# MTrust: Reputation-Based Trust Model for a Mobile Agent System

Suphithat Songsiri
*Dept. of Communication Systems, Fern Universität Hagen*
*Universitätstr.11, D-58084 Hagen Germany*
*suphithat.songsiri@fernuni-hagen.de*

**Abstract.** This research aims to promote MTrust: a reputation-based trust model that will ensure cooperative interactions amongst mobile agents and visited hosts in a mobile agent system. MTrust is composed of a reputation system and a trust formation system. A reputation system relies on two components; a truthful feedback submission algorithm and a set of distributed feedback information storages. The first encourages participants to submit truthful transaction feedbacks and punishes them when pairs of contradictive feedbacks are reported. The second constitutes the distributed storage system, where each storage location is assigned to contain feedback information of one visited host. A trust formation system enables a truster to compute a trustee's trustworthiness. It has two components namely; a feedback aggregation module (*FAM*) and a trust computing module (*TCM*). A *FAM* calculates a trust value from feedback information when there is a lack of direct experiences using Beta distribution. A *TCM* calculates a trust value using Bayesian Network (BN).

## 1. Introduction

The security of a mobile agent paradigm emphasizes on protecting and preventing a mobile agent from malicious hosts' attacks by applying cryptography functions. Unfortunately these countermeasures alone are not enough to protect mobile agents from malicious hosts. We summarize that by deciding to visit a trustworthy host, the probability of a mobile agent being attacked can be reduced. This paper aims to promote cooperative behavior between mobile agents and visited hosts by using trust as a qualitative measurement. A mobile agent utilizes a combination of a host selection scheme (i.e. a method of choosing a visited host based on some criteria such as queue length) and a calculated trust value to decide whether it will migrate and perform tasks on that visited host or not. Logically, a trustworthy host is supposed to have a higher trust value compared to those who are malicious. A host, which possesses a high trust value, tends to be chosen by mobile agents as their service provider. Obtaining a high trust value is an incentive, which encourages a host to act co-ordinately with mobile agents. We conclude that by integrating a trust model in to a mobile agent system will absolutely increase cooperative behavior and cast malicious hosts away from the system. This paper provides a new truthful feedback submission algorithm using incentive-based timely feedback submission and a fair punishment scheme, a *FAM* for deriving a trust value from feedback information and a BN for a trust computing from direct experiences. The rest of this paper is organized as follows: Section 2 presents principles of trust and reputation. Section 3 explains MTrust architecture. Section 4 demonstrates a truthful feedback system. Section 5 describes a trust formation system. Section 6 represents conclusions and future work.

## 2. Principles of trust and reputation

Trust and reputation are two related, but diverse concepts. Despite extensive studies from areas of sociology, psychology and computer science, the lack of coherence in the definition of trust is blatantly evident. Amidst the numerous trust surveys ranging from the area of P2P, reputation system, mobile agent and e-Commerce, the most commonly accepted definition is that of Gambetta [1]. He defined trust as "*a particular level of the subjective probability with which an agent assesses that another agent or a group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action*". Thereby, it can be concluded that trust should have the following properties: *subjective, asymmetric, dynamic, transitive and context-dependent.*

In this paper, trust is defined as a subjective quantified predictor of the expected future behavior of a trustee according to a specific agreement elicited from the outcomes of the previous interactions, both from direct experiences and indirect experiences; also known as feedbacks. A trust value calculated by a truster can be viewed as the likelihood that a trustee will likely perform an action according to an agreement. Trust can be expressed in different ways such as a value in a continuous range between 0 and 1, binary values or a set of discrete values. The visualization of trust normally relies on a method calculating trust values.

Reputation of an individual refers to certain characteristics related to its trustworthiness ascribed by its interactants. Reputation can be obtained from a set of interaction feedbacks, where mobile agents describe a visited host's performance in fulfilling its obligations.

## 3. The MTrust System Architecture

MTrust is composed of a feedback system and a trust formation system as depicted in figure 1.

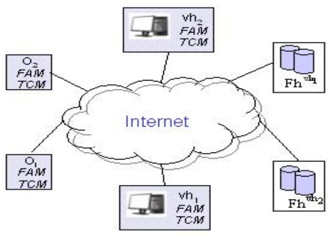


**Figure1 MTrust Architecture**

This paper uses a scenario of data retrieval mobile agents. An owner, denoted by $O_i$, when i=1,2,..,n, implements a feedback aggregation module and a trust computing module for calculating the list of trustworthy hosts. $O_i$ generates a set of mobile agents $A^{O_1} = \{ ma_1^{O_1}, ..., ma_j^{O_1} \}$. Each $ma_j^{O_i}$ represents a unique mobile agent's name, which allows visited hosts to trace the agent's owner. Each mobile agent is provided a list of visitable trustworthy hosts obtained from FAM and TCM computation and will select the visited hosts using the host selection scheme. A visited host (a service host) $vh_i$, provides files requested by the mobile agent. Each visited host is assigned a single feedback storage server $Fh^{vh_i}$ which maintains all feedbacks related to this visited host. Each $Fh^{vh_i}$ has fault-tolerance architecture. Each visited host grants services to a mobile agent according to its reputation. Once a transaction is completed, both $ma_j^{O_i}$ and $vh_i$ must submit their feedbacks according to the protocols described in section 4.1. Public key infrastructure is assumed to be available. The notations used throughout this paper are as follows:

- $O_i^T$ ( $O_i^R$ ): a truster (a rater).
- $E_{O_i}$ is an encryption using a public key of $O_i$ and $Sig_{O_i}$ is a signature using a private key of $O_i$.
- $H(m)$ is a hash of message m.
- Trust: $T^M_{O_i^T \to vh_i} \in [0,1]$: computed using a method from set M, where M $\in$ {Predefined Trust value (PT), General Trust value (GT), Feedback Aggregation method for inexperienced truster ($FA_{in}$), Feedback Aggregation method for experienced truster ($FA_{ex}$), Bayesian Network (BN), A combination of $FA_{ex}$ and BN(CO)}.
- Feedback: is an evaluation of its partner's obligation. A feedback is either 0 (i.e. dissatisfied) or 1 (i.e. satisfied). $\tilde{F}^{t_n,id}_{vh_i \to ma_j^{O_i}}$ denotes a feedback submitted by $vh_i$ at time $t_n$ for service number "id" containing the services provided to $ma_j^{O_i}$. $F^{t_n,id}_{ma_j^{O_i} \to vh_i}$ indicates a feedback from $ma_j^{O_i}$ comprising of $ma_j^{O_i}$'s feedback on $vh_i$'s quality of service.
- Reputation: A visited host has two types of reputation, namely; a good service provider ( $Rep^S_{vh_i}$ ) and an honest rater ( $Rep^R_{vh_i}$ ). In contrary to a visited host, each mobile agent belonging to the same owner possesses a single (group) reputation, as an honest rater, denoted by $Rep^R_{ma_i^{O_i}}$ .
- History: $HIS^{t_0:t_n}_{O_i^R \to vh_j} = \{\psi^{t_0,id}_{ma_j^{O_i},vh_{i+1}}, ...., \psi^{t_n,id}_{ma_j^{O_i},vh_{i+1}}\}$, A history of feedbacks between $O_i^R$ and $vh_i$ is a set of feedback pairs given by raters during time $t_0$ until $t_n$.

## 4. Reputation System

A reputation system [2] represents a promising method for fostering trust among complete strangers and for helping each individual to adjust or update its degree of trust towards its corresponding interaction partner. The fundamental idea of this reputation system is to use an algorithm to encourage transaction participants to evaluate on each others' performances on the previously concurred commitment engagements, by submitting truthful feedbacks. In this paper, the service host is committed to cater service requested by the mobile agent and to submit feedback, whereas the mobile agent is obliged to submit feedback. An integration of a reputation system into a mobile agent environment serves to induce cooperative interactions between visited hosts and mobile agents as well as to encourage entities to 'behave' well, otherwise their reputation would deteriorate, which in turn could lead to rejection of interaction by other entities. Before proceeding to the design of our reputation system, threats found in a reputation system are presented as follows:

- Strategic rater: From [3], a single or a collusive group of raters strategically provide a set of unfair feedbacks aiming to destroy a peer's reputation or boost its partner's reputation.
- Strategically malicious visited host: a host can manage its reputation according to its goal. For instance, it behaves cooperatively until it receives a high reputation then corrupts following transactions or it fluctuates its performance by cooperating or defecting its partners unevenly in an acceptable range so that it can still engage itself in future interactions.
- Whitewasher: From [4], entities that purposely leave or join the system with a new identity in an attempt to conceal any bad reputations, i.e. bad rater or malicious host, they have accumulated under their previous identity.

To repress strategic raters, we apply an incentive-based timely feedback submission as discussed in section 4.1.2 and 4.1.3 and a fair punishment scheme as discussed in section 4.1.4. In case of strategically malicious host, we implement BN as demonstrated in section 5. The prevention of whitewashers will not be discussed in this paper.

## 4.1 Truthful Feedbacks Submission

This subsection presents an algorithm, which inspires each participant in each transaction to faithfully provide its truthful feedback. Whitby et.al. [3] pointed out that elimination of unfair feedbacks can be achieved by increasing the cost of doing or decreasing the incentive to lie. To avert unfair feedbacks, three relevant approaches could be implemented, namely; detecting-based methods [3, 5-9], incentive-based methods [11-14,16] and punishment-based methods [11]. The first method attempts to detect or exclude unfair feedbacks using filtering concepts. The second method introduces incentives to motivate (or even benefit) truthful feedback submission. The last approach induces punishment like temporary transaction cessation for any unfair feedback submission found. This paper incorporates incentive-based timely feedback submission method and a fair punishment scheme. The argument for not implementing detection-based method is simply to avoid high cost accruement from constant observation and exclusion of unfair feedbacks. Our algorithm is explained in detailed in the following sub-sections.

### 4.1.1 Evidence of Legitimate Transaction (ELT)

Prior to any transaction engagement, a mobile agent and $vh_i$ have to agree upon the services $vh_i$ is to provide a mobile agent. This step is to ensure non-repudiated transactions. The presence of ELT does not annihilate the possibility of strategic raters, instead serves only as a tracking mechanism to the number of transactions performed between any mobile agent and $vh_i$. ELT can be acquired as follows:

- Service Request (SR): ***A***; denotes a mobile agent belonging to $O_i$, currently residing at host $vh_i$, requests $vh_{i+1}$ (a trustworthy host) for services by sending an encrypted message M1 containing $vh_i$'s signature on service request to $vh_{i+1}$.

$$A \xrightarrow{M1} vh_{i+1} : E_{vh_{i+1}}[Sig_{vh_i}(\text{service request})] \tag{1}$$

- Service Offer (SO): $vh_{i+1}$ replies ***A*** with types of services and quality of services it will provide.

$$\mathrm{vh}_{i+1} \xrightarrow{M2} A : E_{\mathrm{vh}_i}[\mathrm{Sig}_{\mathrm{vh}_{i+1}}(\text{service offer}, \text{service id\#})] \qquad (2)$$

- Service Agreement (SA): In case ***A*** agrees on vh$_{i+1}$'s services, it sends an acknowledgement for the service agreement to vh$_{i+1}$ and $\mathrm{Fh}^{\mathrm{vh}_{i+1}}$.

$$A \underset{M4}{\overset{M3}{\rightleftarrows}} \mathrm{vh}_{i+1} : \begin{cases} M3 = E_{\mathrm{vh}_{i+1}}[\mathrm{Sig}_{\mathrm{vh}_i}(H(M2))] & (3.1) \\ M4 = E_{\mathrm{vh}_i}[\mathrm{Sig}_{\mathrm{vh}_{i+1}}(H(M3))] & (3.2) \end{cases}$$

$$A \underset{M6}{\overset{M5}{\rightleftarrows}} \mathrm{Fh}^{\mathrm{vh}_{i+1}} : \begin{cases} M5 = E_{\mathrm{Fh}^{\mathrm{vh}_{i+1}}}[\mathrm{Sig}_{\mathrm{vh}_i}(\mathrm{Sig}_{\mathrm{vh}_{i+1}}(\text{service offer}, \text{service id\#}))] & (4.1) \\ M6 = E_{\mathrm{vh}_i}[\mathrm{Sig}_{\mathrm{Fh}^{\mathrm{vh}_{i+1}}}(\mathrm{vh}_{i+1}\text{'s status})] & (4.2) \end{cases}$$

vh$_{i+1}$'s status indicates whether or not vh$_{i+1}$ is under probation period.

- Service Acknowledgement (S_ack): After receiving message M3, vh$_{i+1}$ sends a service agreement M3 and M4 to $\mathrm{Fh}^{\mathrm{vh}_i}$.

$$\mathrm{vh}_{i+1} \underset{M8}{\overset{M7}{\rightleftarrows}} \mathrm{Fh}^{\mathrm{vh}_{i+1}} : \begin{cases} M7 = E_{\mathrm{Fh}^{\mathrm{vh}_{i+1}}}[\mathrm{Sig}_{i+1}(H(M3 \| M4)] & (5.1) \\ M8 = E_{i+1}[\mathrm{Sig}_{\mathrm{Fh}^{\mathrm{vh}_{i+1}}}(H(M7))] & (5.2) \end{cases}$$

After ETL has been completed, ***A*** migrates to vh$_{i+1}$ to perform tasks for its owner.

### 4.1.2 Incentive-based Algorithm

Once vh$_{i+1}$ has performed its obligations for ***A***, both of them are liable to submit feedbacks. Jurca et al. [13] pointed out that an incentive-compatible scheme for truthful feedback submission ensures that it is for the best interests of a rational rater to actually report truthful feedbacks. From [12], the first issue to be considered is the choice of appropriate incentive, which effectively stimulates truthful feedback submission behavior. Wu et al. [14] summarized choices of incentives used in Ad-hoc networks to be reputation-based and price-based. This paper classifies feedbacks from two raters as follows.

- A pair of consistent feedbacks $\psi^{t_n,id}_{A,\mathrm{vh}_{i+1}} = \{\tilde{F}^{t_n,id}_{\mathrm{vh}_{i+1}\to A}, F^{t_n,id}_{A\to \mathrm{vh}_{i+1}}\}$. Both raters provide

feedbacks in the same direction (i.e. either satisfied or dissatisfied). Here, it can be interpreted that both raters are honest or collusive. Klein et al. [15] pointed out the high occurrence tendency of feedback reciprocity; i.e. giving a positive feedback in return to a positive feedback, in reputation systems. In contrast, feedback retaliations i.e. reacting to a negative feedback with a negative, are relative rare.

- A pair of contradictive feedbacks $\tilde{\psi}^{t_n,id}_{A,\mathrm{vh}_{i+1}} = \{\tilde{F}^{t_n,id}_{\mathrm{vh}_{i+1}\to A}, F^{t_n,id}_{A\to \mathrm{vh}_{i+1}}\}$. One rater

contributes a feedback, which is opposite to another feedback. This means that one participant is dishonest.

The elimination of feedback reciprocity or retaliation can eventuate if both raters are unaware of each other's feedbacks prior to feedback submission. This will be demonstrated in subsection 4.1.3. By integrating a fair punishment scheme, the number of contradictive feedback pairs can be reduced. This will be further discussed in subsection 4.1.4.

### 4.1.3 Timely Feedback Submission Algorithm

The concepts of a timely feedback submission rely strongly on an invisibility of feedback submission and a continuity of future transactions. After a transaction is done, $vh_{i+1}$ submits its feedback to $Fh^{vh_{i+1}}$ as described below.

$$vh_{i+1} \overset{M9}{\underset{M10}{\rightleftarrows}} Fh^{vh_{i+1}} : \begin{cases} \tilde{F}^{t_n,id}_{vh_{i+1}\to A} \quad \text{when} \quad \tilde{F}^{t_n,id}_{vh_{i+1}\to A} = E_{Fh^{vh_{i+1}}}[Sig_{vh_{i+1}}(0,1)] & (6.1) \\ E_{vh_{i+1}}[Sig_{Fh^{vh_{i+1}}}(H(M9))] & (6.2) \end{cases}$$

Unlike $vh_{i+1}$, ***A*** submits its feedback to its owner to be used as direct experience and $Fh^{vh_{i+1}}$ to be compared with $vh_{i+1}$'s feedback, only after migrating to the next host $vh_{i+2}$. This is to prevent $vh_{i+1}$ from snooping on ***A***'s feedback.

$$A \overset{M11}{\underset{M12}{\rightleftarrows}} Fh^{vh_{i+1}} : \begin{cases} F^{t_n,id}_{A\to vh_{i+1}} \quad \text{when} \quad F^{t_n,id}_{A\to vh_{i+1}} = E_{Fh^{vh_{i+1}}}[Sig_{vh_{i+2}}(0,1)] & (7.1) \\ E_{vh_{i+2}}[Sig_{Fh^{vh_{i+1}}}(H(M11))] & (7.2) \end{cases}$$

$$A \overset{M13}{\underset{M14}{\rightleftarrows}} O_i : \begin{cases} F^{t_n,id}_{A\to vh_{i+1}} \quad \text{when} \quad F^{t_n,id}_{A\to vh_{i+1}} = E_{O_i}[Sig_{vh_{i+2}}(0,1)] & (8.1) \\ E_{vh_{i+2}}[Sig_{O_i}(H(M13))] & (8.2) \end{cases}$$

If any participant fails to submit its feedback, it will be banned from engaging itself in any future transaction. Both $vh_{i+1}$ and ***A***'s feedbacks will be legally obtainable from $Fh^{vh_{i+1}}$ only, once $Fh^{vh_{i+1}}$ has received them. $Fh^{vh_{i+1}}$ will then compare the feedbacks for consistency or contradiction. For contradictive feedbacks, a fair punishment scheme must be exercised. Should $vh_{i+1}$ or ***A*** fail to submit its feedback, it will be banned from future transactions for a period of time.

### 4.1.4 A Fair Punishment Scheme

The common punishment scheme prohibits both participants from engaging in new transactions for a period of time, should a pair of contradictive feedbacks occur. This scheme will discourage a 'victim' host or mobile agent from making any future transactions with those specific raters, thereby dissuading untruthful feedback submission. In addition to the common punishment scheme, the fair punishment scheme exerts the following:

- On mobile agent: temporary transaction cessation for *all* mobile agents belonging to $O_i$. Upon receiving contradictive feedbacks, $Fh^{vh_{i+1}}$ immediately notifies $O_i$ about its mobile agent and $vh_{i+1}$ participating in the contradictive feedback and broadcasts to all service providers that all $O_i$'s mobile agents are under probation, which implies that no services should be provided to them.

- On service host: temporary transaction cessation. From eq.(4.1) (section 4.1.2), $Fh^{vh_{i+1}}$ reveals $vh_{i+1}$'s probation status, thus discouraging all mobile agents from engaging in a transaction with $vh_{i+1}$.

The effect of the proposed mechanism above is incontestably more effective than the common punishment scheme, as the punishment impact occurs almost immediately after malicious behaviour has been encountered and lasts for some period. Furthermore, the probation status engenders impairment of the continuity of future transaction prospects especially for both $O_i$ and $vh_{i+1}$, as they would most probably evade any future transactions with each other. The scheme is further enhanced if all service hosts collaborate in declining transaction engagements requested by $O_i$'s mobile agents under probation. Nonetheless, the scheme induces additional costs of broadcasting for the storage servers. The fair punishment scheme also introduces collaboration incentive for service hosts, which deny transactions with agents under probation. The incentive is chosen to be reputation. Engaging a transaction with an agent under probation will not improve their reputation, as the feedback submitted to their dedicated storage server will be ignored. For a rational rater each transaction means additional positive reputation, the latter is therefore disadvantageous to rational raters. Hence, on the one hand, the fair punishment scheme encourages truthful feedback submission and on the other hand punishes strategic raters accordingly. A probation period is a function of the number of contradictive feedback pairs that both raters have implicated together.

$$PrP_{A,vh_i}(\mathrm{k}) = \alpha^{\mathrm{k}} \tag{9}$$

k is the number of contradictive feedback pairs between both raters and $\alpha$ is any appropriate integer depending on a system designer. Assuming no communication failure between raters and a feedback storage server, this algorithm is vulnerable to the following problems:

- Incompliant next visited host: host $vh_{i+2}$ impedes ***A*** from submitting its feedback. For this, MTrust is equipped with a mobile agent location update scheme [26]. This scheme assists an owner in determining the status of its mobile agents (i.e. alive or killed). A mobile agent informs its owner about inability of feedback submission.
- Collusive next visited host: before submitting its feedback, $vh_{i+1}$ waits for its collusive partner $vh_{i+2}$ to send ***A***'s feedback to it and strategically submits its own feedback. Should the feedback pair be contradictive, the fair punishment scheme would be enforced.

## 5. Trust Formation System

A trust value can be computed using many methods [17]. To calculate a trustee's trust value, the types of a truster and status of a trustee towards a truster should be considered. A truster can be categorized into an inexperienced truster; i.e. a truster who is new to the system or an experienced truster; i.e. a truster who has had some transactions with some trustees. Status of a trustee towards a truster can be explained as a newcomer to the system (type1), never transacts with a truster but not new to the system (type2), or has committed some transactions with a truster (type3). The set of methods used by a truster according to a combination of types of a truster and status of a trustee is summarized in the following table 1 and 2.

| Truster \ Trustee | Type1 | Type2 |
|---|---|---|
| Inexperienced Truster | PT | $FA_{IN}$ |

**Table1. A set of methods used by an inexperienced trustee**

| Truster \ Trustee | Type1 | Type2 | Type3 |
|---|---|---|---|
| Experienced Truster | GT | $FA_{EX}$ | CO |

**Table2. A set of methods used by an experienced truster**

A PT value is a trust value given to a trustee according to truster's behavior. The FA algorithm derives a trust value from a trustee's reputation. A GT value can be obtained from experiences a truster has with all trustees in the system. A CO is a combination $FA_{EX}$ and BN, is employed to calculate a risk and a trust value of a trustee. The BN utilizes direct experiences with a trustee to place a trust value on a trustee. The following subsections explain each of mentioned method in detail.

## 5.1 Bayesian Network Trust Computing

In [18], Pearl states that Bayesian methods provide reasoning about partial beliefs under conditions of uncertainty. The heart of Bayesian techniques lies in this formula.

$$p(H \mid e) = \frac{p(e \mid H) * p(H)}{p(e)} \qquad (10)$$

Where p(H) is prior probability of hypothesis H, p(H|e) is the posterior probability, p(e|H) is a likelihood probability that e (i.e. evidences) will materialize if H is true and p(e) is the prior probability of evidence e. An advantage of using BN to compute trust is that it can infer trust in various aspects from the corresponding conditional probabilities. BN's definition given by Jensen [19] is that a Bayesian network is composed of a set of variables (i.e. each variable has a finite set of mutually exclusive states) and a set of directed edges between variables. The variables together with the directed edges form a directed acyclic graph (DAG). Every owner develops a simple BN model as described in figure 2. Root node T has two stages; satisfying and unsatisfying, denoted by 1 and 0 respectively. The leaf nodes Q, ST and FS each have two stages; "good" and "bad", "fast" and "slow" and "honest" and "biased", which are represented by 1 and 0 respectively. The structure in figure 2 is called the diverging connection. Evidence cannot pass between all the children of T only if the state of T is known. We apply the d-separation rule to decide for any pair of variables in BN whether they are independent. From the figure, if the state of T (i.e. T is instantiated) is known, then Q, ST and FS become independent. Owner obtains the state of T from feedbacks about trustee sent by its mobile agents.

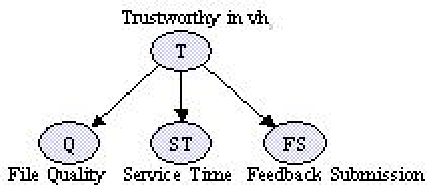


**Figure 2. Bayesian Network for Trust Computing**

A feedback contains an evaluation (i.e. satisfying or unsatisfying) given the result of interaction shown in eq. (11).

$$F^{t_n,id}_{ma_j^{o_i} \to vh_i} = \{1 \text{ or } 0 \mid \textit{result of interaction}\} \qquad (11)$$

A satisfying evaluation is obtained when $Q + ST + FS \geq 2$ . The value of FS is initially assumed to be 1, conferring to truthful feedback submission. For instance, if a visited host provides good file quality, slow service time and is honest on the feedback submission, then Q+ST+FS = 2. Therefore, the transaction is evaluated to satisfying. The feedback for this transaction is shown as follows:

$$F^{t_n,id}_{ma_j^{o_i} \to vh_i} = \{1 \mid Q{=}1, ST{=}0, FS{=}1\} \qquad (12)$$

An owner updates its BN once a feedback is received. In case of contradictive feedback occurrence broadcast from $Fh^{vh_i}$ , the owner re-evaluates the feedback with FS=0, and updates its BN again. BN trust computing allows computation of various conditional probabilities For instance, p(T=1|Q=1,FS=1) is a probability that a trustee is trustworthy in providing good file quality and being an honest rater. To conform to the trust defined in section 2, a trustee's trust value equals the specific conditional probability computed by a truster akin to its interests, as demonstrated in the following equation.

$$T^{BN}_{O_i^T \to vh_i} = p(T = 1 \mid \textit{specific interest}) \qquad (13)$$

A specific conditional probability can be computed by its joint probability. In our BN model, a truster maintains three CPTs; p(Q|T),p(ST|T) and p(FS|T). All tables are generated in the same way.

| p(Q \| T) | T=1 | T=0 |
|---|---|---|
| Q = 1 | p(Q = 1 \| T = 1) | p(Q = 1 \| T = 0) |
| Q = 0 | p(Q = 0 \| T = 1) | p(Q = 0 \| T = 0) |

**Table 3. Conditional Probability Table p(Q|T)**

Please note that the sum of each column must equal to 1. Each table must be updated after a transaction has been evaluated. p(T=1) is a general trust value of a trustee, which can be computed from the number of satisfying transactions S divided by the total number of transactions $S_{Total}$. p(T=1) is updated each time feedback has been received. The following experiments have been conducted to demonstrate the calculation of trust values using BN. Figures 3.1 and 3.2 illustrate the simulation of a trustee's trust value whose transaction feedbacks are evaluated to be "satisfying" even though it provides a set of fluctuating services. Figure 3.1 demonstrates that if a trustee always commits satisfying transactions (T=1) with good file quality (Q=1), its trust value will increase. However, if a trustee commits satisfying transactions (T=1) with bad file quality (Q=0), its trust value (i.e. P(T=1|Q=1) ) generally remains the same as the previous ones. Figure 3.2 presents a trustee who always satisfies transactions with bad file quality, its trust value is equal to its initial trust value.

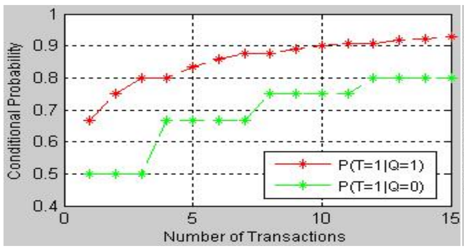


**Figure 3.1**

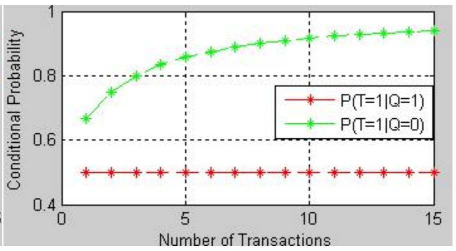


**Figure 3.2**

From the experiments conducted, it is obvious that if a trustee can manipulate its performance such that the evaluation result always comes out to be satisfying, then its trust value would increase with good file quality or remains equal with bad file quality. To evade such exploitations, both p(T=1|Q=1) and p(T=1|Q=0) must be considered when assigning trust value. A trust value is acquired according to the following equation

$$T^{BN}_{O_i^T \rightarrow vh_i} = p_{current}(T=1 \mid Q=1) - \Delta p(T=1 \mid Q=0) \qquad (14)$$

The second term denotes the difference between the latest and previous values of p(T=1|Q=0). By applying eq. (14), a trustee's trust value will not remain unchanged; instead will vary with the quality of service provided to a truster.

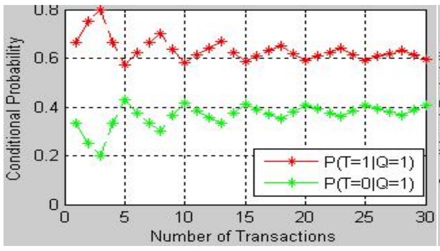


**Figure 3.3**

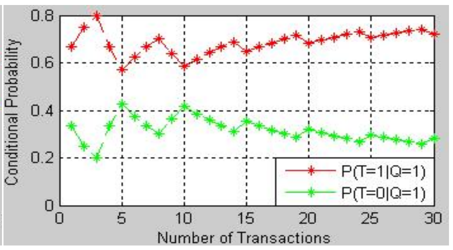


**Figure 3.4**

Figure 3.3 and 3.4 elucidate the cause for the deterioration of trust value and how a strategically malicious trustee manipulates its trust value. In this scenario, each mobile agent either evaluates each result of transaction to be 1 or 0 with good file quality. Deterioration in trust value eventuates when a mobile agent declares a result of evaluation to be dissatisfying when file quality was good. Figure 3.3 exhibits a pattern of service evaluations such that a trustee is voted to be satisfying for three consecutive transactions and dissatisfying for two consecutive transactions. If a trustee maintains this pattern, its trust value will slightly decrease. A strategically malicious trustee can also boost its trust value by appending a pattern of service, which contained a higher number of satisfying transactions and a lower number of unsatisfying transactions, as shown in figure 3.4 with pattern (4,1). To handle a strategically malicious trustee, a truster must take an observation on trustee's file quality. The assignment of trust value to a strategically malicious trustee is performed as follows:

$$T^{BN}_{O_i^T \to vh_i} = \frac{p_{current}(T = 1 \mid Q = 1)}{N_R} \qquad (15)$$

where $N_R$ is the number of repeated patterns (from figure 3.3, $N_R$ =6). This way, a strategically malicious trustee's trust value will dramatically reduce, thereby rendering no incentive for a trustee to perform services which will be evaluated to be a repeated pattern of service. This protocol still lacks in capability to prevent a strategically malicious trustee who dynamically changes its pattern of service. A better solution must be investigated.

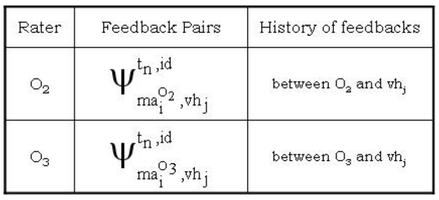
## 5.2 Feedback Aggregation Algorithm

This subsection demonstrates an algorithm to aggregate feedbacks from different raters and to form a single trust value representing trustworthiness of a trustee based on the trustee's reputation. There are some algorithms proposed to combine feedbacks. In [20], Shi et al. present an average algorithm. They argue that averaging feedbacks simplifies the algorithm design and provides low cost in running the system. From [21], Wang suggests that averaging should be used with feedbacks from unknown sources and weighing should be used from known sources. Xiong et al. [22] propose weighing feedbacks by using personalized similarity between rater and source of feedbacks. Yu et al. [23] only use feedbacks from witnesses who have interacted with a target peer. From [24,25], the authors use beta distribution model to calculate a trust value.

Our algorithm relies on beta distribution as it provides more information (variance, expectation) about the trustee's behavior of service provision when compared to a normal average algorithm; rendering less computation complexity as compared to similarity based method. From tables 1 and 2, an inexperienced rater and an experienced rater use an appropriate feedbacks aggregation algorithm to compute a trust value but each method is slightly different. Both algorithms are described as follows;

- Feedbacks Aggregation Algorithm for an inexperienced truster ($FA_{IN}$)

This algorithm applies to a scenario where a trust value is computed from received sets of unknown raters' history of feedbacks about a trustee. $FA_{IN}$ summarizes each trustee's reputation value derived from each unknown rater's history of feedbacks into a single trustee's trust value. Firstly; $O_i^T$ requests sets of history of feedbacks about $vh_j$ from $Fh^{vh_j}$. $Fh^{vh_j}$ will then send back each unknown rater's history of feedbacks of a recent time frame to $O_i^T$ as described in the following data structure shown in table 4. This table will be updated when a new pair of feedbacks arrives.



| Rater | Feedback Pairs | History of feedbacks |
|---|---|---|
| $O_2$ | $\psi^{t_n,id}_{ma_i^{O_2},vh_j}$ | between $O_2$ and $vh_j$ |
| $O_3$ | $\psi^{t_n,id}_{ma_i^{O_3},vh_j}$ | between $O_3$ and $vh_j$ |

**Table 4. Data Structure**

A reason of retrieving a history of feedbacks during a recent time window is that it reflects a recent behavior of a trustee towards a specific rater. An absolutely inexperienced truster should commence by obtaining all the histories of feedbacks. With the histories in hand, $O_i^T$ computes a trust value of vh$_j$ as follows:

- $O_i^T$ analyzes each history of feedbacks received. This is to ensure that the trust value will be computed according to $O_i^T$'s standards, as each rater's degree of requirements may be different. The analysis of a history of feedbacks of one rater is accomplished by checking each task related to each specific pair of feedbacks and comparing it with $O_i^T$'s expectations. This process is called feedback analysis. vh$_j$'s trust value is then computed once feedback analysis of all raters have been completed.
- Subsequently, a truster summarizes each analysed history of feedbacks belonging to each rater into a number of positive consistent feedbacks $N_P$ and a number of negative consistent feedbacks $N_N$. A trustee's reputation value is then calculated using beta distribution. A beta distribution is commonly used as a prior distribution for random variables that take on continuous value between 0 and 1. A beta distribution can be used to model the distribution of binary events (i.e. in this paper, binary event represents positive and negative feedback). The beta distribution can be expressed using gamma function $\Gamma$ (i.e. where $\Gamma(k+1) = \int_0^\infty y^k e^{-y} dy$, when $k > -1$) as follows:

$$f(b \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} b^{\alpha-1}(1-b)^{\beta-1} \qquad (16)$$

for $0 \le b \le 1$, where b is trustee's behavior towards rater and $\alpha, \beta > 0$. Trustee's behavior can be implied from a history of feedbacks. A truster computes each trustee's reputation perceived by each rater as an expectation of beta distribution shown in eq. 17.

$$\mathrm{Rep}_{O_i^R \to vh_i} = E[b \mid \alpha, \beta] = \frac{\alpha}{\alpha+\beta} \qquad (17)$$

where $\alpha = N_p + 1$ and $\beta = N_n + 1$. Undoubtedly, a trustee's reputation from two or more raters could have the same value. To determine the accuracy of reputation values which share the same values, variance is applied. Variance measures the statistical dispersion to which the typical values deviate from the expected values.

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \qquad (18)$$

The trustee's trust value is then computed from both the mean and variance of the beta distribution of each rater's history of feedbacks.

- A graph of mean (y-axis) versus variance (x-axis); where x-axis is divided into N (integer) equal ranges to represent the groups of least accuracy deviating reputation values, is plotted. Finally, a trustee's trust value is calculated as a summation of a normalized reputation from each raters group. A general form of a trustee's trust value is presented in eq.19.

$$T^{FA_{In}}_{O_i^T \rightarrow vh_i} = \sum_{k=1}^{Num} w_k \overline{rep_k} \text{ and } \overline{rep_k} = \frac{\sum_{i=1}^{N_k} Rep_{O_i^R \rightarrow vh_i}}{N_k} \qquad (19)$$

where Num is the number of ranges, $w_k$ is the weight for range k, $\overline{rep_k}$ is the average reputation represented by range k and $N_k$ is the number of raters in range k. $w_k$ is a function of the average variance and the number of raters in the range k. By multiplying $\overline{rep_k}$ with $w_k$, the average trustee's reputation derived from a range containing a large number of raters with a small value of average variance is more reliable than a range containing a less number of raters with a larger value of average variance. $w_k$ is derived as follows:

$$w_k = f(\overline{\sigma^2}, N_k) = \frac{w_k^{'}}{\sum_{k=1}^{Num} w_k^{'}} \text{ and } w_k^{'} = \frac{N_k}{N_{total}\overline{\sigma^2}} \qquad (20)$$

where $N_{total}$ is the total number of raters, and the number of raters dedicated to each range is $N_k$. The trust value derived by feedback aggregation algorithm is given as follows:

$$T^{FA_{In}}_{O_i^T \rightarrow vh_i} = \sum_{k=1}^{Num} \frac{w_k^{'}}{\sum_{k=1}^{Num} w_k^{'}} \overline{rep_K} \qquad (21)$$

For the purpose of simulation, a set of histories of feedbacks containing 19 pairs of positive and negative feedbacks from different raters have been used.

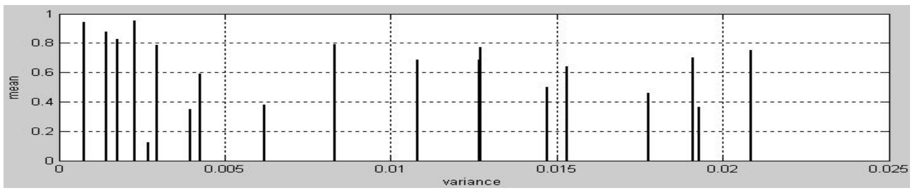


**Figure 4. A graph of mean versus variance**

From figure 4, for demonstration purposes, calculation of reputation from range 1 (variance between 0 and 0.005) results in the following: $N_{total}$=19, Num=5, $N_1$=8,

$\overline{rep_k} = 0.681, \overline{\sigma^2} = 0.000253$, $w_1 = 0.92$. A combination of all ranges of the trust value results in $T^{FA_{In}}_{O_i^T \to vh_i} = 0.63$.

- Feedbacks Aggregation Algorithm for an experienced truster ($FA_{EX}$)

This algorithm is applied to a scenario where a truster receives a set of known and unknown raters' history of feedbacks about a trustee. $FA_{EX}$ is designed to assist a truster in giving each history of feedbacks from different rater a weight factor ($w_f$) which represents the certainty of each history of feedbacks. Certainty means reliability of history of feedbacks in predicting a trustee's future behavior. The assignment of a weight factor to each history of feedbacks requires consideration of types of rater (known and unknown). Initially, each unknown rater possesses a weight factor of 1. After an interaction with a trustee, a truster updates all weight factors. The weight factor can be updated such that if a derived reputation value from a rater implies a trustworthy trustee but the result of transaction is dissatisfying, then a truster decreases a weight factor, otherwise it increases it. This way of a weight factor updating is not practical because a visited host can choose to accommodate bad or good services to specific victims or clients. If a truster is a victim of bad service, the weight factor of a rater who has been experiencing good services will decrease at every weight factor update. To avoid this problem, we assign a weight factor as an average of correctness in predicting all trustees' future behavior. $W_f$ is given as:

$$w_f^{O_i^R} = \frac{\sum_{j=1}^{M} p_j}{M} \tag{22}$$

where M is the total number of histories of feedbacks that a truster has considered so far from a rater $O_i^R$ and $p_j$ is the percentage of having successful transactions from total transactions with a trustee "j". Each $p_j$ must be updated once a transaction with trustee "j" has been completed. The weight factor is then multiplied with a reputation of trustee calculated from a rater's history of feedbacks, shown in eq. (17), to form a weighted reputation of the trustee. To use $FA_{EX}$, there are three situations a truster must consider. The first, if all raters are unknown, then the truster applies $FA_{IN}$ with a weight factor of 1 for each rater. The second, if all raters are known, $FA_{In}$ is utilized with modified reputation value, shown in eq. (23).

$$Rep^*_{O_i^R \to vh_i} = \frac{\alpha}{\alpha + \beta} w_f^{O_i^R} \tag{23}$$

The third, if the raters consist of a combination of known and unknown raters, then a truster separates raters into an unknown and a known group, and applies the appropriate methods as described previously. Subsequently, a truster combines trust values from both groups as shown in eq.(24).

$$T^{FA_{EX}}_{O_i^T \to vh_i} = \phi T_{unknown} + \varphi T_{known} \tag{24}$$

where $\phi + \varphi = 1, \varphi > \phi$. Both $\phi$ and $\varphi$ must be adjusted according to a truster's behavior.

## 5.3 A Predefined Trust Value

A predefined trust value is the one most difficult to calculate, because there is no information about a trustee for a truster to consider. In general, a transaction cannot be started if a truster has a small value of trust in a trustee. Oppositely, an imprudent truster obviously assigns a high predefined value of trust. This might give a chance to a malicious trustee to cheat on a truster. Actually, a predefined trust value is a trust value deduced from a truster's behavior. To make a transaction possible, we suggest that a truster should assign a trust value high enough before making a transaction with a trustee.

## 5.4 A General Trust Value

A general trust value is a trust value concluded from each trust value a truster assigns to each trustee. It can be viewed later as a predefined trust value that an experienced truster designates to a trustee who is a newcomer to the system. A general trust value presents a summary of a truster's experiences obtained from all trustees a truster has been interacting with. A general trust value is computed from an averaging of all trustees' trust values shown in eq. (25).

$$T^{GT}_{O_i^T} = \frac{\sum_{j=1}^{K} T^{M}_{O_i^T \to vh_j}}{K} \tag{25}$$

## 5.5 A combination of BN and Feedback aggregation

An experienced host uses $FA_{EX}$ to enhance its trust value computed by BN. Assuming that a truster has computed a trustee's trust value using its direct experiences at time $t_x$ without receiving any new feedback from its mobile agent, subsequently at time $t_y$, when $t_x<t_y$, a truster requires to calculate this trustee's trust value again. A truster retrieves only histories of feedbacks, which have been committed between its latest feedback with a trustee at time $t_z$ ($t_z<t_x$) and the last feedback of another rater with a trustee before the requesting time $t_v$ ($t_x<t_v<t_y$). A retrieved history of feedbacks will be used to compute a risk using $FA_{EX}$ algorithm. A truster concludes a trustee's trust value as eq. (26).

$$T^{CO}_{O_i^T \to vh_i} = \xi T^{FA_{EX}}_{O_i^T \to vh_i} + \zeta T^{BT}_{O_i^T \to vh_i}, \quad \xi + \zeta = 1 \tag{26}$$

where $\zeta = e^{\frac{t_z - t_v}{\theta}}$ and $\theta$ is the largest time interval between two consecutive feedbacks from its mobile agents about this trustee.

## 6. Conclusions and Future Work

This paper proposes MTrust: a reputation-based trust model for a mobile agent system. It contains a new incentive-based timely feedback submission algorithm and a fair punishment scheme, which enforce truthful feedback submissions. We provide a set of trust computing methods, which embodies five computation algorithms, used by each truster according to its type (i.e. inexperienced or experienced) and status of trustee towards it. We present a Bayesian Network based trust computing (i.e. how trust can be inferred from a conditional probability), its vulnerabilities and propose two algorithms for strategically malicious trustee prevention. The problems of whitewasher and strategically malicious trustee employing a dynamic pattern of services have not been addressed in this paper, however could be further investigated as future work.

## 7. Acknowledgement

The author is currently a doctoral candidate under the supervision of Prof. Dr.-Ing Firoz Kaderali at the department of Communication Systems, Fern Universität Hagen Germany. The author wishes to thank Prof. Dr. rer. nat. habil. Werner Poguntke and Prof. Dr.-Ing Firoz Kaderali for the encouraging discussions that led to the development of this trust model and reviewing this paper. The author is fully funded by the project " Mathematic and Engineering Science Method for Secure Data Transmission and Information Transfer" from DFG (German Research Foundation).

## References

1. D. Gambetta. Can we trust trust? In D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations, chapter 13, pages 213–237. Basil Blackwell, 1988.
2. A. Whitby, A.Jøsang, J. Indulka. Filtering out unfair Ratings in Bayesian Reputation Systems.The Icfain Journal of Management Research, 4(2), pp.48-64, February 2005
3. W.T. Luke Teacy, J. Patel, N. R. Jennings, M. Luck. Coping with Inaccurate reputation Sources: Experimental Analysis of A Probabilistic Trust Model. AAMAS 2005
4. S. Marti, H. Garcia-Molina.Taxonomy of Trust: Categorizing P2P reputation system. March 2005
5. T.D. Huynh,N.R. Jenning, N.R. Shadbolt.On Handling Inaccurate Witness Report.
6. C. Dellarocas. Mechanism for Coping with Unfair Ratings and Discriminatory behavior in Online Reputation Reporting systems.
7. M. Srivatsa, L. Xiong, L. Liu. TrustGuard: Countering Vulnerabilities in Reputation management for Decntralized Overlay Networks. www 2005
8. S. Buchegger, J.Y. Le Boudec. The Effect of Rumor Sprading In Reputation Systems for Mobile Ad-hoc Networks. Wiopt 2003
9. R. Jurca, B. Faltings. " Eliciting Truthful Feedback for Binary Reputation Mechanism" IEEE/WIC/ACM International Conference on Web Intelligence (WI'04).
10. Th. Dariotaki, A. Delis. " Detecting Reputation Variation in P2P Networks" The 6th Workshop on Distributed Data and Structures (WDAS'2004)
11. A. Fernandes,E. Kotsovinos, S. Ostring, B. Dragovic. Pinocchio: Incentives for honest participationin distributed trust management.itrust04
12. P. Obreiter, B. König-Ries, G. Papadopoulos." Engineering Incentive Schemes for Ad Hoc Networks
13. R. Jurca, B. Faltings. " An Incentive Compatible Reputation
14. Q. He, D. Wu, P Khosla. " SORI A secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks"Reference ( feedback aggregation )
15. T. J. Klein, C. Lambertz, G. Spagnolo, K. O. Stahl. Last Minute Feedback.
16. P. Obreitrt, J. Nimis.A Taxonomy Of Incentive Patterns- the Design Space of Incentives for Cooperation. Technical Repost Nr. 2003-9 University of Karlruhe
17. A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision (to appear). Decision Support Systems, 2006.
18. J. Pearl. Probabilistic Reasoning in Intelligent system
19. F. V. Jensen. " Bayesain network and Decision Graph

20. Z.Liang and W. Shi, "**PET**: A **PE**rsonalized **T**rust Model with Reputation and Risk Evaluation for P2P Resource Sharing, Jan 2005
21. Y. Wang. “ Bayesian Network –Based Trust Model in Peer-to Peer Networks.
22. L. Xiong, L. Liu. “ PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Community. IEEE Transaction on Knowledge and Data Engineering july 2004
23. B. Yu, M. P. Singh, k. Sycara. “ Developing Trust in Large Scale Peer-to-Peer Systems.
24. Patel, J., Teacy, W. T. L., Jennings, N. R. and Luck, M. (2005) A Probabilistic Trust Model for Handling Inaccurate Reputation Sources. *itrust05* pp. 193-209
25. A. Jøsang and R. Ismail. *The Beta Reputation System*. In the proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002.
26. S. Songsiri. A Naming Service Architecture and Optimal Periodical Update Scheme for a Multi Mobile Agent System. IAWTIC 2005.