Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 92485, 14 pages
doi:10.1155/2007/92485

Research Article

Globally Decoupled Reputations for Large Distributed Networks

Gayatri Swamynathan, Ben Y. Zhao, Kevin C. Almeroth, and Haitao Zheng

Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

Received 8 March 2007; Accepted 7 June 2007

Recommended by Shigang Chen

Reputation systems help establish social control in peer-to-peer networks. To be truly effective, however, a reputation system should counter attacks that compromise the reliability of user ratings. Existing reputation approaches either average a peer's lifetime ratings or account for rating credibility by weighing each piece of feedback by the reputation of its source. While these systems improve cooperation in a P2P network, they are extremely vulnerable to unfair ratings attacks. In this paper, we recommend that reputation systems decouple a peer's *service provider* reputation from its *service recommender* reputation, thereby, making reputations more resistant to tampering. We propose a scalable approach to system-wide decoupled service and feedback reputations and demonstrate the effectiveness of our model against previous nondecoupled reputation approaches. Our results indicate that decoupled approache significantly improves reputation accuracy, resulting in more successful transactions. Furthermore, we demonstrate the effectiveness and scalability of our decoupled approach as compared to PeerTrust, an alternative mechanism proposed for decoupled reputations. Our results are compiled from comprehensive logs collected from Maze, a large file-sharing system with over 1.4 million users supporting searches on 226TB of data.

1. INTRODUCTION

The explosive growth of Internet connections in the last decade has resulted in an increase in the use and popularity of online peer-to-peer (P2P) communities. While the growth of these communities advances distributed applications like file transfer, remote storage, and computation, it is becoming increasingly critical to manage trust relationships in order to improve the performance of these applications. Large-scale P2P communities, like Gnutella [1], rely on cooperation among network peers. These communities, however, neither enforce cooperation nor centrally control peers. Peers are anonymous and self-interested, behaving only in their best interests. While this open nature of a P2P community is increasing the number of participating peers, it also makes these communities extremely difficult to police and vulnerable to attacks, thereby reducing the performance of the network.

As a popular P2P network, Gnutella is susceptible to a variety of attacks. One common attack is "whitewashing," where a free-riding node repeatedly joins the network under a new identity in order to avoid the penalties imposed on free-riders. A more serious attack is when dishonest peers distribute viruses and Trojan horses hidden as files. The *VBS.Gnutella* worm, for example, stores Trojan-horse executable files on network peers [2]. Meanwhile, a Gnutella worm called *Mandragora* registers itself as an active peer in the network, and provides a renamed copy of itself for download in response to intercepted queries [3]. Finally, dishonest peers often pass corrupted or blank files as legitimate content.

In order to reduce transaction risks and improve performance, P2P networks need to motivate cooperation and honest participation within their networks. Reputation systems help address this need by establishing a trust mechanism that enables peers to decide who to trust before undertaking a transaction. The predictive power of reputation assumes that a peer's past behavior is indicative of its future behavior. Feedback from all the peers that have previously interacted with a peer are aggregated to compute the first peer's reputation. Such a reputation mechanism, consequently, enables a community to police itself in order to establish social control.

A large amount of literature confirms the fact that reputation systems are an effective means of social control [3–9]. Within the bounds of their assumptions, these systems

demonstrate the ability to significantly reduce the number of malicious transactions and improve cooperation in a network. Despite these proposals, designing a robust and reliable reputation system is still largely an open challenge. The trust model used in existing reputation systems is extremely vulnerable to misleading or unfair feedback.

The challenge, therefore, lies in building a reputation system that is effective despite attacks that compromise the reliability of feedback. Most existing reputation systems either average a peer's lifetime ratings, resulting in the "increased trust by increased volume" vulnerability, or correlate service trust to imply feedback trust. The latter assumes that peers reputed to provide trustworthy service, in general, provide trustworthy feedback. While useful as a simple defense, such an assumption can easily fail or be manipulated. For example, colluding nodes can offer honest service for the express purpose of boosting their reputations so they can badmouth other peers. Countering false feedback attacks is a critical challenge that needs to be addressed to improve the performance of reputation systems, and consequently, peer-to-peer networks.

With this research challenge in mind, this paper offers three contributions. First, we discuss the various types of false feedback (or rating) attacks that can compromise existing reputation systems. We then discuss different reputation-based trust approaches that have been proposed for P2P networks and analyze their ability to overcome these false rating attacks. Next, we describe our scalable globally decoupled reputation model, which decouples each per-user reputation rating into a service rating and a feedback rating. The credibility of a peer's feedback in our model is weighed by its reputation as a service recommender, and not as a service provider. Finally, we use extensive evaluations to compare our trust model against some of the existing approaches, including the approach of averaging lifetime ratings, the coupled trust approach, and PeerTrust's personalized similarity trust metric, a system that employs a similar decoupled trust policy [10]. Our simulations show that decoupled trust models provide significantly more accurate reputations by detecting and isolating malicious peers. We also demonstrate the effectiveness and scalability of our decoupled system on peer traces gathered from *Maze*, a large scale peer-to-peer file-sharing system.

The remainder of the paper is organized as follows. We begin by identifying related work in Section 2. In Section 3, we describe our decoupled trust model and present our reputation system. Next, we classify unfair ratings attacks and discuss some reputation-based approaches employed to counter these attacks in Section 4. Our experimental evaluations in Section 5 perform detailed comparisons of our proposed trust model with respect to the different reputation models and different unfair ratings attacks discussed in Section 4. We evaluate our system using trace-driven simulations from the Maze file-sharing system in Section 6. Furthermore, we employ the Maze logs to compare the PeerTrust decoupled algorithm and our decoupling approach and highlight some vulnerabilities inherent to PeerTrust. Finally, we discuss implications of our system in Section 7 and conclude in Section 8.

2. RELATED WORK

Significant prior work has shown that reputation systems, if reliable, can effectively motivate trustworthiness and cooperation [3, 5, 7–13]. Reputation systems can build trust models using two approaches. One approach is to use only firsthand information to evaluate peers. While highly reliable, a firsthand-only approach does not employ all reputation information available in the network, making it highly inefficient and unscalable. Firsthand information proves sufficient if a peer locates honest service providers with which it repeatedly transacts [14].

Almost all reputation systems use global information, that is, peers aggregate opinions of all other peers that have interacted with them in the past. While global reputations are efficient and help to quickly detect misbehavior in the system, they are vulnerable to false ratings and collusion. One technique that incorporates global information is a simple averaging or summarizing of ratings. EBay, the largest online auction site, uses a reputation-based trust scheme where, after each transaction, buyers and sellers rate each other using the *Feedback Forum* [15]. Because EBay uses a central authority to manage all communication and coordination between peers, it essentially eliminates much of the complexity that exists in a decentralized system.

Simple summarizing schemes, in general, are highly vulnerable to malicious participants that increase their transaction volume to hide frequent misbehavior. A peer could increase its trust value by increasing its transaction volume, thereby hiding the fact that it frequently misbehaves at a certain rate. For example, a peer could undertake a thousand good transactions of low value (say, worth $1) and use the accumulated good reputation towards one dishonest transaction of high value (say, worth $1000). Additionally, if all ratings are given an equal weight, Sybil attacks and collusion are encouraged.

Therefore, to incorporate global information effectively, trust models must account for the credibility of the service raters using different trust propagation mechanisms. Examples include *transitive* trust, *coupled* trust, or *decoupled* trust. The underlying principle of *transitive trust* is if A trusts B and B trusts C, then it is likely that A trusts C. Reputation systems employ such web-of-trust chains to establish and propagate trust among peers. In general, longer chains imply greater risk of encountering a malicious link. Schemes like weighing ratings of a transitive chain by the reputation of the least reputed peer in the chain [16], employing a node distrust table [6], and using pretrusted peers [8] have been proposed.

Coupled trust approaches assume that peers reputed to provide trustworthy service, in general, will be likely to provide trustworthy feedback. We cite two well-known examples of reputation systems that rely on the correlated trust assumption. Aberer and Despotovic propose a decentralized reputation system for P2P networks where data is stored on a P-Grid [5]. Their system assumes that most network peers are honest, and reputations in the system are expressed as complaints. EigenTrust is a reputation system for P2P networks designed to combat the spread of fake files [8]. Each

peer is associated with a global trust value that reflects the experiences of all other peers with it. These values are used as a metric of reliability when choosing download sources.

Alternatively, some systems make use of *decoupled trust*, which involves using separate metrics to evaluate service trust and feedback trust [10, 11]. To decouple feedback trust from service trust in PeerTrust [10, 17], peers use a *personalized similarity measure* to more heavily weigh opinions of peers who have provided similar ratings for a common set of past partners. In a large P2P system, however, finding a statistically significant set of such past partners is a challenge. Peers are likely to make choices among a set of candidates for which there is no information. In Section 6.3, we empirically demonstrate how this lack of trust data information can impact the effectiveness and scalability of PeerTrust in its computation of trust values.

Confidant, another decoupled trust mechanism, attacks the problem of false ratings using a Bayesian approach in a mobile ad hoc network [11]. They distinguish between reputation, which they define as how well a node behaves in routing, and trust, which is how well it behaves in the reputation system. A node distributes only firsthand information to other nodes and only accepts other firsthand information if those opinions are similar to its own opinion. Compared to Confidant, where a node's referral is interpreted subjectively per node, our proposal produces a system-wide referrer rating per node.

Reputation ratings are normally associated with peers. But in some cases, a reputation rating is associated with a resource, for example, a file in a file-sharing P2P network [18]. Damiani et al. present a detailed discussion on the advantages and disadvantages of employing a pure resource-based or peer-based reputation and propose combining both reputations [3]. Storage overheads are substantially higher because the number of resources in any system tends to be significantly larger than the number of peers. Also, it is often not possible for a single resource to be widespread enough to have a sufficient number of raters for it. Credence [18] overcomes this problem by employing a web-of-trust in the absence of direct observations.

3. OUR SCALABLE DECOUPLED TRUST MODEL

We now discuss our decoupled reputation framework and describe the trust model used to update reputation ratings.

3.1. Reputation propagation framework

Our reputation system associates two sets of reputation ratings with each peer: an aggregated service rating (SR) and an aggregated feedback rating (FR). The service rating indicates a peer's trustworthiness as a service provider, for example, a peer's overall file-sharing behavior in a P2P file-sharing system. The feedback rating of a peer indicates its overall trustworthiness as a service recommender. Additionally, each peer maintains a list of peers that has rated it and the ratings provided by them. Service reputations are normalized values between 0.0 and 1.0 with 1.0 indicating a perfect service reputation. Similarly, feedback reputations are normalized values that range from 0.0 and 1.0 with 1.0 indicating a perfect rater. Initially, SR and FR are set to 1.0 for all peers.

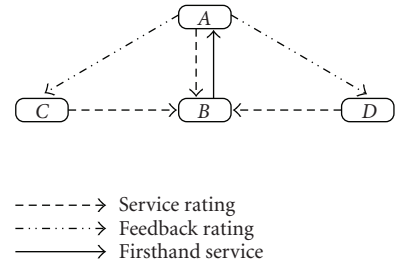

Figure 1: Decoupling service and feedback reputation: after interacting with B, peer A modifies B's service reputation and also modifies the feedback reputations of B's previous raters, C and D.

Consider a peer, A, that queries for a file. In order to make a decision about which responding peer with which to transact, A chooses the peer with the highest aggregated service rating. While this choice can result in an unbalanced load distribution in the network, a probabilistic approach can be employed to distribute load [8]. After finishing a transaction with service provider B, A provides B with either a rating of 0 (unsatisfactory) or 1 (satisfactory), depending on the outcome. This rating is weighted by $\mathrm{FR}(A)$, that is, the feedback rating of A. This weighting implies that A needs to be wellreputed as a feedback provider in order for its opinions to have an effect on B's service reputation. In other words, feedback from peers with higher feedback trust ratings will have more impact than those with lower feedback ratings. While we currently employ binary ratings to rate transaction outcomes, our design framework works for complex ratings schemes as well. For example, subjective ratings like Very Good, Good, OK, Bad, Very Bad can be mapped to quantitative rating values.

At the end of the transaction, A also needs to send feedback rating updates to all peers that had rated B earlier. A provides a rating of 1 to all peers that rated B with a value consistent with A's firsthand experience. In the case where the outcome of A's transaction with B did not match with a prior service rating, A generates a feedback rating of 0 to the originator of the rating. This rating is in turn weighted by A's feedback rating. This process is shown in Figure 1, where A interacts with B, updates B's service reputation, and updates the feedback ratings of C and D, who contributed to B's service reputation.

Feedback reputations pose additional storage and management overhead. By maintaining reputation information for only a window of recent transactions, the amount of overheads can be controlled. Our experiments in Section 5 demonstrate that the increase in reputation accuracy justifies this additional storage overhead.

We do not explicitly address storage and communication issues in our model, since they are largely orthogonal to our problem of decoupling reputation. Our trust model would

work with a number of different storage models. For example, peers can compute and maintain their reputations in a self-storing model [9]. While digital signatures and time stamps need to be built into the reputation system to ensure validity and integrity of the reputation data, such local storage schemes ensure data availability, thus eliminating the problem of a reputation storer being offline while the target is online. Alternatively, peer reputations can be stored and computed independently by third parties using distributed hash table- (DHT-) based approaches [5, 8]. The dissemination of reputations can leverage the communication protocol of the peer-to-peer network [3] or employ lookup schemes for structured storage [10].

3.2. The trust model

In this section, we formalize our trust metric for updating service and feedback reputations.

Our trust model is based on peer evaluation. Each peer observes two kinds of peer evaluations: *service rating*(s) after it serves as a service provider and *feedback rating*(s) after it provides service evaluations. For each transaction between two peers (i, j), where i represents the service provider and j represents the service requester, we define the following terms:

(i) *transaction rating* ($s_{i,j}$): j's firsthand observation of i's service for their most recent transaction (0: unsatisfactory, 1: satisfactory);
(ii) *transaction set*, T_i: let $|T_i|$ represent the cardinality of the set of peers that have transacted with peer i;
(iii) *peer i's average service rating*, $\mathrm{SR}(i)$: the life-time average rating of i based on peer evaluation;
(iv) *peer j's average feedback rating*, $\mathrm{FR}(j)$: the life-time average rating on j's feedback. In other words, this value is the average credibility of j's feedback on service satisfaction.

For each peer, i, its average service rating is an average of all the firsthand observations provided by peers with which it transacted, weighted by their individual feedback credibility. That is,

$$\mathrm{SR}(i) = \frac{1}{|T_i|} \cdot \sum_{j \in T_i} s_{i,j} \cdot \mathrm{FR}(j). \tag{1}$$

Like the service rating, each peer's feedback credibility is based on peer evaluation. In particular, at the end of each transaction (i, j), the service requester, j, not only sends feedback on i's service, but also updates feedback ratings of all peers that have previously transacted and rated i.

We define $f_{k,j}$ as the feedback rating of k by j based on j's current transaction with i and k's most recent transaction with i. If the service ratings are consistent, j rates k's feedback helpful or malicious. That is,

$$f_{k,j} = \begin{cases} 1, & s_{i,j} = s_{i,k}, \\ 0, & s_{i,j} \neq s_{i,k}. \end{cases} \tag{2}$$

The overall (average) feedback rating of k is the average of the most recent feedback ratings from its peers, weighed by their feedback credibility. The derivation is as follows:

$$\mathrm{FR}(k) = \frac{1}{|T_k^f|} \cdot \sum_{j \in T_k^f} f_{k,j} \cdot \mathrm{FR}(j), \tag{3}$$

where T_k^f represents the set of peers that have provided feedback ratings for k.

In this way, we incorporate peer evaluations on both the service provider and the requester in the computation of trust values making them robust to peer maliciousness.

3.3. Handling dynamic peer personalities

In our proposed trust metric, a peer's service or feedback rating is an aggregation of all the firsthand service or feedback observations it has received in its lifetime. This long-term aggregation makes our reputation system slow to react to changes in a peer's "personality." A peer can establish a good reputation in order to behave badly, without the results significantly impacting its reputation. In addition, honest peers can be subverted at any time by attackers and begin behaving badly. Therefore, peer reputations must be representative of more recent behavior rather than old ratings. To address the issue of dynamic behavior, we employ a simple window-based adaptation of our metric, a technique similar to that of PeerTrust [10].

We define the following:

(a) *peer i's reference service rating*, $\mathrm{SR}_{\mathrm{ref}}(i)$: i's service rating averaged over the set of M recent transactions.[1] That is,

$$\mathrm{SR}_{\mathrm{ref}}(i) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{SR}_m(i), \tag{4}$$

where $\mathrm{SR}_m(i)$ represents i's average service rating after the $(M - m)$th most recent transactions provided by i. $\mathrm{SR}_{\mathrm{ref}}(i)$ provides a guideline to regulate i's dynamic behavior. In particular, the following actions are performed (in order) after each transaction with i:

$$\mathrm{SR}(i) = \frac{1}{|T_i|} \cdot \sum_{j \in T_i} s_{i,j} \cdot \mathrm{FR}(j)$$

{calculate average service rating},

$$\mathrm{SR}_{\mathrm{ref}}(i) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{SR}_m(i)$$

{calculate reference service rating},

$$\mathrm{SR}(i) = \begin{cases} \mathrm{SR}_{\mathrm{ref}}(i), & \mathrm{SR}(i) - \mathrm{SR}_{\mathrm{ref}}(i) > \epsilon \\ \mathrm{SR}(i), & \text{otherwise} \end{cases}$$

[1] M is a design parameter. We assume that, in our system, transactions occur periodically. Hence, M also directly relates to the length of the average time window.

$$\{\text{check whether performance has dropped recently}\},$$
$$\mathrm{SR}_{m+1}(i) = \mathrm{SR}_m(i), \quad m = 1 \cdots M - 1$$
$$\{\text{update ratings over } M \text{ recent transactions}\},$$
$$\mathrm{SR}_1(i) = \mathrm{SR}(i); \tag{5}$$

(b) *peer j's reference feedback rating*, $\mathrm{FR}_{\mathrm{ref}}(j)$: j's service rating averaged over the set of M recent feedback ratings provided to j. $\mathrm{FR}_{\mathrm{ref}}(j)$ is derived in a manner similar to $\mathrm{SR}_{\mathrm{ref}}(i)$.

In (3.3), if $\mathrm{SR}_{\mathrm{ref}}(i)$ is smaller than $\mathrm{SR}(i)$ by a specified threshold ϵ, it means that a peer's performance has dropped significantly within some recent time frame. In this case, $\mathrm{SR}_{\mathrm{ref}}(i)$ is assigned as the peer's new reputation rating. This time-based adaptation allows reputations to be more sensitive to recent behavior and helps our system to quickly adapt to changes in peer behavior. This approach proves particularly adept in counteracting oscillating peers who alternate between honest and dishonest behavior in order to build and abuse good reputations.

4. SAFEGUARDING REPUTATIONS

The reliability of reputation systems is compromised by a variety of attacks. In this section, we discuss some important attacks to reputations, namely, unfair ratings attacks, dynamic peer personalities, and collusion. We also discuss three statistical-based approaches, namely, conventional, coupled, and decoupled PeerTrust PSM, used to safeguard reputation systems. Our experimental evaluations in Section 5 offer a detailed comparison of our proposed trust model with the three reputation models and with respect to the different reputation attacks discussed in this section.

4.1. Attacks to reputation systems

(i) *Unfair ratings.* An *honest* peer is one that is honest in providing service recommendations to other peers. A *malicious* peer, on the other hand, tries to subvert a system by falsely rating a bad transaction as good, and vice versa. This behavior could be due to jealousy, competition, or other malicious reasons. A malicious peer with a static personality (e.g., always behaving badly) is easily detected in the network. A *strategic* peer, on the other hand, is a malicious peer that may choose to behave honestly with some probability, in order to confuse other peers and cheat the reputation system.

(ii) *Dynamic (oscillating) peer personalities.* Some peers can exhibit a dynamic personality, that is, switching between an honest and dishonest behavior. Behavior changes can be based on the type or value of the transaction or the party involved at the other end. Reputation *milkers*, or *oscillating* peers, are one type of peer personality that builds a good reputation and then takes advantage of it to do harm.

(iii) *Collusion.* Dellarocas identifies four scenarios in which peers can intentionally try to "rig the system," resulting in biased reputation estimates [19]. In *ballot stuffing*, a colluding group inflates the colluder's reputation which then allows the colluder to use its good reputation towards other malicious motives. Similarly, in *badmouthing*, a malicious collective conspires against one or more peers in the network by assigning unfairly low ratings to the target peers, thereby, hurting their reputation. Finally, positive (and negative) *discrimination* arises when peers provide good (and poor) service to a few targeted peers. Controlled anonymity has been shown to avoid badmouthing and negative discrimination, while cluster filtering can be used to reduce ballot stuffing and positive discrimination [19].

(iv) *Sybil attacks.* Douceur has shown that unless there is a centrally trusted party, it is impractical to establish distinct identities (i.e., one identity for one entity) in a large-scale decentralized network [20]. There are no certificate authorities in a P2P network, and peers are free to generate their own identities. This availability of free identities results in the *whitewashing* attack, where a free-riding malicious peer rejoins the network under a new identity to avoid imposed penalties on its behavior. A peer can also generate a large number of identities or "Sybils" to maliciously increase the reputation of one or more of its identities. *Sybil-proofing* reputation systems is an open challenge to current reputation systems [21].

Current reputation schemes are highly vulnerable to tampering via ratings attacks including the above-mentioned unfair ratings attacks, collusion, and Sybil attacks. These vulnerabilities limit the reliability of reputations in predicting a peer's trustworthiness. By decoupling each per-peer reputation rating into a service rating and a feedback rating, a peer is accountable for its behavior both as a service provider and service recommender. Assuming that a majority of network peers are honest in nature, malicious peers that provide poor ratings to honest peers will have little agreement with the network as a whole. Their feedback credibility, consequently, will be low. In order to cheat the reputation system, an intelligent (or strategic) malicious peer may occasionally concur with the network majority and rate an honest peer correctly, thereby, getting itself a good feedback reputation. But, by ensuring that a good feedback and service reputation is difficult to gain and easy to lose, our decoupled trust approach forces a strategic malicious peer (or an oscillating peer) to constantly rebuild its reputation rating. A malicious peer spends a greater amount of time rebuilding its reputation rather than performing malicious transactions. In general, a greater number of feedback disagreements with honest peers results in a more rapid decline in maliciously acquired reputations. This discourages peers from providing incorrect service and feedback reputation ratings.

Similarly, a collusive group will give good ratings to peers within the group and false ratings to the outside network. Even one transaction with an honest peer, however, can bring down the service reputation of the malicious service provider and feedback reputations of the collusive group. Honest peers not only rate a colluding peer poorly for bad service, but also rate other colluding peers poorly for their incorrect feedback. A greater number of interactions with honest peers outside the colluding group results in a more rapid decline of reputations within the group. Additionally, by ensuring that a good reputation is difficult to gain and easy to lose, a malicious peer spends a greater amount of time colluding and

rebuilding its reputation rating rather than performing malicious transactions. Therefore, by reducing the productivity of unfair raters and colluders, our decoupled trust mechanism is able to curtail ratings attacks.

4.2. Reputation-based trust models

We now identify three statistical reputation-based trust approaches and discuss the effectiveness of these approaches in the presence of false feedback attacks. We will perform a detailed comparison of our proposed trust model with respect to these reputation models in our evaluation.

(i) *Conventional approach.* This approach is the simple technique of averaging service ratings in order to measure the trustworthiness of peers [7, 15]. The service trust rating of peer, i, denoted by $\mathrm{SR}_{\mathrm{conv}}(i)$, for the conventional approach, is derived as

$$\mathrm{SR}_{\mathrm{conv}}(i) = \frac{1}{|T_i|} \cdot \sum_{j \in T_i} s_{i,j}. \tag{6}$$

Here, $s_{i,j}$ is the rating given by peer j to i, valued at 0 (unsatisfactory) or 1 (satisfactory). T_i indicates the set of nodes that have previously transacted with i.

A simple averaging approach is flawed in several respects. A peer can easily increase its transaction volume to hide an occasional, or even, frequent misbehavior. Incremental ratings do not affect a peer once it has established a good reputation, thereby, giving a peer little incentive to behave honestly. For such a scheme, *decaying* a peer's reputation is important, that is, a peer's reputation should be representative of recent behavior rather than old behavior.

(ii) *Coupled approach.* This approach weighs the credibility of a peer's feedback by its reputation as a service provider [5, 8]. The service trust rating of peer i, denoted by $\mathrm{SR}_{\mathrm{coupled}}(i)$, for the coupled approach, is derived as

$$\mathrm{SR}_{\mathrm{coupled}}(i) = \frac{1}{|T_i|} \cdot \sum_{j \in T_i} s_{i,j} \cdot \mathrm{SR}_{\mathrm{coupled}}(j). \tag{7}$$

By taking into account the credibility of the feedback source, a coupled approach performs better than the simple averaging approach. However, a good service provider cannot be assumed to always provide good recommendations, and a bad service provider cannot be assumed to always provide bad recommendations. A peer providing honest service may provide false feedback about other peers' service due to jealousy or competition.

(iii) *PeerTrust personalized similarity measure.* PeerTrust PSM decouples feedback trust from service trust [10, 17]. PeerTrust uses a personalized similarity measure to more heavily weigh opinions of peers who have provided similar ratings for a common set of past partners. Each peer, x, in PeerTrust maintains a local copy of all feedback provided by the peer. This information is accessed up by a peer, y, that wishes to evaluate its feedback similarity with x. The root mean square or standard deviation (dissimilarity) of x's and y's feedbacks is used to compute their feedback similarity. While statistically hard to find a significant set of such overlapping past partners in a large-scale network, PeerTrust is reasonably robust compared to other approaches in handling unfair attacks and peer collusion.

In the following sections, we perform two sets of detailed experiments to evaluate the effectiveness, benefits, and scalability of our decoupled trust approach. The first set of experiments compares the effectiveness and benefits of our decoupled approach with the afore-mentioned approaches for reputation modeling, namely, conventional averaging, coupled service and feedback trust approach, and PeerTrust PSM. The second set of experiments, performed on Maze transaction logs, demonstrates the effectiveness and scalability of our reputation system on peers in that system. We also employ the Maze logs to compare the PeerTrust decoupled algorithm to our system-wide decoupling approach and highlight some of the vulnerabilities inherent in PeerTrust.

5. EXPERIMENTAL EVALUATION

We evaluate our decoupled trust model using a number of simulated and trace-driven experiments. In the following section, we describe our simulation setup, and examine the accuracy of our system in a variety of environments ranging from a network with malicious peers, strategic peers, peers with oscillating behavior, and colluding peers.

5.1. Simulation setup

We have implemented our simulator in *C* using tools included with the Stanford Graph-Base (SGB) [22]. We use graphs in the SGB platform to represent members of a P2P community. For our simulations, we use graphs of 100 to 5000 peers generated from the GT-ITM topology generator [23]. Table 1 summarizes the main parameters related to the peer model and the simulation. Our results are generated from a simulated community of 64 to 100 peers. We also run our experiments on a community of 5000 peers but observed no qualitative difference in the results.

Our network simulations proceed in cycles. We assume, for simplicity, that every peer in the network makes one transaction in each query cycle. With the help of the trust-based selection scheme, the peer requesting the service initiates a transaction with the peer that has the highest trust value. At the end of the transaction, the requesting peer gives the service provider peer a rating of either 0, indicating a bad transaction, or 1, indicating a satisfactory transaction.

The peer model includes the five types of behavior patterns discussed in section, namely, *honest*, *malicious*, *strategic*, *oscillating*, and *colluding*. Our experiments illustrate the effectiveness of our model against all these types of attacks and also show better performance when compared to the three existing reputation-based trust approaches, namely, the *conventional* approach of averaging lifetime ratings, the *coupled* approach of weighing feedback by the service reputation of its source, and PeerTrust's *personalized similarity measure.*

5.2. Metrics and experiments

In the first set of experiments, we employ three metrics in our simulations, namely, transaction success rate, trust

Table 1: Simulation parameters.

	Parameter	Value range	Default value
Peer model	Number of peers in the network	50–5000	100
	Percentage of honest peers	0–100	75
	Percentage of malicious peers	0–100	25
	Percentage of strategic peers	0–100	25
	Percentage of oscillating peers	0–100	5
	Percentage of colluding peers	0–25	10
	Percentage of peers responding to a query request	0–20	5
Simulation	Number of query cycles in one experiment	100–10000	100
	Number of experiments over which results are averaged	5	5

computation error, and computed reputation rating. First, we define the *transaction success rate* as the ratio of the number of successful transactions over the total number of transactions in the system. This metric enables us to illustrate the benefit of a trust-based peer selection scheme over a random peer selection process.

Second, we define *trust computation error* [10] to evaluate the effectiveness of our trust model against malicious and strategic peers. The trust computation error is the root mean square (RMS) of the computed trust value for all peers and the actual probability of those peers performing a satisfactory transaction, that is, 1 for good peers, and 0 for malicious peers. The RMS error is a value that ranges from 0 to 1. A lower RMS error indicates a more accurate reflection of a peer's trustworthiness.

Finally, in order to evaluate the effectiveness of our system against peer oscillation and collusion, we measure the *average reputation value* of the oscillating and colluding peers as transactions are being performed by them in the network. Periodically, measuring the reputation values enables us to measure the increase and decrease in reputation values for these peers as they perform malicious, collusive, and honest transactions.

5.3. Simulation results

We now present the results of our experiments and demonstrate the effectiveness of our decoupled trust model as compared to other reputation models. Data points in our figures represent an average of results from five randomized runs.

5.3.1. Benefit of trust-based peer selection

The first experiment demonstrates the benefits of a trust-based reputation scheme in a peer-to-peer system. A transaction is deemed successful if, at the end of the transaction, the service provider is given a rating of 1. We define the transaction success rate to be the ratio of the number of successful transactions over the total number of transactions. The experiment proceeds by having honest peers repeatedly initiate transactions. This method ensures completely honest feedback in the reputation system. As seen in Figure 2, without a trust model, there is a 50% probability of a transaction being satisfactory. When any type of trust model is used, however, a much higher success rate is achieved. All of the models have a success rate close to 100%. Clearly, a peer community with a higher transaction success rate is more productive. By avoiding transactions with untrustworthy peers, the number of unsatisfactory transactions is reduced. This experiment, therefore, indicates that having any kind of trust model in place provides a significant benefit for a peer-to-peer system.

5.3.2. Effectiveness against malicious behavior

In this experiment, our objective is to evaluate the effectiveness of our decoupled model against malicious peers as compared to other trust models. A malicious peer is defined as one that provides dishonest service and dishonest feedback at all times. We perform the evaluations after conducting 6,400 transactions over 100 peers, that is, an average of 100 transactions per peer. The percentage of malicious peers varies from 10% to 70% and the trust computation error is measured. We define the trust computation error [10] as the root mean square (RMS) of the computed trust value for all peers and the actual probability of those peers performing a satisfactory transaction, that is, 1 for good peers and 0 for malicious peers.

As Figure 3 shows, the conventional approach of averaging ratings is not effective against malicious behavior. This result occurs because the approach does not consider the credibility of the feedback provider and is particularly vulnerable to unfair ratings. We note that the correlated trust approach performs well when a small percentage of network peers is malicious. In the correlated approach, ratings assigned to the service provider at the end of a transaction are weighed by the service rating of the rater. That is, the feedback from those peers with higher service ratings will be weighed more than those with lower service ratings. When malicious peers exhibit static personalities, the assumption that a dishonest peer will provide dishonest feedback holds true. Hence, traditional reputation models that correlate service and feedback trust work well. Malicious peers are easily detected and avoided, resulting in a low trust computation error.

As malicious peers become the majority (>50%), however, they begin to overwhelm honest nodes, resulting in a significant increase in the trust computation error. Both our decoupled approach and the PeerTrust PSM models exhibit

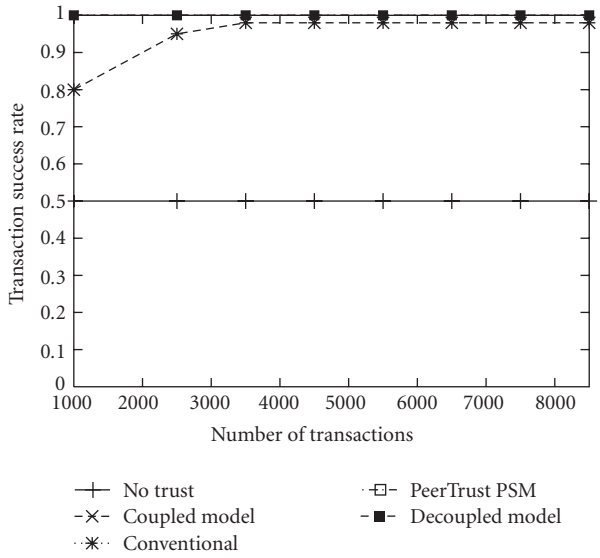

Figure 2: The benefit of a trust-based peer selection scheme.

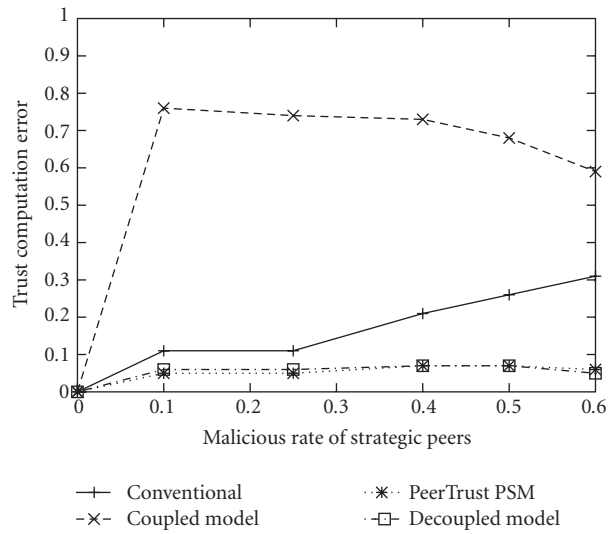

Figure 4: Trust computation error in a network with varying malicious rate of strategic peers.

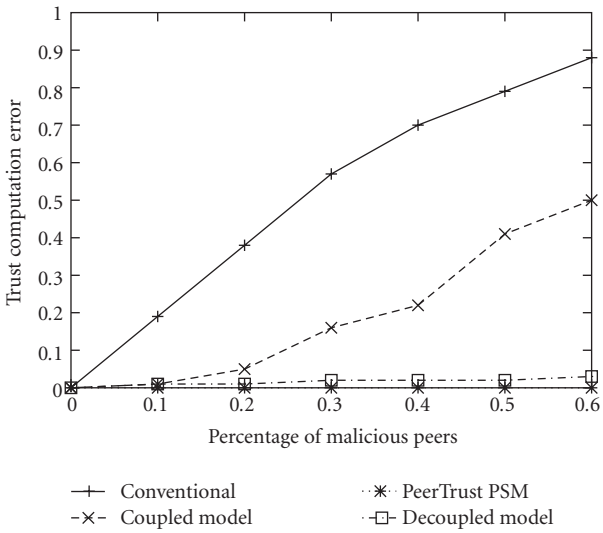

Figure 3: Trust computation error in a network with a varying percentage of malicious peers.

a trust computation error of nearly 0. This result occurs because both models are sensitive to the credibility of the feedback source. Again, as the malicious peers become the majority (> 50%), there is a slight increase in the trust computation error with our decoupled approach. This result demonstrates the natural collusion between dishonest nodes when they form a network majority.

5.3.3. Effectiveness against strategic behavior

The objective of this experiment is to evaluate the benefits of decoupling service and feedback trust compared to the correlated trust approach and the conventional approach. We introduce strategic behavior in this experiment. Strategic peers are malicious peers that try to "rig the system" by providing honest feedback, instead of dishonest feedback, in some cases. We use a network with 25% strategic peers and 75% honest peers. We vary the percent of maliciousness, X, such that a strategic peer will act maliciously, for service and feedback, for 10% to 100% of the total number of its transactions. It will provide honest feedback for the rest of its transactions. For this experiment, we define the trust computation error as the root mean square of the computed trust value for all peers and the actual probability of those peers performing a satisfactory transaction, that is, 1 for good peers, and 1-X for malicious peers.

A number of interesting observations can be made from Figure 4. Our decoupled approach and the PeerTrust PSM significantly outperform the conventional and the correlated trust approaches by reducing the number of malicious transactions and having a low trust computation error. By weighing service ratings with the credibility of the feedback source, both these approaches are able to detect strategic behavior. As the rate of malicious transactions by malicious peers increases beyond 60%, the trust computation error becomes nearly 0. An increase in the rate of malicious transactions results in a corresponding increase in the trust computation error for the conventional approach of averaging ratings. As this approach does not weigh the credibility of the feedback provider, it is unable to counter the strategic peers.

In the correlated approach, feedback from those peers with higher service ratings will weigh more than those with lower service ratings. By giving bad service but occasional good feedback, strategic peers are able to take advantage of the correlated trust assumption and fool the system. We also note that once the rate of malicious transactions increases beyond 40%, the trust computation error for the correlated model decreases. This result occurs because malicious peers are dishonest in providing service and feedback for more than 40% of their transactions, and hence confuse the system

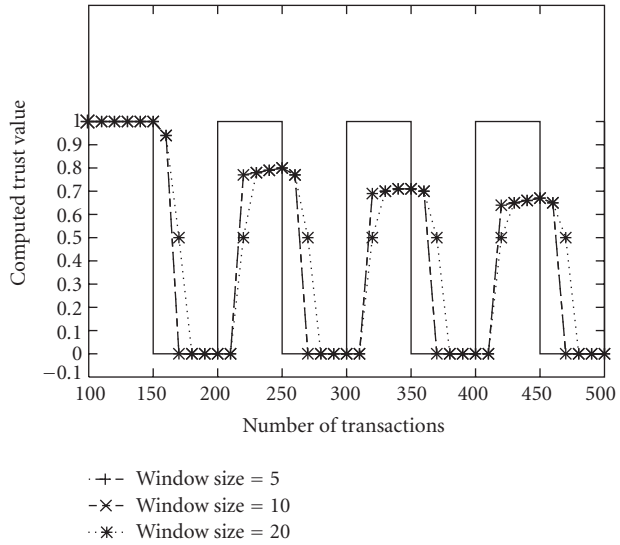

Figure 5: Effectiveness of our decoupled trust model against oscillating peer personalities.

less often. The coupled model, nonetheless, performs poorly in the presence of strategic peers. On the other hand, our decoupled model associates two ratings with each peer, one for its role as a service provider, and the other for its role as a service recommender. It is able to correctly identify peers as malicious service providers or sources of malicious feedback.

5.3.4. Effectiveness against oscillating behavior

Once a peer has established a good reputation in the network, it can abuse it by cheating occasionally. Honest peers can be subverted at any time and begin to behave badly. In order to motivate peers to perform honestly at all times, a peer's reputation must be representative of more recent behavior rather than an old established reputation. To address this issue, we calculate two reputation ratings for each peer. The first reputation rating is calculated over all the transactions undertaken by a peer, while the other is calculated over a subset of the ratings acquired by that peer in a recent window of transactions. As explained in (3.3), if the latter reputation value is smaller that the first value by a certain threshold, we assign it as the peer's new reputation rating.

The objective of this experiment is to illustrate the effectiveness of our decoupled approach against oscillating peer personalities. An oscillating peer is a malicious peer that builds and then abuses its reputation periodically. We simulate a community of 100 peers, with one oscillating peer and the rest completely honest. Each peer performs an average of 500 transactions. For this experiment, the oscillating peer is simulated to change its behavior every 50 transactions and we periodically measure the trust value of that peer as computed by an honest peer. Additionally, we vary the window size in our experiment in order to understand its effect on the computed trust value.

A number of interesting observations can be made from Figure 5. First, by employing our approach, a peer's reputation rating can drop quickly, but is hard to rebuild afterward. Second, a smaller window results in a more rapid reputation decay as compared to a larger window. With a large window, a peer is still able to take advantage of its available reputation and conduct malicious transactions. Performing continuous malicious transactions, however, results in an eventual breakdown of that peer's reputation. We also observe from the figure that a peer can never build its reputation on previous high levels once it cycles through periods of building and abusing its reputation.

5.3.5. Effectiveness against collusion

Service requesters and/or providers can intentionally "rig the system," resulting in biased reputation estimates [19]. In *ballot stuffing*, a colluding group inflates the reputations of its members, who can then leverage their good reputations for attacks. Similarly, in *badmouthing*, a malicious collective conspires against one or more participants in the network by assigning unfairly low ratings to them and, thereby, bringing down their reputation.

We ran experiments to observe the effectiveness of our model against a ballot stuffing type of collusion. The objective of the first experiment is to observe the reputation of a colluding group as computed by an honest peer. A colluding peer is a malicious peer that provides dishonest service and feedback at all times. When transacting with each other, however, two colluding peers boost each other's reputation rating. For this experiment, we use a network of 100 peers with 10% colluders and 90% honest peers. We observe similar results for a higher percentage of colluding peers. The experiment proceeds with each peer randomly performing transactions with other peers. We vary the number of transactions from 100, that is, an average of 1 transaction per peer, to 10 000, that is, an average of 100 transactions per peer. Colluding peers collude with one another at a fixed rate of 10%, 50%, or 80% of their total transactions. We monitor the average reputation of the colluding group per query cycle.

As seen in Figure 6, the service reputation rating of the colluding group declines rapidly in the presence of honest peers. In the decoupled approach, honest peers not only rate a colluding peer poorly for bad service, but also rate other colluding peers poorly for their incorrect feedbacks.

The objective of our next experiment is to determine the effectiveness of our decoupled approach against varying rates of collusion. We use a network of 100 peers with 10% colluders and 90% honest peers. The experiment proceeds as each peer randomly performs transactions with other peers. We vary the probability, X, that a colluding peer undertakes a collusive transaction. A collusive transaction implies that both the colluding requester and provider boost each other's service and feedback reputations, respectively. A colluding peer will behave dishonestly, in feedback and service, for any noncollusive transaction. We measure the average service reputation of the colluding group at the end of all the transactions.

As seen in Figure 7, a decoupled approach is more sensitive to peer collusion compared to a coupled approach. A

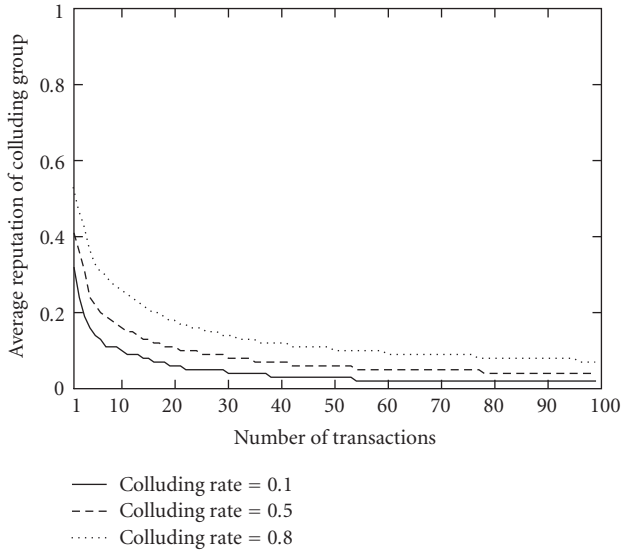

Figure 6: Effectiveness of our decoupled trust model against collusion.

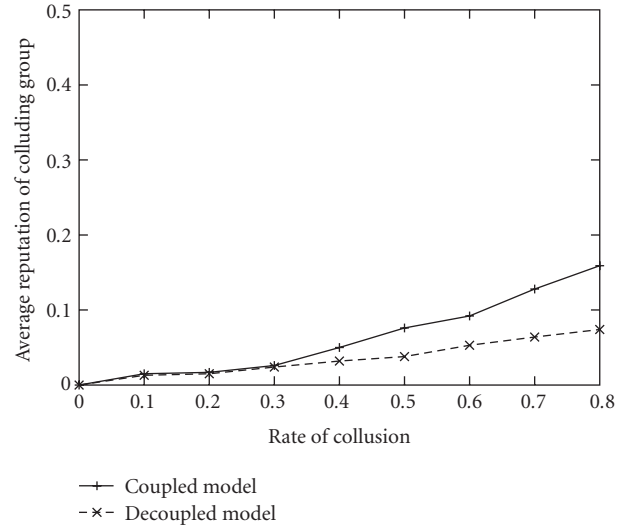

Figure 7: Effectiveness of our decoupled trust model with varying rates of collusion.

collusive rate of about 90% results in an average rating of less than 0.1 for the group. With the coupled approach, on the other hand, the colluding group is able to maintain about twice that rating for the same collusive rate. The assumption of static personalities holds true for low collusive rates as colluding peers fool the system to a lesser extent. However, as the collusive rate increases, a colluding peer exhibits a more dynamic personality, which the coupled approach is unable to handle. Also, a higher rate of collusion implies that a colluding peer spends a greater amount of time colluding and rebuilding its reputation rating rather than performing malicious transactions with honest peers. Clearly, our decoupled trust approach results in less productivity for colluding peers as compared to the coupled approach.

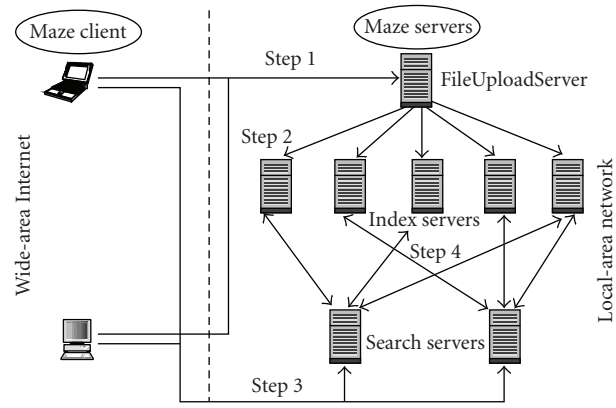

Figure 8: Architecture of the Maze file-sharing network.

6. TRACE-DRIVEN EXPERIMENTS

The previous section described our results conducted on a simulated peer community. In this section, we use transaction logs from *Maze*, a large deployed P2P file-sharing network, to drive our simulated experiments. We give a brief background on Maze [24] before presenting our results.

Maze is a popular Napster-like peer-to-peer network designed, implemented, and deployed by an academic research team at Peking University, Beijing, China. Maze is currently deployed across a large number of hosts inside China's internal network. Maze currently includes a user population of 1.4 million users and supports searches on 140 million files, 20 million of which are unique, totaling over 226TB of data. At any given time, there are over 50 000 users online, and over 1.3 million file transfers per day [24].

Maze uses a simple centralized architecture where metadata for all user files is stored on a set of central index servers. As shown in Figure 8, clients forward metadata to a FileUploadServer, which is then forwarded to index servers. Other clients issue queries to search servers. Because all transactions go through central servers managed by the team at Peking University, they monitor and log all users, metadata, and transaction records.

We perform our experiments using a sequence of transaction logs gathered from February 19, 2005 to March 24, 2005. While this log includes more than 32 million file transfers, we limit our analysis to a truncated data set that includes transactions conducted between the first 5000 users. On average, each peer performs 15 transactions.

The format of the Maze transaction logs is presented in Table 2. *UIDc* and *UIDs* refer to the file requester and provider, respectively. The *GlobalTime* and *downloadSize* fields refer to the transfer end time and the transfer size for a given session. If *downloadSize* is less than *totalSize*, it implies an incomplete transfer. *End/start* is the session time. Related work has shown that significant amount of colluding behavior can be observed from Maze transactions logs [24]. While we cannot conclusively determine the full extent of colluding behavior, we take a pessimistic approach, and define a file transfer (transaction) as failed (or malicious) if

Table 2: Maze transaction logs.

UIDc UIDs	GlobalTime	Download size	Totalsize	End/start
799 141532 302	1109 606 402	2959 816	2959 816	42
799 141532 302	1109 606 402	9240 638	9240 638	97
572 989318 395	1109 606 402	600 000	222 119 423	1
572 989318 395	1109 606 402	600 000	161 026 413	8
621 848435 436	1109 606 403	600 000	165 177 856	287
841 802538 283	1109 606 403	7491 756	7491 756	105
655 843791 333	1109 606 404	1684 920	1864 920	548
805 000278 815	1109 606 404	1800 000	3514 636	522
295 726634 900	1109 606 404	600 000	13 258 424	259 237

the *downloadsize* field equals zero or is less than half of the *totalsize* field.

We now evaluate the effectiveness of our reputation-based trust model against malicious and strategic peers in Maze. We employ the trust computation error metric to evaluate our model.

6.1. Effectiveness against malicious behavior

In this experiment, our objective is to evaluate the effectiveness of our decoupled approach against malicious peers in Maze. A malicious peer always behaves dishonestly when providing feedback and service. We identify a malicious peer as one that has failed in more than X% of its total transactions as a service provider. The extreme case is when X is 100%, that is, when a peer is defined as malicious only if it has failed in every single transaction. The total percentage of malicious peers, in this worst case scenario, is about 40% of all the peers.

We vary the percentage of malicious peers in the community by varying X, from 5% to 40%. The experiment proceeds as each peer randomly performs transactions with another peer. At the end of about 75 000 transactions covering 5000 peers, an honest peer is chosen to evaluate the trustworthiness of all peers and the trust computation error is measured.

As seen in Figure 9, when a small percentage of network peers malicious, the correlated trust approach performs as well as our decoupled ratings approach. In the correlated approach, ratings assigned to the service provider at the end of a transaction are weighed by the service rating of the rater. Hence, when peers exhibit static personalities, the assumption that a dishonest peer will provide dishonest feedback holds true and reputation models that correlate service and feedback trust work well. We note that these observations made from the Maze transaction logs are similar to the results observed in our simulated community. The trust computation error in Figure 3, however, is lower than in Figure 9 since there are more transactions per peer in the former experiment. This result confirms that the performance of a reputation system increases with an increasing number of transactions in the community. Fewer transactions per peer results in less accurate trust ratings.

This experiment also shows that when a peer community like Maze employs a trust management scheme, it observes a higher transaction success rate and greater productivity.

6.2. Effectiveness against strategic behavior

We now discuss the impact of strategic peers on trust computation. The objective of this experiment is to evaluate the effectiveness of our decoupled approach against strategic peers in Maze. A strategic peer is a malicious peer that tries to fool the reputation system by behaving honestly in some of its transactions. We use a network with 40% strategic peers and 60% honest peers. We vary the rate, X, that a strategic peer will act maliciously, for service and feedback, from 10% to 100%.

The experiment proceeds as each peer performs random transactions with other peers. At the end of about 75 000 transactions over 5000 peers, an honest peer is chosen to evaluate the trustworthiness of all peers. We define the trust computation error as the root mean square of the computed trust value for all peers and the actual probability of those peers performing a satisfactory transaction, that is, 1 for good peers, and 1-X for malicious peers.

Figure 10 clearly indicates that our decoupled approach outperforms the correlated trust approach by having a lower trust computation error. Our approach is able to correctly identify peers as malicious service providers or malicious recommenders. The trust computation error is nearly 0 in most cases. On the other hand, strategic peers are able to take advantage of the correlated trust assumption and fool the coupled model. Once the rate of malicious transactions increases beyond 50%, however, the trust computation error for the correlated model decreases. This result occurs because malicious peers become more predictable when they are dishonest in more transactions than honest.

The last two experiments evaluated the effectiveness of our decoupled trust model against malicious and strategic peers in Maze. The goal of our next set of experiments is to employ the complete Maze dataset to compare the PeerTrust decoupled algorithm with our globally decoupled trust model and highlight some of the vulnerabilities inherent in PeerTrust.

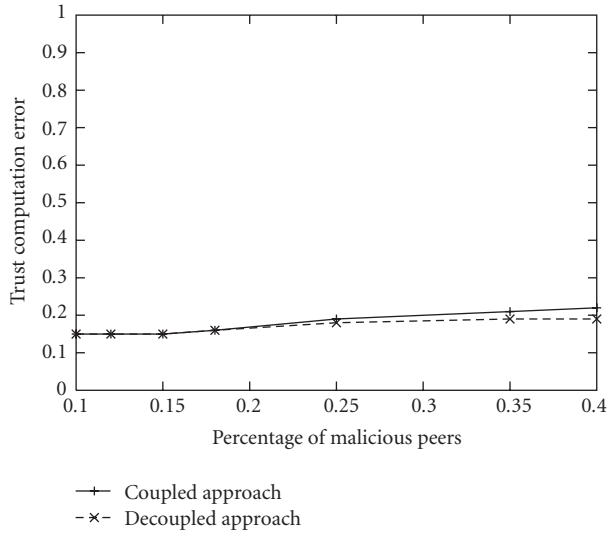

Figure 9: Trust computation error with respect to percentage of malicious peers.

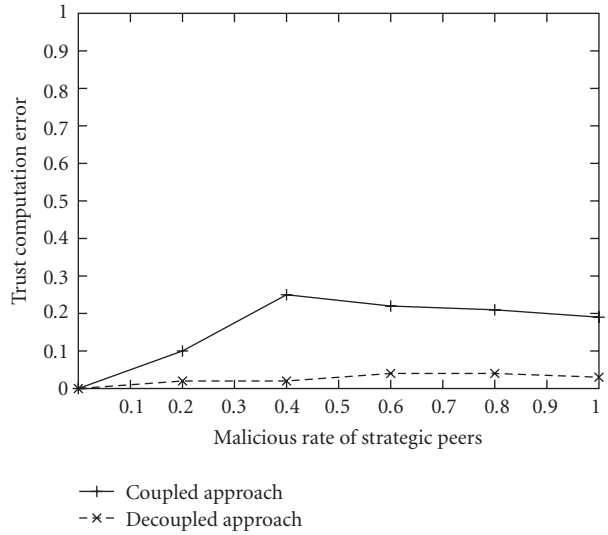

Figure 10: Trust computation error in a network with a varying malicious rate of strategic peers.

6.3. Comparison with PeerTrust's personalized similarity metric

An alternate mechanism proposed for decoupled reputations is that of PeerTrust. PeerTrust proposes the use of a personalized similarity measure (*PeerTrust PSM*) to more heavily weigh opinions of peers who have provided similar ratings for a common set of past partners. To measure the feedback credibility of any peer, x, a peer, y, computes the feedback similarity between y and x over the common set of peers with which they have interacted in the past.

While this personalized credibility provides PeerTrust with its flexibility and robustness against maliciousness, it is limited by the availability of trust data required for computing reputations. In this section, we empirically demonstrate two vulnerabilities: (a) PeerTrust's vulnerability to an overlapping set of past partners, and (b) network churn's effect on the availability of data needed for trust computations and, consequently, impact of the effectiveness and scalability of PeerTrust in large distributed P2P systems. The following experiments employ transaction logs from Maze, a P2P file-sharing network deployed in China, to compare the performance of PeerTrust PSM and our decoupled approach in overcoming the abovementioned vulnerabilities.

6.3.1. Overlapping set of past partners

In large P2P systems, finding a statistically significant set of overlapping partners can be challenging. In general, a greater number of common past partners will result in an increasingly accurate personalized feedback similarity measure for PeerTrust. But an absence of such common past partners can result in peers making arbitrary decisions among a set of candidates for which there is no information.

In the first experiment, we randomly sample 15.5 million unique transaction Maze peers. As Figure 11 illustrates, 92% of the pairs do not share even a single common partner. Approximately 4% of the pairs share only one common past partner and 2% of pairs share two common past partners. Clearly, there is not enough experience with a breadth of peers for an accurate measure of PeerTrust PSM.

6.3.2. Network churn

The second limitation of PeerTrust PSM, and of reputation systems in general, is their vulnerability to network churn. Each peer, x, in the PeerTrust algorithm maintains a local copy of all feedback provided by it. This information is accessed by a peer, y, that wishes to evaluate its feedback similarity with peer x. High peer turnover (or churn) in P2P systems impacts the availability of trust information needed to dynamically compute feedback credibilities. PeerTrust proposes an approximate trust calculation algorithm (PSM/ATC) where cached service trust and feedback similarity values are stored by each peer for reference in future transactions. However, such cached copies cannot be generated and maintained for every peer in the network.

In the second experiment, we evaluate the accuracy of the PeerTrust algorithm and our decoupled algorithm in computing trust while experiencing churn as modeled by the Gnutella churn trace. We conduct 13,517 transactions (one transaction per Gnutella peer) over a 60-hour interval (the time interval for the Gnutella trace logs). We map transaction histories of 13,517 random Maze peers to the Gnutella peers. Each time a Gnutella requester, x, wishes to evaluate a provider, y, it searches for local feedback data stored by y's previous transaction partner set, Z. If $z \in Z$ is unavailable at the transaction time, then x is unable to compute its credibility similarity with z. For this experiment, we assume that x does not hold a locally cached credibility measure for z.

We define the *percentage of successful computations* for a given provider, y, as the ratio of the number of y's past transaction partners online at the given time with respect to

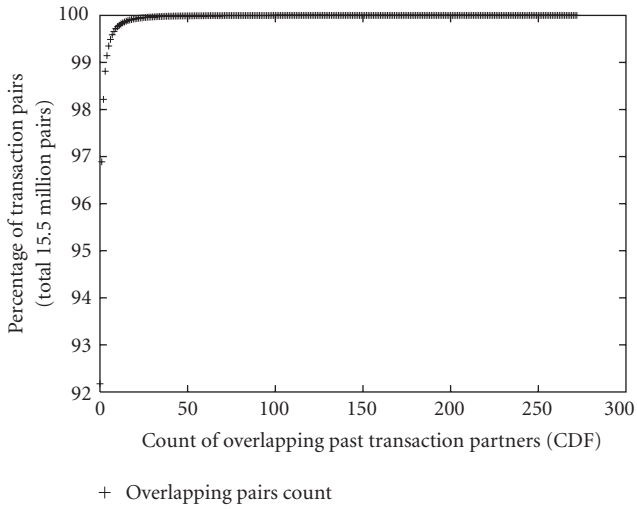

FIGURE 11: Vulnerability of PeerTrust algorithm to overlapping past transaction histories.

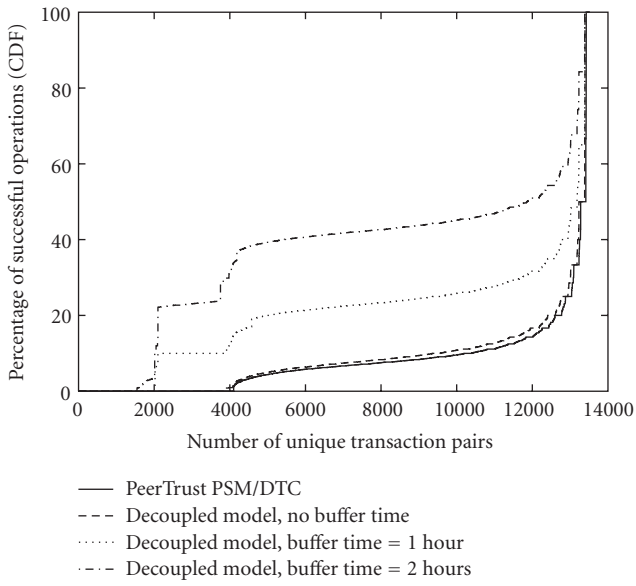

FIGURE 12: Performance of PeerTrust and our decoupled approach in overcoming network churn.

the total number of y's past transaction partners required to compute y's true trust value as observed by requester x. As Figure 12 illustrates, 30% of transactions employing the PeerTrust algorithm have 0 successful trust computations. This implies that requesters in 30% of the cases found no past partners online for dynamic trust computations. About 9% of all transactions had around 5% of the transactors online for trust computations. Only 1% of the transactions had 50% successful operations, and less than 0.70% had 100% successful operations for computing their PeerTrust PSM values.

The two experiments clearly indicate the vulnerability of the PeerTrust algorithm to a lack of trust data. On the other hand, our decoupled approach lends itself reasonably well to networks experiencing churn. At the end of each transaction, a requester, x, provides feedback updates to all peers that previously transacted with provider y. We define *buffer time* as the amount of time a requester will hold feedback update values if the target is not online. We vary the buffer time from one to two hours. A requester periodically probes targets for online availability every 60 seconds. Here, the percentage of successful operations is given by the ratio of total number of successful feedback updates over the total number of feedback updates that need to be communicated by our decoupled algorithm. As seen in Figure 12, our decoupled approach is equally vulnerable to network churn as PeerTrust. However, we overcome the problem of churn by employing a buffer time window which enables requesters to communicate their feedback update values to a greater number of target peers. As expected, the larger the buffer time window, the greater the number of successful feedback update operations.

7. DISCUSSION

Reputations are not a guaranteed solution to the problem of maliciousness in peer-to-peer networks. They only serve as a risk-management technique, reducing the chances of a peer deceiving another in an online transaction. A reputation system assumes that a peer's past behavior is indicative of its future behavior. This assumption, however, proves to be incorrect when a peer is compromised. By ensuring that a good reputation is difficult to gain and easy to lose, our decoupled approach is robust to common attack strategies. We safeguard our reputation system from malicious, colluding, and oscillating peers.

We note several implications of using a decoupled reputation system. First, the use of dual service and feedback reputations is likely to impact the way users choose with whom they transact. In a traditional reputation system, a peer requesting a service checks the service reputations of available peers to select a trustworthy peer with which to transact. Because feedback reputations are also available in our decoupled system, service providers now have an incentive to choose from whom they accept requests. A provider might avoid peers that have poor feedback reputations, since those peers might inaccurately rate the service provider's performance. In an actual system, this behavior will likely lead to the isolation of both nodes who perform bad service and nodes who give bad ratings.

The dual reputation approach imposes additional storage and management overhead for the feedback reputation. However, we note that the feedback reputation can be stored and managed using the same mechanisms as service reputations. The additional overhead is clearly justified given the increase in reputation accuracy and higher transaction success rates. Finally, while users generate service ratings, feedback ratings are generated automatically without user involvement.

8. CONCLUSIONS

Reputation systems establish peer trustworthiness in P2P networks. A number of feedback attacks, however, compromise the reliability of reputations generated by a reputation

system. In this paper, we discuss these attacks and the different reputation-based trust approaches that have been proposed to counter these attacks. After a detailed theoretical and experimental analysis of existing reputation mechanisms, we recommend that decoupled trust approaches be employed as they result in more robust reputations.

In this paper, we propose our own scalable globally decoupled trust model and demonstrate, using simulation-based and trace-based experiments, reputation improvement by removing the assumption of correlation between service quality and feedback quality. We demonstrate the effectiveness and scalability of our decoupled approach as compared to PeerTrust, an alternative mechanism proposed for decoupled reputations. Our decoupled approach incorporates global reputations of both the service provider and the requester in the computation of trust values and, in this way, makes our model more robust to peer maliciousness.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful feedback. This work is supported in part by the National Science Foundation under Career Award no. CNS-0546216 and by DARPA under the Control Plane program.

REFERENCES

[1] Gnutella, "The gnutella protocol specification v0.4," 2001.

[2] Symantec, "Vbs.gnutella worm," 2000, http://securityresponse.symantec.com/avcenter/venc/data/vbs.gnutella.html.

[3] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 207–216, Washington, DC, USA, November 2002.

[4] K. Walsh and E. G. Sirer, "Fighting peer-to-peer SPAM and decoys with object reputation," in *Proceeding of the 3rd Workshop on Economics of Peer-to-Peer Systems (P2PECON '05)*, pp. 138–143, Philadelphia, Pa, USA, August 2005.

[5] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM '01)*, pp. 310–317, Atlanta, Ga, USA, November 2001.

[6] K. Burton, "Design of the openprivacy distributed reputation system," May 2002, http://www.peerfear.org/papers/openprivacy-reputation.pdf.

[7] P. Dewan and P. Dasgupta, "Pride: peer-to-peer reputation infrastructure for decentralized environments," in *Proceedings of the 13th International Conference on World Wide Web (WWW '04)*, pp. 1212–1213, New York, NY, USA, May 2004.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 640–651, Budapest, Hungary, May 2003.

[9] B. C. Ooi, C. Y. Liau, and K.-L. Tan, "Managing trust in peer-to-peer systems using reputation-based techniques," in *Proceedings of the 4th International Conference on Advances in Web-Age Information Management (WAIM '03)*, pp. 2–12, Chengdu, China, August 2003.

[10] L. Xiong and L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[11] S. Buchegger and J. Le Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems (P2PECON '04)*, Cambridge, Mass, USA, June 2004.

[12] R. Sherwood, S. Lee, and B. Bhattacharjee, "Cooperative peer groups in NICE," *Computer Networks*, vol. 50, no. 4, pp. 523–544, 2006.

[13] Z. Zhang, S. Chen, and M. Yoon, "MARCH: a distributed incentive scheme for peer-to-peer networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1091–1099, Anchorage, Alaska, USA, May 2007.

[14] S. Marti and H. Garcia-Molina, "Identity crisis: anonymity vs. reputation in P2P systems," in *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P '03)*, pp. 134–141, Linkvping, Sweden, September 2003.

[15] eBay, "ebay," 2005, http://www.ebay.com/.

[16] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proceedings of the 5th ACM Conference on Electronic Commerce (EC '04)*, vol. 5, pp. 102–111, New York, NY, USA, May 2004.

[17] M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, pp. 422–431, Chiba, Japan, May 2005.

[18] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *Proceedings of the 3rd Symposium on Networked System Design and Implementation (NSDI '06)*, San Jose, Calif, USA, May 2006.

[19] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00)*, pp. 150–157, Minneapolis, Minn, USA, October 2000.

[20] J. Douceur, "The sybil attack," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pp. 251–260, Cambridge, Mass, USA, March 2002.

[21] A. Cheng and E. Friedman, "Sybilproof reputation mechanisms," in *Proceedings of the 3rd Workshop on Economics of Peer-to-Peer Systems (P2PECON '05)*, pp. 128–132, Philadelphia, Pa, USA, August 2005.

[22] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, ACM Press, New York, NY, USA, 1993.

[23] K. L. Calvert, M. B. Doar, and E. W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, 1997.

[24] M. Yang, H. Chen, B. Y. Zhao, Y. Dai, and Z. Zhang, "Deployment of a large-scale peer-to-peer social network," in *Proceedings of the 1st Workshop on Real Large Distributed Systems (WORLDS '04)*, San Francisco, Calif, USA, December 2004.