A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities

Li Xiong
College of Computing
Georgia Institute of Technology
lxiong@cc.gatech.edu

Ling Liu
College of Computing
Georgia Institute of Technology
lingliu@cc.gatech.edu

Abstract

Peer-to-Peer eCommerce communities are commonly perceived as an environment offering both opportunities and threats. One way to minimize threats in such an open community is to use community-based reputations, which can be computed, for example, through feedback about peers' transaction histories. Such reputation information can help estimating the trustworthiness and predicting the future behavior of peers. This paper presents a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system. There are two main features of our model. First, we argue that the trust models based solely on feedback from other peers in the community is inaccurate and ineffective. We introduce three basic trust parameters in computing trustworthiness of peers. In addition to feedback a peer receives through its transactions with other peers, we incorporate the total number of transactions a peer performs, and the credibility of the feedback sources into the model for evaluating the trustworthiness of peers. Second, we introduce two adaptive factors, the transaction context factor and the community context factor, to allow the metric to adapt to different domains and situations and to address common problems encountered in a variety of online communities. We present a concrete method to validate the proposed trust model and report the set of initial experiments, showing the feasibility and benefit of our approach.

1 INTRODUCTION

Peer-to-peer (P2P) electronic commerce (eCommerce) communities can be seen as truly distributed computing applications in which peers (members) communicate directly with one another to exchange information, distribute tasks, or execute transactions. P2P eCommerce communities can be implemented either on top of a P2P network [24, 1, 26] or using a conventional client-server platform. Gnutella is an example of P2P eCommerce communities that are built on top of a P2P computing platform. Person-to-person online auction sites such as eBay and many business-to-business (B2B) services such as supply-chain-management networks are examples of P2P communities built on top of a client-server computing architecture.

In eCommerce settings P2P communities are often established dynamically with peers that are unrelated and unknown to each other. Peers of such communities have to manage the risk involved with the transactions without prior experience and knowledge about each other's reputation. One way to address this uncertainty problem is to develop strategies for establishing trust and develop systems that can assist peers in accessing the level of trust they should place on an eCommerce transaction. For example, in a buyer-seller market, buyers are vulnerable to risks because of potential incomplete or distorted information provided by sellers. Trust is critical in such electronic markets as it can provide buyers with high expectations of satisfying exchange relationships. A recent study [9] reported results from both an online experiment and an online auction market, which confirmed that trust can mitigate information asymmetry (the difference between the amounts of information the two transacting parties possess) by reducing transaction-specific risks, therefore generating price premiums for reputable sellers.

Recognizing the importance of trust in such communities, an immediate question to ask is how to build trust. There is an extensive amount of research focused on building trust for electronic markets through trusted third parties or intermediaries [14, 23, 8]. However, it is not applicable to P2P eCommerce communities where peers are equal in their roles and there are no entities that can serve as trusted third parties or intermediaries.

Reputation systems [21] provide a way for building trust through social control without trusted third parties. Most research on reputation-based trust utilizes information such as community-based feedbacks about past experiences of

peers to help making recommendation and judgment on quality and reliability of the transactions. Community-based feedbacks are often simple aggregations of positive and negative feedbacks that peers have received for the transactions they have performed and cannot accurately capture the trustworthiness of peers. In addition, peers can misbehave in a number of ways, such as providing false feedbacks on other peers. The challenge of building a trust mechanism is how to effectively cope with such malicious behavior of peers. Another challenge is that trust context varies from communities to communities and from transactions to transactions. It is important to build a reputation-based system that is able to adapt to different communities and different situations.

Furthermore, there is also a need for experimental evaluation methods of a given trust model in terms of the effectiveness and benefits. Most traditional trust models only give an analytical model without any experimental validation due to the subjective nature of trust. There is a need of general metrics for evaluating the effectiveness and benefits of trust mechanisms.

With these research problems in mind, we develop PeerTrust, a peer-to-peer trust model for quantifying and assessing the trustworthiness of peers in P2P eCommerce communities. Our goal is to build a general trust metric that provides an effective measure for capturing the trustworthiness of peers, addresses the fake or misleading feedbacks, and has the capability to adapt to different communities and situations.

A unique characteristic of our trust model is the identification of five important factors for evaluating the trustworthiness of a peer in an evolving P2P eCommerce community: (1) the feedback in terms of amount of satisfaction a peer obtains from other peers through transactions, (2) the feedback scope, such as the total number of transactions that a peer performs with other peers in the community, (3) the credibility factor for the feedback source, (4) the transaction context factor for discriminating mission-critical transactions from less or non-critical ones, and (5) the community context factor for addressing community-related characteristics and vulnerabilities. A general trust metric is defined to combine these trust parameters in computing trustworthiness of peers (see Section 3). Most existing reputation-based trust models only take into account the first factor – the amount of satisfaction (feedbacks) that others peers have over the given peer. By analyzing a variety of common problems encountered in today's electronic markets and online communities, we demonstrate that the feedback only approach is not only inaccurate but also vulnerable when applied to evaluating the trustworthiness of a peer. In addition, we present a concrete metric to illustrate the importance of the trust assessment factors (see Section 4); validate the proposed trust model, and report the set of initial experiments, showing the feasibility and benefits of our approach (see Section 5).

2 RELATED WORK

There are a few existing online reputation systems such as the feedback system of eBay, Yahoo!Auction, and Auction Universe. Most of these systems use the single factor of feedbacks as the reputation measure. As we have pointed out, the feedback only approach cannot capture the trustworthiness of users effectively. We will analyze the common problems encountered in these communities in detail in Section 3.1 and discuss how our approach addresses the problems.

A number of reputation systems and mechanisms are proposed for online environments and agent systems in general[28, 4, 27]. Most of them assume the feedback is always given honestly and with no bias and paid little attention to handle the situation where peers may conspire to provide false ratings.

A few proposals attempted to address the issue of quality of the feedbacks. The proposal for computing and using reputation for Internet ratings by Chen et al. [10] differentiates the ratings by computing a reputation for each rater based on the quality and quantity of the ratings it gives. However, the method is based on the assumption that the ratings are of good quality if they are consistent to the majority opinions of the rating. Adversaries who submit fake or misleading feedbacks can still gain a good reputation as a rater in their method simply by submitting a large number of feedbacks and becoming the majority opinion. Dellarocas [12] proposed mechanisms to combat two types of cheating behavior when submitting feedbacks. The basic idea is to detect and filter out exceptions in certain scenarios using cluster-filtering techniques. The technique can be applied into feedback-based reputation systems to filter out the suspicious ratings before the aggregation. In comparison, our trust model is more general. We use the credibility of the feedback source as one of the basic trust parameters when evaluating the trustworthiness of peers. The credibility factor can be also used to detect fake or misleading ratings.

There is some research on reputation and trust management in P2P systems. Aberer and Despotovic [6] proposed a complaint-only trust management method for a distributed P2P system, due to the lack of incentives for submitting feedbacks. The complaint-only trust metric works in very limited cases and is over-sensitive to the skewed distribution of the community and to several misbehaviors of the system. Another work is the P2PRep proposed by Cornelli et al [11] It is a P2P protocol where servents can keep track of information about the reputation of their peers and share them with others. Their focus is to provide a protocol com-

plementing existing P2P protocols, as demonstrated on top of Gnutella. However, there is no formalized trust metric and no experimental results in the paper validating their approach. Our work differs from them in a number of ways. First, we take a coherent approach to analyze the trust problems in eCommerce communities and identify the important trust parameters in addition to the feedbacks in order to effectively evaluate the trustworthiness of peers and to address various malicious behaviors in a P2P community. Second, we also consider other context factors to allow the general trust metric to adapt to different communities under different transactional or community-specific contexts. Furthermore, we present a method for experimental evaluation of our approach in a distributed P2P environment.

Another closely related research area is collaborative filtering and recommendation systems [20, 22, 13, 15]. In collaborative filtering based recommendation systems the ratings are about static products instead of peers and there is no notion of transaction-based ratings. Although both the reputation systems and recommendation systems are based on collaborative feedbacks, reputation systems have problems that are unique and do not apply to recommendation systems. For example, the reputation systems need to reflect different transaction contexts and adapt to the changing behavior of peers, which do not apply in recommendation systems.

3 THE TRUST MODEL

In this section we first present a list of common problems observed in today's electronic markets and online communities. Then we introduce the three basic factors with two adaptive factors in evaluating trustworthiness of peers. We illustrate the general trust metric through discussion on each of the five factors and their roles in addressing the common problems.

3.1 Common Problems in Current Electronic Communities

A variety of electronic markets and online community sites have reputation management built in, such as eBay, Amazon, Yahoo!Auction, Edeal, Slashdot, Entrepreneur.com. However, to our knowledge, there are no comprehensive surveys of all sites that use reputation management systems. From our experience with online auction sites, and the survey provided by Malaga in [16], we summarize a list of common problems observed.

1. Most systems rely solely on the positive or negative feedbacks to evaluate and determine the reputation of peers. The feedback only approach suffers from inaccurate reflection of past experiences of peers in the respective community.
2. Most systems assume feedbacks are honest and unbiased and lack ability to differentiate feedbacks obtained from less trustworthy peers and those from trustworthy peers.
3. Most systems lack ability to set up various context sensitive feedback filters.
4. Most systems lack temporal adaptivity by either counting all the transaction history of a peer without decaying the importance of old transactions in the far past or only count the recent transactions.
5. Most systems do not provide incentives for a peer to rate others.

In the rest of the section we present our trust model and discuss how each of the above-mentioned problems is avoided or reduced.

3.2 Trust Parameters – Overcoming Inaccuracy and Non-flexibility

With the above problems in mind, we design and develop PeerTrust model. In PeerTrust, a peer's trustworthiness is defined by an evaluation of the peer in terms of the level of reputation it receives in providing service to other peers in the past. Such reputation reflects the degree of trust that other peers in the community have on the given peer based on their past experiences in interacting with the peer. We identify five important factors for such evaluation:

- the feedback in terms of amount of satisfaction a peer obtains through transactions with others,
- the number of transactions the peer has performed with other peers,
- the credibility of the feedbacks submitted by peers,
- the transaction context factor, addressing the impact of transaction characteristics (such as values or types of the transactions) on the trustworthiness of the peers, and
- the community context factor, addressing the impact of community-specific properties on the trustworthiness of peers.

In the rest of this section we illustrate the importance of these parameters through a number of example scenarios and address the problems with feedback-only methods. We formalize these factors, and show that they play an equally important role in evaluating the trustworthiness of a peer.

Feedback in Terms of Amount of Satisfaction

Reputation-based systems rely on feedbacks to evaluate a peer. In a P2P eCommerce community, the feedbacks in terms of amount of satisfaction a peer receives regarding its service comes primarily from the transactions other peers

have had with this peer and reflects how well this peer has fulfilled its part of the service agreement. Most existing reputation based systems uses this factor alone and computes a peer u's trust value by a summarization of all the feedbacks u receives through its transactions with other peers in the community. For example, in eBay, buyers and sellers can rate each other after each transaction (+1, 0, -1) and the overall reputation is the sum of these ratings over the last 6 months.

We can clearly see that these feedback-only metrics are flawed. A peer who has performed dozens of transactions and cheats on 1 out of every 4 cases will have a steadily rising reputation in a given time duration whereas a peer who has only done 10 transactions during the given time duration but is completely honest will be treated as less reputable if the reputation measures of peers are computed by a simple aggregation of the feedbacks they receive.

Number of Transactions

With a skewed transaction distribution, i.e. some peers have a higher transaction frequency than other peers, the trustworthiness of a peer is not captured fairly when a simple aggregation of feedbacks is used to model the trustworthiness of peers without taking into account the number of transactions. A peer may increase its trust value by increasing its transaction volume to hide the fact that it frequently misbehaves at a certain rate. So the number of transactions is an important scope factor for comparing the feedbacks in terms of amount of satisfaction among different peers. An updated metric can be defined as the ratio of the total amount of satisfaction peer u receives over the total number of transactions peer u has, i.e. the average amount of satisfaction peer u receives for each transaction.

However, this is still not sufficient to measure a peer's trustworthiness. When considering reputation information we often account for the source of information and context.

Credibility of Feedback

The feedback peer u receives from another peer v during a transaction is simply a statement from v regarding how satisfied v feels about the quality of the information or service provided by u. The trust model should consider potential threats. For example, a peer may make false statements about another peer's service due to jealousy or other types of malicious motives. Consequently a trustworthy peer may end up getting a large number of false statements. Without a credibility factor built in, this peer will be evaluated incorrectly because of false statements even though it provides satisfactory service in every transaction. Therefore, the feedback from those with better credibility should be weighted more heavily in the trust metric. Intuitively incorporating credibility factor for feedbacks represents the need to differentiate the credible amounts of satisfaction from the less credible ones in computing the reputation of peers. If we consider reputation-based trust as an important mechanism to address threats of untrustworthy peers and their malicious behaviors in the P2P community, then we can see credibility of feedbacks as a mechanism to address the risk of using potentially false feedbacks to rate peers' reputation. The concrete formula to determine credibility of peers in filing feedbacks will be discussed in Section 3.4.

Transaction Context Factor

Transaction context is another important factor when aggregating the feedbacks from each transaction as transactions may differ from one another even within the same eCommerce community. For example, if a community is business savvy, the size of a transaction is an important context that should be incorporated in the trust metric to weight the feedback for that transaction. It can act as a defense against some of the subtle malicious attacks, such as a seller develops a good reputation by being honest for small transactions and tries to make a profit by being dishonest for large transactions.

Community Context Factor

Various community contexts can be taken into account to address some of the common problems we listed such as lack of the temporal adaptivity. In a pop music sharing community, it may be desirable to only consider the recent transaction histories of a peer to reflect the current trend. However, in a business community, one may wish to use the recent transaction history of a peer and at the same time consider the historical ratings a peer receives in the past but with a lower weight than the recent history in order to evaluate the peer based on its consistent behavior. This historical behavior of a peer is one type of community context that is important to be incorporated into the trust model to give the trust system a temporal adaptivity.

The feedback incentive problem can be also alleviated by adding a reward as a community context for peers who submit feedbacks.

The community context can be also used to adapt the trust system to different communities and address problems that are specific to the community. For instance, free riding is a common challenge with online file sharing communities [7, 19] The total number of files a peer shares can be seen as a type of community context and be taken into account when evaluating the trustworthiness of a peer. With such a community context factor, a peer that shares a large number of files with the rest of the peers in the community will have a higher trust value than the free riders and alleviate the free riding problem.

3.3 General Trust Metric

We have discussed the importance of each trust parameter we identified. In this section we formalize these parameters and present a general trust metric that combines them in a coherent manner.

Let $I(u)$ denote the total number of transactions performed by peer u during the given period, $p(u,i)$ denote the other participating peer in peer u's ith transaction, $S(u,i)$ denote the normalized amount of satisfaction peer u receives from $p(u,i)$ in its ith transaction, $Cr(p(u,i))$ denote the credibility of the feedback submitted by $p(u,i)$, $TF(u,i)$ denote the adaptive transaction context factor for peer u's ith transaction, and $CF(u)$ denote the adaptive community context factor for peer u during the given period. The trust value of peer u during the period, denoted by $T(u)$, is defined as:

$$T(u) = \alpha * \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i)) * TF(u,i)}{I(u)} + \beta * CF(u) \quad (1)$$

The metric consists of two parts. The first part is the average amount of credible satisfaction a peer receives for each transaction. It may take into account transaction context factor to capture the transaction-dependent characteristics. This history-based evaluation can be seen as a prediction for peer u's likelihood of a successful transaction in the future. A confidence value can be computed and associated with the trust metric that may reflect the number of transactions, the standard deviation of the ratings depending on different communities.

The second part of the metric adjusts the first part by an increase or decrease of the trust value based on community-specific characteristics and situations. α and β denote the normalized weight factors for the two parts.

This general trust metric may have different appearances depending on which of the parameters are turned on and how the parameters and weight factors are set. The design choices depend on characteristics of communities. We argue that the first three parameters – the feedback, the number of transactions, and the credibility of feedback source are the important basic trust parameters that should be considered in computation of a peer's trustworthiness in any P2P eCommerce communities.

3.4 The Basic Metric

We first consider the basic form of the general metric by turning off the transaction context factor ($TF(u,i) = 1$) and the community context factor ($\alpha = 1$ and $\beta = 0$):

$$T(u) = \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i))}{I(u)} \quad (2)$$

This metric computes the trust value of a peer u by an average of the credible amount of satisfaction peer u receives for each transaction performed during the given period.

The feedbacks in terms of amount of satisfaction are collected by a feedback system. PeerTrust model uses a transaction-based feedback system, where the feedback is bound to each transaction. The system solicits feedback after each transaction and the two participating peers give feedback about each other based on the current transaction. Feedback systems differ with each other in their feedback format. They can use a positive format, a negative format, a numeric rating or a mixed format. $S(u,i)$ is a normalized amount of satisfaction between 0 and 1 that can be computed based on the feedback.

Both the feedbacks and the number of transactions are quantitative measures and can be collected automatically. Different from these two basic parameters, the third trust parameter – credibility of feedback is a qualitative measure and needs to be computed based on past behavior of peers who file feedbacks. Different approaches can be used to determine the credibility factor and compute the credible amount of satisfaction. One way is to solicit separate feedbacks for feedbacks themselves. This makes the problem of reputation-based trust management more complex. A simpler approach is to infer or compute the credibility value of a peer implicitly. For example, one may use a function of the trust value of a peer as its credibility factor so feedbacks from trustworthy peers are considered more credible and thus weighted more than those from untrustworthy peers. This solution is based on two assumptions. First, untrustworthy peers are more likely to submit false or misleading feedbacks in order to hide their own malicious behavior. Second, trustworthy peers are more likely to be honest on the feedbacks they provide.

It is widely recognized that the first assumption is generally true but the second assumption may not be true at all time. For example, it is possible (though not common) that a peer may maintain a good reputation by performing high quality services but send malicious feedbacks to its competitors. In this extreme case, using a function of trust to approximate the credibility of feedbacks will generate errors. This is because the reputation-based trust in PeerTrust model is established in terms of the quality of service provided by peers, rather than the quality of the feedbacks filed by peers. Therefore it cannot handle the situation of inconsistent behavior, such as peers offering good services but providing false feedbacks to jeopardize its competitors.

We believe that the study of what determines the precision of credibility of feedbacks is by itself an interesting and hard research problem that deserves attention of its own. Given that one of the design goals of the PeerTrust model is to emphasize on the roles of different trust parameters in computing trustworthiness of peers, in the rest of the paper we will use a function of trustworthiness of a peer to approximate the credibility of feedbacks filed by this peer.

3.5 Adapting the Metric Using Context Factors

We have discussed the motivations and scenarios for incorporating the adaptive context factors into our general trust metric. In this section we focus on the concrete formula when these factors are turned on in the general metric and address some of the common problems by setting proper context factors in the general metric.

Incorporating Transaction Contexts

For a business savvy community, we can incorporate the size of a transaction i in terms of dollar amount, denoted by $D(u,i)$, into the general trust metric to weight the feedback for that transaction. If we only turn on the transaction context factor, and keep the community context factor off, we have the adapted trust metric of the following form:

$$T(u) = \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i)) * D(u,i)}{I(u)} \quad (3)$$

Adding Temporal Adaptivity

The historical records of a peer's performance within a community can be an important factor for evaluation of trustworthiness of this peer in a consistent manner. When this is the case, the community context factor can be defined as the evaluation of the peer's historical behavior since the time when peer u enters the community. Through PeerTrust formula, such temporal adaptivity can be incorporated into the trust metric seamlessly. By assigning a proper weight, the past history of the peer can be taken into account but with a lower weight than the recent history. Let $I_h(u)$ denote the total number of transactions peer u has historically. If we only turn on the community context factor, and keep the transaction context factor off, we have the adaptive trust metric of the following form:

$$T(u) = \alpha * \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i))}{I(u)} + \\ + \beta * \frac{\sum_{i=1}^{I_h(u)} S(u,i) * Cr(p(u,i)}{I_h(u)} \quad (4)$$

Providing Incentives to Rate

The incentive problem of reputation systems can be addressed by building incentives into the metric through community context factor. This can be accomplished by providing a small increase in reputation whenever a peer provides feedback to others. The community context factor can be defined as a ratio of total number of feedbacks peer u give others during the given time period, denoted as $F(u)$, over the total number of transactions peer u has. The weight factors can be tuned to control the amount of reputation that can be gained by rating others. If we turn off the transaction context factor, we have the adapted metric:

$$T(u) = \alpha * \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i))}{I(u)} + \beta * \frac{F(u)}{I(u)} \quad (5)$$

Alleviating Free Riding Problem

The free riding problem in file sharing communities can be also addressed by building incentives for sharing files into the metric through community context factor. The community context factor can be defined as a ratio of total number of transactions in which peer u uploaded a file during a time period, denoted as $U(u)$, over the total number of transactions peer u has. If we turn off the transaction context factor, we have the adapted metric:

$$T(u) = \alpha * \frac{\sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i))}{I(u)} + \beta * \frac{U(u)}{I(u)} \quad (6)$$

3.6 Using the Trust Value

The value given by the trust metric gives a measure that helps peers to form a trust belief or action on other peers or to compare the trustworthiness of other peers. A higher value of $T(u)$ indicates that peer u is more trustworthy in terms of the collective evaluation of u by the peers who have had transactions with u and other community context factors.

There are several usages of the trust value in P2P eCommerce communities. First, a peer w can derive trust relationship with another peer u to determine whether to perform the next transaction with peer u. A decision rule is needed to derive a trust relationship based on the trust value and the situation. Each peer must consider to which degree the value of $T(u)$ with the associated confidence value will make it trust u given a specific situation. Different peers may have different perception over the same value. A simple rule for peer w to form a trust action on peer u can be conducted as:

$$\text{if } T(u) > T_{threshold}(w), \text{ then trust } u \quad (7)$$

where $T_{threshold}(w)$ is the threshold trust value for peer w to trust another peer.

The factors that determine the threshold $T_{threshold}(w)$ include how much peer w is willing to trust others. A more tolerant peer may have a lower threshold. It is a manifest of what is called dispositional trust [18], the extent to which an entity has a consistent tendency to trust across a broad spectrum of situations and entities. Other factors include the context of the potential transaction. For example, a more expensive transaction may require a higher threshold.

More complex decision rules can be applied and are not our focus in this paper. Interested readers may refer to [17]

for a number of models that derive a trust relationship from different parameters in an eCommerce environment.

A second usage is to compare the trustworthiness of a list of peers. For example, in a file sharing community like Gnutella, a peer who issues a file download request can compare the trustworthiness of the peers that respond to its request based on their trust value and choose the peer with the highest trust value to download the file.

Furthermore, the trust values of peers can be used to compute the aggregate trust values of a peer group in order to derive a trust relationship for a task that requires a group of peers.

4 AN EXAMPLE TRUST METRIC

We have presented a general trust metric for evaluating the trustworthiness of peers in a P2P eCommerce community. In this section we present a concrete metric with its computation to illustrate our trust model. The metric is also used to conduct experiments and study the feasibility, effectiveness, and benefits of our trust model.

This metric uses a complaint based feedback system and assumes peers are rational in a game theoretic sense, i.e. trustworthy peers do not file fake complaints and untrustworthy peers file fake complaints when they misbehave during a transaction. With a complaint system, if a peer u receives a complaint from another peer during its ith transaction, it simply means the peer receives 0 amount of satisfaction for this transaction and $S(u,i)$ is set to 0, otherwise the peer is considered to have a satisfactory performance and $S(u,i)$ is set to 1. The trust value of a peer is used as the credibility factor to weight the complaints the peer files against other peers. Thus, the total credible amount of satisfaction peer u receives can be measured as $I(u) - C(u,v) * T(v)$ where $C(u,v)$ is the total number of complaints peer u receives from v during the time period. We turn off the transaction context factor and community context factor and derive a complaint-based trust metric from equation 2:

$$T(u) = 1 - \frac{\sum_{v \in P, v \neq u} C(u,v) * T(v)}{I(u)} \tag{8}$$

We can write above equation in a matrix form as:

$$\begin{pmatrix} T(1) \\ \vdots \\ T(u) \\ \vdots \end{pmatrix} = 1 - \begin{pmatrix} \frac{C(1,1)}{I(1)} & \cdots & \frac{C(1,v)}{I(1)} & \cdots \\ \vdots & \ddots & \vdots & \ddots \\ \frac{C(u,1)}{I(u)} & \cdots & \frac{C(u,v)}{I(u)} & \cdots \\ \vdots & \ddots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} T(1) \\ \vdots \\ T(u) \\ \vdots \end{pmatrix} \tag{9}$$

The trust values can be computed by solving the above equation. We can start by setting each element in the trust value vector on the right side of the equation to a default value, say 1. As we collect more transaction histories for each peer, we repeatedly compute the trust vector until it converges.

We can easily see that this computation is very expensive in a distributed environment where there is no central database to manage the trust data and the trust data are distributed and dynamically maintained over the peer network. Every time when a peer is interested in evaluating the trustworthiness of another peer or a small subset of peers, it has to retrieve the trust data of all peers in the community. To address this high communication cost, we propose an approximate computation by maintaining a trust cache at each peer to provide a more cost-effective computation.

Each peer maintains a trust cache that keeps the trust values it has computed for other peers in the past and uses the available cached trust values as the credibility factors when computing this concrete trust metric. It thus eliminates the recursive computation. The trust value of peer u is computed at peer w as follows:

$$T(u) = 1 - \frac{\sum_{v \in P, v \neq u} C(u,v) * T_{cache}(w,v)}{I(u)} \tag{10}$$

where $T_{cache}(w,v)$ is the trust value of peer v in peer w's cache or a default trust value if it is not available in the cache.

5 EXPERIMENTAL EVALUATION

We performed three sets of initial experiments to evaluate PeerTrust approach and show its feasibility, effectiveness, and benefits. The first set of experiments evaluates PeerTrust in terms of its accuracy. The second set of experiments demonstrates the benefit of PeerTrust model when it is used in a distributed community. The last one compares an example of the adapted metrics with the basic metric to show the effects and benefits of adapting the metric using the context factors.

5.1 Simulation Setup

We implemented a simulator in Mathematica 4.0 and this subsection describes the general simulation setup, including the community model, the threat model, the transaction model, and a list of simulation parameters.

Community Model

Our initial simulated community consists of N peers. We start with a small number of peers for the first set of experiments and continue with larger number of peers. Among these peers, some are trustworthy and some are untrustworthy. The percentage of untrustworthy peers is denoted by k. An untrustworthy peer may not act malicious during every transaction. We use $mrate$ to model the frequency that

an untrustworthy peer acts malicious (see the threat model below for more detail).

Threat Model

The threat comes from the untrustworthy peers when they act malicious. A peer fails to provide the requested service or information when acting malicious during a transaction. It further files a fake complaint against the other peer to hide its own malicious behavior. A peer also generates random trust data in response to queries for the data it is responsible for storage when acting malicious for the data storage function. The overall malicious behavior percentage in the community is captured by $M = k * mrate$.

Transaction Model

The transactions are randomization-based, i.e. peers are randomly picked to perform transactions with one another or to initiate transactions and respond to transaction requests. During each transaction, a trustworthy peer always cooperates and only files a complaint when the other peer fails to provide the requested service or information. An untrustworthy peer acts malicious at a certain rate specified by $mrate$. We use Sk to model the transaction skew in the community which means some peers have a higher transaction frequency than others. Concretely, half randomly chosen peers have a transaction frequency Sk times higher than the other half peers. When $Sk = 0$, each peer has about the same transaction frequency. The average number of transactions each peer has up to current time t is denoted by $I_{Ave}(t)$.

Table 1 summarizes the main parameters which we will use throughout our simulations.

Trust Mechanism Implementation

There is no central database or server in the community. Each peer stores a small portion of the trust data (transaction history and feedbacks) and a distributed P2P data location scheme, P-Grid [5], is used for the data routing and lookup. A peer collects the trust data and conducts the trust evaluation on the fly when needed. The approximate computation is used for the trust evaluation. For further implementation details, please refer to [25].

5.2 Trust Evaluation Accuracy

The objective of this set of experiments is to evaluate the effectiveness of the trust model with basic parameters and understand how the malicious behavior and transaction skew in the community affect its performance. We compare PeerTrust approach to the conventional approach in which only the first parameter, i.e. the amount of satisfaction, is used to measure the trustworthiness of a peer.

Evaluation Metric

We define trust evaluation accuracy as a metric to evaluate how well the trust model helps peers in making trust decisions. A trust evaluation is considered correct when a trustworthy peer is evaluated as trustworthy and an untrustworthy peer is evaluated as untrustworthy. In contrast, a trust evaluation is considered incorrect if a trustworthy peer is evaluated as untrustworthy or an untrustworthy peer is evaluated as trustworthy. For the first case, the peer that requests a service may miss an opportunity to interact with a trustworthy peer. For the second case, the peer that requests a service may end up interacting with an untrustworthy peer and running into the risk of misbehavior from the other peer. The trust evaluation accuracy is defined as the ratio of the correct evaluations over the total number of evaluations.

Simulation Design

We set the total number of peers to 128 ($N = 128$). For the first experiment, we vary the malicious behavior factor in the community (M) by varying the percentage of untrustworthy peers with a fixed malicious rate of $1/4$ ($mrate = 1/4$). The transaction skew factor is set to 0 ($Sk = 0$). For the second experiment, we vary the transaction skew factor. The percentage of untrustworthy peers is set to $1/2$ ($k = 1/2$) and malicious rate of untrustworthy peers is set to $1/4$ ($mrate = 1/4$).

The experiments proceed as peers are randomly chosen to perform transactions with each other. After 6400 transactions in the community, i.e. an average of 100 transactions for each peer ($I_{Ave}(t) = 100$), 4 peers are chosen to evaluate the trustworthiness of 100 randomly chosen peers from the remaining 124 peers. A simple decision rule is used at the evaluating peer u, as shown in Equation 7, with the threshold set to be 0.8 ($T_{threshold}(u) = 0.8$), to decide whether a peer is trustworthy based on the computed trust value. We then compute the trust evaluation accuracy of the evaluations.

For comparison purpose, we use the complaint-based trust method described in [6] as an example of the conventional method. The method uses the number of complaints only as the trust measure. It only supports a binary trust output, i.e. whether the peer is trustworthy or not. We refer to this method as the complaint-only approach.

Simulation Results

Figure 1 represents the trust evaluation accuracy of the two models with respect to the malicious behavior factor in the community. We can make a number of interesting observations. First, PeerTrust and the complaint-only approach perform almost equally well when the malicious behavior factor is low. This is because the complaint-only approach relies on there being a large number of trustworthy peers who offer honest statements to override the effect of the false statement provided by the untrustworthy peers and thus achieves a high accuracy. Second, as the malicious behavior factor increases, PeerTrust stays effective while the performance of the complaint-only approach deteriorates. This can be explained as follows. When the malicious behavior factor in the community increases, the chances for

Parameter	Description
N	Number of peers in the community
k	Percentage of untrustworthy peers in the community
$mrate$	Rate that an untrustworthy peer acts malicious
M	Malicious behavior factor in the community ($M = k * mrate$)
Sk	Transaction skew factor in the community
t	The current time
$I_{Ave}(t)$	Average number of transactions each peer has up to time t

Table 1. Simulation Parameters

trustworthy peers to interact with untrustworthy peers and receive fake complaints increase. Since the complaint-only approach only uses the number of complaints for computing the trustworthiness of peers and does not take into account the credibility of the complaints, the trustworthy peers with fake complaints will likely be evaluated as untrustworthy incorrectly. On the contrary, PeerTrust uses the credibility factor to offset the risk of fake complaints and thus is less sensitive to the misbehaviors of untrustworthy peers.

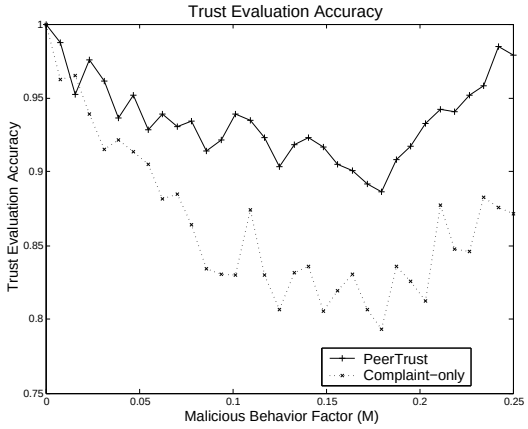

Figure 1. Trust Evaluation Accuracy with Malicious Behavior ($N = 128$, $Sk = 0$, $I_{Ave}(t) = 100$)

Figure 2 represents the trust evaluation accuracy of the two models with respect to the transaction skew factor in the community. When the transaction skew factor increases, PeerTrust stays effective while the performance of the complaint-only approach deteriorates. This demonstrates the importance of the number of transactions when computing the trustworthiness of peers. The complaint-only approach is very sensitive to the transaction skew because it does not take into account the number of transactions in their trust metric.

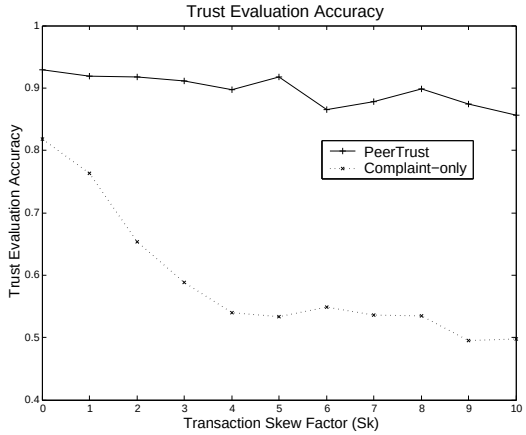

Figure 2. Trust Evaluation Accuracy with Transaction Skew ($N = 128$, $M = 0.125$, $I_{Ave}(t) = 100$)

5.3 Benefit of the Trust Mechanism

This set of experiments simulates an application scenario where peers use the trust mechanism to compare the trustworthiness of peers and choose the peer with the highest trust value to interact with. The objective is to use the scenario to show how a trust mechanism benefits a P2P community.

Evaluation Metric

We define transaction success rate as a metric to measure the productivity and security level of a community. A transaction is considered successful if both of the participating peers cooperate. Otherwise one or both of the peers is faced with the risk of malicious behaviors from the other peer. The successful transaction rate is defined as a ratio of the number of successful transactions over the total number of transactions in the community up to a certain time. A community with a higher transaction success rate has a higher productivity and a stronger level of security. We expect that a community with an effective trust mechanism should have

a higher transaction success rate as peers are able to make informed trust decisions and avoid unreliable and dishonest peers.

Simulation Design
We set the number of peers to be 1024 ($N = 1024$), the percentage of untrustworthy peers to be 1/2 ($k = 1/2$), the malicious rate of an untrustworthy peer to be 1/4 ($mrate = 1/4$), and the transaction skew to be 0 ($Sk = 0$).

The experiment proceeds by repeatedly having randomly selected peers initiating transactions. A selected peer (source peer) initiates a transaction by sending out a transaction request and a certain number of randomly selected peers respond. The source peer needs to select a peer from the peer candidates to perform the transaction. The selection process differs in a community that has no trust mechanism and a community that has a trust mechanism. In the first case, the source peer randomly selects a peer from the peer candidates. In the second case, the source peer evaluates the trustworthiness of each peer in the peer candidates and selects the peer with the highest trust value. The two peers then perform the transaction and cooperate or defect according to their trustworthiness status and malicious rate. We record whether the transaction succeeds and compute the transaction success rate when the experiment proceeds.

We simulated three communities, the first with PeerTrust mechanism, the second with the complaint-only mechanism for comparison, and the last without any trust mechanism for reference.

Simulation Result
Figure 3 shows the transaction success rate with the average number of transactions each peer has at current time. The graph presents a number of interesting observations. First, we see an obvious gain of the transaction success rate in both communities equipped with a trust mechanism. This confirms that supporting trust is an important feature in a P2P community. Second, the complaint-only trust metric is not as effective as PeerTrust. This also matches the results from the previous experiment. Third, it is also interesting to observe that the transaction success rate increases over the time in the community with PeerTrust and then stays fairly stable. This is because as peers interact with each other over the time, peers successfully select trustworthy peers to interact with. The untrustworthy peers are deterred from participating in transactions. On the other hand, the transaction success rate increases first and then drops before going stable in the community with complaint-only method. This is because peers make wrong evaluations due to the limitations of the method and in turn choose untrustworthy peers to interact with before the system gets stable.

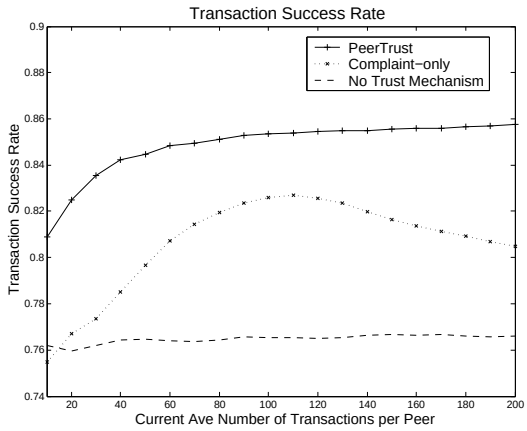

Figure 3. Transaction Success Rate ($N = 1024$, $M = 0.125$, $Sk = 0$)

5.4 Adapting the Metric Using Context Factors

The goal of this experiment is to use an example adapted metric to show the effects and benefits of adapting the metric using the adaptive context factors.

Simulation Design
We set the number of peers to be 128 ($N = 128$), the percentage of untrustworthy peers to be 1/2 ($k = 1/2$), the malicious rate of an untrustworthy peer to be 1/4 ($mrate = 1/4$), and the transaction skew factor to be 0 ($Sk = 0$).

The experiment proceeds similarly as the previous one except that one randomly selected peer u acts maliciously with $mrate$ for a period of time and then starts acting trustworthy at a certain time point. We record the computed trust value of peer u when the experiment proceeds.

We compare the basic example metric defined in Equation 8 with the time window set to be 50 ($I(u) = 50$), which only counts the feedbacks from the recent 50 transactions, and the adapted trust metric defined in Equation 4, which takes into account the historical behavior of peers from the very beginning by using the community context factor.

Simulation Result
Figure 4 shows the computed trust value of peer u by the basic metric and the adapted metric. The dashed line shows the changing point of the peer. We can see both trust values computed by the two metrics show an increase corresponding to the change of the peer. However, with the historical community context factor, the adapted metric shows a more gradual change as it also takes into account the historical behavior of the peer so a peer cannot simply increase its trust value quickly by acting good for a short recent period. Another observation is that the adapted trust metric is more consistent than the basic metric and has fewer spikes that

indicate an inaccurate trust value.

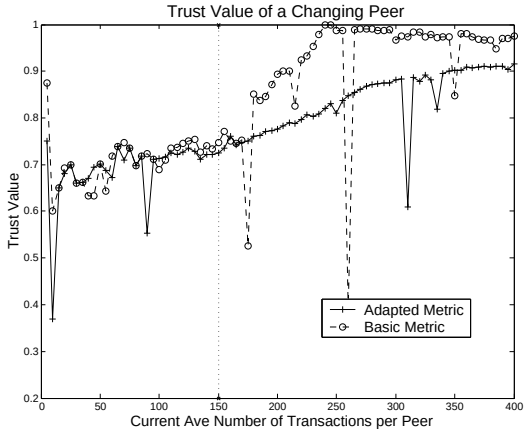

Figure 4. Trust Value of a Changing Peer ($N = 128$**,** $k = 1/4$**,** $mrate = 1/4$**)**

6 CONCLUSION

We presented an adaptive reputation-based trust model for P2P electronic communities. We identified the five important trust parameters and developed a coherent trust metric that combines these parameters for quantifying and comparing the trustworthiness of peers. We also reported a set of initial experimental results, demonstrating the feasibility, effectiveness, and benefits of our trust model.

Our research on PeerTrust continues along several directions. First, we are investigating different threat models of P2P eCommerce communities and exploring mechanisms to make PeerTrust model more robust against malicious behaviors such as collusion among peers. We are also interested in combining trust management with intrusion detection to address concerns of sudden and malicious attacks. Second, we are interested in testing the approach with real workload data. Finally, we are working towards incorporating PeerTrust into two P2P applications that are currently under development in Georgia Tech, namely PeerCQ [3] and HyperBee [2].

Acknowledgement

We would like to thank Karl Aberer and Zoran Despotovic for providing us the source code of P-Grid and the simulator of their trust model for our experimental comparison. This research is supported partially by a NSF ITR grant. The second author would like to acknowledge the partial support from a NSF CCR grant, a DOE SciDAC grant, and a DARPA ITO grant.

References

[1] Gnutella. http://www.gnutella.com.

[2] HyperBee. http://www.hyperbee.com.

[3] PeerCQ. http://disl.cc.gatech.edu/PeerCQ.

[4] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *33rd Annual Hawaii International Conference on System Sciences (HICSS-33)*, 2000.

[5] K. Aberer. P-grid: A self-organizing access structure for p2p information systems. In *Cooperative Information Systems, 9th International Conference, CoopIS 2001*, 2001.

[6] K. Aberer and Z. Despotovic. Managing trust in a peer-to-peer information system. In *2001 ACM CIKM International Conference on Information and Knowledge Management*, 2001.

[7] E. Adar and B. A. Hubeman. Free riding on gnutella. *First Monday*, 5(10), 2000.

[8] Y. Atif. Building trust in e-commerce. *IEEE Internet Computing*, 6(1), 2002.

[9] S. Ba and P. A. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS Quarterly*, 26(3), 2002.

[10] M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *3rd ACM Conference on Electronic Commerce*, 2001.

[11] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a P2P network. In *Eleventh International World Wide Web Conference*, 2002.

[12] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *2nd ACM Conference on Electronic Commerce*, 2000.

[13] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *1995 ACM Conference on Human Factors in Computing Systems (CHI '95)*, 1995.

[14] S. Ketchpel and H. García-Molina. Making trust explicit in distributed commerce transactions. In *16th International Conference on Distributed Computing Systems*, 1996.

[15] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Kerlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to usenet news. 40(3), 1997.

[16] R. A. Malaga. Web-based reputation management systems: Problems and suggested solutions. *Electronic Commerce Research*, 1(4), 2001.

[17] D. W. Manchala. E-commerce trust metrics and models. *IEEE Internet Computing*, 4(2), 2000.

[18] D. H. McKnight and N. L. Chervany. The meanings of trust. Technical Report WP9604, University of Minnesota Management Information Systems Research Center, 1996.

[19] L. Ramaswamy and L. Liu. Freeriding: A new challenge for peer-to-peer file sharing systems. In *36th Annual Hawaii International Conference on System Sciences (HICSS-36)*, 2003.

[20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *CSCW '94*, 1994.

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

[22] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *1995 ACM Conference on Human Factors in Computing Systems (CHI '95)*, 1995.

[23] M. Shepherd, A. Dhonde, and C. Watters. Building trust for e-commerce: Collaborating label bureaus. In *Second International Symposium on Electronic Commerce, ISEC 2001*, 2001.

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

[25] L. Xiong and L. Liu. PeerTrust: A trust mechanism for an open peer-to-peer information system. Technical Report GIT-CC-02-29, Georgia Institute of Technology, College of Computing, 2002.

[26] J. E. Youll. Peer to peer transactions in agent-mediated electronic commerce. Master's thesis, Massachusetts Institute of Technology, 2001.

[27] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *Cooperative Information Agents, 7th International Conference, CoopIS 2000*, 2000.

[28] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(8), 2000.