small number) and can forward resource requests to them. From another perspective this amounts to constructing an application specific *overlay network* that allows to address resources by their application-specific keys, instead of using application-independent physical peer addresses. The problem is of how to realize such an overlay network efficiently, i.e. of how to realize basic operations of overlay network maintenance and routing (respectively search) with low consumption of physical resources, in particular network bandwidth.

This reasoning leads to the following abstract definition of P2P systems:

Definition A.1 (P2P System). A P2P system is a tuple (P, R, K, G, RA, MA), where P is a set of peers, R is a resource set to be distributed among the peers P, K is a set of keys used to identify resources, G is a directed graph (P, V) representing the overlay network, RA and MA are algorithms operating on the graph G and enabling resource lookup and network maintenance.

The algorithm RA is implemented by using a mapping $forward : P \times K \rightarrow 2^P$ such that $forward(p, k)$ is a subset of the peers reached by outgoing edges of p in G. The outdegree of the nodes in the graph G is usually but not necessarily low, since the purpose of the maintenance algorithm MA is to keep the outgoing links consistent in the presence of peer joins, leaves and failures.

Based on this abstract view two important classes of P2P systems can be distinguished.

- Unstructured P2P systems: In unstructured P2P systems the distribution of resources over the peers and the structure of the overlay network G are not correlated. Thus searches are performed in an exhaustive fashion, e.g. by using broadcasts. Examples are Gnutella (2001) and Lv, Cao, Cohen, Li, and Shenker (2002).

- Structured P2P systems: The distribution of resources of the peers and the graph structure are correlated. Thus searchs can be performed in a directed fashion, e.g. by greedy routing. Examples are FreeNet [Clarke, Miller, Hong, Sandberg, and Wiley (2002), Clarke, Sandberg, Wiley, and Hong (2000)], Chord [Dabek, Brunskill, Kaashoek, Karger, Morris, Stoica, and Balakrishnan (2001), Stoica, Morris, Karger, Kaashoek, and Balakrishnan (2001)], CAN [Ratnasamy, Francis, Handley, Karp, and Shenker (2001)], Pastry [Rowstron and Druschel (2001)], Tapestry [Rhea, Wells, Eaton, Geels, Zhao, Weatherspoon, and Kubiatowicz (2001)], Viceroy [Malkhi, Naor, and Ratajczak (2002)], Symphony [Manku, Bawa, and Raghavan (2003)] (based on the small world graphs theory of Kleinberg (2000)) and PGrid [Aberer, Punceva, Hauswirth, and Schmidt (2002), Aberer (2001)].

In the sequel we will adopt this division and give high level overviews of the two mentioned paradigms. To precisely evaluate each of them we will consider how they deal with the following questions.

- What is the structure of the overlay network?

- How efficient and flexible is search in the overlay network?

- How is the overlay network maintained efficiently and reliably?

- How autonomuous are the peers?

For a more detailed analysis of these questions as well as for analyzing other important properties, such as load balancing, security and proximity-based routing, just to name a few, a number of overview papers have been recently appearing or are about to appear [Gummadi, Gummadi, Gribble, Ratnasamy, Shenker, and Stoica (2003), Manku (2003), Aberer and Hauswirth (2004)]. It also important to recognize that lookup of $(key, value)$ pairs, as needed for reputation-based trust management, is only one among many possible applications of P2P systems. Others that have recently been discussed in the literature are publish-subscribe systems, data broadcast, and data aggregation.

A.1 Unstructured P2P Networks

As a distinctive property of unstructured P2P systems, as compared to structured P2P systems, peers do not commit or restrict themselves to manage any specific type of resource, since the distribution of resources over the peers and the structure of the overlay network G are not correlated. Thus any peer could hold any resource from R. In order to obtain access to resources managed by other peers, peers use the graph G to implement a request forwarding scheme that has to be designed such that it eventually reaches any peer in the P2P network.

Definition A.2 (Unstructured P2P System). An unstructured P2P system is a P2P system in which the distribution of the resources R among the peers and the structure of the overlay network G are independent.

Example A.1. An early example of an unstructured P2P network has been Gnutella. In Gnutella each peer is connected to a constant number of neighbors, a typical value being 4. Resource requests are flooded through the whole network, by forwarding them recursively to all known neighbors. However, in order to control the number of messages, requests are only forwarded a bounded number of times, known as the time-to-live (TTL) and a typical value being 7. Request messages carry identifiers such that requests returning along a cycle to the same peer can be dropped. The number of messages generated by broadcast is thus linear in the size of the network. However, it is important to realize that the search latency depends on the diameter of the overlay network, which is typically logarithmic in the number of nodes, such as for random graphs and small world graphs. For constructing and maintaining an unstructured network Gnutella uses a similar mechanism. Peers joining the network flood the network with a discovery message. From the responding peers the joining peer then selects its local neighbors. If peers fail to operate they are dynamically replaced from a list of known peers in the network. It has been shown that this mechanism leads approximately to a power law distribution of incoming links, as peers tend to attach preferentially to stable peers. Similar behaviors have been discovered for many self-organizing networks, including the web.

Taking advantage of the emergent structure of Gnutella-like P2P networks, there have been several attempts to improve search cost, measured in number of messages, by optimizing the request forwarding scheme. These approaches include the use of random walkers, where a depth-first search strategy is used, and taking advantage of results from percolation theory, to precisely estimate the number of links that need to be traversed in order to reach the whole network. Also a number of replication and caching strategies have been studied in order to improve search performance. However, the fundamental problem of lack of knowledge which peer is managing which resource makes search in unstructured P2P networks inherently expensive in terms of communication overhead.

On the other hand unstructured P2P networks are very efficient in terms of updates to the resources and maintenance of network structure since there exist no dependencies among peers and among peers and resources and thus these operations can be performed by the peers autonomously. Also, since routing of resource requests does not depend on the type of request, unstructured P2P networks can be easily employed to handle complex requests, such as complex queries.

A.2 Structured P2P Networks

In structured P2P networks peers commit to manage a specific type of resource, since the distribution of resources over the peers and the structure of the overlay network G are correlated. The correlation among peers and resources is established by associating the peers with a key taken from key space K and to associate with this key a partition of the key space such that the peer becomes responsible to manage all resources identified by keys from the associated partition. Typically the key partition consists of all keys closest to the peer key in a suitable metric. Thus in structured P2P systems the key space K is equipped with a distance function d. For forwarding resource requests peers form a routing network by taking into account the knowledge on the association of peers with key partitions.

Thus we have the following definition:

Definition A.3 (Structured P2P System). A structured P2P system is a P2P system in which the distribution of the resources R among the peers and the overlay network G are correlated. The correlation is present through the following three functions: a function $key : P \to K$ that associates peers with keys and, given $key(P)$, a function $partition : K \to 2^K$ associating peers with partitions of K and a function $neighbors : K \to 2^P$ that associates peers with their neighbors in graph G.

The function *neighbors* can be either deterministic or randomized, which leads to the further distinction among non-randomized and randomized structured P2P networks. Using the metric of the key space, typically peers maintain short-range links to all peers with neighboring keys and in addition long-range links to some selected peers, where the probability of having a long-range link to a peer decreases with the distance to the peer's key. The standard approach that has been used in most of the structured P2P networks is to choose long-range links with exponentially decreasing probability depending on the distance. Using the routing network peers then forward resource requests in a directed manner to the closest peers that they know from their routing table. The standard structured P2P overlay networks achieve by virtue of this construction lookup with a number of messages logarithmic in the size of network by using routing tables which are also logarithmic in the size of the network. However, there are also some works that achieve constant outdegree graph topologies and consequently constant sized routing tables, e.g Viceroy Malkhi, Naor, and Ratajczak (2002).

The specific designs of these structures, frequently termed as distributed hash tables, depend on the choice of key space, key partitioning, and linking strategy. They have been subject of intensive research over the recent years and resulted in numerous designs of structured overlay networks. We summarize in Table 2 the main properties of some representative solutions that have been proposed.

Deterministic Routing Topologies				
P2P System	Keys	Topology	# Links	Avg. Latency
CAN	d-dimensional	Torus	$O(d)$	$O(n^{1/d})$
Chord	Fixed-length binary strings	Hypercube	$O(\ln n)$	$O(\ln n)$
Pastry, Tapestry	Fixed-length strings	Hypercube	$O(\ln n)$	$O(\ln n)$
Viceroy	Fixed-length binary strings	Hypercube	$O(1)$	$O(\ln n)$
Randomized Routing Topologies				
P2P System	Keys	Topology	# Links	Avg. Latency
Symphony	d-dimensional	Grid	$O(d)$	$O(\ln^2 n)$
P-Grid	Variable-length binary strings	Binary trie	$O(\ln n)$	$O(\ln n)$

Table 2: Main properties of various structured P2P systems

For constructing and maintaining a structured P2P network peers have to deal in particular with the problem of node joins and failures. Since the freedom to choose neighbors in a structured P2P network is constrained by the conditions imposed by the function $neighbors$, maintenance algorithms are required to re-establish the consistency of routing tables in the presence of network dynamics. Depending on the type of guarantees given by the network different deterministic and probabilistic maintenance strategies have been developed. Maintenance actions can be triggered by various events, such as periodical node joins and leaves or routing failures due to inconsistent routing tables. The different maintenance strategies trade-off maintenance cost versus degree of consistency and thus failure resilience of the network.

Despite the apparent advantages of structured P2P networks in terms of reducing network bandwidth consumption for resource location one has to understand that this comes at various costs. First, dependencies among peers are introduced and network maintenance is required as has been described before. The main reason to keep the sizes of routing tables small is actually not the required storage overhead, but rather the maintenance cost induced by having large routing tables. Second, as opposed to unstructured P2P networks, changes to the resources themselves, i.e. insertions, updates and deletions, are not localized to the peers that maintain the resources, but affect other peers as well. Third, resource request are constrained to simple key-based lookups. Even slight generalizations, such as requesting ranges of keys require non-trivial algorithms for implementation. From another perspective, in structured P2P networks the peer autonomy is more constrained as in unstructured P2P networks. This is the result of requiring commitments of peers to manage specific resources.

B Game Theory Overview

B.1 Game Theory Basics

Game theory describes mathematical models of conflicting and cooperative interactions between rational, utility maximizing, decision makers (players in the parlance of game theory). The presence of interaction is the most important ingredient of this definition - the utilities of the players are affected not only by their own strategic choices but also by those of all other players as well.

In this text we will concentrate on, so called, *strategic games*, in which players choose their actions simultaneously, at the same time.

Definition B.1 (Strategic game). A *strategic game* consists of: a set of players $N = \{1, 2, \dots, n\}$; for each player i, a pure strategy (or action) set A_i and a utility function $u_i : A \to \mathbb{R}$, where $A = \times_{i \in N} A_i$.

The definition is illustrated in the following example.

Example B.1 (Stag hunt game). Consider a situation two hunters might face. Assume that they have the possibilities to hunt a stag (together) or hares (separately). Thus we have that the set of players is two-element set $N = \{1, 2\}$ and that the players' strategy sets are $A_1 = A_2 = \{\mathsf{Stag}, \mathsf{Hare}\}$. To have a fully defined game we need also to define the players' utility functions on the set $A_1 \times A_2$. They can be neatly presented by a table, as done in Table 3. Each cell in the table contains two entries corresponding to the utilities players 1 and 2 respectively receive when the corresponding combination of actions has been chosen.

The above definition also introduces some notational conventions we will be using. Apart from the symbols given there we use the symbol $\Delta(B)$ to denote the set of all probability distribution over the set B. For any player i the set $\mathscr{A}_i = \Delta(A_i)$ will be called the set of player i's *mixed* strategies. The set of all mixed strategy combinations, $\times_{i \in N} \Delta(A_i)$ will be denoted $\mathscr{A}$. Any mixed strategy profile $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathscr{A}$ induces a probability distribution over

		Player 2	
		Stag	Hare
Player 1	Stag	2, 2	0, 1
	Hare	1, 0	1, 1

Table 3: Two-player Stag Hunt game.

the set of all pure strategy profiles A in the obvious way. For any such mixed strategy profile α we define $u_i(\alpha)$ as the expected value of player i's payoff with respect to this distribution. Also, any strategy profile $(a_1, \ldots, a_i, \ldots, a_n)$ will be denoted (a_i, a_{-i}) when we want to emphasize that player i uses action a_i.

We now ask the question: what strategies maximize the players' utilities? This questions is central to game theory and the solution concept game theorists prescribe is that of *Nash equlibrium*, formally defined as follows.

Definition B.2 (Nash equilibrium). A strategy profile $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*) \in \mathscr{A}$ is a Nash equilibrium if no player can gain by unilateraly deviating. So, denoting $\alpha^* = (\alpha_i^*, \alpha_{-i}^*)$, we have that for any player i and any of her strategies $\alpha_i \neq \alpha_i^*$ the following must be satisfied:

$$u_i(\alpha_i^*, \alpha_{-i}^*) \geq u_i(\alpha_i, \alpha_{-i}^*) \tag{B.1}$$

By inspecting Table 3 we can see that the profiles (Stag, Stag) and (Hare, Hare) are Nash equlibria of the stag hunt game. For instance, given that one player plays Stag the other player cannot increase his utility by switching to any randomization between Stag and Hare that puts a positive probability on Hare. But, apart from these two there is another equilibrium involving mixed strategies, in which both players randomize between their two pure strategies with probabilities of 0.5.

The question of the equilibria existence is resolved by the following theorem.

Theorem B.1. For any finite game there is at least one Nash equilibrium in $\mathscr{A} = \times_{i \in N} \mathscr{A}_i$.

B.1.1 Bayesian Games

In many games of interest different players may have different information about some important aspects of the game. To model such situations the concept of *Bayesian games* (or games with incomplete information), introduced by Harsanyi (1968), has been widely adopted. Informally, central to Bayesian games are *types* of the players by which the players' private information is modeled. Each player learns his own type at the beginning but the types of other players remain unknown to him. The utility functions now specify ordinal payoffs for combinations of chosen strategies and realized types rather than strategies only. Another important assumption is that the types are drawn from a common prior probability distribution, known to and agreed upon by all the players, so that every player can derive the probability distributions of the combinations of the other players' types given any of his own types. Then, the goal of every player is to maximize his payoff for any of his types. The concept of Bayesian game and its equlibria are defined formally as follows.

Definition B.3 (Bayesian game). A *Bayesian game* consists of a set of players $N = \{1, 2, \ldots, n\}$ and for each player i: a set of possible actions A_i and a set of possible types T_i, a probability function $p_i : T_i \to \Delta(T_{-i})$, where $T_{-i} = \times_{j \in N-i} T_j$ and a utility function $u_i : A \times T \to \mathbb{R}$, where $A = \times_{i \in N} A_i$ and $T = \times_{i \in N} T_i$.

Definition B.4. Any strategy profile $\sigma^* \in \times_{i \in N} \times_{t_i \in T_i} \Delta(A_i)$ is a Bayesian equlibrium if for any type t_i of any player i mixed strategy $\sigma^*(a_i|t_i)$ optimizes player i's expected payoff where the expectation is taken over all combinations of types of the other players.

Example B.2 (First-price sealed-bid auction). Auctions present a typical example of Bayesian games. In this example we will describe the first price sealed-bid auction. Let us assume that n bidders are competing to buy an auctioned item by submitting sealed bids. The owner of highest bid wins and pays the price of his bid. Assume also that any bidder has a private valuation of the item, denoted v_i, and that each player's valuation is independent of those of the other players. However, the distribution from which the valuations are drawn are known. Here we assume the uniform distribution on the interval $[0, 1]$. So, it should be obvious that this makes a Bayesian game in which the item valuations are the players' types and all distributions p_i for any player i are uniform on $[0, 1]$. The utility functions are such that for any player i and any valuation-bid tuple $(v, b) = (v_1, \ldots, v_n, b_1, \ldots, b_n)$

$$u_i(v, b) = \begin{cases} v_i - b_i & \text{if } b_i = \max(b_1, \ldots, b_n) \\ 0 & \text{otherwise.} \end{cases}$$

It can be shown that in this setting the optimal bid for any bidder i with valuation v_i is $\frac{n-1}{n} v_i$. In other words, the equilibrium is made of the set of n functions $B(v) = \frac{n-1}{n} v$ for n bidders and $v \in [0, 1]$.

B.2 Repeated Games

To model repeated interactions game theorists use the concept of repeated games in which the same stage game is repeated finite or infinite number of times whereby the sets of players who participate in the stages can vary from stage to stage. In this section our primary concern will be infinitely repeated games with discounting, in which the stage game is repeated infinitely many times and the players discount their future payoffs as compared to those received at present. Our main goal will be to show that repeated play can differ greatly from one-shot encounters in the sense that it can allow for a whole range of equilibria which are not normally found in the constituent, one-shot games.

To define a repeated game and its equilibria we need to define the players' strategy sets and payoffs for the entire repeated game given the strategies and payoffs of its constituent stage game. Therefore, assume that the stage game is an n-player strategic form game and that the action set of each player i, A_i, is finite. Let The u_i denote the stage game payff of player i. Assuming that the players observe each other's realized pure strategies at the end of each stage we can proceed as follows: let $a^t = (a_1^t, \ldots, a_n^t)$ be the pure strategy profile realized in period t and $h^t = (a^0, \ldots, a^{t-1})$ the history of these profiles for all periods before t. Let, finally, H^t denote the set of all such period-t histories. Then a period-t mixed strategy of player i in the repeated game is any mapping $\sigma_i^t : H^t \to \mathscr{A}_i$. A mixed strategy player i for the whole repeated game is then a sequence of such maps $\sigma_i = \{\sigma_i^t\}_{t=1}^{\infty}$.

To define the repeated game payoffs we must have a way to compare, possibly infinite, sequences of the stage game payoffs. In this text we concentrate on so called discounting criterion for this purpose. So if u_i^t is the payoff player i receives in the period t then the δ-discounted $(0 \leq \delta < 1)$ payoff of this player is defined as:

$$g_i = (1 - \delta) \sum_{t=0}^{\infty} \delta^t u_i^t \tag{B.2}$$

The reasoning behind this definition is that the players see future gains as less valuable than those of the present or that there is uncertainty regarding the ending time of the game with $1-\delta$ being the probability that the next stage will be the last. For obvious reasons, a repeated game in which the players discount their future payoffs with a common discounting factor δ will be denoted $G(\delta)$.

Any strategy profile $\sigma = (\{\sigma_1^t\}_{t=1}^{\infty}, \ldots, \{\sigma_N^t\}_{t=1}^{\infty})$ induces a probability distribution over the set of all infinite histories. We define the players' payoffs in the repeated game as the expected values of δ-discounted payoffs the players receive when the paths of play follow each of these histories. With this clarification we finally define that a Nash equilibrium of the repeated game $G(\delta)$ is every strategy profile $(\{\sigma_1^t\}_{t=1}^{\infty}, \ldots, \{\sigma_N^t\}_{t=1}^{\infty})$ such that no player can gain by switching to a different strategy given that all other players implement it.

An important problem in the theory of repeated games is what payoff sets can be supported in equilibria of the repeated game given the structure of its constituent, stage game. Intuitively, because the players can condition their future play on the past play of their opponents and retaliate if the opponents do not play in a specific way, many payoff allocations, not supportable in the one shot game, might become supported in an equilibrium of the repeated game provided the future is not discounted too much. The theorems formalizing this intuition are known in the literature as *folk theorems*. As seen from the perspective of engineering of online reputation mechanisms, they are important as the starting points determining what outcomes are feasible for a given mechanism. If socially desirable outcomes are not among the feasible ones then the mechanism must be redesigned as to include them.

We will state here a folk theorem corresponding to the most commonly studied type of repeated interactions: the one with all players participating in all stages of the game (hereafter such players will be termed long-run). To state it precisely we need to introduce several notions. We define the *minmax value* of player i to be

$$\underline{v}_i = \min_{\alpha_{-i} \in \times_{j \in N-i} \mathscr{A}_j} \max_{\alpha_i \in \mathscr{A}_i} u_i(\alpha_i, \alpha_{-i}). \tag{B.3}$$

Thus, player i's minmax value is the lowest payoff he can achieve in the stage game provided he correctly foresees the choice of actions of his opponents. Any payoff $v_i > \underline{v}_i$ is called individually rational for player i.

Further, putting $u(a) = (u_1(a), \dots, u_n(a))$, we can define the set of feasible payoffs as

$$V = \text{convex hull}\{v | \exists a \in A \text{ s.t. } u(a) = v\}. \tag{B.4}$$

Theorem B.2 (Fudenberg and Maskin (1986)). For any feasible payoff allocation v that is individually rational for all players there is a $\underline{\delta}$ such that for all $\delta \in (\underline{\delta}, 1)$ there is a Nash equilibrium of the repeated game $G(\delta)$ with payoffs v.

Example B.3. As an example, let us apply this theorem to the Prisoner's Dilemma game, shown in Table 4. It is easy to see that the minmax values of both players are $\underline{v}_1 = \underline{v}_2 = 1$ and that the payoff allocation $(5, 5)$ is in the set V. Thus, we can conclude that the socially most desirable outcome of both players cooperating in every stage of the repeated game is possible in long term interactions provided the players are sufficiently patient (their discount factors are sufficiently close to 1). It is also easy to construct such a strategy profile that results in this cooperative equilibrium. For instance, if the players implement the strategy "cooperate as long as no player cheated in the past stages, otherwise cheat" then no player has incentive to deviate if his short-term gain from the deviation is less then his long-term gain from cooperation. This will be the case if receiving 5 utils in every stage is better than receiving 6 today and 1 in all future stages, or if $5 \geq (1 - \delta)6 + \delta$, which implies $\delta \geq \frac{1}{5}$. For an elaborate exposition of this phenomenon and its empirical confirmations we refer to Axelrod (1984).

We stress that similar results obtain for other types of repeated interactions in which some players may participate at specific stages only, as opposed to the participation in the whole repeated game. Of particular importance for application to online world is the special case with one long-run player facing a (possibly infinite) sequence of short-run players each of whom plays the stage game only once as opposed to the long-run player who stays in the game till its end. This model closely approximates a typical online setting, P2P in particular,

	Cooperate	Cheat
Cooperate	5, 5	0, 6
Cheat	6, 0	1, 1

Table 4: Payoff Matrix of the Prisoner's Dilemma Game

in which a large number of participants makes multiple interactions between the same players highly improbable. However, for the lack of space, we will not cite here the folk theorem corresponding to this setting. Instead, we point the interested readers to Fudenberg, Kreps, and Maskin (1990).

References

Aberer, K. (2001): "P-Grid: A self-organizing access structure for P2P information systems," in *Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS 2001)*, Trento, Italy.

Aberer, K., and Z. Despotovic (2001): "Managing Trust in a Peer-2-Peer Information System," in *Proc. of the IX International Conference on Information and Knowledge Management*, Atlanta, Georgia.

Aberer, K., and M. Hauswirth (2004): "Peer-to-Peer Systems," in *The Practical Handbook of Internet Computing*, ed. by M. P. Singh. CRC Press, Forthcoming.

Aberer, K., M. Punceva, M. Hauswirth, and R. Schmidt (2002): "Improving Data Access in P2P Systems," *IEEE Internet Computing*, 6(1), 58–67.

Abreu, D., D. Pearce, and E. Stachetti (1990): "Toward a Theory of Discounted Repeated Games with Imperfect Monitoring," *Econometrica*, 58, 1041–1064.

Axelrod, R. (1984): *The Evolution of Cooperation.* Basic Books, New York.

Beth, T., M. Borcherding, and B. Klein (1994): "Valuation of Trust in Open Networks," in *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, pp. 3–18, Brighton, UK. Springer-Verlag.

Bolton, G., E. Katok, and A. Ockenfels (2002): "How effective are online reputation mechanisms? An experimental investigation," Discussion paper 25, 2002, Max Planck Institute for Research into Economic Systems, Germany.

Buchegger, S., and J. Y. Le Boudec (2003): "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks," in *Proc. of WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France.

Clarke, I., S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley (2002): "Protecting Free Expression Online with Freenet," *IEEE Internet Computing*, 6(1), 40–49.

Clarke, I., O. Sandberg, B. Wiley, and T. W. Hong (2000): "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in *ICSI Workshop on Design Issues in Anonymity, Berkeley, CA*.

Dabek, F., E. Brunskill, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan (2001): "Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service," in *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, pp. 81–86.

Dellarocas, C. (2000): "Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior," in *Proceedings of the 2nd ACM conference on Electronic Commerce*, Minneapolis, USA.

——— (2003a): "The Digitization of Word-of-Mouth: Promise and Challenges of Online Feedback Mechanisms," Working paper, 4296-03, MIT Sloan School of Management.

——— (2003b): "Efficiency and Robustness of Binary Feedback Mechanisms in Trading Environments with Moral Hazard," Working paper 4297-03, MIT.

Despotovic, Z., and K. Aberer (2002): "Trust-Aware Delivery of Composite Goods," in *International Workshop on Agents and Peer-To-Peer Computing*, Bologna, Italy.

——— (2004): "Maximum Likelihood Estimation of Peers' Performances in P2P Networks," in *2nd Workshop on the Economics of Peer-to-Peer Systems*, Cambridge, MA, USA. Available at http://www.eecs.harvard.edu/p2pecon/.

Fehr, E., and S. Gächter (2002): "Altruistic punishment in humans," *Nature*, 14, 137–140.

Fudenberg, D., D. Kreps, and E. Maskin (1990): "Repeated Games with Long-run and Short-run Players," *Review of Economic Studies*, 57, 555–574.

Fudenberg, D., and D. Levine (1989): "Reputation and Equilibrium Selection in Games with a Patient Player," *Econometrica*, 57(4), 759–778.

——— (1992): "Maintaining a Reputation when Strategies are Imperfectly Observed," *Review of Economic Studies*, 59, 561–579.

FUDENBERG, D., AND E. MASKIN (1986): "The Folk Theorem in Repeated Games with Discounting or Incomplete Information," *Econometrica*, 54, 533–556.

GNUTELLA (2001): "Clip2. The Gnutella Protocol Specification v0.4 (Document Revision 1.2)," http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

GUMMADI, P. K., R. GUMMADI, S. D. GRIBBLE, S. RATNASAMY, S. SHENKER, AND I. STOICA (2003): "The impact of DHT routing geometry on resilience and proximity," in *SIGCOMM 2003*, Karlsruhe, Germany.

HARSANYI, J. (1968): "Games with Incomplete Information Played by 'Bayesian' Players," *Management Science*, 14, 159–182, 320–334, 486–502.

HOUSER, D., AND J. WOODERS (2001): "Reputation in Auctions: Theory and Evidence from eBay," Working paper, University of Arizona.

JURCA, R., AND B. FALTINGS (2002): "Towards Incentive-Compatible Reputation Management," in *AAMAS 2002 Workshop on Deception, Fraud and Trust in Agent Societies*, Bologna, Italy.

KAMVAR, S. D., M. T. SCHLOSSER, AND H. GARCIA-MOLINA (2003): "EigenRep: Reputation Management in P2P Networks," in *Proceedings of the World Wide Web Conference*, Budapest, Hungary.

KANDORI, M. (2002): "Introduction to Repeated Games with Private Monitoring," *Journal of Economic Theory*, 102, 1–15.

KLEINBERG, J. (2000): "The Small-World Phenomenon: An Algorithmic Perspective," in *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC 200)*, pp. 163–170.

KREPS, D., AND R. WILSON (1982): "Reputation and Imperfect Information," *Journal of Economic Theory*, 27, 253–279.

LV, Q., P. CAO, E. COHEN, K. LI, AND S. SHENKER (2002): "Search and replication in unstructured peer-to-peer networks," in *International Conference on Supercomputing*, pp. 84–95, New York, USA.

MALKHI, D., M. NAOR, AND D. RATAJCZAK (2002): "Viceroy: A scalable and dynamic emulation of the butterfly.," in *Proc. 21nd ACM Symposium on Principles of Distributed Computing, PODC 2002*, pp. 183–192, Monterey, CA, USA.

MANKU, G. S. (2003): "Routing networks for distributed hash tables," in *Proc. 22nd ACM Symposium on Principles of Distributed Computing, PODC 2003*, pp. 133–142, Boston, MA, USA.

Manku, G. S., M. Bawa, and P. Raghavan (2003): "Symphony: Distributed Hashing in a Small World," in *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)*, Seattle, WA, USA.

Melnik, M. I., and J. Alm (2002): "Does a Seller's Ecommerce Reputation Matter? Evidence from eBay Auctions," *Journal of Industrial Economics*, 50(3), 337–349.

Miller, N., P. Resnick, and R. Zeckhauser (2002): "Eliciting Honest Feedback in Electronic Markets," Working paper, SITE02 workshop, Available at: http://www.si.umich.edu/ presnick/papers/elicit/.

Mui, L., M. Mohtashemi, and A. Halberstadt (2002): "A Computational Model of Trust and Reputation," in *Proceedins of the 35th Hawaii International Conference on System Science (HICSS)*, Hawaii, USA.

Page, L., S. Brin, R. Motwani, and T. Winograd (1998): "The PageRank Citation Ranking: Bringing Order to the Web," Discussion paper, Stanford University, Stanford, CA.

Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Shenker (2001): "A Scalable Content-Addressable Network," in *Proceedings of ACM SIGCOMM '01*, pp. 161–172.

Resnick, P., and R. Zeckhauser (2002): "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," in *The Economics of the Internet and E-Commerce*, ed. by M. R. Baye, vol. 11 of *Advances in Applied Microeconomics*. Amsterdam, Elsevier Science.

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

Rhea, S., C. Wells, P. R. Eaton, D. Geels, B. Y. Zhao, H. Weatherspoon, and J. Kubiatowicz (2001): "Maintenance-Free Global Data Storage," *IEEE Internet Computing*, 5(5), 40–49.

Richardson, M., R. Agrawal, and P. Domingos (2003): "Trust Management for the Semantic Web," in *Proceedings of the Second International Semantic Web Conference*, pp. 351–368, Sanibel Island, FL.

Rowstron, A. I. T., and P. Druschel (2001): "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *ACM International Conference on Distributed Systems Platforms (Middleware)*, pp. 329–350.

Sandholm, T. W. (1996): "Negotiation Among Self-Interested Computationally Limited Agents," Ph.D. thesis, University of Massachusetts.

Stoica, I., R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan (2001): "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications," in *Proceedings of the 2001 ACM SIGCOMM Conference*, pp. 149–160.

Usunier, J.-C. (2001): "Trust Management in Computer Information Systems," Working paper, IUMI, HEC, University of Lausanne, Switzerland.

Xiong, L., and L. Liu (2004): "PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities," *IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Issue on Peer-to-Peer Based Data Management*, 16(7), 843–857.

Yu, B., and M. P. Singh (2000): "A Social Mechanism of Reputation Management in Electronic Communities," in *Proceedings of the 4th International Workshop on Cooperative Information Agents (CIA)*, pp. 154–165, Boston, USA.

Zacharia, G., A. Moukas, and P. Maes (1999): "Collaborative reputation mechanisms in electronic marketplaces," in *Proceedings of 32nd Hawaii International Conference on System Sciences*, Hawaii, USA.