A Novel Feedback Based Fast Adaptive Trust Model for P2P Networks

Anupam Das and M. Mahfuzul Islam
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.
Email: anupamdas@cse.buet.ac.bd,mahfuz@cse.buet.ac.bd

***Abstract*—Peer-to-peer (P2P) networks have shown great potentials in providing a wide range of services starting from simple file sharing to distributed computing. However, P2P systems present ominous threats due to its anonymous and dynamic nature. One feasible way to minimize the threats is to evaluate the trust and reputation of the interacting peers. Trust models have often been deployed in determining the trust of peers in the network with the view to avoiding the malicious ones. Most of the existing trust models can successfully isolate malicious peers when the peers behave in a predictable way while others even fail to do so. On the other hand, these models suffer greatly when peers start to behave in a unpredictable way. Moreover, these models are ineffective in providing quick response to a peer's dynamic personality. To cope with such strategically altering behavior we present in this paper, a feedback based fast adaptive trust model which takes into account various factors in computing the trust of peers including recent trend, historical trend, sudden deviation of trust and so on. Simulations show that our model compared to other existing models can effectively identify and isolate the dynamic behavioral change of malicious peers.**

***Index Terms*—Peer-to-Peer(P2P), trust, reputation, trust model, malicious peer.**

I. Introduction

P2P systems have evolved a lot in the recent decade and are now being used in many distributed applications including file-sharing [1], digital content delivery and even grid computing [2]. The main features of P2P system is that as peers join the system, the total ability of the system increases because all of the peers provide not only their own data, but also their storage space and computing power. However, the dynamic, anonymity and autonomy of P2P systems have made it a challenge for researchers to make P2P a secured environment. Conventional security policies such as authentication and authorization can not be applied to P2P network. As a result, trust and reputation have been used in assessing peers before making any transaction decision. A reputation based trust system [3] collects, distributes and aggregates trust information about participants to assist peers in determining the level of trust it should place upon other peers.

Most of the existing trust models use transaction history of peers to evaluate the peers in the system. However, the main challenge arise when malicious peers start to strategically alter their behavior in a way that benefits them such as behaving maliciously after attaining high reputation. Another well recognized threat is the shilling attack where malicious peers submit false feedback and form collusive groups with each other to boost their own ratings while disparaging non-malicious peers. Almost all the trust models fail to cope with these strategic malicious behaviors. So, a dynamic trust model that can quickly adapt to any abrupt change in peer behavior is required to sustain the reliability of P2P systems.

With this research problem in mind, we propose a dynamic trust model which can effectively detect sudden strategic alteration in malicious behavior by peers. We have also considered various aspects in computing trust like transaction context factor, incentive factor, decay of trust with time and sudden deviation in trust.

The rest of the paper is organized as follows. Section II reviews the existing related works. In section III we formally introduce our trust model. To show the effectiveness of our model, we present simulations in section IV. Finally we conclude the paper.

II. Related Work

In this section we give a brief description of the related research work done on P2P trust models. We categorize the related works mainly into two groups: local trust models and global trust models. In local trust models [4], [5] a peer collects trust information from some pre-trusted super peers. So, in local trust model the scope of information aggregation is limited to only a few number of peers. In case of global trust models [6], [7], [8], [9] a peer has a view of the network which is wider than its own experience. In other words, in global trust models a peer aggregates feedback from all the peers who have ever interacted with the target peer.

EigenTrust [6] computes the global trust of each peer by considering the peer's entire history of uploads, and presents a distributed and secured method for computing the global trust values based on power iteration. EigenTrust's main flaw is that it relies on some pre-trusted peers i.e., peers that are fully trusted by all the other peers in the system and the selection of these pre-trusted peers is not feasible from a practical point of view. PeerTrust [7] uses personalized similarity measure to compute the credibility of the recommenders and it uses this credibility measure to weight each feedback submitted by the recommenders. PeerTrust's drawback is that it has to retrieve all the transactions within the recent time window (which may contain a large number of transactions) to compute the trust

978-1-4244-8389-1/10/$26.00 ©2010 IEEE

of a peer. So the trust evaluation process is computationally and spatially expensive. The concept given by Dou *et al.* [10] is similar to EigenTrust, but without the consideration of using pre-trusted peers. Dou's model reduces iteration cost and punishes malicious behavior, but doesn't consider the punishment for dishonest feedback. The Bayesian network based trust model [4] assumes that trust is multi-facet and peers need to develop differentiated trust in different aspects of other peers' behavior. This model uses Bayesian probability to calculate credibility which is based on the peer's own subjective judgement and this can sometimes be biased by the peer calculating it.

The recommendation model provided by *Wang et al.* [11] combines feedback mechanism and punishment mechanism to prevent deception and conspiracy by malicious peers, but this model uses average successful transaction as local trust rating which fails to give any relative time significance to the transactions. FCTrust [12] proposes a feedback credibility measure for quantifying and evaluating the trustworthiness of participants. It uses transaction density and similarity measure to define the credibility of any recommender. However, FCTrust's main drawback is that it stores all the transactions performed within a time frame which might be a significantly long period and this results in storage overhead. SFTrust [13] uses two trust metric, one for service trust and the other for feedback trust, with the view to take full advantage of all the peers' service abilities even in the presence of fluctuating feedbacks. *Wen et al.* [14] combine both direct trust and indirect trust in computing the trust value of a peer. In computing indirect trust the model uses both direct and referral credibility. In case of referral credibility it uses the transitive law of trust. However, the model's main flaw lies in the assumption that all paths to the target peer have equal weight which is not logical since longer the path the more possibility that a malicious peer lies in the path.

The models defined in [4], [6], [7], [8], [9], [10], [11], [12], [13] do not address a critical aspect of trust theory which is decay of trust value with the lapse of time. Since, at present, the network is highly dynamic and unpredictable, trust values should decay with the lapse of time. Some models [14], [15] address the above mentioned issue by incorporating their own decay functions. However, these models fail to simulate a realistic decay function which should have a small decay rate at the initial phase while having a higher decay rate as more and more time elapses. We have incorporated such decay function in our trust model.

III. Our Trust Model

Our model is a global trust model that combines both local trust and recommendation from other peers to compute the global trust value of a given peer. Our trust model also provides effective measure against sudden strategic alteration in malicious behavior. First of all, we will discuss the basic components used in calculating the trust of a peer and at the same time we will briefly describe the mathematical expressions used for each component. Many of these components have previously been discussed in [7], [13], [12], [11], [14] but none of them can cope with the true scenarios faced in real life. So we have redefined some of the components and at the same time defined some new ones. Moreover, none these models have considered all the components that we have integrated in our trust model. For the following sections we assume that peer ***u*** (known as evaluator) is calculating the trust value of peer ***v*** (known as the target peer).

A. Satisfaction

The satisfaction metric combines the local satisfaction level of all the transactions a peer makes with another peer by using an exponential averaging update function. By using exponential averaging update function we not only reduce the storage overhead but also at the same time assign time relative significance to the transactions i.e., recent transactions are given higher importance than past transactions. It is a natural tendency to rely more on the recent transactions than past transactions. Let, $Sat_{t,i}(u,v)$ represent the amount of satisfaction peer *u* places upon peer *v* based on its service up to *i* transactions in the *t-th* time interval. The satisfaction update function is defined as follows-

$$Sat_{t,i}(u,v) = \alpha Sat_{cur} * TCF_{t,i}(u,v) + (1-\alpha) Sat_{t,i-1}(u,v) \tag{1}$$

where $Sat_{t,0}(u,v) = S_{t-1,last}(u,v)$ and $Sat_{0,0}(u,v) = 0$. Here, $TCF_{t,i}(u,v)$ represents the transaction context factor. Transaction context factor is important when we are aggregating feedback from different types of transactions as different communities may have different requirements. For example some might give emphasis on the speed of transaction while others might give emphasis on transaction data size. We have therefore, defined $TCF_{t,i}(u,v)$ as the weighed summation of different factors related to transaction quality. Let, $\{a_1, a_2, ..., a_n\}$ represents the different factors related to transaction quality and let $\{w_1, w_2, ..., w_n\}$ be their respective weight where $\sum_{k=1}^{n} a_k = 1$ and $\sum_{k=1}^{n} w_k = 1$. Then $TCF_{t,i}(u,v)$ is defined as follows-

$$TCF_{t,i}(u,v) = \sum_{k=1}^{n} w_k a_k \tag{2}$$

Sat_{cur} represents the satisfaction value for the most recent transaction. Here, we have used a binary feedback system where a peer rates other peers with either 1 or 0 based on whether the transaction is satisfactory or not.

$$Sat_{cur} = \begin{cases} 0, & \text{if recent transaction is not satisfactory} \\ 1, & \text{if recent transaction is satisfactory} \end{cases} \tag{3}$$

The weight α changes based on the accumulated deviation $\zeta_{t,i}(u,v)$.

$$\alpha = threshold_{\alpha} + c\frac{\xi_{t,i}(u,v)}{1+\zeta_{t,i}(u,v)} \tag{4}$$

$$\xi_{t,i}(u,v) = |Sat_{t,i-1}(u,v) - Sat_{cur}| \tag{5}$$

$$\zeta_{t,i}(u,v) = c\xi_{t,i}(u,v) + (1-c)\zeta_{t,i-1}(u,v) \tag{6}$$

$\zeta_{t,0}(u,v) = \zeta_{t-1,last}(u,v)$ and $\zeta_{0,0}(u,v) = 0$. Here c is a user define constant which controls to what extent we will react to the recent error ($\xi_{t,i}(u,v)$). The $threshold_\alpha$ is used to prevent α from saturating to a fixed value. For example if α ever becomes 0 then it will totally ignore the satisfaction level of the current transaction (Sat_{cur}) and as a result $Sat_{t,i}(u,v)$ (from eqn 1) will always remain constant irrespective of whatever kind of transactions follow. Initial value of α is set to 1 and $threshold_\alpha$ is set to 0.1.

B. Similarity

Similarity metric measures to what extent two peers are similar. One way to compute similarity is to determine the difference in satisfaction rating over the common set of interacted peers and then use that computed difference to define the degree of similarity. Let,$TS(x)$ represent the set of peers with whom peer *x* has made transaction. Then $CS(u,v) = TS(u) \bigcap TS(v)$ denotes the set of peers with whom both peer *u* and peer *v* have interacted. The difference in satisfaction rating between peer *u* and peer *v* denoted as $Diff_{t,i}(u,v)$ is defined as follows-

$$Diff_{t,i}(u,v) = \sqrt{\frac{\sum_{x \in CS(u,v)} (Sat_{t,i}(a,x) - Sat_{t,i}(o,x))^2}{|CS(u,v)|}} \tag{7}$$

To compute similarity between peer *u* and peer *v* ($S_{t,i}(u,v)$), we first compare $Diff_{t,i}(u,v)$ with a difference deviation constant(τ) and then assign similarity according to the following function-

$$S_{t,i}(u,v) = \theta * \left(\frac{e - e^{Diff_{t,i}(u,v)}}{e-1}\right) \tag{8}$$

$$\theta = \begin{cases} 1, & \text{if } Diff_{t,i}(u,v) < \tau \\ \frac{\tau}{Diff_{t,i}(u,v)}, & \text{otherwise} \end{cases} \tag{9}$$

If however, $|CS(u,v)| = 0$ then $Diff_{t,i}(u,v) = 0$ and $S_{t,i}(u,v) = 0.5$ (default value).

C. Credibility

Credibility is a measure of the degree of reliability a peer has upon the feedback of a recommender. Peer that provides good service might not necessarily provide honest feedback so credibility metric is used to filter out inaccurate feedbacks submitted by different recommenders. Let, $Cre_{t,i}(u,v)$ represent the credibility of peer *v* from peer *u*'s perspective.

$$Cre_{t,i}(u,v) = \begin{cases} Cre_{t,i-1}(u,v) + \frac{1 - Cre_{t,i-1}(u,v)}{\mu}, \\ \quad \text{if } S_{t,i}(u,v) < Cre_\theta \\ Cre_{t,i-1}(u,v) - \frac{Cre_{t,i-1}(u,v)}{\psi}, \\ \quad \text{otherwise} \end{cases} \tag{10}$$

where Cre_θ represents the threshold value for credibility while μ represents rewarding factor and ψ represents punishment factor and both can be changed dynamically depending on the system requirement. In equation (10) we see that when the similarity rating is less than Cre_θ credibility increases and otherwise it decreases. Here $\mu, \psi \in \Re$ and we must make sure that that $\mu > \psi$ because the degree of punishment should be greater than that of reward. By doing so we incorporate the principle of "slow rise and quick decline".

D. Local Trust

Local trust represents the portion of the trust that a peer computes from its own experience with the target peer. Let, $LT_{t,i}(u,v)$ represent the local trust that peer *u* has upon peer *v* up to *i* transactions in the *t-th* interval. We have used the satisfaction measure along with the credibility of the target peer to define local trust.

$$LT_{t,i}(u,v) = Sat_{t,i}(u,v) * Cre_{t,i}(u,v) \tag{11}$$

where $Cre_{t,i}(u,v)$ represents the credibility of peer *v* from peer *u*'s perspective.

E. Recommendation

To a get a better view of the target peer, a peer should utilize the experience gained by other peers in the network. To do so, a peer requests other peers to provide recommendation about the target peer. The requesting peer takes into consideration the credibility of the recommender while receiving recommendation. Let, $R_{t,i}(u,v)$ represent the recommendation that peer *u* computes about peer *v*.

$$R_{t,i}(u,v) = \frac{\sum_{x \in W - \{u\}} Cre_{t,i}(u,x) * LT_{t,i}(x,v)}{|W - \{u\}|} \tag{12}$$

Here $W = IS(v)$, represents the set of peers who have ever interacted with peer *v*. If $|W - \{u\}| = 0$ then $R_{t,i}(u,v) = 0$.

F. Recent Trust

Recent trust reflects the recent trend in peer behavior. We have computed recent trust as the weighted summation of local trust and recommendation. The weight of local trust adjusts dynamically based on interaction count. Local trust is given higher weight as more and more interactions occur with the target peer i.e., the evaluator becomes more confident about its own experience over the experience gained by others. Let, $RT_{t,i}(u,v)$ represent the recent trust peer *u* has upon peer *v*.

$$RT_{t,i}(u,v) = \beta * LT_{t,i}(u,v) + (1 - \beta) * R_{t,i}(u,v) \tag{13}$$

where β represents the weight of local trust which can be calculated as follows-

$$\beta = \lambda^\kappa \tag{14}$$

$$\kappa = \sqrt{\frac{Mean_t(u,v)}{N_t(u,v)}} \tag{15}$$

$$Mean_t(u,v) = \frac{\sum_{x \in W - \{u\}} N_t(x,v)}{|W - \{u\}|} \tag{16}$$

Here $W = IS(v)$, represents the set of peers who have ever interacted with peer *v* and $N_t(u,v)$ represents the number of interactions peer *u* has conducted with peer *v* in the *t-th* interval. Then, $Mean_t(u,v)$ represents the mean number of interactions that other peers (peers other than *u*) have conducted with peer *v*. With λ ($\lambda < 1$) being a constant β depends on the value of κ and as $N_t(u,v)$ increases (compared

to $Mean_t(u,v)$) the value of κ decreases which in turn increases the value of β. So we see that as own experience increases the value of β also increases. If $|W-\{u\}|=0$ then $Mean_t(u,v)=0$ and if $N_t(u,v)=0$ then $\beta=0$.

G. Past Trust

Past trust reflects long term behavioral pattern and it is built from past experience. As recent trust reflects recent trend and with the elapse of time recent trend become historical trend. We can compute past trust from previous recent trusts, but instead of storing previous recent trusts we have defined past trust using an exponential averaging function. Let, $PT_{t,i}(u,v)$ represent the past trust that peer *u* has about peer *v*.

$$PT_{t,i}(u,v)=\rho*(PT_{t,i-1}(u,v)+RT_{t,i-1}(u,v)) \tag{17}$$

where $\rho(0\leq\rho\leq 1)$ is the forgetting factor (determining to what extent we are forgetting the older experiences). With past trust present malicious peers cannot suddenly wipe out their bad reputation and start behaving good.

H. Global Trust

Global trust reflects expected behavior of the target peer and it is obtained by aggregating both recent and past behavior. Let, $GT_{t,i}(u,v)$ represent the global trust of peer *v* from peer *u*'s perspective. Global trust is calculated as follows-

$$GT_{t,i}(u,v)=\begin{cases}0, & \text{if neither RT nor PT available}\\ \eta RT_{t,i}(u,v)+(1-\eta)PT_{t,i}(u,v), & \\ & \text{if both RT and/or PT available}\end{cases} \tag{18}$$

Initially η is set to 0.5 but η adjusts dynamically based on the difference of recent and past trust (deviation factor ε). Here we are allowing a peer with the scope of improving its global trust after an unintentional past accident by increasing η when recent trust exceeds past trust by a given threshold (ε).

$$\eta=\begin{cases}\eta+0.1, & \text{if } RT_{t,i}(u,v)-PT_{t,i}(u,v)>\varepsilon\\ \eta-0.1, & \text{if } RT_{t,i}(u,v)-PT_{t,i}(u,v)<-\varepsilon\\ \eta, & \text{if } -\varepsilon<RT_{t,i}(u,v)-PT_{t,i}(u,v)<\varepsilon\end{cases} \tag{19}$$

I. Decay model

Due to easy access and abundant resource peers today are highly unpredictable and as a result their trust value should decline in absence of interaction. In other words, if a peer does not make a transaction with the network for a long period, its trust value will gradually degrade with the lapse of time. Now, in our model local trust depends on satisfaction and recommendation is calculated from the local trust of recommenders, so we apply a decay function on satisfaction metric. Similarly past trust also under goes decay with the progress of time and this can be modeled by the properties of a decay function. The decay functions are given as follows-

$$\widehat{Sat}_{t,i}(u,v)=\frac{Sat_{t,i}(u,v)}{(\chi)^{\Delta t}} \tag{20}$$

$$\widehat{PT}_{t,i}(u,v)=\frac{PT_{t,i}(u,v)}{(\chi)^{\Delta t}} \tag{21}$$

$$\Delta t=t_{current}-t_{previous} \tag{22}$$

where $\widehat{Sat}_{t,i}(u,v)$ and $\widehat{PT}_{t,i}(u,v)$ represent the modified value of satisfaction and past trust after decay respectively. Here, $\chi(\chi>1)$ represents a system dependent decay rate and its controls how quickly the value will degrade. $\Delta t(\Delta t\geq 0)$ represents the interval between the current interaction and the last interaction.

J. Deviation Factor

Malicious peers have evolved with time and they are now much smarter than they were before. Malicious peers tend to incorporate dynamic personality where they first build up their reputation and then milk it. They can also oscillate between good and malicious nature with the intent of not getting noticed by others while ripping benefit for themselves. So, some kind of measurement is required to handle such fluctuation. For this we have defined a deviation factor which is a measure of how much deviation we are willing to tolerate. Deviation factor handles trust fluctuation by keeping record of sudden misuse of trust by peers. Here, we introduce the component *accumulated trust fluctuation* (denoted as $ATF_{t,i}(u,v)$).

$$ATF_{t,i}(u,v)=\begin{cases}ATF_{t,i-1}(u,v)+\frac{RT_{t,i}(u,v)-PT_{t,i}(u,v)}{\omega}, & \\ \quad\text{if } RT_{t,i}(u,v)-PT_{t,i}(u,v)>\varphi & \\ ATF_{t,i-1}(u,v)+RT_{t,i}(u,v)-PT_{t,i}(u,v), & \\ \quad\text{if } RT_{t,i}(u,v)-PT_{t,i}(u,v)<-\varphi & \\ ATF_{t,i-1}(u,v), \text{otherwise} & \end{cases} \tag{23}$$

where φ represents the tolerated margin of error in the evaluation of trust and ω ($\omega>1$) represents the punishment factor for sudden rise in trust. In equation (23) we are considering both sudden rise and fall of trust by peers. However, we are punishing more for sudden fall than sudden rise as we are encouraging peers to raise their trust through effective transactions. Initial value of accumulated trust fluctuation $ATF_{0,0}(u,v)=0$.

Deviation factor uses the accumulated trust fluctuation to measure the deviation in peer behavior. Deviation factor (denoted $DF_{t,i}(u,v)$) is defined by the following equation-

$$DF_{t,i}(u,v)=\begin{cases}\cos(\frac{\Pi*ATF_{t,i}(u,v)}{2*maxAT}), & \text{if } ATF_{t,i}(u,v)<maxAT\\ 0, & \text{otherwise}\end{cases} \tag{24}$$

where $maxAT$ represents the maximum tolerable trust fluctuation. Initial value of deviation factor $DF_{t,i}(u,v)=1$.

K. Incentive factor

One of the main threats of P2P system is the free riding problem [16] where peers only utilize resources without providing any service or feedback. Several remedies have been suggested for the incentive problem of reputation systems in [17]. We are incorporating one of the easiest solutions which is to assign a metric for the amount of uploads/feedbacks/services a peer provides. The incentive factor is defined as follows-

$$IF_{t,i}(u,v)=\frac{SP(v)}{N(v)} \tag{25}$$

where $SP(v)$ represents the total number of feedback/upload/services provided by peer *v* and $N(v)$ represents the total number of transaction requested by peer *v*.

L. Trust Metric

The trust metric is the overall value assigned to a peer after considering all the components which we have discussed. Let, $T_{t,i}(u,v)$ represent the final trust value peer *u* places upon peer *v*.

$$T_{t,i}(u,v) = \delta * [GT_{t,i}(u,v) * DF_{t,i}(u,v)] + (1-\delta) * IF_{t,i}(u,v) \quad (26)$$

where δ is a user defined constant. A peer will use equation (26) to select the target peer with the highest trust value.

IV. Experimental Evaluation

In this section we will illustrate the performance of our trust model and show its effectiveness and robustness against malicious behavior. First, we evaluate its trust computation accuracy in the presence of opportunistic malicious peers. Then in the next set of experiments we demonstrates the effectiveness of our model compared to other existing trust models under different scenarios.

A. Simulation Setup

We have developed our simulation in java programming language using JBuilder and the discrete event simulation toolkit SimJava. In this section we will describes the general simulation settings and the performance evaluation index used for comparison.

Our simulated environment contains P peers where P is set to 100 in almost all the experiments. However, in one experiment we have varied the value of P to show the scalability of our trust model. In our simulation there are mainly two types of peers- good and malicious. Good peers cooperate in providing both good service and honest feedback where malicious peers provide both ineffective service and false feedback. The percentage of malicious peers in the network is controlled by the parameter *mrate* which is varied in different experiments.

In our simulation we have studied mainly three types of malicious behavioral pattern namely- noncollusive, collusive and strategic alteration. In noncollusive setting malicious peers deceive other peers and provide false feedback. The collusive setting is similar to the noncollusive setting except that the malicious peers form conspiracy groups among themselves to boost each others rating to the external community. We have used the parameter *cm* to denote the percentage of malicious peers forming collusive groups. In the strategically altering setting a malicious peer may occasionally decide to cooperate in order to fool the system. We use the parameter *mfback* to model the rate of dishonest feedback by a malicious peer.

Table I summarizes the the different parameters related to the simulation setting and trust computation. The table also lists the default value of the different parameters used. These default values have been empirically tuned.

For comparing the performance of our trust model with other existing trust models we compute the commonly used evaluation index called success rate (SR). SR is described as the ratio of the number of successful transactions to the total number of transactions. Since different trust models compute trust differently, other evaluation index such as trust computation error is not suitable for comparison. Rather the relative ranking of peers based on their computed trust value is comparable and thats why we only evaluate SR for comparison with other models. In the following section we will determine SR against the variation of *mrate*, *mfback* and *cm*. All the experimental results are averaged over 10 runs of the experiment and for simplicity we assume that $TCF = 1$ and $\delta = 1$ i.e., we are ignoring both transaction context factor and incentive factor.

B. Trust Computation Accuracy

In this set of experiments we will show how accurately our model can compute the trust value of peers even in the presence of various malicious activities. The experiments starts as peers randomly start interacting with each other and after each peer performs on average 500 transactions a good peer is randomly selected to compute the trust value of all the other peers. The trust computation error is calculated by taking the root-mean-square (RMS) of the computed trust value of all the peers against their binary rating which is 1 for good peers and 0 for malicious peers. So a low RMS indicates better performance. This experiment is performed under both noncollusive and collusive setting.

In the first experiment we set *mfback* to 100% and we vary the percentage of malicious peers (*mrate*). Fig. 1 represents the trust computation error with respect to *mrate* under noncollusive (*cm*=0%) and collusive (*cm*=100%) setting. In both cases we see that our model becomes more effective in the presence of large percentage of malicious peers. The reason behind this is that the credibility measure effectively filters out the dishonest feedbacks submitted by malicious peers or collusive groups.

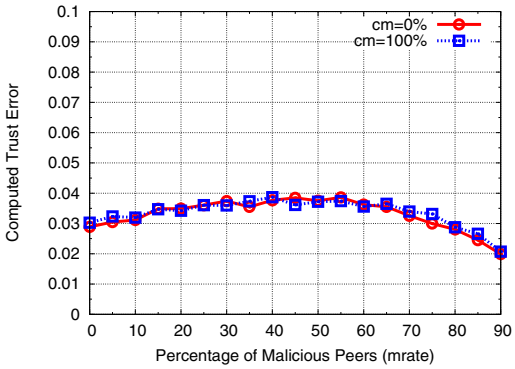

Fig. 1. Trust computation error with respect to percentage of malicious peers in noncollusive and collusive setting.

In the second experiment we set *mrate* to 40% and vary *mfback*. Again the experiment is conducted under both noncol-

TABLE I
SIMULATION PARAMETERS SETTINGS

	Parameter	Description	Default value
Environment Setting	P	# of peers in the system	100
	mrate	% of peers malicious in the system	40%
	mfback	% of time a malicious peers gives false feedback	100%
	response	% of peers who respond to a transaction request	5%
	cm	% of malicious peers forming a collusive group	0%
Trust Computation Setting	α	contribution factor for recent satisfaction	1
	$threshold_\alpha$	minimum threshold of α	0.25
	c	user defined constant	0.9
	β	weight of direct trust in computing recent trust	0.5
	λ	constant factor	0.85
	ρ	forgetting factor	0.9
	η	weight of recent trust in computing global trust	0.5
	ε	deviation factor used in global trust	0.2
	Cre_θ	threshold for credibility update	0.5
	μ	reward factor for credibility	20
	ψ	punishment factor for credibility	4
	χ	decay rate	1.15
	τ	difference rating deviation	0.25
	φ	tolerated margin of error	0.25
	ω	punishment factor for sudden rise in trust	2
	$maxAT$	threshold for accumulated misused trust	10
	δ	contribution of incentive factor	0.3

lusive (*cm*=0%) and collusive (*cm*=100%) setting. From Fig. 2 we see that in the noncollusive setting the error is slightly lower when *mfback* varies from 0% to 45% signifying that malicious peers can confuse the system slightly when they behave cooperatively. For collusive setting we see a much steady result which signifies that our model can successfully discard false rating provided by collusive groups.

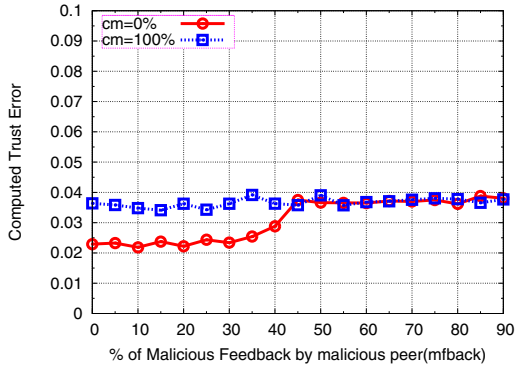

Fig. 2. Trust computation error with respect to percentage of malicious response by malicious peers in noncollusive and collusive setting.

In the third experiment we set *mrate* to 40% and *mfback* to 100% and vary *cm*. Again from Fig. 3 we see that our trust model can effectively discards the impact of the collusive interaction due to the sensitive credibility metric.

C. Comparison with other Trust Models

In this set of experiments we will compare the efficiency of our trust model against other existing trust models. In these experiments a peer performs selective transactions rather random transactions where it first computes the trust values of the responding peers and then selects the peer with the highest trust value for interaction. A transaction is considered successful if the participating peer is cooperative i.e., it is a good peer. In all the experiments we compute success rate as the

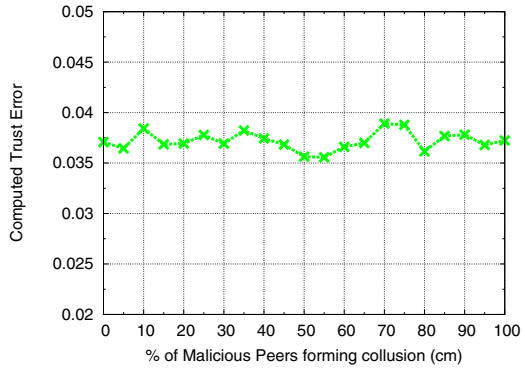

Fig. 3. Trust computation error with respect to percentage of malicious peers forming collusive group.

evaluation index under different scenario. In each experiment we perform a total of 100 iterations where in each iteration each peer in the system initiates one transaction. Transactions initiated by malicious peers have been discarded from the calculation of SR. Since the responders to a transaction request is generated at random we take the mean of 10 experiments for each scenario. We compare our model with SFTrust [13], FCTrust [12], P2P Recommendation Trust model (for short we will use Reco-Trust) [11], Trust model of users' behavior (for short we will use User-Trust) [14] and PeerTrust [7].

First, we set *mfback* to 100% and *cm* to 0% and compute SR against the variation of *mrate*. From Fig. 4 we see that both our model and PeerTrust show superiority over the remaining trust models as the amount of malicious peers in the network increase beyond 40%. Due to the ease of availability and accessibility networks today are full of malicious peers of different characteristics, especially the Internet is home to a large number of malicious peers. So, in such adverse environment our model would provide a means of secured communication.

Now we will observe the impact of collusion on SR. For

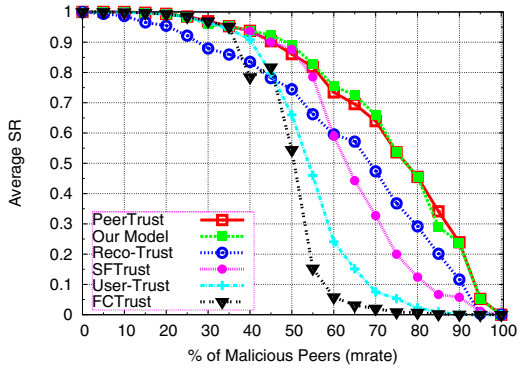

Fig. 4. Comparing our model with other existing trust models in terms of average SR against *mrate*.

this experiment we want the number of malicious peers in the network relatively high because the impact of collusion will then be noticeable. So, we set *mate* to 60% and *mfback* to 100%. Fig. 5 represents the computed SR against *cm*. As we can see that again, PeerTrust and our model show superiority over others. The main reason behind this is the credibility measure which filters out false feedbacks. Here false high ratings come from peers (namely malicious peers) with low credibility as a result they have no impact on SR. The low credibility itself results from the personalized similarity measure.

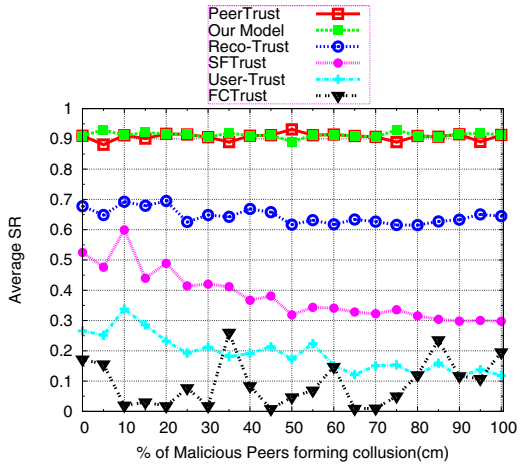

Fig. 5. Comparing our model with other existing trust models in terms of average SR against *cm*.

Malicious peers tend to fool other peers by oscillating between good and malicious nature. So, in the third experiment we will analysis the impact of *mfback* on SR. In this experiment we set *mrate* set to 60% while *cm* is set to 0%. Fig. 6 represents the computed SR against *mfback*. From the figures it is evident that our model out performs all the other trust models by quite a significant margin in presence of high percentage of malicious peers and PeerTrust suffers the most. The reason for this is that our model keeps track of sudden rise and fall of trust by peers and penalizes any peer showing frequent trust fluctuation. Thus, our model can effectively resist strategic alteration in malicious behavior.

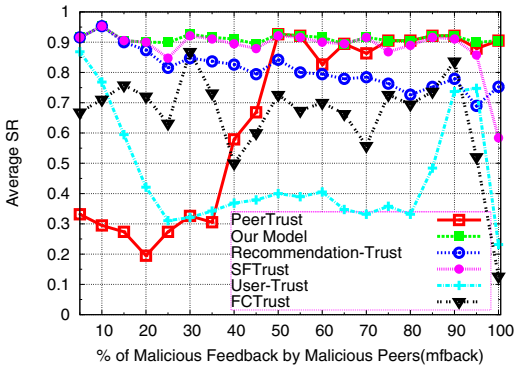

Fig. 6. Comparing our model with other existing trust models in terms of average SR against *mfback*.

In the last experiment we will test the stability of the trust models under oscillating behavior. Here we run the experiment for a total of 500 iterations with *mrate* set to 50% and *cm* set to 0%. But we divide the 500 iterations into four equal slots and so each slot contains 125 iterations. Malicious peers oscillate between benevolent and malicious nature from one slot to the next starting with benevolent nature. Then we compute the number of times malicious peers are selected as service providers to transactions initiated by only good peers. From Fig. 7 we see that in the initial slot malicious peers are selected numerous times. This is because in the first slot they start off by behaving good so there is no reason to reject them. But in the following slots this number should decline as we now know their true nature. We see that our model performs best in isolating the malicious peers from providing service compared to the other trust models.

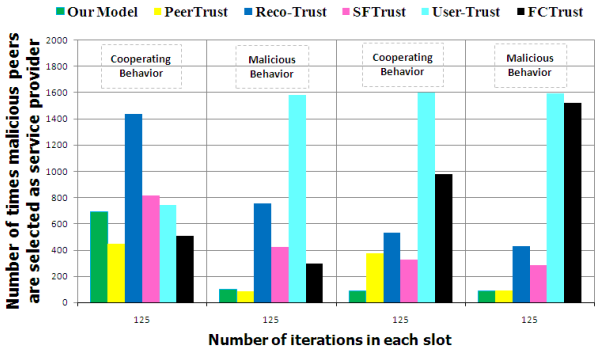

Fig. 7. Comparing our model with other existing trust models in terms of number of times malicious peers are selected as service providers.

D. Testing Scalability

The objective of this experiment is to show that our trust model is scalable with the increase of peers in the network. For this purpose we computed average SR against the number of peers in the network. We have set *mrate* to 40%, *mfback*

to 100% and *cm* to 0%. Fig. 8 shows that the computed SR against the number of peers in the network and from the figure it is evident that our model remains unaffected in terms of performance as the number of peers in the network increase.

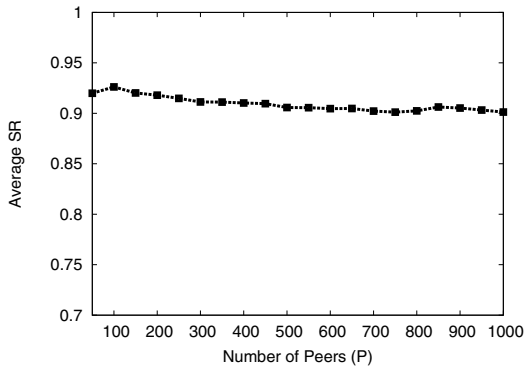

Fig. 8. Scalability of our trust model.

E. Decay of trust in absence of interaction

This experiment emphasizes the importance of attenuation of trust with lapse of time in absence of interaction. Now a days peers show highly dynamic personality as result trust values should not remain static in absence of interaction. By including a time decay model we are establishing the principle "the more recent the transaction the more reliable it is". Fig. 9 shows the comparison among no decay and the attenuation function defined in [14] with our decay model. It is observable that the attenuation function defined in [14], has higher declination rate at the beginning than later one. However, it should be the reverse i.e., initially the declination rate should be smaller and as more and more time passes the declination rate should increase which our decay function incorporates.

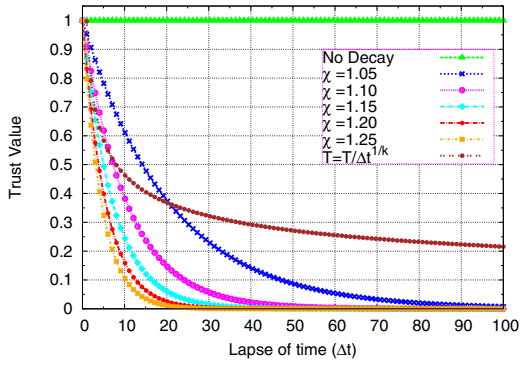

Fig. 9. Decay of Trust Value with lapse of time.

V. Conclusion

In this paper we have presented a fast adaptive dynamic trust model which can effectively identify the strategic change in malicious behavior in P2P system. We have also provided comprehensive mathematical analysis of the different factors related to the evaluation of trust. Simulation results show, compared with other existing trust models our trust model is more robust and effective against attacks from malicious peers.

References

[1] (2000) Gnutella. [Online]. Available: http://www.gnutella.com

[2] Q. Zhang, Y. Sun, Z. Liu, X. Zhang, and X. Wen, "Design of a Distributed P2P-Based Grid Content Management Architecture," in *Proceedings of 3rd Annual Conference on Communication Networks and Services Research (CNSR)*. Halifax, Novia Scotia, Canada: IEEE Computer Soc. Press, 2005, pp. 339–344.

[3] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[4] Y. Wang and J. Vassileva, "Bayesian Network-Based Trust Model," in *Proceedings of IIEEE/WIC International Conference on Web Intelligence (WI)*. Halifax, Canada: IEEE Computer Soc. Press, October 2003, pp. 372–378.

[5] F. Cornelli, E. Damiani, S. D. Capitani, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a p2p network," in *In Proceedings of the 11th World Wide Web Conference (WWW)*. Hawaii, USA: ACM, May 2002, pp. 376–386.

[6] 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 World Wide Web conference (WWW)*. Budapest, Hungary: ACM Press, 2003, pp. 640–651.

[7] L. Xiong and L. Li, "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.

[8] Z. Runfang and H. Kai, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–476, 2007.

[9] 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)*. ACM, 2005, pp. 422–431.

[10] D. Wen, W. Huaimin, J. Yan, and Z. Peng, "A Recommendation-Based Peer-to-Peer Trust Model," *Journal of Software*, vol. 15, no. 4, pp. 571–583, 2004.

[11] X. Wang and L. Wang, "P2P Recommendation Trust Model," in *Proceedings of IEEE 8th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE Computer Soc. Press, 2008, pp. 591–595.

[12] J. Hu, Q. Wu, and B. Zhou, "FCTrust: A Robust and Efficient Feedback Credibility-Based Distributed P2P Trust Model," in *Proceedings of IEEE 9th International Conference for Young Computer Scientists (ICYCS)*. IEEE Computer Soc. Press, 2008, pp. 1963–1968.

[13] Y. Zhang, S. Chen, and G. Yang, "SFTrust: A double Trust Metric Based Trust Model in Unstructured P2P Systems," in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (ISPDP)*. IEEE Computer Soc. Press, 2009, pp. 1–7.

[14] L. Wen, P. Lingdi, L. Kuijin, and C. Xiaoping, "Trust model of Users' behavior in Trustworthy Internet," in *Proceedings of IEEE WASE International Conference on Information Engineering (ICIE)*. IEEE Computer Soc. Press, 2009, pp. 403–406.

[15] Y. Zhang, K. Wang, K. Li, W. Qu, and Y. Xiang, "A Time-decay Based P2P Trust Model," in *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2. IEEE Computer Soc. Press, 2009, pp. 235–238.

[16] E. Adar and B. A. Huberman, "Free Riding on Gnutella," *First Monday*, vol. 5, no. 10, 2000.

[17] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation Systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.