Taxonomy of Trust: Categorizing P2P Reputation Systems [⋆]

Sergio Marti and Hector Garcia-Molina

Stanford University
{*smarti, hector*}*cs.stanford.edu*

Abstract

The field of peer-to-peer reputation systems has exploded in the last few years. Our goal is to organize existing ideas and work to facilitate system design. We present a taxonomy of reputation system components, their properties, and discuss how user behavior and technical constraints can conflict. In our discussion, we describe research that exemplifies compromises made to deliver a useable, implementable system.

Key words: Trust, reputation, peer-to-peer, security

1 Introduction

The development of any complex computer architecture can be a challenge. This is especially true of a complex distributed algorithm that is run by autonomous untrusted agents, yet is expected to be relatively reliable, efficient, and secure. Such is the task of designing a complete reputation system for use in peer-to-peer networks. To accomplish the task, it is necessary to break down the problem into separate simpler problems of constructing a mechanism that provides a specific set of functions or properties, allowing developers to "divide and conquer" the problem of reputation system design.

Our primary goal is to provide a useful taxonomy of the field of peer-to-peer reputation design. To accomplish this goal, we identify the three basic components of a reputation system, break them down into the necessary separate mechanisms, and categorize properties we feel the mechanisms need to provide

[⋆] This research is supported in part by NSF Grant (IIS-9817799).

Table 1
Breakdown of Reputation System Components

Reputation Systems		
Information Gathering	Scoring and Ranking	Response
Identity Scheme	Good vs. Bad Behavior	Incentives
Info. Sources	Quantity vs. Quality	Punishment
Info. Aggregation	Time-dependence	
Stranger Policy	Selection Threshold	
	Peer Selection	

in order for the reputation system to fulfill its function. For each mechanism we list possible design choices proposed by the research community.

In the process, we give examples of research in the area of trust and reputation. A variety of research papers and implementations are referenced to illustrate ideas and provide the reader avenues for further investigation. We often draw on work done by the Peers research group [1] at Stanford University and do not pretend to produce a complete survey of the research area. We feel this overview will be of particular interest to those who are unfamiliar with the breadth of issues relating to reputation system design for peer-to-peer networks.

Taxonomies related to trust and reputation systems (either in part or as a whole) have been proposed by others (e.g O'Hara et al. [2]) and will be discussed in the text when appropriate.

1.1 Taxonomy Overview

The following section defines terms we use throughout the paper. We begin our taxonomy by classifying the assumptions and constraints of the system in Section 3. These assumptions include expected user behavior, as well as the goals of adversaries in the system and their capabilities. How effectively a reputation system can deal with adversaries may be constrained by the the technical limitations imposed on the implementation by the target system environment. These issues determine the necessary properties and powers of the reputation system.

Next, we break down the functionality of a reputation system into the three components shown in Table 1. In general, a reputation system assists agents in choosing a reliable peer (if possible) to transact with when one or more have offered the agent a service or resource. To provide this function, a reputation system collects information on the transactional behavior of each peer (information gathering), scores and ranks the peers based on expected reliability (scoring and ranking), and allows the system to take action against malicious peers while rewarding contributors (response). Each component requires separate system mechanisms (listed in Table 1). For each mechanism we study the possible desired properties and then discuss the implementation limitations and trade-offs that may prevent some of the properties from being met. In the discussion we will reference existing solutions or research to illustrate how different mechanism designs achieve certain properties within the given system constraints.

The three functionalities, gathering, scoring and response are covered in turn in Sections 4, 5 and 6.

2 Terms and Definitions

Before discussing the various taxonomies we would like to define certain terms we will be using throughout the paper:

Transactions Peer-to-peer systems are defined by interactions between autonomous agents or peers. These interactions may include swapping files, storing data, answering queries, or remote CPU usage. In addition, money may be exchanged when purchasing the desired resource. We refer to all interactions in general as *transactions* between two parties.

Cooperate/Defect When well-behaved peers carry out transactions correctly, we say they *cooperate*. Bad peers, however, may at times attempt to cheat or defraud another peer, in which case they *defect* on the transaction. We will use these terms (when applicable) when discussing general system/peer behavior.

Structured vs Unstructured P2P network architectures tend to be categorized as either structured or unstructured, depending on how the overlay topology is formed. Structured networks use a specific protocol to assign network IDs and establish links to new peers and are exemplified by the class of systems called *Distributed Hash Tables (DHTs)* (e.g. [3–5]). In purely unstructured topologies new users connect randomly to other peers. A hybrid approach is to assign certain peers as supernodes (or ultrapeers) that form an unstructured network and all peers connect to supernodes. Such organization is used in most popular file-sharing systems (e.g. [6,7]). However, for simplicity, we will classify supernode networks as unstructured networks [8].

Strangers Peers that appear to be new to the system. They have not interacted with other peers and therefore no trust information is available.

Adversary A general term we use to apply to agents that wish to harm other peers or the system, or act in ways contrary to "acceptable" behavior. This may include accessing restricted information, corrupting data, maliciously attacking other nodes in the network, or attempting to take down the system services.

3 Assumptions and Constraints

The driving force behind reputation system design is providing a service that severely mitigates misbehavior while imposing a minimal cost on the well-behaved users. To that end, it is important to understand the requirements imposed on system design by each of the following: the behavior and expectations of typical good users, the goals and attacks of adversaries, and the technical limitations resulting from the environment where the system is deployed. We discuss each of these here. The choices made here will impact the necessary mechanism properties discussed in Sections 4, 5, and 6.

3.1 User Behavior

A system designer must build a system that is accessible to its intended users, provides the level of functionality they require and does not hinder or burden them to the point of driving them away. Therefore, it is important to anticipate any allowable user behavior and meet their needs, regardless of added system complexity.

Examples of user behavior and requirements that affect distributed mechanism design include:

Node churn The rate at which peers enter and leave the network, as well as how gracefully they disconnect, affects many areas from network routing to content availability. Higher levels of churn require increased data replication, redundant routing paths, and topology repair protocols [9].

Reliability For most applications, users require certain guarantees on the reliability or availability of system services. For example, a distributed data storage application would want to guarantee that data stored by a user will always be available to the user with high probability and that it will persist in the network (even if temporarily offline) with a much higher probability [10]. The situation is more difficult in peer-to-peer networks where adversaries are actively attempting to corrupt the content peers provide. Group auditing techniques may help detect or prevent data loss [11].

Privacy Along with reliability, users that store data in an untrusted distributed system would also want to protect the content from being accessed by unauthorized users. One solution is to encrypt all data before storing [10].

However, in some applications access to unencrypted data is necessary for processing. Separating sensitive data from subject identities, or using legally binding strict privacy policies may be sufficient [12–14].

Anonymity As a specific application of privacy, users may only be willing to participate if a certain amount of anonymity is guaranteed. This may vary from no anonymity requirements, to hiding real-world identity behind a pseudonym, to requiring that an agent's actions be completely disconnected from both his real-world identity and his other actions. Obviously, a reputation system would be infeasible under the last requirement.

3.2 Threat Model

The two primary types of adversaries in peer-to-peer networks are selfish peers and malicious peers. They are distinguished primarily by their goals in the system. *Selfish peers* wish to use system services while contributing minimal or no resources themselves. A well-known example of selfish peers are "freeriders" [15] in file-sharing networks, such as Kazaa and Gnutella. To minimize their cost in bandwidth and CPU utilization freeriders refuse to share files in the network.

The goal of *malicious peers*, on the other hand, is to cause harm to either specific targeted members of the network or the system as a whole. To accomplish this goal, they are willing to spend any amount of resources (though we can consider malicious peers with constrained resources a subclass of malicious peers). Examples include distributing corrupted audio files on music-sharing networks to discourage piracy [16] or disseminating virus-infected files for notoriety [17].

Reputation system designers usually target a certain type of adversary. For instance, incentive schemes that encourage cooperation may work well against selfish peers but be ineffective against malicious peers. The number or fraction of peers that are adversaries also impact design. Byzantine protocols, for example, assume less than a third of the peers are misbehaving [18].

3.2.1 Adversarial Powers

Next, a designer must decide what techniques he expects the adversaries to employ against the system and build in mechanisms to combat those techniques. The following list briefly describes the more general techniques available to adversaries.

Traitors Some malicious peers may behave properly for a period of time in order to build up a strongly positive reputation, then begin defecting. This technique is effective when increased reputation gives a peer additional privileges, thus allowing malicious peers to do extra damage to the system when they defect. An example of traitors are eBay merchants that participate in many small transactions in order to build up a high positive reputation, and then defraud one or more buyers on a high-priced item. Traitors may also be the computers of well-behaved users that have been compromised through a virus or trojan horse. These machines will act to further the goals of the malicious user that subverted them.

Collusion In many situations multiple malicious peers acting together can cause more damage than each acting independently. This is especially true in peer-to-peer reputation systems, where covert affiliations are untraceable and the opinions of unknown peers impacts ones decisions. Most research devoted to defeating collusion assume that if a group of peers collude they act as a single unit, each peer being fully aware of the information and intent of every other colluding peer [11].

Front peers Also referred to as "moles" [19], these malicious colluding peers always cooperate with others in order to increase their reputation. They then provide misinformation to promote actively malicious peers. This form of attack is particularly difficult to prevent in an environment where there are no pre-existing trust relationships and peers have only the word and actions of others in guiding their interactions [20].

Whitewashers Peers that purposefully leave and rejoin the system with a new identity in an attempt to shed any bad reputation they have accumulated under their previous identity [21]. Whitewashers are discussed in depth in later sections (Sec. 4.3).

Denial of Service (DoS) Denial of Service attacks are particularly malicious threats. Whether conducted at the application layer or network layer, DoS attacks usually involve the adversary bringing to bear large amounts of resources in order to completely disrupt service usage. Using Internet worms however, malicious users are able to minimize their own personal resource usage while amplifying the damage done through Distributed DoS attacks. Much work has been done on detecting, managing, and preventing DoS attacks. P2P-specific applications include [22,23] in Gnutella networks and [18] in DHT networks. Not only would we like reputation systems to detect DoS attackers, but such attacks could be used against the reputation mechanism itself.

As we discuss different mechanisms, we will reference these tactics and explain how certain system properties can help against them. Most of the existing research does not claim to handle malicious peers that bring to bear all these attacks at once. In fact, much of the work focuses solely on independent selfish peers.

3.3 Environmental Limitations

The primary division among system component architectures is centralized versus decentralized. Implementing certain functionality at a single trusted entity can simplify mechanism design and provide a more efficient system. As we will see, some component properties can only be attained using the management and auditing capabilities afforded by a single point of trust. Of course centralization also has several drawbacks. It may be infeasible to have a single entity all agents trust. A centralized server becomes a single point of failure as well as a bottleneck. Providing performance and robustness requires the controlling entity to unilaterally invest large sums of money. It also makes for a single point of attack by adversaries, either by infiltration, subversion, or DoS attacks.

Between purely centralized and purely decentralized is a spectrum of hybrid architectures. For simplicity, we will refer to proposed mechanisms as centralized if they require one (or a small number) entity that is trusted by all users to handle some service for the entire system, even if they do not need to be always available, only intermittently. Otherwise, the mechanism is decentralized.

4 Gathering Information

The first component of a reputation system is responsible for collecting information on the behavior of peers, which will be used to determine how "trustworthy" they are (either on an absolute scale or relative to the other peers).

4.1 System Identities

Associating a history of behavior with a particular agent requires a sufficiently persistent identifier. Therefore, our first concern is the type of identities employed by the peers in the system. There are several properties an identity scheme may have, not all of which can be met with a single design. In fact, some properties are in direct conflict of each other. The properties we focus on are:

Anonymity As previously mentioned in Section 3.1, the level of anonymity offered by an identity scheme can vary from using real-world identities to preventing any correlation of actions as being from the same agent.

Most peer-to-peer networks, such as Kazaa [7], use simple, user-generated pseudonyms. Since peers connect directly to one another, their IP addresses are public, providing the closest association between the agent's actions and their real-world identity. To hide their IP addresses users can employ redirection schemes, such as Onion routing [24]. A P2P-specific solution using anonymizing tunnels is Tarzan [25]. Frequently changing pseudonyms and routing tunnels disassociates the user's actions from each other.

Though full anonymity prevents building user reputation, some peer-to-peer reputation systems, such as TrustMe [26], use pseudo-anonymity to encourage honest information sharing without fear of retribution. Each peer is assigned two identifiers; one for transactions and another for reporting reputation information and scores. A centralized login server minimizes fraud and whitewashing.

Spoof-resistant To prevent adversaries from impersonating other peers identities must be resistant to spoofing. One common solution is the use of public/private key pairs. If a peer uses its public key as its identifier, other peers can verify that any communication comes to or from that peer, assuming the use of nonces to defeat replay attacks. However, initially transmitting ones public key may still be susceptible to man-in-the-middle attacks. Certificates signed by an a priori trusted certificate authority (CA) can help, but requires a centralized mechanism.

Unforgeable In addition to being spoof-resistant, unforgeable identities protect against whitewashers and Sybil attacks [27], where a single user poses as several distinct peers in the network. Unforgeable identities are usually generated by a trusted system entity and given to new users as they join. These identifiers can be proven to have been generated by this trusted entity and only that entity. Notice a user's public/private key pair is not sufficient. A certificate for that public key issued by a trusted CA is. Login servers can also authenticate users as they enter. The CA or login server may require real-world identity proof to ensure that each user receives only one system identifier, perhaps using credit card verification [28,18]. These solutions are necessarily centralized. Decentralized solutions usually require identifiers that are costly to produce, though not strictly unforgeable. Costly identifiers help slow the rate of whitewashing or generating multiple identities, but does not eliminate it. [27].

The effectiveness of any solution at providing a given property lies on a cost scale (e.g. cycles, bandwidth, dollars). An adversary with infinite resources can compromise any property. For example, most resilient unforgeable or spoof-resistant identity schemes often rely on public/private key privacy. Given enough CPU power, an adversary can crack the key and therefore negate its intended purpose. An informal representation of the spectrum of identity choices is presented in Figure 1.

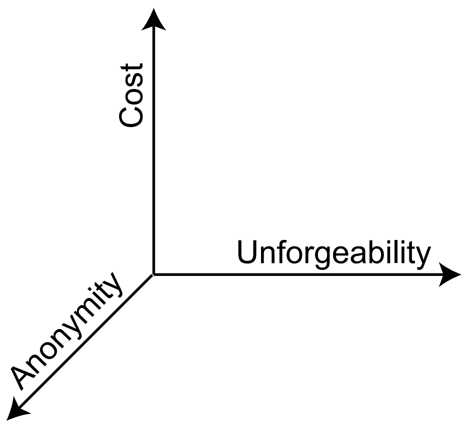

Fig. 1. Representation of primary identity scheme properties.

4.2 Information Sharing

Using established network identities, a reputation system protocol collects information on a given peer's behavior in previous transactions in order to determine their reputation. Examples of useful information include reports on the success or failure of a transaction by one or both parties, as well as the quality of the service/resource provided. This information collection may be done individually by each peer in a reactive method, or proactively by all peers collating together their experiences. In this section we discuss the sources from which information is collected, what quality of information agents can expect to collect, and how peers combine information from different sources.

4.2.1 Sources of Information

In general, quantity and quality of information are diametrically opposed. As the amount of information gathered increases, the credibility of each piece of information usually decreases.

The most cautious individuals may only want to rely on their own personal experience and use only *local information* when determining whether to transact with a given peer. Of course without additional information, the individual risks being cheated the first time they interact with each adversary. However, local information may be sufficient if the agent locates a few well-behaved peers able to repeatedly provide good service [29].

To increase the information sources a cautious agent can collect the opinions of users whom they have *a priori trust relationships* with externally from the system. These may include their friends from their personal lives, coworkers or business relationships, or even members of social networks [30,31] they trust.

Even with personal experience and the opinions of friends (that are currently online), an agent is unlikely to have any information on a particular random peer. To gather more opinions an agent can ask peers it has met in the P2P network, such as its neighbors in the overlay network, or peers who have already provided good service, proving themselves reputable. The question now is how many peers to query for their opinions (We discuss how to aggregate these opinions in Sec. 4.2.2.) Asking a small number of peers limits the communication overhead on the network [20], while asking a larger number improves the chances of collecting useful information on a specific peer [32].

If the number of personally-proven reputable peers is small, then an agent may request that each of those peers collect the opinions of other peers they believe are reputable, recursively. Each additional step exponentially increases the information sources. Information located through a *transitive trust* chain may be more reliable than asking a random peer [33,34,19].

Finally, there are the *global history* reputation systems that collect information about all peers from all peers. These solutions are the most comprehensive as well as the most complex to implement. While the probability that any single opinion is fraudulent may be greater, the collective sum of all opinions is likely to be accurate, even when a large fraction of the peers are malicious colluding adversaries.

While previous information-sharing techniques are easily decentralized, global history systems tend not to be. Perhaps the most widely used reputation system is that of eBay [35], which consists of a single trusted entity that collects all transaction reports and rates each user. Global history systems proposed for P2P networks tend to be more distributed. TrustMe [26] relies on a centralized server to assign unforgeable identities, but reputation adjustments and lookups are handled purely between peers. EigenTrust [36] offers a fully decentralized solution using weak identity leaving it more vulnerable to whitewashing.

In conclusion, a peer's reputation is based on information collected about that peer from one or multiple sources. The primary sources are: personal experience, external trusted sources, one-hop trusted peers, multi-hop trusted peers, and a global system. Each source provides increasingly more information about peers. However, that information becomes increasingly less credible as well.

In [2], O'Hara et al. categorize "trust strategies" for the Semantic Web based on how agents react to peers they have no personal experience with. Their five basic strategies include: optimistically assuming all strangers are trustworthy unless proven otherwise, pessimistically ignoring all strangers until proven trustworthy, investigating a stranger by asking trusted peers, transitively propagating the investigation through friends of friends, or using a centralized reputation system. Their taxonomy mirrors that presented here.

4.2.2 Information Integrity

One major problem with reputations systems is guaranteeing the validity of opinions. It is impossible to enforce honest, accurate reporting on transaction outcomes by all peers. Most reputation systems do not attempt to verify the integrity of information collected. Instead they assume the majority of users are honest and well-behaved, and that collecting information from a large number of peers will result in a relatively accurate assessment of a peer's behavior.

Reputation systems that hope to combat colluding adversaries and front peers that promote each other while denigrating good users, use reputation to weigh the information and opinions collected. Instead of considering the opinions of each peer, or each reported transaction experience, equally, these systems weigh the information based on the trustworthiness of the source when compiling a peer's reputation rating. For example, information provided by personal friends would likely be considered two or three times more accurate than that of a seemingly reputable, but unknown peer in the network. Of course, when available, personal experience would be valued the most [20]. Often the opinions of system peers are weighted by their previously determined reputation scores. Information collected through transitive trust may be weighted by the reputation rating of the least reputable peer in the trust chain [19]. Or, if reputation ratings lie between 0 and 1, the opinion would be weighted by the product of the ratings of the peers in the chain.

Even global history reputation systems apply reputation-based weighting. EigenTrust [36] uses a distributed algorithm similar to PageRank [37] to compute a global reputation rating for every peer using individual transaction reports weighted by the rating of the reporting peer. However, even this algorithm was found to be vulnerable to widespread collusion. Therefore, the authors suggest each agent separately weigh a globally computed rating with the personal opinions of trusted peers, when available.

Some systems attempt to improve the accuracy of the transaction reports by requiring proof of interaction. TrustMe [26], for example, requires that both parties in a transaction sign a transaction certificate that is then presented when reporting on the outcome of the transaction. While this may not prevent malicious peers from lying about the outcome of a transaction, it does prevent adversaries from submitting fraudulent reports about peers they have not interacted with in order to besmirch their reputation.

4.3 Dealing with Strangers

Of course, with new users joining the system periodically, agents will often encounter peers with no transaction history available at any source. As the number of sources an agent gathers information from increases, the frequency of encountering a *local stranger* (a peer whom the agent has no direct or indirect experience with or knowledge of) decreases. In the global history systems all local strangers are also *global strangers* (peers whom no agent in the system has interacted with).

When no reputation information can be located, an agent must decide whether to transact with a stranger based on its *stranger policy*. As mentioned previously, two simple strategies are to optimistically trust all strangers, or pessimistically refuse to interact with them. Both have their drawbacks. Optimistic agents may frequently be defrauded, especially in systems with high levels of whitewashing. However, in pessimistic systems, new users will be unable to participate in transactions and will never build a reputation.

Feldman et al. have done extensive work in analyzing the problem of stranger policies and whitewashing in P2P networks [21,19,38]. They suggest a "stranger adaptive" strategy. All transaction information on first-time interactions with any stranger is aggregated together. Using a "generosity" metric based on recent stranger transactions, an agent estimates the likely probability of being cheated by the next stranger and decides whether to trust the next stranger using that probability. This probabilistic strategy adapts well to the current rate of whitewashing in the system.

5 Reputation Scoring and Ranking

Once a peer's transaction history has been collected and properly weighted, a reputation score is computed for that peer, either by an interested agent, a centralized entity, or by all peers collectively, as in EigenTrust [36]. We will refer to the method by which the score is computed as a general reputation score function.

The primary purpose of the reputation score is to help an agent decide which available service provider in the network it should transact with. The two typical scenarios are:

i) Agent A is offered a resource or service by peer P. A decides if transacting with P is worth the expected risk of defection, based on P's reputation score.

ii) In response to A's request for a certain resource or service, multiple service providers ($P1$, $P2$,...) respond. A uses the reputation scores of each responder to rank them in order of how likely they are to provide proper service. A then chooses the highest ranked provider. Should that transaction fail, A may try again with the next highest ranked peer.

In the next two sections, we consider the inputs and outputs of the reputation score function. What statistics gathered from a peer's transactional history will most benefit in computing its trustworthiness? How should reputation scores be represented?

5.1 Inputs

Regardless of how a peer's final reputation rating is calculated, it may be based on various statistics collected from its history. But what statistics should be used in computing the ranking score? Ideally, both the amount a peer cooperates and defects would be taken into account. However, often the amount a peer defects may be unknown. While a malicious peer may openly defect on an agreed transaction by providing bad service or no service, selfish peers usually defect "silently". For example in file-sharing networks, freeriders refuse to share their files and ignore queries they could answer. Other peers cannot determine how often a peer selfishly ignores a request. However, as suggested in [19], peers can calculate the rate at which an agent contributes to the network. The contribution rate is a reputation rating based solely on good work.

When defection information is available, this statistic is usually more useful than cooperations. Notice that visible defections usually constitute malicious behavior, which is more harmful than selfish behavior. While both good and bad behavior can be taken into account, the negative impact of bad behavior on reputation should outweigh the positive impact of good behavior.

When only information on positive contributions are available, the reputation will have to be based solely on the amount of good work done. However, if a history of peer's cooperations and defections is available, should the peer's reputation be based on the quality of the work its done? Or should the quantity also matter? Our previous work has shown that while quality alone is useful [20], a score that properly combines quality and quantity is much more effective and flexible under a variety of adversarial techniques [39]. Included in quantity should be the value of each transaction. Intuitively, a peer that defects on one \$100 transaction should have a lower reputation than one who defects on two or three \$1 transactions.

If a system wishes to defend against traitors, then reputations scores must consider time. More recent transaction behavior should have a greater impact on a peer's score than older transactions. For example, a weighted transaction history could be used. This would allow system agents to detect peers who suddenly "go bad" and defend against them.

5.2 Outputs

In the end, the computed reputation rating may be a binary value (trusted or untrusted), a scaled integer (e.g. 1 to 10), or on a continuous scale (e.g. [0,1]). The choice will be application dependent, although a binary value would likely be insufficient in a P2P environment where all peers are untrusted, but we want to rank peers based on how reliable they are likely to be.

Both scenarios detailed above imply a single scalar value is obtained for each candidate and is compared either against other candidates' ratings or against a trust threshold determined by the transaction. However, it is useful to maintain a peer's reputation as multiple component scores. Applying different functions to the scores allows a peer to calculate a rating best suited for the given situation. Many proposed systems suggest maintaining multiple statistics about each peer. For example, keeping separate ratings on a peer's likelihood to defect on a transaction and it's likelihood to recommend malicious peers helps mitigate the effects of front peers. The TRELLIS system [40] keeps separate ratings for the likelihood a peer cooperates on a transaction (referred to as its "reliability") and the accuracy of its opinions or recommendations (its "credibility"). Reliability would correspond to the reputation score as discussed here, while the credibility score would be used for weighing information sources, as discussed in Section 4.2.2. Guha et al. [41] suggest maintaining separate scores for trust and distrust.

5.3 Peer Selection

Once an agent has computed reputation ratings for the peers interested in transacting with it, it must decide which, if any, to choose. If there is only one peer, and the question is whether to trust it with the offered transaction, the agent may decide based on whether the peer's reputation rating is above or below a set *selection threshold* [29].

If multiple peers are offering the same resource, the agent would likely go with the peer with the highest reputation rating. However, even with many peers available, an agent may decide to refuse all their transaction requests if all their reputations lie below the selection threshold. It may not be uncommon in certain systems, such as document-sharing systems, for all peers responding to a rare document request to be malicious. Malicious peers disseminating inauthentic or virus-infected files can reply to any request, while well-behaved peers will only reply if they have the queried document. A selection threshold is necessary to protect against malicious spam responses [29].

6 Taking Action

In addition to guiding decisions on selecting transactional partners, reputation systems can be used to motivate peers to positively contribute to the network and/or punish adversaries who try to disrupt the system.

6.1 Incentives

Mechanisms used to encourage cooperation in the system are referred to as *incentive schemes*. They are most effective at combatting selfishness as they offset the cost of contribution with some benefit. However, incentive schemes can mitigate some maliciousness if access to system services requires an adversary provide good resources first. Such a reciprocative procedure raises the cost of misbehavior.

Most suggested incentive schemes offer one of two types of incentives: improved service or money, with service improvements further decomposing into three general categories:

Speed Agents that contribute resources to the network may be rewarded with faster download speeds or reduced response latency for their requests and queries. An example of this incentive can be seen in Bittorrent [42], a common P2P application for downloading popular files by allowing downloaders to share parts of files as they are received. Applying the principle of "tit-for-tat", the client application throttles upload speeds to a peer based on the download speed it is receiving from that peer. Therefore, peers that are willing to devote more upload bandwidth are rewarded with a higher download speed.

Quality Some systems may provide content at varying levels of quality, depending on a peer's contribution rate. For example, a P2P streaming movie service could provide movies at different resolutions depending on a customer's subscription plan. This approach is already used by many online video providers (e.g. IFILM [43]).

Quantity Similar to quality, the amount of information, content, or service providers available to a peer would be determined by the amount the peer contributes. This approach is also used by many online services that provide a limited amount of content for free but require payment for access to all their content. Similar ideas have been proposed for use in P2P systems. For instance, some solutions encourage peers to route network messages for other peers (e.g. [44,45]).

Money Currently, peer-to-peer systems are used to share files and resources that require little or no cost for the contributing peer to produce and distribute. However, to support the exchange of more valuable content will require a payment mechanism that allows an agent to pay the content creator and provider upon acquiring it. Most of the content will likely carry a low price since the cost of distribution is spread over the users. Therefore a low-weight micropayment mechanism is needed, allowing clients to make payments of a few cents (or fractions of cents) without incurring a larger billing fee. Several papers have proposed low-cost micropayment mechanisms for P2P systems (e.g. [46,47]).

6.2 Punishment

While incentives are very useful at discouraging selfishness, curtailing misbehavior requires the ability to punish malicious peers. As discussed earlier, the primary function of reputation systems is to inform agents as to which peers are likely to defect on a transaction. Not only does adversary avoidance benefit well-behaved peers, but it punishes malicious peers who will quickly find themselves unable to disseminate bad resources or cheat other peers. E-commerce sites, such as eBay [35], use reputation systems not only to provide good customers information on sellers, giving buyers a sense of security, but also to discourage misbehavior in the first place.

If the reputation system can identify actively malicious peers it may retaliate in several ways beyond simply warning other users. Overlay network neighbors can disconnect from the adversary, immediately ejecting it from the network. Depending on the type of identifiers used, the adversary may be kicked from the network for a period of time, or permanently banned. To reenter the system, the adversary would need to acquire a new valid identifier, which may be costly or impossible.

Finally, P2P systems tied to financial institutions for monetary payments could fine a malicious peer for each verified act of misbehavior. Of course, such a solution should be used cautiously as adversaries could use them to wreak havoc in the system by falsely implicating well-behaved peers of misbehavior.

7 Conclusion

Developing an implementable reputation system is an art involving many separate design problems and choices. A reputation system is generally composed of three basic components: gathering behavioral information, scoring and ranking peers, and rewarding or punishing peers. In turn, each component requires a combination of mechanisms to function. We believe a proper dissection of the overall design problem will allow researchers to develop efficient solutions to each separate part without losing sight of the overall goal.

We have demonstrated that many desirable system properties lead to implementation conflicts; many of which stem from the requirement for a purely decentralized architecture. However, continuing investigation and innovation will generate solutions that overcome these conflicts. We believe future research will provide mechanisms capable of all the functionality, reliability and security of a centralized trusted reputation system in an anonymous peer-to-peer network.

8 Acknowledgements

The authors would like to thank Neil Daswani and Kevin Lai for their input and comments.

References

[1] Stanford Peers research group, http://www-db.stanford.edu/peers/.

[2] K. O'Hara, H. Alani, Y. Kalfoglou, N. Shadbolt, Trust Strategies for the Semantic Web, in: ISWC'04 Workshop on Trust, Security and Reputation on the Semantic Web, 2004.

[3] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Trans. Netw. 11 (1) (2003) 17–32.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A Scalable Content-Addressable Network, in: Proceedings of the ACM SIGCOMM Symposium on Communication, Architecture, and Protocols, ACM SIGCOMM, San Diego, CA, U.S.A., 2001, pp. 161–172.

[5] A. Rowstron, P. Druschel, Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (2001) 329–350.

[6] Gnutella specification, www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[7] KaZaA Home Page, http://www.kazaa.com/.

[8] B. Yang, H. Garcia-Molina, Comparing hybrid peer-to-peer systems, in: The VLDB Journal, 2001, pp. 561–570.
URL `citeseer.nj.nec.com/461877.html`

[9] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, I. Stoica, The impact of DHT routing geometry on resilience and proximity, in: Proc. ACM SIGCOMM, 2003.

[10] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao, OceanStore: An Architecture for Global-scale Persistent Storage, in: Proceedings of ACM ASPLOS, ACM, 2000.
URL `citeseer.nj.nec.com/kubiatowicz00oceanstore.html`

[11] P. Maniatis, M. Roussopoulos, T. Giuli, D. S. H. Rosenthal, M. Baker, Y. Muliadi, Preserving peer replicas by rate-limited sampled voting, in: 19th ACM Symposium on Operating Systems Principles (SOSP 2003), 2003.

[12] M. K. Reiter, A. D. Rubin, Crowds: Anonymity for web transactions, in: ACM Transactions on Information and System Security, 1998.

[13] G. Agarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, Y. Xu, Vision Paper: Enabling Privacy for the Paranoids, in: VLDB, 2004.

[14] G. Agarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu, Two Can Keep a Secret: A Distributed Architecture for Secure Database Services, in: CIDR, 2005.

[15] E. Adar, B. A. Huberman, Free riding on gnutella, First Monday 5 (10).

[16] L. McClain, RIAA posting bad music files to deter illegal downloaders, The Daily Texan. 2/6/2004.

[17] BBC NEWS, Viruses turn to peer-to-peer nets, BBC NEWS. 1/20/2004.

[18] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, D. S. Wallach, Secure routing for structured peer-to-peer overlay networks, in: Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, 2002.

[19] M. Feldman, K. Lai, I. Stoica, J. Chuang, Robust Incentive Techniques for Peer-to-Peer Networks, in: ACM Conference on Electronic Commerce (EC'04), 2004.

[20] S. Marti, H. Garcia-Molina, Limited Reputation Sharing in P2P Systems, in: ACM Conference on Electronic Commerce (EC'04), 2004.

[21] K. Lai, M. Feldman, I. Stoica, J. Chuang, Incentives for Cooperation in Peer-to-Peer Networks, in: Workshop on Economics of Peer-to-Peer Systems, 2003.

[22] N. Daswani, H. Garcia-Molina, Query-Flood DoS Attacks in Gnutella, in: ACM Conference on Computer and Communications Security, 2002.

[23] N. Daswani, H. Garcia-Molina, Pong-Cache Poisoning in GUESS, in: ACM Conference on Computer and Communications Security, 2004.

[24] P. Syverson, D. Goldschlag, M. Reed, Anonymous Connections and Onion Routing, in: Proceedings of the IEEE Symposium on Security and Privacy, 1997.

[25] M. Freedman, R. Morris, Tarzan: A Peer-to-Peer Anonymizing Network Layer, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002.

[26] A. Singh, L. Liu, TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems, in: IEEE 3rd International Conference on Peer-to-Peer Computing (P2P 2003), 2003.

[27] J. R. Douceur, The Sybil Attack, in: Proc. of the International Workshop on Peer-to-Peer Systems, 2002.

[28] E. Friedman, P. Resnick, The social cost of cheap pseudonyms, Journal of Economics and Management Strategy 10 (2) (1998) 173–199.

[29] S. Marti, H. Garcia-Molina, Identity Crisis: Anonymity vs. Reputation in P2P Systems, in: IEEE 3rd International Conference on Peer-to-Peer Computing (P2P 2003).

[30] S. Marti, P. Ganesan, H. Garcia-Molina, SPROUT: P2P Routing with Social Networks, in: International Workshop on Peer-to-Peer Computing & DataBases (P2P&DB 2004), 2004.

[31] T. Hogg, L. Adamic, Enhancing Reputation Mechanisms via Online Social Networks, in: ACM Conference on Electronic Commerce (EC'04), 2004.

[32] F. Cornelli, E. Damiani, S. D. Capitani, Choosing Reputable Servents in a P2P Network, in: Proc. of the 11th International World Wide Web Conference, 2002.

[33] B. Yu, M. P. Singh, A social mechanism of reputation management in electronic communities, Cooperative Information Agents (2000) 154–165.

[34] S. Lee, R. Sherwood, B. Bhattacharjee, Cooperative Peer Groups in NICE, in: Proceedings of the IEEE INFOCOM, 2003.

[35] eBay - The World's Online Marketplace, http://www.ebay.com/.

[36] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina, The EigenTrust Algorithm for Reputation Management in P2P Networks, in: Proceedings of the Twelfth International World Wide Web Conference, 2003.

[37] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web, Tech. rep., Stanford Digital Library Technologies Project (1998).

[38] M. Feldman, C. Padimitriou, J. Chuang, I. Stoica, Free-Riding and Whitewashing in Peer-to-Peer Systems, in: ACM SIGCOMM 2004, Workshop of Practice and Theory of Incentives and Game Theory in Networked Systems, 2004.

[39] S. Marti, H. Garcia-Molina, Modeling Reputation and Incentives in Online Trade (extended), Tech. rep. (2004. dbpubs.stanford.edu/pub/2004-45).
URL `dbpubs.stanford.edu/pub/2004-45`

[40] Y. Gil, V. Ratnakar, Trusting information sources one citizen at a time, in: Proceedings of the First International Semantic Web Conference (ISWC), 2002.

[41] R. Guha, R. Kumar, P. Raghavan, A. Tomkins, Propagation of trust and distrust, in: Proceedings of the 13th World Wide Web Conference (WWW2004), 2004.

[42] B. Cohen, Incentives Build Robustness in BitTorrent, in: Workshop on Economics of Peer-to-Peer Systems, 2003.

[43] IFILM Corp., IFILM, http://www.ifilm.com (2004).

[44] B. Yang, T. Condie, S. Kamvar, H. Garcia-Molina, Addressing the Non-Cooperation Problem in Competitive P2P Systems, in: Workshop on Economics of Peer-to-Peer Systems.

[45] A. Blanc, Y.-K. Liu, A. Vahdat, Designing Incentives for Peer-to-Peer Routing, in: Workshop on Economics of Peer-to-Peer Systems.

[46] B. Horne, B. Pinkas, T. Sander, Escrow Services and Incentives in Peer-to-Peer Networks, in: Proceedings of 3rd ACM Conference on Electronic Commerce, 2001.

[47] B. Yang, H. Garcia-Molina, PPay: Micropayments for Peer-to-Peer Systems, in: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), 2003, Washington D.C.

[48] N. Daswani, Personal communication (2004).

[49] K. Lai, Personal communication (2004).