International Journal of Digital Content Technology and its Applications
Volume 4, Number 7, Octber 2010

An Efficient Feedback-based Trust Model for Pervasive Computing

[1*]Jianguo Chen, [2]Stefan D. Bruda
[1*]*College of Information Engineering, China Jiliang University, China, jgchen@cjlu.edu.cn*
[2]*Department of Computer Science, Bishop's University, Canada, bruda@cs.ubishops.ca*
doi: 10.4156/jdcta.vol4.issue7.22

Abstract

In pervasive computing environments, pervasive devices should collaborate effectively such that the vision of pervasive computing will come true. However, without trust, pervasive devices cannot collaborate effectively. Distributed trust systems (DTS) may support trust and thus foster collaboration in hostile pervasive computing environments. The challenge for DTS is how to aggregate the local trust values without a centralized storage and management facility. Another challenge is that such systems should have minimal overhead in terms of computation, infrastructure, storage, and message complexity, especially in real-time embedded systems and wireless networks. We focus on the computation and storage of DTS in this paper. The computation time in many existing DTS will increase with the amount of data to be processed. The efficiency of the existing DTS ranges from O (t^2) to O (log t). We propose an efficient trust model based on the incremental Proportional-Integral-Derivative (PID) controller and the efficiency of our model is O(1) that uses only four data and needs four multiplications, four additions and two subtractions. Besides, this paper reviews some trust models and discusses the robust effects for some important issues such as consistent node behavior, sudden fluctuations in node behavior and unintentional errors. The modified version of our trust model for slow increasing for new entity and quick dropping for misbehaving is also investigated.

Keywords*: Trust, Incremental PID Controller, Security, Pervasive Computing, Distributed System.*

1. Introduction

Pervasive computing, also as known as ubiquitous computing, is the third wave of computing technologies to emerge since computers first appeared. Pervasive computing is moving beyond the personal computer to everyday devices with embedded technology and connectivity as computing devices become progressively smaller and more powerful. In pervasive computing environments, pervasive devices should collaborate effectively such that the vision of pervasive computing will come true. However, without trust, pervasive devices cannot collaborate effectively.

Since the computation capacity and storage of pervasive devices are usually limited, the applications for pervasive computing have put in evidence the importance of minimizing the computation time and storage for any additional feature, in particular security and trust services.

Security and trust are two properties of modern computing systems that are the focus of much recent interest. They play an increasingly significant role in the requirements for modern computing systems. A significant amount of security research has been devoted to addressing the vulnerabilities in computer systems. Trust is intensively studied too and it is not intended to replace any of security research, but it is intended to complement the existing security mechanisms.

One technology to realize the pervasive computing environments is to build a peer-to-peer (P2P) network. To foster the collaboration or cooperation among peers, i.e., pervasive devices, the reputation systems in P2P networks have gained immense popularity in the recent years.

The computation time in many existing DTS will increase with the amount of data to be processed. For example, some DTS such as eBay will keep the data for each entity for six months. The efficiency of the existing DTS ranges from O (t^2) to O (log t).

We propose an efficient trust model based on the incremental Proportional-Integral-Derivative (PID) controller and the efficiency of our model is O(1) that uses only four data and needs four multiplications, four additions and two subtractions. Besides, we review some trust models and discuss the robust effects for some important attacks such as sudden fluctuations in node behavior and tolerating unintentional errors. The deviation of our trust model for slow increasing for new entity and quick dropping for misbehaving is also investigated.

The rest of this paper is organized as follows. Section 2 briefly discusses P2P computing, trust definitions and formal expressions. The PID controller or algorithm is introduced in section 3. Section 4 is the analysis of trust models, followed by our trust model based on the incremental PID (IPID) controller in section 5. Common attacks and issues to be studied are discussed in section 6. Section 7 is the experiment evaluations of our model and shows the effects against some attacks. We present related work in section 8 and conclude in section 9.

2. P2P computing, trust definitions and formal expressions of trust features

2. 1. P2P computing

P2P computing is the sharing of computer resources and services by direct exchange between the peers in the systems. All the peers in P2P networks have the same role and there is no peer with a special responsibility to monitor or supervise the network behavior so that each peer acts both as a client and as a resource provider. The peers in a P2P network can be anything, ranging from handhelds to powerful desktop computers, i.e., pervasive devices.

The resources and services in P2P networks include the exchange of information, processing cycles, cache storage and disk storage for files. However, amidst the benefits, there are some risks. One main risk is that each peer has to interact with some unrelated and unknown peers without a trusted third authority. Non-authority and no central server open the door to possible misuses and abuses. P2P networks need to be robust and fault tolerant since it is almost a certainty that selfish or malicious peers will be joining the network.

One way to minimize such risks in the P2P community is to use feedback-based reputation systems that can help estimate trustworthiness and predict future behavior of peers.

2. 2. Trust definitions

Recent works demonstrated that using a feedbacks-based trust system is an effective way for peers to minimize the threats and protect the system [1, 2]. Feedbacks will provide an efficient and effective way to build trust relationship amongst peers in open environments.

Bolton, Katok and Ockenfels [4] studied and compared electronic reputation mechanisms with and without online feedback. They concluded that mechanisms with feedbacks are more efficient than those without feedbacks. The reputation system with feedbacks can help participants decide whom or what to trust, encourage trustworthy behavior, and deter dishonest participation.

The key to the success of reputation systems is to set up a proper trust model. Then, what is trust? In computer science, trust is not a new research topic, spanning areas as diverse as security and access control in computer networks, reliability in distributed systems, game theory and agent systems, and policies for decision making under uncertainty. Trust is mandatory to support the dependable implementation of distributed protocols and services. In addition, it is much more fundamental when the services are implemented through collaboration of mutually untrustworthy entities. However, the concept of trust in these different communities varies in how it is represented, computed, and used [5].

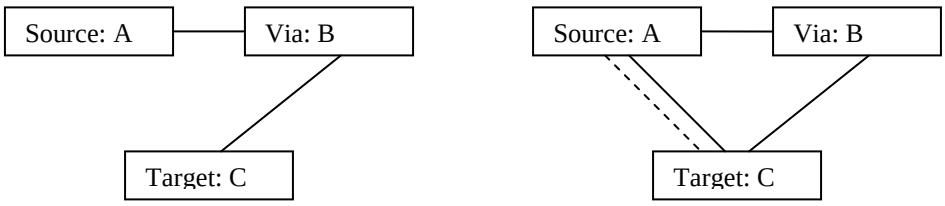
Trust is generally interpreted as one's reputation, one's opinion, and/or a probability of honesty. Grandison and Sloman defined trust as "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context" [6]. Distrust is also a useful concept to specify as a means of revoking previously agreed trust. Jøsang treated trust as the belief that an entity had about other entities and such a belief was formulated based upon past experiences, knowledge about the entity's nature and/or recommendations from other trusted entities [7].

In this paper, we will unite these two definitions and propose that trust is a belief that combines the direct feedbacks and the reputation, i.e., the recommendations or indirect feedbacks, acts dependably, securely and reliably within a specified context. Here, the term reputation is the opinion of the public towards a person, a group of people, an organization, or a resource. In the context of pervasive computing environments such as P2P systems, the reputation represents the opinions that the nodes in the system have about their peers and peer-provided resources.

2. 3. Formal expressions of trust features

According to our trust definition, trust model should consist of certain features such as transition, i.e., recommendation, and composition. We will assume some terms: entities A, B, C, R(A,B) as a trust relationship from A to B, P(A, B, C) as a trust relationship path between A and C via B, and f(x) is a trust function of the parameter x. The constant α, β, λ are trust values. Formal expressions for the features of transition and composition are as follows:

1) Transition. If A trusts B, B trusts C, A does not know C, then A could trusts C via B: if $\exists R(A, B), R(B, C), P(A, B, C), \nexists R(A, C)$, since A trusts B and B trusts C, B can recommend C to A and A will trust C. The C's trust value in A is less or equal than the recommended value multiplied by the B's trust value in A: $\exists R(A, B), R(B, C), P(A, B, C) \mid R(A, B) = \alpha \wedge R(B, C) = \beta \Rightarrow R(A, C) \leq \alpha * \beta$) for each P(A, B, C), see Fig.1 part 1.



1) Transition 2) Composition

Figure 1. Trust transition and composition

2) Composition. The trust can be updated by combining the direct feedbacks at different times and direct feedbacks with the reputation.

- If R(A,B) at transaction time t-1 and t are α and β, the updated trust R'(A,B) will combine these two values: $\exists R_{t-1}(A, B) = \alpha \wedge R_t(A, B) = \beta \Rightarrow R'(A, B) = f(\alpha + \beta)$;
- Further with the reputation, we assume that we ask for the final updated R'(A,C), see the dashed line in Fig. 1 part 2. There are R(A, C), R(A, B), R(B, C) and P(A, B, C). The R'(A,C) will be: $\exists R(A, C), R(A, B), R(B, C), P(A, B, C) \mid R(A, C) = \alpha \wedge R(A, B) = \beta \wedge R(B, C) = \lambda \Rightarrow R'(A,C) = f(\alpha, \beta * \lambda)$. Here, the first part of f(x, y) is the direct feedback and the second one is the reputation.

Other trust properties are reflexive, non-symmetrical and dynamic. Following [8], we have:

3) Reflexive. Each entity trusts itself: $\forall A \mid R(A, A) = 1$.

4) Non-symmetrical:

- If A trusts B, it is not necessarily B trusts A: $\exists R(A, B) \not\Rightarrow \exists R(B, A)$;
- Furthermore, if A trusts B and B trusts A, B's trust is not necessarily equal to A's trust: $\exists R(A, B), R(B, A) \mid R(A, B) = \alpha \wedge R(B, A) = \beta \not\Rightarrow \alpha = \beta$.

5) Dynamic. Trust can change (either increase or decrease) along time, depending on the future action a: $\exists R_{t-1}(A, B) = \alpha$, action a $\Rightarrow R_t(A, B) \geq \alpha$) or $R_t(A, B) \leq \alpha$.

3. Proportional-integral-derivative (PID) controller

The PID controller is widely used in feedback control of industrial processes. The acronyms are also used at the element level: the proportional element is referred to as the "P element," the integral element as the "I element," and the derivative element as the "D element." The PID controller was first placed on the market in 1939 and has remained the most widely used controller in process control until today.

Three functions for the three elements will produce outputs with the following natures:

- P element: proportional to the present data at the instant t;
- I element: proportional to the integral of the data up to the instant t, which can be interpreted as the accumulation of the past data;
- D element: proportional to the derivative of the data at the instant t, which can be interpreted as the prediction of the "future" data.

Typical version of the PID controller is described by:

$$u(t) = K(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d * \frac{de(t)}{dt}) \tag{1}$$

where u(t) is the output of the controller and e(t) is the input data. The controller parameters are proportional gain K, integral time T_i, and derivative time T_d.

In application, designers have freedom of using the three functional elements (P, I, and D) of the PID controller in whatever combination they consider most appropriate for their problems. Theoretically, there exist seven action modes. Among them, the five combinations listed in Table 1 are important in practice.

Table 1. Action modes of PID controllers

Action mode	**Element(s) used**
Proportional (P)	P element only
Integral (I)	I element only
Proportional- Integral (PI)	P and I elements
Proportional- Derivative (PD)	P and D elements
Proportional- Integral-Derivative (PID)	All 3 elements

To simply the basic PID algorithm in equation 1 and make the terms proper to the trust model, we change some notations in using the full PID format as follows:

$$TR(t) = \alpha * T(t) + \beta * \int_0^t T(x)dx + \gamma * \frac{dT(t)}{dt} \tag{2}$$

where TR(t) is the updated trust value at time t, T(t) is the raw trust value of peer n at time t. The first component (proportional) refers to the contribution of the current reports of trust received at time t, the second component (integral) represents the past performance of the peer (history information) and the third component (derivative) reflects the sudden changes in the trust value of a peer in the very recent past. Here, the parameters α, β and γ are three weights for three components respectively and they can be calculated from equation 1 as follows: $\alpha = K$; $\beta = K/T_i$; $\gamma = K*T_d$.

Choosing α larger value for biases the trust value to the reports currently received. A larger value of β gives heavier weight to the performance in the past. Similarly, a larger value of γ amplifies sudden changes in behavior of the peer in the recent past.

To discretize the PID-based trust model in equation 2, we assume that the trust values of peers are updated periodically within one time unit and the successive time periods (intervals) are numbered with consecutive integers starting from one. Therefore the discrete trust model is as follows:

$$TR(t) = \alpha * T(t) + \beta * \sum_{1}^{t} T(i) + \gamma * \frac{(T(t) - T(t-1))}{1} \quad (3)$$

4. Analysis of trust models

Reputation based trust uses personal experience or the experiences of others, possibly combined, to make a trust decision about a peer. The basic idea is to let a peer rate each other, for example, after the completion of a transaction, and to use the aggregated ratings about a given peer to derive a trust score which can assist other peers in deciding whether or not to transact with that peer in the future. Many reputation systems have been developed for P2P systems due to the development of many collaborative P2P applications such as CORE [9], EigenTrust [10] and TrustGuard [3].

4. 1. Simplest trust metric

In a number of reputation management systems, including eBay, Yahoo! Auctions and Auction Universe, the trust metric is calculated by aggregating all feedback scores:

$$R_i(t) = \sum_{k=1}^{t} R_i(k) \quad (4)$$

where $R_i(t)$ = a peer i's reputation at transaction t , $R_i(k)$ = a peer i's reputation rating for k^{th} specific transaction.

4. 2. Basic Trust Metric

A simple sum of all negative and positive reputation scores does not accurately reflect a user's reputation. Thus there is a need for a more reliable reputation equation. Some other sites have devised equations that are based on a ratio between the total amounts of reputation points over the number of ratings. This method is used by a number of electronic market and online community sites such as Amazon.com auctions and Exp.com.

$$R_i(t) = \frac{\sum_{k=1}^{t} R_i(k)}{t} \quad (5)$$

4. 3. Trust metric with credibility

In the physical world, we often account for the source of the information when considering reputation information. The input from peers who have a better reputation should be weighed more heavily in calculating reputation. On the contrary, a peer who may make false statements about another peer's service due to jealousy or other types of malicious motives should be weighted less. Therefore, a credibility factor should be built into the trust model and leads to equation 6.

$$R_i(t) = \frac{\sum_{k=1}^{t} \sum_{j=1}^{n} R_{ij}(k) * C_j(k)}{t} \quad (6)$$

where $R_{ij}(k)$ = a peer i's reputation rating from rater j, $1 \leq j \leq n$, (supposed that there are n peers

interacting with peer i during each time interval) for k^{th} transaction and $C_j(k)$ is the credibility of the feedback submitted by rater j at the k^{th} transaction. The difficulty with such an equation is how to determine the credibility factor.

A simpler approach is to use a function of the trust value of a peer as its credibility factor so that the feedbacks from trustworthy peers are considered more credible and thus weighted more than those from untrustworthy peers, i.e., $C_j(k) = f(R_j(k))$. In some exceptional cases, the above approach will generate errors. For example, it is possible that a peer may maintain a good reputation by performing high quality services, but send malicious feedback to its competitors. Therefore, more precise methods for credibility are needed. To filter out the false data, similarity measure can be used to rate the credibility of reported feedback. If the feedback data is similar to direct experience and other received feedback, it will be used in the reputation calculation [3].

4. 4.Trust metric combining recommendations

According to our trust definition, the trust is the belief that an entity has about other entity from past experiences and recommendations from trusted entities. The updated trust metric $T_i(t)$ for the peer i should combine the information based on own experience R_{id} (interaction derived or first-hand information) and the reputation R_{ir} (second-hand information):

$$T_i(t) = \omega R_{id}(t) + (1-\omega) R_{ir}(t) \tag{7}$$

where ω is the weight of $R_{id}(t)$ and $(1 - \omega)$ is the weight of $R_{ir}(t)$.

4. 5. Trust model for fading historic data

In e-business systems like eBay, the buyers may place much more weight on the recent few feedbacks for a seller than they do on their priors about seller. Therefore a decay function can be used to assign more weights to recent interactions and less weight to previous interactions.

The decay weight function dω(t) is a timing discount function and it can be described as follows:

$$d\omega(t) = \rho^{t-t'}, 0 < \rho \le 1, 1 \le t' \le t \tag{8}$$

where t is the current time and t' the old time, ρ is a normalized weight. Equation 8 shows that for the oldest transaction, i.e., t'=1, dω(t) will decay most. For the current feedback, t' = t and dω(t) = 1, no decay occurs. The trust metric with fading weights will be equation 9:

$$T'_i(t) = d\omega(t) * T_i(t) \tag{9}$$

To be simple, we will omit the superscript and subscript for $T'_i(t)$, i.e., T(t) stands for $T'_i(t)$ in the rest sections of the paper.

5. Incremental PID trust algorithm

I element in Equation 3 describes a scenario where one sums up all the transaction history of a peer. As Srivatsa et al. noted in [3], this may not be a feasible solution because the number of trust values held on behalf of a long standing member of the system can become extremely large and the computation time of equation 3 increases with the amount of data to be processed. The storage of so many data is another big problem. To solve above two problems, we propose the IPID controller.

5. 1. Standard IPID

We assume that we were at time t-1, the trust value would be TR(t-1):

$$TR(t-1) = \alpha * T(t-1) + \beta * \sum_{1}^{t-1} T(i) + \gamma * \frac{(T(t-1) - T(t-2))}{1} \quad (10)$$

The difference between TR(t) and TR(t-1) will be:

$$\begin{aligned} \Delta TR(t) &= \alpha * (T(t) - T(t-1)) + \beta * T(t) + \gamma * ((T(t) - T(t-1)) - (T(t-1) - T(t-2))) \\ &= \alpha * (T(t) - T(t-1)) + \beta * T(t) + \gamma * (T(t) - 2 * T(t-1) + T(t-2)) \end{aligned} \quad (11)$$

From (11), we know that ΔTR(t) can easily be calculated using three data: T(t), T(t-1) and T(t-2), in addition to the three parameters α, β and γ. Thus, equation 11 needs only four multiplications, three additions and two subtractions without worrying how large the history is. The updated TR(t) is the sum of TR(t-1) and ΔTR(t), therefore only one more addition operation will be needed to get TR(t):

$$TR(t) = TR(t-1) + \Delta TR(t) \quad (12)$$

In a word, only four data, four multiplications, four additions and two subtractions are needed to get the final trust value by our standard IPID model.

5. 2. Modified IPID for trust slow building and fast degrading

In the reputation systems, the negatives should be much more consequential than positives in affecting a seller's overall reputation. We modify the parameter γ in D element of PID controller to reflect this requirement as follows.

$$\begin{aligned} \gamma &= \gamma_1 \quad \text{if } \Delta T(t) \geq 0 \\ &= \gamma_2 \quad \text{if } \Delta T(t) < 0 \end{aligned} \quad (13)$$

To reflect the need for trust slow building, e.g., especially for the new peers and fast degrading for the malicious attackers, we set $\gamma_1 < \gamma_2$ and change equation 11 into equation 14:

$$\Delta TR(t) = \alpha * (T(t) - T(t-1)) + \beta * T(t) + \gamma_t * (T(t) - T(t-1)) - \gamma_{t-1}((T(t-1) - T(t-2)) \quad (14)$$

where γ_t and γ_{t-1} can be γ_1 or γ_2 according to the values of ΔT(t) and ΔT(t-1), respectively. However, γ_2 should be chosen carefully due to the sensitivity of the formulation to negative feedback that may be issued by the malicious attackers.

6. Common attacks and issues to be studied

The reputation system should be robust to malicious attacks. There are several classes of attacks such as self-promoting, whitewashing, slandering, orchestrated and denial of service [11].

- Self-promoting: attackers falsely increase their own reputation.
- Slandering: attackers report false data to lower the reputation of the victim nodes.
- Whitewashing: attackers escape the consequence of abusing the system by leaving and re-entering the systems by new identities.
- Orchestrated: attackers employ several of the above strategies.
- Denial of service: attackers cause denial of service by preventing the calculation and dissemination of reputation values.

However, we will only discuss the first three attacks and show the effect of our trust model on following important issues by the experiment evaluations:

- Evaluations 1: to reflect the consistent node behavior.
- Evaluations 2: to reflect the sudden fluctuations in node behavior.
- Evaluations 3: to tolerate the unintentional errors.

6. 1. Self-Promoting

Some forms of such attack and the relative defending technologies are as follows:

- Attack form 1: an attacker fabricates fake positive feedback about itself.

Solution 1: the reputation system is required to provide the accountability and proof of successful transactions.

- Attack form 2: disparate identities or a single physical identity acquiring multiple identities collude to promote each other by real transactions and real positive feedbacks.

Solution 2: the reputation system should be able to limit or prevent an attacker from obtaining multiple identities or use the heuristic-based solutions.

6. 2. Slandering

In this kind of attack, one or more identities falsely produce negative feedback about other identities. The reason for such attack is the lack of authentication and high sensitivity of the formulation to negative feedback. Therefore, the defending technologies include:

- Applying stricter feedback authentication mechanisms.
- Setting the proper sensitivity for negative feedbacks.
- Limiting the number of identities malicious nodes.
- Lowering the credibility factor to the peers that issue the attack as in equation 6.
- Paying more attention to the direct information as in equation 7.

6. 3. Whitewashing

The peers in such an attack will attempt to re-enter the system with a new identity with a fresh reputation. A reputation system is vulnerable if the trust formulation relies exclusively on long-term history without discriminating between old and recent actions. The solutions are as follows:

- Using the fading factor in equation 8 to discriminate the old data from new ones.
- Setting zero trust value for newcomers, set low γ_1 in equation 13 to slow down the trust building.
- Limiting users from quickly switching identities or obtaining multiple identities
- Taking into account limited history such as using the sliding windows of the recent data.

7. Experiment evaluation

In this section, we report our experiment results. Though our model is independent of history size, we set a sliding data window size as k = 64. To simplify the evaluation, we only consider the direct feedback, i.e., setting weight factor ω in equation 7 to be one. We first assume that there is only one transaction during each time interval and each peer is supposed to be honest, i.e., the credibility factor in equation 6 is also one.

7. 1. Evaluations 1- consistent node behavior

From Fig. 2, the trust value TR(t) stays in the high consistent trust zone, i.e., TR(t)>0.75. We apply the standard IPID algorithm and pay more attention on PD elements by lowering the weight for I element. Compared with the parameters of $\alpha(0.2)$, $\beta(0.9/k)$, $\gamma1(0.05)$ and $\gamma2(0.2)$ in [3], we assign the three parameters experimentally as follows: $\alpha = 0.2$, $\beta = 0.1/k$, $\gamma = 0.05$.

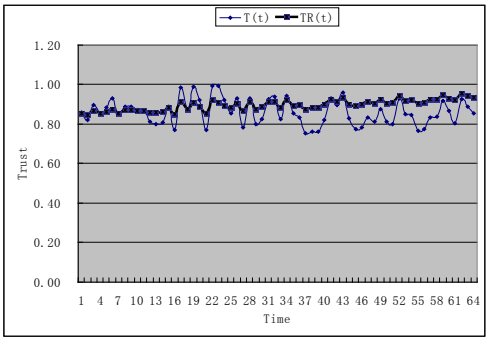


Figure 2. Consistent node behavior

7. 2. Evaluations 2 - sudden behavior fluctuations

For the strategic peer behaviors such as the sudden fluctuations, saying T(18) = 0, T(19) = 0.88 and T(20) = 0, we can apply IPID algorithm to get the results as TR(18) = 0.66, TR(19) = 0.92 and TR(20) = 0.66. From Fig. 3, we know that even the feedbacks change dramatically, TR(t) changes mildly due to the averaging nature of I element.

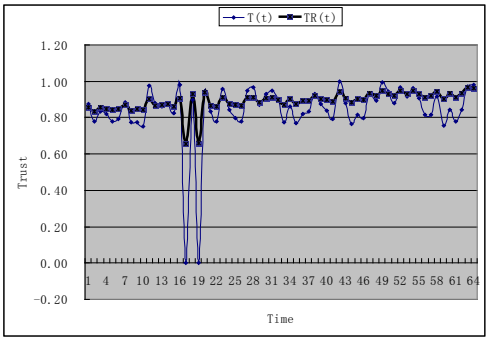


Figure 3. Effect of sudden fluctuations

7. 3. Evaluations 3- unintentional errors

To test tolerating unintentional errors, we change T(50) from 0.88 to 0.3. The feedbacks at t<50 and t>50 are normally good feedbacks, i.e., between 0.75 and 1. We can see from Fig. 4 that TR(50) = 0.78 which still remains in the normal trust zone.

From Fig. 3 and 4, we see that the trust value TR(t) keeps almost consistent after sudden fluctuations and can tolerate unintentional errors.

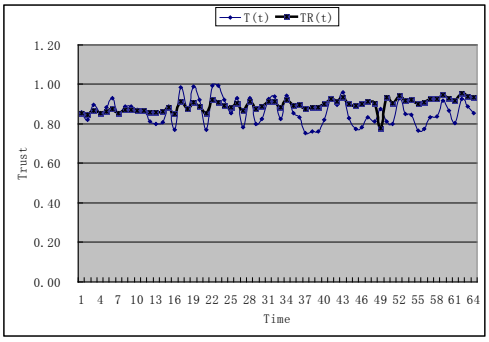


Figure 4. Effect of unintentional errors

8. Related work

The existing common reputation systems that compute the trust value by first-hand information such as equations 1-3 requires a weighted average of all previously known historical values, giving an efficiency of O(t), where t is the number of values retained. The efficiency of some reputation systems can be reduced to O(log t) by optimization such as fading memories [3].

Table 2 presents the efficiency of some other common reputation systems. From Table 2, we know that the efficiency of most reputation systems ranges from $O(t^2)$ such as Guha to O(log t) such as TrustGuard.

Table 2. Reputation systems and efficiency

Reputation Systems	Efficiency
Zimmermann	$O(t^2)$
Guha	$O(t^2)$
eBay	O(t)
EigenTrust	O(t)
Feldman	O(log t)
P-GRID	O(log t)

Not many reputation systems use PID controller. To our knowledge, the only trust model that uses PID controller is presented in TrustGuard by Srivatsa et al. in [3]. However, the basic PID controller usually presents heavy overheads to the system because of the of the integral value computation. In [3], the model represents the trust values using $\log_2 t$ values, where t represents system time intervals and allows the strategic guard calculations to be deterministically performed with an efficiency of O(log t) instead of O(t).

In this paper, we present a more frugal and efficient trust calculation by IPID controller which applies only four data to compute the trust metric. The computation for the trust model will be four multiplications, four additions and two subtractions for standard IPID, which leads to the efficiency of our model to be O(1).

The differences between TrustGuard and our IPID are as follows:

- TrustGuard uses a term H[i] to represent the past reputation history and computes the derivative component D with T[t] - H[t]. Besides, TrustGuard basically applies the basic PID.
- Our trust model uses the standard discrete IPID algorithm instead of PID and uses T[t] - T[t-1] as the D element. We also modify the IPID to suit for the requirement of trust slow building and fast degrading.

9. Conclusions

It is new approach to incorporate the PID controller into a trust model. The proportional value is used to know the reaction to the current trust, the integral value to determine the reaction based on the sum of previous trusts, and the derivative value to set up the reaction based on the rate at which the trust has been changing. By "tuning" three parameters for three components in the PID algorithm, such a trust model could bias the trust calculations for different occasions. Further, we present an IPID trust model to solve the shortage of computation power and storage for the pervasive computing devices.

The efficiency of our trust model O(1) that uses only four data and needs four multiplications, four additions and two subtractions without worry about the number of transactions for the standard version of IPID. The modified version of our trust model for slow increasing for new entity and quick dropping for misbehaving is also investigated.

Besides, this paper reviews some trust models and discusses the robust effects for some important issues such as consistent node behavior, sudden fluctuations in node behavior and unintentional errors.

From Table 2, we believe that our approach is more frugal and can efficiently/effectively be applied in many existing reputation-based trust systems. Our IPID trust model can also be used the time and storage limited applications such as real-time embedded systems and wireless networks.

Our model needs improving since the weights for three components in IPID controller should be

empirically determined. Our future work will deal with some more trust formations such as whether a seller's product and its price, geographic location, written comments should be merged into the trust model. Besides, to cope with the other attacks introduced into the P2P reputation-based trust mechanisms will be our future work.

10. Acknowledgments

This work was supported by the key course project, high education project and the scientific research startup project of China Jiliang University, higher science & engineering education & teaching reform and practice project of Education Department of China, Natural Sciences and Engineering Research Council of Canada and Fond Québécois de recherché sur la nature et les technologies.

11. References

[1] P. Resnickand, R. Zeckhauserand, E. Friedman, and K. Kuwabara, “Reputation systems”, Journal of Communications of the ACM, vol. 43, no. 12, pp. 45-48, 2000.
[2] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, P. Samarati and F. Violante, “A reputation-based approach for choosing reliable resources in peer-to-peer networks”, In Proceedings of the 9th ACM conference on Computer and Communications Security, pp. 207-216, 2002.
[3] 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, pp. 422-431, 2005.
[4] G.. Bolton, E. Katok, and A. Ockenfels, “How effective are electronic reputation mechanisms? An experimental investigation”, Journal of Management Science, vol. 50, no. 11, pp. 1587–1602, 2004.
[5] D. Artz and Y. Gil, “A survey of trust in computer science and the semantic web”, Journal of Web Semantics, vol. 5, no. 2, pp.58-71, 2007.
[6] T. Grandison and M. Sloman, “A survey of trust in internet applications”, Journal of IEEE Communications Surveys and Tutorials, vol. 4, no. 4, pp. 2-16, 2000.
[7] A. Jøsang, “An algebra for assessing trust in certification chains”, In Proceedings of the Network and Distributed Systems Security Symposium, San Diego, USA, The Internet Society 1999.
[8] F. Almenarez, A. Marin, C. Campo and C. Garcia, “PTM: A Pervasive Trust Management Model for Dynamic Open Environments”, In First Workshop on Pervasive Security, Privacy and Trust, 2004.
[9] P. Michiardi and R. Molva, “CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,” in Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, pp. 107–121, 2002.
[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The EigenTrust algorithm for reputation management in P2P networks,” In Proceedings of the 12th international conference on World Wide Web, pp. 640–651, 2003.
[11] K. Hoffman, D. Zage, and C. Nita-Rotaru. “A survey of attack and defense techniques for reputation systems”, Technical report, Purdue Univ., 2007. http://www.cs.purdue.edu/homes/zagedj/docs/reputationsurvey.pdf