Enhanced Reputation Mechanism for Mobile Ad Hoc Networks

Jinshan Liu, Valérie Issarny

Enhanced Reputation Mechanism for Mobile Ad Hoc Networks

Jinshan Liu and Valérie Issarny

INRIA - Rocquencourt,
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France
Jinshan.Liu,Valerie.Issarny@inria.fr
http://www-rocq.inria.fr/arles/

Abstract. Interactions between entities unknown to each other are inevitable in the ambient intelligence vision of service access anytime, anywhere. Trust management through a reputation mechanism to facilitate such interactions is recognized as a vital part of mobile ad hoc networks, which features lack of infrastructure, autonomy, mobility and resource scarcity of composing light-weight terminals. However, the design of a reputation mechanism is faced by challenges of how to enforce reputation information sharing and honest recommendation elicitation. In this paper, we present a reputation model, which incorporates two essential dimensions, time and context, along with mechanisms supporting reputation formation, evolution and propagation. By introducing the notion of recommendation reputation, our reputation mechanism shows effectiveness in distinguishing truth-telling and lying agents, obtaining true reputation of an agent, and ensuring reliability against attacks of defame and collusion.

1 Introduction

The pervasiveness of lightweight terminals (e.g., handhelds, PDAs and cell phones) with integrated communication capabilities facilitates the ambient intelligence vision of service access anytime, anywhere. This necessitates interactions between terminals belonging to different authorities, which are marginally known or completely unknown to each other. Trust management to enable such interactions has thus been recognized as a vital part of mobile ad hoc networks (MANET), which features lack of infrastructure, openness, node mobility, and resource scarcity (e.g., network, energy and storage space) of composing lightweight terminals.

In closed networks, trust establishment is managed by an authentication mechanism that assigns roles to agents. By *agent*, we mean a software entity working for and representing a node in MANET; each agent also has some reachable neighbor agents named *peers*. In an open environment such as MANET, fixed role assignment has to be be replaced by dynamic decisions. An important factor affecting the decision making is an agent's *reputation*.

C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 48–62, 2004.

Reputation assessment requires knowledge, information and evidence about the evaluated agent, which can be derived from an agent's own experiences. However, openness implies significant opportunities of meeting with *strangers* an agent has never encountered before. Furthermore, more accurate estimation of an agent's reputation becomes possible with sharing of reputation information among peers. Reputation mechanism has been widely used and implemented in electronic market places [1,2] and online communities [3]. For example, visitors at "amazon.com" or eBay usually read previous customers' reviews and feedbacks before deciding whether to make transactions.

However, the design of a reputation mechanism is faced by a number of challenges, including: (i) the "free-rider" problem, i.e., agents do not share reputation information with peers; and (ii) the honest elicitation problem, i.e., agents may report false reputation information. There are multiple reasons for agents to be reluctant to report evaluations or to do so honestly [1]. Agents may withhold positive evaluations if a seller's capacity is limited, e.g., wise parents are reluctant to reveal the names of their favorite baby-sitters. Agents may be reluctant to give positive recommendations because it lifts the reputation of the evaluated agent, which is a potential competitor. Agents may wish to be considered "nice", or be afraid of retaliation for negative feedbacks. And last but not least, the reputation information agents provide only benefits other peers.

Therefore, it is necessary to build a reputation mechanism to enforce both active reputation information sharing and truthful recommendation elicitation, which are necessary for a reputation system to operate effectively [4]. Our target reputation mechanism aims to defend against the following three kinds of attacks:

- Inactivity: This refers to agents' free-ride activities by not sharing reputation information with peers.
- Defame: This refers to agents' activities of propagating a victim's reputation that is lowered on purpose.
- Collusion: This refers to agents' activities of propagating good reputation to promote each other.

Hence, the desired properties of a reputation system for MANET are:

1. Valid: The system is effective in the sense that agents are able to distinguish honest from dishonest agents through the reputation system.
2. Distributed: The system should not assume access to any trustworthy entity (e.g., Certificate Authority), or centralized storage of reputation values.
3. Robust: The system is robust to the attacks listed above.
4. Timely: The system should be dynamic and be able to reflect the trustworthiness of an entity in an up-to-date manner.
5. Resource-saving: The reputation system should take into account the limited computation power and storage space of each terminal in MANET.

Existing reputation systems either do not address the aforementioned incentive problems (e.g., [5,6]), or depend on some (centralized) trustworthy entity (e.g.,[1,7]). Our approach, which is targeted at mobile ad hoc networks, does

not depend on any trustworthy entity or any centralized reputation storage, and possesses the aforementioned desired properties. Our contribution includes: (1) a reputation model that incorporates two dimensions, time and context, which captures reputation's time-sensitivity and context-dependence; (2) a simple yet effective reputation mechanism that enforces active and truthful reputation information sharing; (3) validation of the effectiveness and robustness of the proposed reputation mechanism via simulation tests. Our work targets service provision among agents in MANET. The *service* notion here is general[1], referring to not only services like Web services [8], packet forwarding services [6,5], but also activities like providing information (e.g., providing cuisine recipes) in online discussion forums.

In the following, Section 2 gives definitions and properties of reputation. Section 3 describes our reputation model, together with related mechanism supporting reputation formation, evolution and propagation. Section 4 presents results of simulation tests. Section 5 surveys related work. Finally, the paper finishes with conclusion and future work.

2 Reputation

Reputation is always associated, and often confused with *trust*. Therefore, in order to have a precise view of reputation, it is necessary to grasp the meaning of trust. Trust is a complex concept relating to belief in the honesty, truthfulness, competence, reliability, etc., of the trusted person or service [2]. Precisely defined, "*...trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or 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*" [9]. Trust towards an agent can been seen as a prediction on that agent's future action. An important factor affecting the prediction is then the reputation of the agent.

2.1 Defining Reputation

Mui *et al.* define reputation as "perception that an agent creates through past actions about its intentions and norms" [10]. This definition is precise except that it does not reflect the fact that reputation of an agent is created from the point of view of other agents. An agent can affect its own reputation by acting honestly or the other way, but it is unable to decide its reputation. To emphasize the "passive" property of reputation, we define reputation as follows:

> Reputation of an agent is a perception regarding its behavior norms, which is held by other agents, based on experiences and observation[2] of its past actions.

[1] Similar to the notion of *resource* in resource discovery

[2] As explained later, observation here refers to *indirect observation* through peers' recommendations.

The reputation assessment of an evaluated agent by an evaluator agent requires collecting related evidences beforehand. The sources of reputation include: (i) The evaluator's own interaction experiences with the evaluated agent; if the evaluator has first-hand experience of interacting with the evaluated agent, the interaction histories can serve as a strong reference for reputation evaluation. (ii) Recommendation from peers who have interacted with the evaluated agent before; note that recommendations of recommending agents are based on the agents' own experiences only, and do not include recommendations obtained from peers. This is necessary to prevent double counting that leads to rumors.

The node mobility and openness of MANET augment the opportunities for nodes to interact with nodes they never encountered before. This increases the agents' reliance on the latter source of reputation (i.e., recommendations from peers).

2.2 Properties of Reputation

Trust is widely deemed *subjective* [11,12]. Reputation, a perception of the trustworthiness of an agent based on experiences and recommendations, is also subjective [10] – because the same behavior can cause different impressions on different agents. It implies that one agent is likely to have different reputations in the view of different peers. We denote $Rep_a(o)$ as the reputation of the agent o, from the point of view of agent a. We represent reputation with a numeric value in the range $[-1.. + 1]$. The value of reputation ranges from *completely untrustworthy* (-1) to *completely trustworthy* $(+1)$. The larger the value is, the trustworthier the agent is. One value in the range that is worth mentioning is *ignorance*, which describes the reputation of agents about whom the evaluator has no knowledge. *Ignorance* bears the value 0[3]. Also we define *very trustworthy* (0.8), *trustworthy* (0.2), *untrustworthy* (-0.2) and *very untrustworthy* (-0.8). These labels do not stand for the only possible values of reputation. Instead, they are used to attach semantic meanings to numeric values. For example, if an agent's reputation value is 0.5, it is then considered to be between very trustworthy and trustworthy.

Reputation is also *context-dependent* [13,14]. For example, David enjoys a reputation of being a very talented painter, but he may not have as high reputation as a cook. So *context* is an important dimension for reputation.

Reputation is also *dynamic* – disreputable agents should be able to improve their reputations by acting honest; reputable agents' reputation should get lower if they become deceitful. Dynamics of reputation is also reflected by its timeliness: reputation is aggregate in the time scale by taking into account recent behavior and past histories. Hence, *time* is also a necessary dimension for reputation.

In the next section, we present our reputation model to depict the aforementioned properties together with associated mechanism of reputation formation, evolution and propagation.

[3] As pointed out by [12,10] and discussed at the end of this paper, this assignment does not differentiate new comers from agents whose 0 reputation value results from previous behaviors.

3 Reputation Model

To build a reliable reputation mechanism that enforces reputation information sharing and honest recommendation elicitation, our model includes the following elements:

1. Separate reputation for expertise (providing good service) and reputation for helpfulness (providing fair recommendation), respectively denoted as *service reputation (SRep)* and *recommendation reputation (RRep)*.
2. Agents derive the *SRep* of another agent according to their experiences (*SExp*) and recommendations (*Rec*) of peers whom they consider trustworthy in service recommendation; the trustworthier a peer is, the more weight its recommendations are assigned.
3. Reputations are both timely (i.e., evolve with time) and dynamic (i.e., adjust with behaviors); especially, recommenders' *RRep* are adjusted according to the *SRep* value of the recommended agent.
4. Agents exchange reputation information, but only with peers they consider helpful (i.e., with good *RRep*).

The above elements motivate truthful recommendations because untruthful and inactive recommendations lead to low *RRep* and thus loss of peers' recommendations; peers' recommendations are an important knowledge source for evaluating an agent's *SRep*, especially a stranger's *SRep*.

3.1 Reputation Definition

Given reputation's properties of being time-sensitive and context-dependent, an accurate reputation model needs to capture the two dimensions by integrating them seamlessly into reputation's definition, formation, evolution and propagation.

Time-sensitive Reputation. Reputation builds with time. A reputation at time t can be very different from the reputation at another time t'. With respect to the time dimension, we denote reputation of agent o in the view of agent a at time t as $Rep_a(o)^t$. Reputation is aggregate in the sense that it integrates peers' recommendations and the evaluator's own experiences, which are also aggregate. The weights assigned to recent behavior and past histories decide how fast the reputation builds up. For example, if recent behavior is assigned a very high weight, an agent's reputation tears down very fast after a few misbehaviors. We assign more weight to recency, as suggested by the results of psychological studies in [15] and empirical studies of ebay feedback mechanism [16], by adopting a parameter named *fading factor* ρ_e:

$$Rep_a(o)^t = Rep_a(o)^{t'} * \rho_e^{t-t'} + \text{New Behavior} * (1 - \rho_e^{t-t'}) \ . \tag{1}$$

Value of ρ_e falls into range [0..1]: the lower value ρ_e has, the more quickly histories are forgotten. When ρ_e equals 0, histories are completely forgotten; while

when ρ_e equals 1, the oldest history is forever remembered. This formula will be substantiated in the evolution of reputation (§3.3).

The representation of reputation assumes a single value with a timestamp stating the time of formation. More information is available if more history records (e.g., the last 10 reputation values) are kept. However, it consumes more space. Our representation with a single timestamped value saves storage space, which is a scarce resource for light-weight terminals, while still reflecting the time-sensitivity of reputation.

Context-dependent Reputation. As reputation is context-dependent, it is necessary to integrate context as a dimension into reputation. As stated, $SRep_a(o)^t$ in context C can be derived by information (i.e., a's experience and other peers' recommendation) in the context of C[4]. But, there are cases when there is no or not enough information in the context of C, but there are plenty in a related context of C'. It is good practice to be able to derive reputation from these related evidences. But, this is challenged by the question of how to capture the relevance of two contexts. This can be measured by the *distance* between two contexts, which is a quantitative parameter for describing the relation between the two contexts.

Context itself is a multi-dimensioned concept, it can include factors such as, importance and utility of a service [12] (e.g., transactions dealing with 10 euros vs. transactions of 10 thousand euros), service category (driving a car vs. flying a plane), and so on. We limit the context to service category in our work, which leads to the question: *how to measure the distance given two service categories*? For example, assuming an agent provides excellent service in providing cuisine recipes, but we need to know whether it is also as good in giving diet tips. The question becomes how far it is between providing cuisine recipes and giving diet tips.

The comparison of services can done in a syntactic way, e.g., comparison of interfaces, attributes and so on; or in a semantic way. The former is managed by comparing service signatures. The latter is currently undertaken by the Semantic Web activity of W3C[5], which proposes languages for service description such as Resource Description Framework (RDF), and Web Ontology Language (WOL). The DARPA Agent Markup Language (DAML), an extension of XML and RDF, is able to provide sophisticated classification and property definition of resources. We thus make use of an ontology tree of services using DAML-S[6], with each node in the tree representing a type of service. Each node is a subcategory (subclass) of its parent node. To save space, we assume each agent is able to obtain a part of the ontology tree that defines the services it is interested in. Given two nodes in the tree, the distance of the two nodes is defined as the least number of intermediate nodes for one node to traverse to another node. For example, in Fig. 1, service *s1* and *s2* has a distance of 3.

[4] For simplicity, we don't discuss context-dependent recommendation reputation here.
[5] http://www.w3.org/2001/sw/
[6] http://www.daml.org/services/

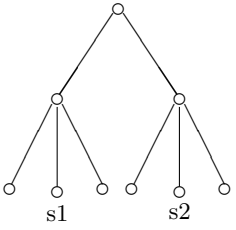

Fig. 1. A service ontology tree

Thus, reputation on context C can be calculated as:

$$SRep_a(o,C)^t = \frac{\sum_{C' \in Tree_a} SRep_a(o,C')^t * \rho_c^{|C'-C|}}{\sum_{C' \in Tree_a} \rho_c^{|C'-C|}} \quad . \tag{2}$$

Similar to (1), ρ_c is a fading factor reflecting an agent's reliance on context-related reputations. When ρ_c equals 0, it means the agent does not consider context-related reputations; while when ρ_c equals 1, the agent takes into account all context-related reputations, all of which have the same impact factor no matter how related or unrelated they are.

In the following, we denote *SRep* of agent o held by agent a at time t as $SRep_a(o)^t$, instead of $SRep_a(o,C)^t$ for simplicity of denotation, except during discussions of context-dependent reputations. However, it always applies that reputation in a certain context can be derived from reputation in other related contexts according to Equation (2). Table 1 summarizes the notations we have introduced so far.

Table 1. Notations used in the model

Label	Value Range	Meaning
$SRep_a(o)^t$	$[-1..+1]$	service reputation of agent o held by agent a at time t
$RRep_a(o)^t$	$[-1..+1]$	recommendation reputation of agent o held by agent a at time t
$SExp_a(o)^t$	$[-1..+1]$	Reputation of o derived from a's interaction experiences with o
$Rec_a(o)^t$	$[-1..+1]$	Recommendation made by agent a regarding agent o's reputation at time t. For honest agent a, $Rec_a(o) = SRep_a(o)$
ρ_e,ρ_c	[0..1]	Fading factor, representing agent's reliance on recent behaviors or related contexts

Having integrated *time* and *context* dimensions into our reputation model, we explore the related mechanism supporting reputation formation, evolution and propagation.

3.2 Reputation Formation

Reputation formation is implemented by the following components running on each node: an experience manager, a recommendation manager and a reputation manager.

Experience Manager

The experience manager is in charge of recording the previous experiences of service provision with other peers. The records include the service category (i.e., context C), the timestamp of last experience (t), and an aggregate value of experience (i.e., $SExp_a(o, C)^t$). The aggregation process of experience value will be further explored in Sec. 3.3.

Recommendation Manager

The recommendation manager implements three functions: (1) storing recommendations from other peers, (2) exchanging reputation information with other peers, and (3) managing a table of *RRep*s of recommenders.

Recommendations from peers regarding an agent's reputation need to be combined together by some means. Dynamic Weight Majority (DWM) [17] is a learning algorithm for tracking *concept drift*, which predicts using a weighted-majority vote of "experts", and dynamically creates and deletes experts in response to changes in performance. Our approach tracks "an agent's reputation" by consulting recommendations (votes) from peers (experts), and dynamically changes their recommendation reputation according to their prediction accuracy. We do not delete peers from the recommender list, however, but we ignore a peer's recommendation if its *RRep* falls below some threshold value.

Reputation Manager

The reputation manager administers and calculates the *SRep* of a peer, taking into account inputs from both experience manager and recommendation manager. Reputation manager assigns different weights to experiences and recommendations, namely, greater weight for its own experience and less weight for recommendations from peers. This is due to the reason that agents tend to rely on their own experience more than on other peers' recommendation, as suggested by experimental studies of Kollock [18].

Consider agent a has recommendations regarding agent o from a group of peers P; the peers considered untrustworthy in service recommendation (i.e., with low *RRep*) have been excluded from P. We get the following formula for *SRep* evaluation:

$$SRep_a(o)^t = \alpha * SExp_a(o)^t + (1 - \alpha) * \frac{\sum_{p \in P}(RRep_a(p) * Rec_p(o))}{\sum_{p \in P} RRep_a(p)} \quad . \qquad (3)$$

where α is a parameter that reflects the agent's degree of reliance on its own experience. As discussed above, usually $\alpha > 0.5$.

3.3 Reputation Evolution

After every interaction, agents can give a score of satisfaction for the interaction. The score of satisfaction for a service in real world is so subjective that it can depend on factors such as provided service quality, service quality expectation, environment (place, weather) and even mood. In order to evaluate subjective degree of satisfaction, we apply a method of quantifying degree of satisfaction based on the Quality of Service (QoS)[7] an agent a receives from another agent o. Given n dimensions of QoS (e.g., availability, service latency) d_i $(i = 1..n)$ which agent a cares about, a states in its request $(b_1, b_2,..,b_n)$ in which b_i is the value (either minimum or maximum) for dimension d_i. As a result of the service, the quality of service that a receives is represented by $(r_1, r_2,..,r_n)$, in which r_i is the value for dimension d_i. The degree of satisfaction of this interaction $(sat_a(o))$ can thus be obtained by:

$$sat_a(o) = \sum_{1 \leq i \leq n} \pi(r_i, b_i) * w_i \ . \tag{4}$$

where $\pi(r_i, b_i)$ is a function to calculate one-dimensioned degree of satisfaction with respect to requested and obtained QoS. It can take the following forms:

1. $\pi(r_i, b_i) = r_i/b_i$ when dimension i is quantitative and stronger with bigger values, for example, availability[8].
2. $\pi(r_i, b_i) = b_i/r_i$, when dimension i is quantitative and stronger with smaller values, for example, latency.
3. $\pi(r_i, b_i) = 1 - (r_i \otimes b_i)$ when dimension i is qualitative and bears boolean values, for example, confidentiality[9].
4. for dimensions whose value space is literals (e.g., *level of service* can have values of *deterministic*, *predictive* and *best-effort*), literals can be ordered from weak to strong and assign numeric values accordingly[10].

In the above equation, w_i refers to *relative importance* of a dimension to an agent (e.g., *availability* may be more important than *latency* to an agent) as defined in [19].

Experience Update

With the newest interaction, agents can update their experience value with each other. Similar to (1), updating of agent a's experience of agent o at time t (denoted as as $Exp_a(o)^t$) is as follows:

$$SExp_a(o)^t = SExp_a(o)^{t'} * \rho^{(t-t')} + sat_a(o) * (1 - \rho^{(t-t')}) \ . \tag{5}$$

where t' is the timestamp of last experience formation.

[7] If the provided service does not meet functionality requirement, it is considered completely unsatisfactory.

[8] Normalization is necessary here because r_i/b_i does not fall into $[-1, 1]$, one normalization way is to define a perfect value (i.e., 1), e.g., five times the requested value. All values higher than perfect is considered perfect.

[9] $\otimes$ represents XOR function, i.e., $x \otimes y = 0$ if x equals y, and 1 otherwise.

[10] For example, weakest value is mapped to 1, the second weakest to 2, and so on.

Reputation Update

With a new interaction, agent can then update the reputation value of the other according to (3), taking into account the newly updated experience.

Recommendation Update

Reputation varies with time. Hence, an agent's recommendation of another agent's trustworthiness also varies with time. It is thus possible for an agent a to receive recommendation from the same peer p regarding agent o (i.e., $Rec_p(o)$) again. It is necessary for agent a to update $Rec_p(o)$ with the new recommended value. Note that we do not apply (1) here because recommendations from peers (which is supposed to be based on their *SRep*) already take into account the past behaviors.

Recommendation Reputation Update

With a new experience available, agent a can update the *RRep* of the recommender p who has recommended the newly interacted peer o.

Let us denote the difference between the newest experience value and the recommended value being $diff = |Rec_p(o) - SExp_a(o)|$. For an honest peer p, we have $Rec_p(o) = SExp_p(o)$. As stated above, reputation is subjective, but we argue that it is not arbitrary, i.e., although same kind of behavior may be of different experience to different agents, we do not expect the experience to be very contrastive. Therefore, similar to each agent's definition of threshold of trust and distrust, we propose definition of a threshold of recommendation tolerance for each agent, which defines the maximal tolerance of agent for recommendation bias (denoted δ_a in the following). The value of $diff$ reflects the accuracy of recommendations, which needs to be normalized: $diff = \frac{1-diff}{\delta_a}$.

Then the recommendation reputation is updated as follows:

$$RRep_a(o)^t = RRep_a(o)^{t'} * \rho^{(t-t')} + diff * (1 - \rho^{(t-t')}) \ . \tag{6}$$

It can be seen that with false recommendation (i.e., negative $diff$), the *RRep* tears down with time. In order to make it possible for a disreputable agent's *RRep* to improve, we supplement the equation with an update method when $RRep_a(o)$ is already below σ_a, i.e., $RRep_a(o)^t = \sigma_a + \epsilon + diff * \rho^{(t-t')}$, where σ_a is an agent-defined reputation threshold value for being considered trustworthy in service recommendation, and ϵ is a small positive value.

With our reputation evaluation as shown above, it is possible that an honest recommender whose "taste" is very different from the evaluator agent a (i.e., $diff > \delta_a$) is mistaken as a dishonest agent. This does not affect our model's validity because those agents' recommendations are of little value to agent a anyway. The power of our reputation system to deter inactivity lies in the dynamics of agents' behavior (e.g., trustworthy agents become deceitful) . If an agent never recommends (i.e., never exchanges reputation information with other peers), its *RRep* will remain as *ignorance*. Although ignorance bears the value of 0, it is highly possible that many agents are reluctant to exchange reputation information with agents whose *RRep* bears the value of *0* (it is not considered

trustworthy either way). If an inactive agent did recommend but stays lazy after, it is likely that its recommended agents change their behavior, which makes its recommendation inaccurate and its *RRep* low. Therefore, the only way to maintain decent *RRep* is to recommend actively and honestly.

Reputation Propagation

For every some period[11], the recommendation manager tries to contact peers – preferably the agents with good *RRep* – for reputation information exchange. In the mean time, if a recommendation manager receives a recommendation exchange request from a peer, it will first check the requester's *RRep*. The exchange proceeds only if the requester's *RRep* is above the agent-defined threshold value.

4 Reputation Mechanism Evaluation

In order to evaluate the effectiveness of our reputation mechanism to help agents distinguish honest and dishonest agents, and interact with unfamiliar agents, we carry out three sets of simulation tests.

Experiment Setting

Our experiment is set up with 100 agents including:

1. Agents A: it includes 30 agents which are trustworthy in both service provision and recommendation.
2. Agents B: it includes 30 agents which are trustworthy in service provision but untrustworthy in recommendation.
3. Agents C: it includes 40 agents which are untrustworthy in both service provision and recommendation.

We track agents' reputation in *nRound* rounds. For each round, $nInt * 2$ agents are randomly selected to interact with each other (before the interaction happens, they evaluate each other's *SRep* to decide whether to have the interaction); and $nRec * 2$ agents are randomly picked to exchange recommendation (similarly, they evaluate each other's *RRec* to decide whether to exchange).

RRec vs. SRec

The first experiment aims to show the advantages of having separate reputation for service provision and service recommendation. We set $nRound = 100, nInt = 30$, and set $nRec$ to 5, 10, 15,..,50. We are interested in the number of resulting mistakes during the interactions. A *mistake* occurs when one agent misjudges another agent and mistakenly interacts with an untrustworthy agent or avoids a trustworthy agent. To simulate the openness of the network, every agent evaluates another peer only by the recommendations obtained from

[11] The length of period depends on the agent's recent interactions. For example, if the agent meets strangers frequently in the recent period, it implies that it has to rely more on recommendations from peers. The need for reputation information from peers becomes stronger and the length is decreased accordingly.

its peers (otherwise most of the interactions are between agents who have encountered each other before). Figure 2 shows the different number of mistakes occurred with or without using *RRep* in the last 50 rounds[12].

We can see from the figure that, with increasing exchanges of reputation information, mistakes are decreasing for both cases. However, mistakes are less with the use of *RRep*, due to the impact of 30 agents (Agents B) which are honest in service provision but deceitful in recommendation. And with full exchange of reputation information (i.e., $nRec$=50, which means in each round, each agent exchanges reputation information with another agent), the number of mistakes decrease from 507 to 172 out of a total of 3000 interactions.

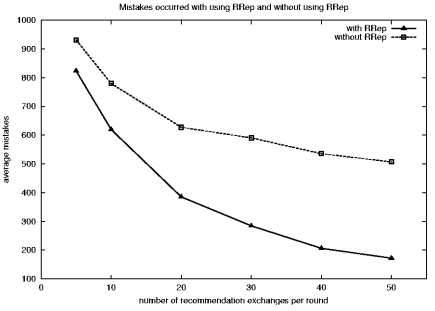

Fig. 2. Mistakes with and without RRep

Defense against Dynamic Behaviors

The second experiment aims to show the robustness of our reputation mechanism against dynamic behaviors of agents (e.g., some honest agents become deceitful). It exhibits the power of our mechanism to incentivize active reputation information exchange.

nRound is set to 500. In order to simulate the behavior dynamics, it is set that at round 50, agents B become honest in service recommendation and agents A become inactive and do not exchange reputation information with peers. We benchmark the average *RRep* of agents A, which indicates the trustworthiness in service recommendation of agents A in the view of their peers. Figure 3 shows the evolution of the average *RRep*s of agents A when they are active and inactive. Although the average *RRep* of agents A declines in both cases after agents B change their behaviors at time 50, it can be seen that if agents A stay active exchanging reputation information with other peers, their average *RRep* picks up after some time; otherwise, their average *RRep* keeps dropping.

Defense against Dishonest Recommendation

The third experiment aims to show the robustness of our reputation mechanism against dishonest recommendations. It shows our mechanism's capability to incentivize honest recommendation.

[12] In the initial phase, agents have no information of each other. Thus we only consider the last 50 rounds when each agent has built up a knowledge base for reputation evaluation.

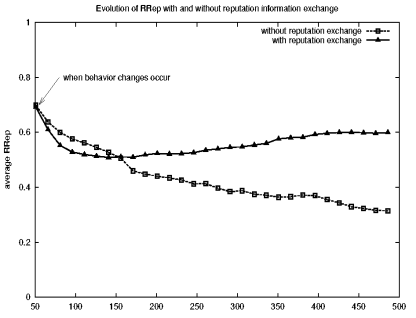

Fig. 3. Changes of RRep with active and inactive without exchange

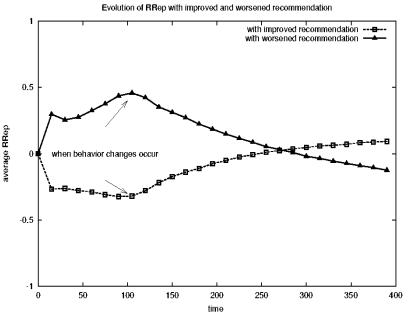

Fig. 4. Changes of RRep with higher and lower trustworthiness of RRep

The experiment set includes 500 rounds (i.e., $nRound = 500$). At round 100, agents B become trustworthy in service recommendation. Similar to the above experiment set, we benchmark agents B's average *RRep*. It can be seen from Fig. 4 that agents B have established good service recommendation reputation by round 300. Similarly, suppose at round 100, agents A become deceitful in service recommendation (other agents stay unchanged). Figure 4 shows that the average *RRep* of agents A falls below 0 by round 250. This proves the dynamics of reputation in our model: reputable recommenders' *RRep*s tear down if they recommend falsely and vice versa.

5 Related Work

Marsh [12] is among the first to present a formal trust model, incorporating properties of trust from psychology and sociology. It is well-founded yet complex model; it does not model reputation in the trust model. Mui, *et al*, [14] review the existing work on reputation across diverse disciplines and give a typology of reputation, classified by the source of reputation. Our reputation model incorporates two types of reputation: *interaction derived reputation* and *propagated reputation*.

Many reputation systems do not differentiate the reputation of service provision and recommendation [3,20,5], or assume the truthfulness of recommendations [6]. Some systems allow only positive recommendations [6] or only negative recommendations [5].

Abdul-Rahman and Hailes [21] present a trust model, incorporating direct trust based on interaction experiences and recommender trust, which is similar to our recommendation reputation. False recommendation are dealt by recording the difference between the recommended value and the experienced value. The difference is then applied to obtain a "true" value. The result is, however, uncertain when the difference is not fixed but varied. Additionally, their work does not provide incentives to give recommendations or punishment for those giving false information.

Pretty Good Privacy (PGP) [22] proposes a *Web of Trust* decentralized authentication scheme, by associating a public key (i.e., a recommender) with its trustworthiness of recommending name-public key binding. Agents can validate an unknown name-public key binding, or peers' credentials [23], through aggregate trust of recommendation (e.g., if a binding is recommended by a *completely trusted* key, it is considered valid). However, the degree of trustworthiness is static and assigned by users subjectively. Thus, it does not apply to dynamic scenarios. Reputation in our work evolves with behavior and time.

Jurca and Faltings [7] propose an incentive-compatible reputation system by introducing special broker agents named *R-agents*, which sell reputation information to and buy reputation information from agents. The payoff for an agent selling reputation information to an R-agent depends on whether its provided information coincides with the future reports on the same agent. The effectiveness of the proposed mechanism lies greatly on the integrity of R-agents, which assumely always exist in the system. In addition, collusion is not considered. Our mechanism defends against both collusion and defame attack by associating a reputation with each agent's recommendation behavior. Dishonest recommenders suffer low recommendation reputation, and thus their recommendations are either excluded or considered very trivial (i.e., assigned a small weight).

6 Conclusion and Future Work

In this paper, we have presented an enhanced reputation mechanism for mobile ad hoc networks by modeling reputation with two important dimensions, time and context, and incorporating reputation formation, evolution and propagation. Our scheme is distributed, effective and storage-saving without reliance on any trustworthy party or centralized storage.

Besides looking into incentive counterpart in sociology and psychology, our future work also includes a more formal analysis of context. As discussed, context is a multiple-facet notion, and can depend on many factors, whether subjective or objective.

We notice the problem of scalability issue with our approach. Although our mechanism does take care of the storage problem, it may still overload nodes given large distributed networks of tens of thousands of terminals. An intuitive approach is to incorporate a caching scheme with some replacement algorithm. However, discarding reputation information can be costly and requires careful tradeoff consideration.

Like most reputation systems, another unaddressed issue is changing of identities. Most online reputation systems protect privacy and each agent's identity is normally a pseudonym. It causes problems because pseudonym can be changed easily [3,10]. When a user ends up having a reputation lower than that of a new comer, the user is tempted to discard her initial identity and start from the beginning. This suggests the necessity of special treatments of new users. We plan to incorporate defense against this kind of attack in our future work.

References

1. Miller, N., Resnick, P., Zeckhauser, R.: Eliciting honest feedback in electronic markets. Working Paper (2002)
2. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Communication Surveys **3** (2000)
3. Zacharia, G., Maes, P.: Trust management through reputation mechanisms. Applied Artificial Intelligence **14** (2000) 881–907
4. Resnick, P., Zeckhauser, R., Friedman, E., Kuwabara, K.: Reputation systems. Communications of the ACM **43** (2000) 45–48
5. Buchegger, S., Boudec, J.Y.L.: Performance analysis of the CONFIDANT protocol. In: Proc. of MobiHOC. (June 2002)
6. Michiardi, P., Molva, R.: CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: CMS'2002. (August 2002)
7. Jurca, R., Faltings, B.: An incentive compatible reputation mechanism. In: Proceedings of IEEE International Conference on E-Commerce, CA, USA (2003)
8. Issarny, V., et al.: Developing Ambient Intelligence Systems: A Solution based on Web Services. JASE (2004, to appear)
9. Gambetta, D.: Can we trust trust? In: Trust, Making and Breaking Cooperative Relations. basil blackwell (1990) 213–237
10. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proceedings of the 35th HICSS. (2002)
11. Misztal, B.: Trust in Modern Societies. Polity Press, Cambridge, MA, USA (1996)
12. Marsh, S.P.: Formalising Trust as a Computational Concept. PhD thesis, University of Stirling (1994)
13. Cahill, V., Gray, E., Seigneur, J.M., et al.: Using trust for secure collaboration in uncertain environments. IEEE Pervasive Computing **2** (2003)
14. Mui, L., Halberstadt, A., Mohtashemi, M.: Notions of reputation in multi-agents systems: A review. In: Proceedings of AAMAS-02. (2002) 280–287
15. Karlins, M., Abelson, H.I.: Persuasion, how opinion and attitudes are changed. Crosby Lockwood & Son (1970)
16. Dellarocas, C.: The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms. MIT Working Paper (2003)
17. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: A new ensemble method for tracking concept drift. In: Proc. of the 3rd IEEE Int' Conf. on Data Mining. (2003)
18. Kollock, P.: The emergence of exchange structures: An experimented study of uncertainty, commitment, and trust. American Journal of sociology **100** (1994)
19. Liu, J., Issarny, V.: QoS-aware service location in mobile ad hoc networks. In: Proceedings of MDM 2004. (Jan. 2004, to appear)
20. Xiong, L., Liu, L.: Building trust in decentralized peer-to-peer electronic communities. In: Proc. of ICECR-5, Montreal, Canada (2002)
21. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proc. Hawaii Int'l Conf. System Science HICSS-33. (2000)
22. Zimmermann, P.R.: The Official PGP User's Guide. MIT press (1995)
23. Keoh, S.L., Lupu, E.: Trust and the establishment of ad-hoc communitie. presentation in 2nd Internal iTrust Workshop (September, 2003)