Towards Expanded Trust Management for Agents in Vehicular Ad-hoc Networks

Umar Farooq Minhas[1], Jie Zhang[2], Thomas Tran[3], and Robin Cohen[4]

[1]PhD Student at the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
E-mail: ufminhas@cs.uwaterloo.ca

[2]Assistant Professor at the School of Computer Engineering, Nanyang Technological University, Singapore
E-mail: zhangi@ntu.edu.sg

[3]Assistant Professor at the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada
E-mail: ttran@site.uottawa.ca

[4]Professor at the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada
E-mail: rcohen@ai.uwaterloo.ca

Abstract: *In this paper, we examine the challenge of designing intelligent agents to enable the sharing of information between vehicles in mobile ad hoc vehicular networks (VANETs). Our focus is on developing a framework that models the trustworthiness of the agents of other vehicles, in order to receive the most effective reports. We develop a multi-faceted trust modeling framework that incorporates role-based trust, experience-based trust and majority-based trust and that is able to restrict the number of reports that are received. We include an algorithm that proposes how to integrate these various dimensions of trust, along with experimentation to validate the benefit of our approach, emphasizing the importance of each of the different facets that are included. In addition, we clarify how our approach is able to meet various critical challenges for trust modeling in VANETs. The result is an important methodology to enable vehicle to vehicle communication via intelligent agents.*

I. INTRODUCTION

Vehicular ad-hoc networks (VANETs) arise as vehicles equipped with GPS and Wi-Fi devices find it valuable to communicate with each other. Artificial intelligence researchers have begun to explore the benefits of equipping each vehicle with an intelligent agent that is able to model the other agents in the environment, for the purpose of assisting each driver [8]. A central concern is how best to model the trustworthiness of each agent. Although various trust and reputation models have been studied for multiagent system environments, we claim that there are unique characteristics of VANETs that make it difficult for existing models to be applied directly. These characteristics include the need for real-time decision making, the sparsity of connections between agents, the importance of time and location in determining the expertise of an agent and the requirement that trust modeling be performed in a totally distributed manner. This leads to the focus of the research presented in this paper: a novel multi-faceted approach for modeling trust, for use in VANET environments. We present this model in detail, demonstrating its value in simulated vehicular settings and discussing clearly how it addresses the particular characteristics for trust modeling that are required. The result is an important first step towards the delivery of effective intelligent vehicular communication, one that is sensitive to the trustworthiness of the vehicular agents.

II. EXPANDED TRUST MANAGEMENT

In this section, we first present the design of our expanded trust model. We then provide a detailed description and formalization of the model's computation steps.

(A) Design of Our Model

In order to capture the complexity that arises between interacting agents in VANET, there is a need to have several different trust metrics with various key characteristics. We also propose that in order to derive a rather complete and comprehensive view of trust for agents in VANET environments, we will need to integrate security solutions (at the system level) for trust management, i.e. secure storage of role identities for role-based trust in our proposal.

Figure 1 illustrates the design of our expanded trust model. The core of the model is grouped by the rounded rectangle in the middle. This core consists of two parts.

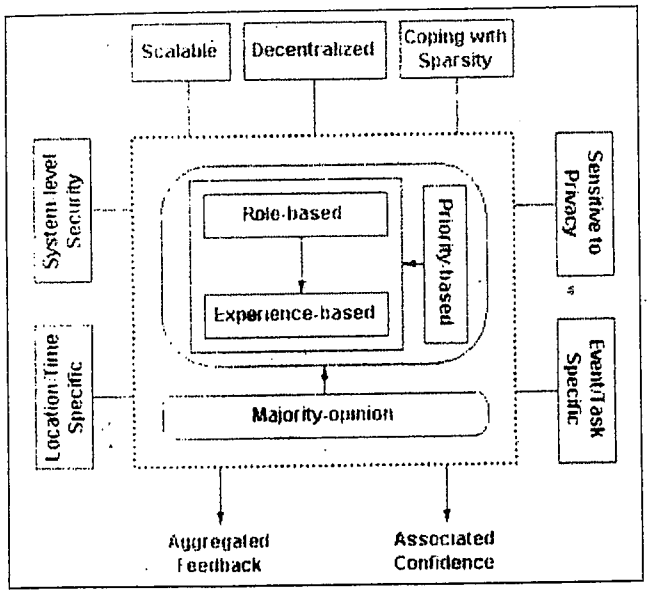

Figure 1: Expanded Trust Management

One part maintains trustworthiness of agents in order for trusted agents (advisors) to be chosen to ask for their feedback. More specifically, in this part, the trustworthiness of agents is modeled based on role-based trust and experience-based trust, which are both combined into the priority-based model that can be used to choose proper advisors.

Our role-based trust exploits certain predefined roles that are enabled through the identification of agents (vehicles). Agents can put more trust in certain agents as compared to others, i.e. agents identified as law enforcing authorities or owned by government [14]. Our experience-based trust represents a component of trust that is based on direct interactions. It is in the same spirit of incorporating evidence from direct interactions into trust calculation through Interaction Trust as proposed by [19] or the Individual Dimension of trust in the model proposed by [16]. Formalization of these two trust metrics will be presented in Section II-B.

The other part of the core is a majority-opinion approach to aggregate feedback from selected advisors. Detailed procedures for these processes will be further discussed in Section II-B. More importantly, our management of trust has several key properties represented by rectangles around the core in the figure. It aims to be decentralized, location/time specific, event/task specific, able to cope with the data sparsity problem, cumulative in order to be scalable, sensitive to privacy concerns, and able to support system-level security. These properties will be extensively discussed in Section VI after the model is clearly described.

The outcome of our trust management is aggregated feedback for a certain request/event and an associated confidence value. The aggregated feedback is eventually affected more heavily by highly trusted advisors. The confidence would depend on the reliability of estimated experience-based trust of each other agent and the maximum acceptable error rate for the aggregated feedback. In general, a higher value of confidence, i.e. a value closer to 1, would result from considering more evidence or metrics having high reliability, for a fixed error rate. We can view confidence as a parameter that adds another dimensionality to the output of the model allowing the agents to have a richer notion of trust and finally decide how to react on the reported event.

(B) Computation Procedure

An agent in a VANET environment may receive reports from other agents about an event, i.e. traffic or collision ahead of the agent. Once it receives a report, it may need to verify (double check) if the information given by the sender is reliable by asking other trusted agents. The agent will need to aggregate senders' reports. Values calculated in this manner can then be used by the agent to decide whether to believe a particular report and take corresponding actions. For this purpose, each agent in our system keeps track of a list of other agents. This agent updates all report senders' trustworthiness after the truth of their reported events is revealed. The above two processes of aggregating reports and updating trust will take into account the context in general, the agent's notion of which other agents it is interacting with, the notion of which group the other agents belong to or the roles assigned to the other agents, the time of reported event together with the time of message arrival, the relative locations of the other agents, and the actual contents of the message to evaluate task/event specific trust etc. Next, we provide detailed description and formalization of each step in our computation procedure.

(1) Computation Steps: *Four elements are incorporated into our overall trust model as its core, shown in Figure 1: (1) Experience-based trust; (2) Role-based trust; (3) Majority opinion (or social network of trust); (4) Priority-based trust. Our computation procedure consists of four steps.*

Step 1: Depending on the task at hand, set a value n = number of agents whose advice will be considered. This incorporates task-based trust. For example, if the agent needs a very quick reply, it may limit $n = 2$ or 3; if the agent is planning ahead and has time to process responses, n could potentially be larger.

Step 2: Using n, construct an ordered list of agents to ask. The list will be partitioned into groups as follows:

$$\begin{bmatrix} G_1: & a_{11}, & a_{12}, & a_{13}, & \ldots, & a_{1k} \\ G_2: & a_{21}, & a_{22}, & a_{23}, & \ldots, & a_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ G_j: & a_{j1}, & a_{j2}, & a_{j3}, & \ldots, & a_{jk} \end{bmatrix}$$

where $jk = n$.[1] This priority list is ordered from higher roles to lower roles, i.e. G_1 is the highest role. Within each group of agents of similar roles, the group is ordered[4] from higher (experience-based) trust ratings to lower ratings. Hence, role-based trust and experience-based trust are combined into this priority-based approach. These two trust metrics will be further discussed later in this section.

Step 3: When an agent requires advice, the procedure is to ask n agents the question, receive the responses and then perform some majority-based trust measurement.

Step 3B: The processing of the responses is as follows: if there is a majority consensus on the response, up to some tolerance that is set by the asker (e.g. I want at most 30% of the responders to disagree), then this response is taken as the advice and is followed. We will formalize this majority-based trust later in this section.

Step 3C: Once this advice is followed, the agent evaluates whether this advice was reliable and if so, personal experience trust values of those agents are increased; if not, personal experience trust values of those agents are decreased. Detailed formalization of this process will also be given below in this section.

Step 3D: If a majority consensus cannot be reached, then requiring majority consensus for advice is abandoned. Instead, the agent relies on role-based trust and experience-based trust (e.g., taking the advice from the agent with highest role and highest experience trust value).

Step 4: To eventually admit new agents into consideration, when advice is sought, the agent will ask a certain number of agents beyond agent in the list. The responses here will not count towards the final decision, but will be scrutinized to update experience-based trust values and some of these agents may make it into the top n list, in this way.

(2) *Role-based Trust:* Our role-based trust exploits certain predefined roles assigned to all agents in the system. The underlying assumption here is that the agents identified by authorities are more closely monitored and are expected to behave in a certain way. We can also conceptualize roles as an expected behavior of a certain group or class of agents where all the agents belonging to a group would behave similarly. We propose a role-based approach because the expected number of possible roles and the rules to assign these rules would be very few in the domain of VANETs and thus can be manually managed and/or updated by a trusted authority (for example, the vehicle licensing authorities upon the registration of a vehicle). Note that the concept of seniority (expertise in a certain context/task, for instance) could be incorporated into role-based trust.

To demonstrate our role-based approach, let's consider a simple system that recognizes the following four different roles listed in decreasing order[2], i.e. from the highest role to the lowest one: (1) authority, (2) expert, (3) seniority, and (4) ordinary. Each role level may also be associated with a role-based trust value where higher level roles have larger values. The rules for assigning and authenticating these roles can be structured as follows:

(1) Agents representing authorities such as traffic patrols, law enforcement, state or municipal police etc. assume the authority role.

(2) Agents specialized in road condition related issues such as media (TV, radio or newspaper) traffic reporters, government licensed and certified instructors of driving schools etc. receive the expert role.

(3) Agents familiar with the traffic or road conditions of the area in consideration, e.g. local people who commute to work on certain roads or highways or have many years of driving experience with a good driving record (e.g. taxi drivers), are given the seniority role.

(4) All other agents are considered having the ordinary role.

All agents should possess certificates issued by a trusted certificate authority for authentication purpose. Note that we need a way for an agent to tell if another agent is indeed having the role that it is claiming to have. One possible solution to this problem is to make use of public-key certificates in an asymmetric cryptosystem as follows: Each agent should have a public key certificate, which can simply be a document containing the agent's name, its role and its public key. That document is signed by a trusted certificate authority (with the certificate authority's private key) to become the agent's public key. Everyone can verify the authority's signature by using the authority's public key. When agent A sends a message to agent B, A must sign the message with its private key. B then can verify (using A's public key) that the message was truly sent by A.

(3) *Experience-based Trust:* We track experience-based trust for agents in the system, which is updated over time, depending on the agent's satisfaction with the

advice given, when asked. Our experience-based trust is cumulative in the sense that it updates agents' trust recursively. Thus, only the most recent trust values and the number of interactions between agents are needed to be stored, to make the system scalable. We here formalize the computation of this trust.

If we define the range of all personal experience trust values to be the interval (–1, 1), where 1 represents absolute trust and -1 represents absolute distrust, then we can use the following scheme to update an agent's personal experience trust value, as suggested by [21].

Let $T_A(B) \in (-1, 1)$ be the trust value indicating the extent to which agent A trusts (or distrusts) agent B according to A's personal experience in interacting with B. After A follows a piece of advice from B, if the advice is evaluated as reliable, then the trust value $T_A(B)$ is increased by

$$T \leftarrow \begin{cases} T + \alpha(1-T) & \text{if } T \geq 0 \\ T + \alpha(1+T) & \text{if } T < 0 \end{cases} \quad (1)$$

where $0 < \alpha < 1$ is a positive increment factor. Note that we substitute $T_A(B)$ by T to simplify the notations.

Otherwise, if B's advice is evaluated as unreliable, then $T_A(B)$ is decreased by

$$T \leftarrow \begin{cases} T + \beta(1-T) & \text{if } T \geq 0 \\ T + \beta(1+T) & \text{if } T < 0 \end{cases} \quad (2)$$

where $-1 < \beta < 0$ is a negative decrement factor.

The absolute values of α and β are dependent on several factors because of the dynamics of the environment, such as the data sparsity situation and the event/task specific property. For example, when interaction data is sparse, these values should be set to be larger, giving more weights to the available data. For life-critical events (i.e. collision avoidance), $|\alpha|$ and $|\beta|$ should be larger, in order to increase or decrease trust values of reporting agents more rapidly. Also note that we may set $|\beta| < |\alpha|$ by $|\beta| = \mu|\alpha|$ and $\mu < 1$ to implement the common assumption that trust should be difficult to build up, but easy to tear down.

We also incorporate a forgetting factor $\lambda (0 < \lambda < 1)$ in Equations 1 and 2, allowing A to assign less weight to older interactions with B. This is to cope with the possible changes of B's behavior over time. If we define t as the time difference between the current interaction and the previous one[3], the equations then become

$$T \leftarrow \begin{cases} \lambda^t(1-\alpha)T + \alpha & \text{if } T \geq 0 \\ \lambda^{-t}(1+\alpha)T + \alpha & \text{if } T < 0 \end{cases} \quad (3)$$

$$T \leftarrow \begin{cases} \lambda^t(1-\beta)T + \beta & \text{if } T \geq 0 \\ \lambda^{-t}(1+\beta)T + \beta & \text{if } T < 0 \end{cases} \quad (4)$$

The trust values A has of B will increase/decrease more slowly than those in Equations 1 and 2 because older interactions between them are discounted and have less impact on the current trust values.

(4) *Majority Opinion and Confidence:* Suppose agent A in VANET receives a set of m reports $\mathcal{R} = \{R_1, R_2, ..., R_m\}$ from a set of n other agents $\mathcal{B} = \{B_1, B_2, ..., B_m\}$ regarding an event. Agent A will consider more heavily the reports sent by agents that have higher level roles and larger experience-based trust values. When performing majority-based process, we also take into account the location closeness between the reporting agent and the reported event, and the closeness between the time when the event has taken place and that of receiving the report. We define C_t (time closeness), C_l (location closeness), T_e (experience-based trust) and T_r (role-based trust). Note that all these parameters belong to the interval (0, 1) except that T_e needs to be scaled to fit within this interval.

For each agent B_i $(1 \leq i \leq n)$ belonging to $\mathcal{B}(R_j) \subseteq \mathcal{B}$ that reports the same report $R_j \in \mathcal{R}$ $(1 \leq j \leq m)$, we aggregate the effect of its report according to the above factors. The aggregated effect $E(R_j)$ from reports sent by agents in $\mathcal{B}(R_j)$ (can be formulated as follows:

$$E(R_j) = \sum_{B_i \in \mathcal{B}(R_j)} \frac{T_e(B_i)T_r(B_i)}{C_t(R_j)C_l(B_i)} \quad (5)$$

Note that location closeness $C_l(B_i)$ depends only on the location of agent B_i while time closeness $C_t(R_j)$ depends on the time of receiving the report R_j.

For the effect of all reports, the majority opinion is then

$$M(R_j) = \arg\max_{R_j \in \mathcal{R}} E(R_j) \quad (6)$$

A majority consensus can be reached if

$$\frac{M(R_j)}{\sum_{R_j \in \mathcal{R}} E(R_j)} \leq 1 - \varepsilon \quad (7)$$

where $\varepsilon \in (0, 1)$ is set by agent A to represent the maximum error rate that A can accept.

If the majority consensus is reached, the majority opinion is associated with a confidence measure. This measure takes into account the number of interactions taken for modeling experience-based trust values of reporting agents and the maximum accepted error rate ε. We define $N(R_j)$ as the average of the number of

interactions used to estimate experience-based trust values of the agents sending the majority report R_j. Based on the Chernoff Bound theorem [11], the confidence of the majority opinion can be calculated as follows:

$$\gamma(R_j) = 1 - 2e^{-2N(R_j)\varepsilon^2} \quad (8)$$

III. EXAMPLES

In this section, we demonstrate several example scenarios to go through the important calculations in Section II-B. In these scenarios, some agents are deceptive and provide unreliable advice to other agents. We also adjust different parameters to show how they affect the results of our calculations.

(A) Experience-based Trust and Forgetting Factor

To illustrate how experience-based trust values are updated according to Equations 1 and 2, let us consider a simple scenario where an agent A_0 asks, among other agents, 3 agents namely A_1, A_2, and A_3, about whether the traffic is clear on a route. Suppose agents A_1 and A_2 say that the route is clear while agent A_3 says it is not. The agent asking for information (A_0), based on the advice collectively received from the asked agents, decides to travel the route and discovers that the route is indeed clear (as said by agents A_1 and A_2). Agent A_0 now wants to update its trust values for agents A_1, A_2, and A_3. Suppose that agent A_0's previous trust values (T') for these 3 agents are 0.4, –0.1, and 0.1, respectively. Table I below summarizes the information.

Table I
An Example of Experience-based Trust Updating without using Forgetting Factor

Agents	T'	*Reliability*	T
A_1	0.4	Yes	0.58
A_2	-0.1	Yes	0.17
A_3	0.1	No	-0.35

The last column of Table I shows the current (updated) trust values (T) that agent A_0 has for agents A_1, A_2, and A_3. In this example, agent A_0 sets the increment factor α to 0.3 and the decrement factor β to –0.5. The current trust values are calculated using Equations 1 and 2 as follows:

$$T_{A_0}(A_1) = 0.4 + 0.3 \times (1 - 0.4) = 0.58$$

$$T_{A_0}(A_2) = -0.1 + 0.3 \times (1 - 0.1) = 0.17$$

$$T_{A_0}(A_3) = -0.1 - 0.5 \times (1 - 0.1) = -0.35$$

As we can see, the trust values for agents A_1 and A_2 are increased, but that of agent A_3 is decreased. As a result, agents A_1 and A_2 can be moved up in the ordered list of agents maintained by agent A_0 (as described in Step 2 of Section II-B) and have a higher chance to be consulted by agent A_0 in the future; on the other hand, agent A_3 will be moved down in the list and have a lower chance to be consulted by A_0, or even will not be consulted at all depending on how A_0 sets the value n (as described in Step 1 of Section II-B).

To demonstrate the use of the forgetting factor λ in Equations 3 and 4 when updating experience-based trust, consider an example where an agent A_0 has currently interacted once with each of 8 other agents $\{A_1, A_2, ..., A_8\}$. It asks for advice from these agents about whether the traffic is clear ahead. Some of the agents provide reliable advice, while others provide unreliable ones. While agent A_0 has interacted with these agents previously, the time difference between the previous interaction and the current interaction varies for some of these agents. We summarize in Table II the information about agent A_0's (experience-based) trust T' in the other agents after the previous interaction, the time from the previous interaction, and whether each agent's current advice is reliable.

From Table II, we can see that agent A_0 previously had a positive trust value 0.5 for agents A_1, A_2, A_3, and A_4, but a negative trust value -0.5 for agents A_5, A_6, A_7, and A_8. The time difference between the previous interaction and the current interaction is the same (0.1) for A_1, A_3, A_5, and A_7, but is greater (0.5) for agents A_2, A_4, A_6, and A_8. The advice from agents A_1, A_2, A_5, and A_6, are reliable, but those from agents A_3, A_4, A_7, and A_8, are not. We show how agent A_0's current trust T for each of the other agents is updated.

Take the update of the trustworthiness of agents A_1 and A_2 as a demonstration. Note that in this example α is set to 0.2, β is set to -0.5, and the forgetting factor λ is set to 0.6. The current trust values agent A_0 has for agents A_1 and A_2, namely $T_{A_0}(A_1)$ and $T_{A_0}(A_2)$, can be calculated according to Equation 3 as follows:

$$T_{A_0}(A_1) = 0.6^{0.1} \times (1 - 0.2) \times 0.5 + 0.2 = 0.25$$

$$T_{A_0}(A_2) = 0.6^{0.5} \times (1 - 0.2) \times 0.5 + 0.2 = 0.51$$

We can see that A_1 has a higher trust value than A_2. This is because the time difference between A_1's previous advice and the current advice is less than that of A_2. A_1's previous trust value has not been forgotten very much, but A_2's has been forgotten a lot. The same trend can be seen from Table II for each other pair of agents (i.e. A_3 and A_4, A_5 and A_6, and A_7 and A_8).[4]

Consider another example where agent A_0 sets the forgetting factor λ to 0.9. Let agents A_9 and A_{10} be the same as agents A_1 and A_2, respectively. For example, similar to A_1, agent A_9's advice to A_0 is also reliable, agent A_0's experience-based trust T' in A_9 after the previous interaction is 0.5, and the time from the previous interaction between A_9 and A_0 is 0.1. The current trust values and can be updated according to Equation 3 as follows:

$$T_{A_0}(A_9) = 0.9^{0.1} \times (1-0.2) \times 0.5 + 0.2 = 0.60$$

$$T_{A_0}(A_{10}) = 0.9^{0.5} \times (1-0.2) \times 0.5 + 0.2 = 0.58$$

By comparing $T_{A_0}(A_9)$ and $T_{A_0}(A_{10})$ with $T_{A_0}(A_1)$ and $T_{A_0}(A_2)$, we can see that the trust values that agent A_0 has for agents A_9 and A_{10} have been forgotten less than those for agents A_1 and A_2 because the forgetting factor λ is set to be larger for A_9 and A_{10}.

(B) Calculation of Majority Opinion

We also demonstrate an example of calculating majority opinion trust and determining whether a majority consensus can be reached, according to Equations 5, 6, and 7. In this example, an agent A receives two different reports about whether the traffic ahead is clear, from 6 other agents $\{A_1, A_2, ..., A_6\}$. Agents A_1, A_2, and A_3 say in their reports R that the traffic is clear, but the other agents say otherwise in their reports R'. We assume that these 6 agents have the same experience-based trust and role-based trust, which are 0.5 respectively. Table III summarizes the information about the agents' time closeness and location closeness. Agents A_1, A_2 and A_3 have time closeness and location closeness values of 0.1, while agents A_4, A_5 and A_6 have values of 0.5. This implies that agents A_1, A_2 and A_3 are closer to the location of the event, and the time for agent A to receive their reports is sooner.

Table II
An Example of Experience-based Trust Updating using Forgetting Factor

Agents	*T'*	*Time Difference*	*Reliability*	*T*
A_1	0.5	0.1	Yes	0.58
A_2	0.5	0.5	Yes	0.51
A_3	0.5	0.1	No	0.21
A_4	0.5	0.5	No	0.08
A_5	-0.5	0.1	Yes	-0.43
A_6	-0.5	0.5	Yes	-0.57
A_7	-0.5	0.1	No	-0.74
A_8	-0.5	0.5	No	-0.82

Table III
An Example of Experience-based Trust Updating using Forgetting Factor

Agents	*Report*	*Time*	*Location*	*Total Effect*
A_1, A_2, A_3	Clear	0.1	0.1	75
A_4, A_5, A_6	Not Clear	0.5	0.5	3

The aggregated effect from reports R sent by agents A_1, A_2 and A_3 can then be calculated based on Equation 5, as follows:

$$E(R) = \sum_{A_i \in A_{1,2,3}} \frac{T_e(A_i) T_r(A_i)}{C_t(A_i) C_l(A_i)} = 3 \times \frac{0.5 \times 0.5}{0.1 \times 0.1} = 75$$

The aggregated effect from reports R' sent by agents A_4, A_5 and A_6 can also be calculated as follows:

$$E(R') = \sum_{A_i \in A_{4,5,6}} \frac{T_e(A_i) T_r(A_i)}{C_t(A_i) C_l(A_i)} = 3 \times \frac{0.5 \times 0.5}{0.5 \times 0.5} = 3$$

The majority opinion is then the report R with the aggregated effect of 75. In this example, agent A sets the maximum error ε to 0.1[5]. We can see that majority consensus can be reached:

$$\frac{E(R)}{E(R) + E(R')} = \frac{75}{75+3} = 0.96 > 1 - \varepsilon = 1 - 0.1 = 0.9$$

Therefore, agent A will trust report R, which says that the traffic on the route is clear.

Now, we assume that agents A_4, A_5 and A_6 have both time closeness and location closeness values of 0.2. The aggregated effect from reports R' sent by agents A_4, A_5 and A_6 is now:

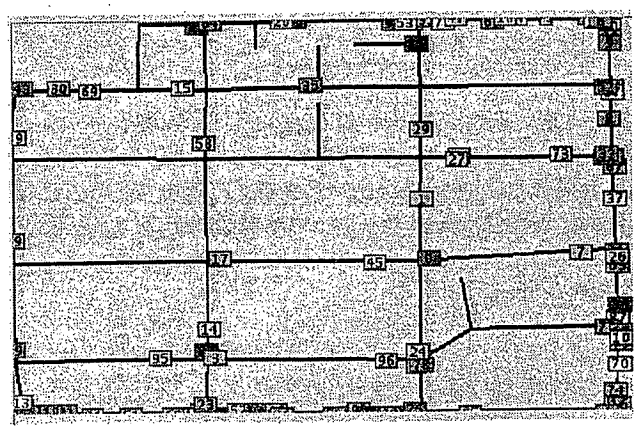

Figure 2: Simulating VANET using SWANS Simulator with STRAW Mobility Model

$$E(R') = \sum_{A_i \in A_{4,5,6}} \frac{T_e(A_i)T_r(A_i)}{C_t(A_i)C_l(A_i)} = 3 \times \frac{0.5 \times 0.5}{0.2 \times 0.2} = 18.75$$

The majority opinion is still report R, but with ε set to 0.1 majority consensus cannot be reached in this case[6]:

$$\frac{E(R)}{E(R)+E(R')} = \frac{75}{75+18.75} = 0.8 < 1-\varepsilon = 1-0.1 = 0.9$$

IV. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of our trust model in detail. Note that in this paper we only experiment with the role-based and experienced-based dimensions of our trust model while leaving a more comprehensive experimental evaluation for future work. Due to prohibitive costs, it is neither easy nor feasible to conduct VANET experiments in a real world setting; therefore, we rely on simulations to evaluate our work.

(A) VANET Simulator

Out of the few possible choices of simulators available for our use [5], [13], we choose SWANS (Scalable Wireless Ad-hoc Network Simulator). SWANS is a wireless ad-hoc network simulator that is based on Java in Simulation Time (JIST) platform proposed at the Cornell University [6]. SWANS is entirely implemented in Java, hence portable, and can potentially handle simulations involving thousands of nodes while using incredibly small amount of memory and processing power. In fact, for simulations with a large number of nodes, it has been shown to outperform the more popular simulator i.e., the Network Simulator (*ns-2*) [13]. However, performance is not the only reason why we choose SWANS. *ns-2* is tailored for research on performance aspects of different protocols such as TCP and multicast protocols but we are interested in the applications running on top of the wireless network rather than the underlying protocols. SWANS offers easier extensibility and more control over the application layer and hence is better suited for VANET simulations.

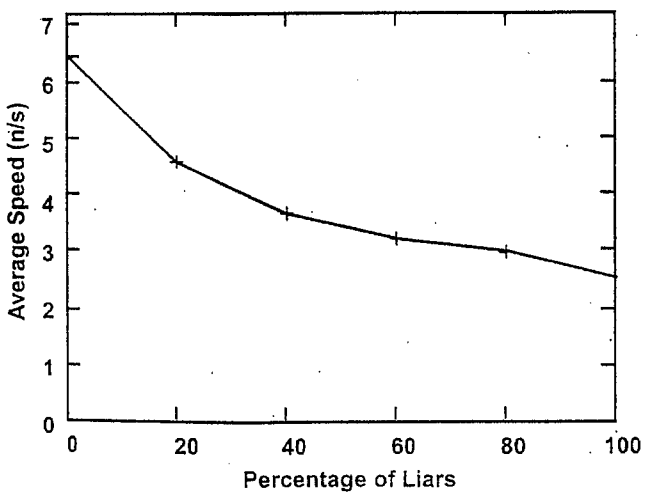

Figure 3: Average Speed of All Cars When There are Different Percentages of Liars

(B) Mobility Model

In order to run VANET simulations on a general purpose ad-hoc wireless simulator such as SWANS, we need to integrate a street mobility model with the simulator. Nodes in a general purpose network simulation are not bounded by rules. A street mobility model makes the simulated nodes follow rules such as speed limits, traffic signals, and stop signs, allowing for more realistic VANET simulations. Movement of nodes is also restricted to follow streets (or roads) as opposed to fairly free movement in a general simulation model. For our evaluation, we use the STRAW (STreet RAndom Waypoint) mobility model [2] integrated with SWANS. STRAW allows to simulate real world traffic by using real maps with vehicular nodes that follow traffic rules. Figure 2 shows a snapshot of one of our simulation runs. The map shown here is a small portion of a real world map of north Boston, Massachusetts, USA. Each node (car) in the simulation is represented as a small colored rectangle and is assigned a node id between 0 to N, where N is the total number of nodes in the simulation. Note how the STRAW mobility model simulates a real world setting by always confining the nodes to move along roads, still further restricting their movement according to current traffic conditions e.g., congestion.

(C) Experimental Settings

For all our experiments we use a single PC based desktop machine with 2x AMD Athlon 64 X2 Dual Core Processor 1.0Ghz, 2GB RAM, running Ubuntu Linux 9.04. We fix the total number of nodes to 100 and run each simulation for a total duration of 900 seconds of simulation framework time. These settings are sufficient for the experiments that we present in this paper. To test the scalability of our model, a large scale study with simulations containing 10 to 100 times more nodes would be valuable. We leave this for our future work.

(D) Performance Metric

We measure the performance of our proposed trust model by observing to what extent it can cope with deceptive information sent by malicious agents. Malicious agents in the network may send untruthful traffic information, to mislead other agents and cause traffic congestion. According to [2], we can measure congestion based on

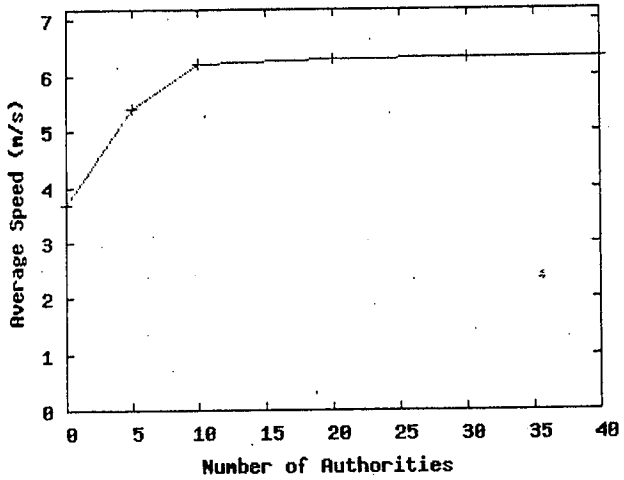

Figure 4: Average Speed of All Cars When There are Different Numbers of Authorities

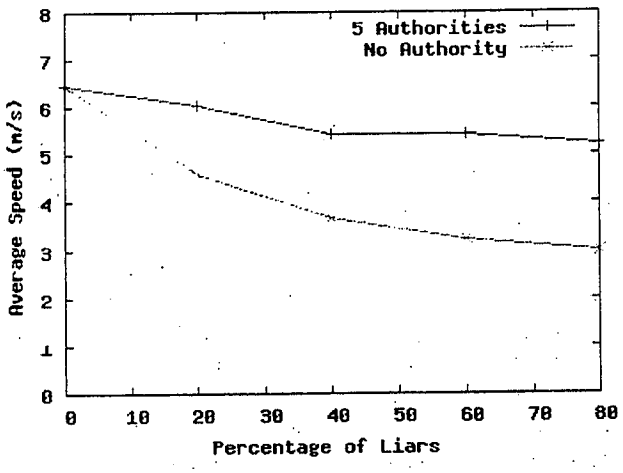

Figure 5: Average Speed of All Cars When There are Five Authorities

the average speed of vehicles. Lower average speed implies more traffic congestion. The performance of our model can then be measured as the increase in average speed of all agents by incorporating our model under the environment where malicious agents exist.

(E) Effect of Malicious Agents on Traffic Congestion

The goal of our first experiment is to quantify the effect that malicious nodes can have on the average speed of vehicles. We choose a lying strategy for the malicious nodes where they always lie about congestion on a particular road segment i.e., report congestion when there is no congestion and vice versa. We first run a simulation where we have no liars in the system and measure the overall average speed. We then perform five more simulation runs, with 20, 40, 60, 80, and 100% malicious nodes in the environment, measuring the change in average speed of vehicles in the network in each case. The results are presented in Figure 3. With no liars in the system, the overall average speed of vehicles is about 6.5 meters/second; this is our baseline. With 20% malicious nodes, the average speed drops to about 5.6 meters/second which is about 13.8% slower as compared to the baseline. We see a similar trend moving forward with 40, 60, and 80% malicious nodes. As expected, we observe the slowest average speed of about 2.5 meters/second with 100% malicious nodes in the system. This is a slowdown of about 61.5% from the baseline. This experiment proves that malicious nodes can cause congestion in vehicular networks, thus motivating the use of a trust model like the one we propose in this work to counter their effects.

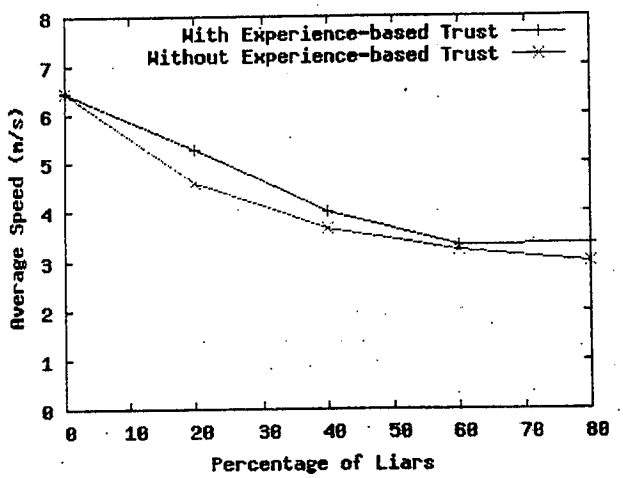

Figure 6: Average Speed of All Cars With Experience-based Trust

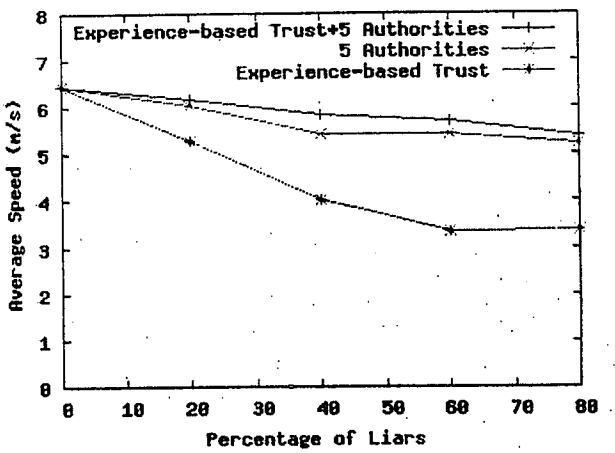

Figure 7: Average Speed of All Cars With Role-based and Experience-based Trust

(F) Reducing Traffic Congestion with Role-based Trust

In order to test the effectiveness of role-based trust we conduct an experiment where we introduce some agents in the environment with the role of authorities such as traffic patrols, as mentioned in Section II-B2. In our

simulation, authorities are assumed to be always trustworthy i.e., we set the value of T_r to 1 for authorities. In this experiment, we fix the number of malicious agents to be 40% and then vary the number of agents with the role of authority between 0 and 40. The results are presented in Figure 4. With no authorities in the system, and 40% liars, the average speed of vehicles is about 3.8 meters/second. After introducing 5 authorities, the average speed increases to 5.4 meters/second, a speedup of about 42%. Further increasing the number of authorities to 10 results in an average overall speed of about 6.3 meters/second, which is already very close to the baseline average speed of 6.5 meters/second from Figure 3 for the case where there are no liars in the system. From Figure 4, we can see that the average speed reaches a maximum with about 20 authorities. This shows that our role-based trust successfully counters the affect of malicious agents, thus reducing traffic congestion.

Next we conduct another experiment where we fix the number of authorities to five while varying the number of malicious agents from 0 to 80%. We present the results in Figure 5 This graph shows that, even with a small number of agents with a role of authority in the system, we can still effectively cope with an increasing percentage of malicious nodes.

(G) Reducing Traffic Congestion with Experience-based Trust

In the third experiment, we employ only the experience-based dimension of trust. Again, we vary the percentage of liars from 0 to 80% and measure the overall average speed of vehicles. As we can see from Figure 6 using experience-based trust results in an increase in the average speed of vehicles. This trend is consistent for all percentages of liars in the system which shows that experience-based trust is able to cope with the lying behavior of malicious agents. Even though experience-based trust results in a reduction of traffic congestion, gains are not very significant. As stated earlier, we run each simulation for a duration of 900 seconds of simulation framework time. However, during this early period agents are still lacking experience for experienced-based trust to work more effectively. With long running simulations we should see more gains from experience-based trust.

(H) Combining Role- and Experience-based Trust

From Figure 5 and 6, we can see that even though experience-based trust results in an increase in the average speed of vehicles in the network with the presence of malicious agents, role-based trust does this job more effectively. In this experiment, we combine both dimensions together and measure the average speed. These results are presented in Figure 7 As we can see, by combining these two dimensions we can achieve an average speed which is higher than when using any one of these two dimensions individually. This shows that a trust model for agents in VANETs can greatly benefit by combining several dimensions of trust as proposed in this work.

V. COMPARISON TO RELATED WORK

In this section, we discuss related work on trust models in multiagent systems. Three main categories of trust models have been proposed in multiagent systems by researchers in artificial intelligence, including learning and evolutionary trust models, reputation (social) models, and socio-cognitive models of trust. These models derive for an agent some beliefs about the honesty or reciprocative nature of its interaction partners. Some key issues of these models are highlighted to emphasize the challenges in developing an effective trust model in VANET environments, some of which motivate our proposal of a multi-faceted trust modeling framework.

(A) Learning and Evolutionary Trust Models

In learning and evolutionary trust models such as those presented in [3], [11], [17], [21], [12], [9], trust between agents emerges as a result of evolution of strategies over multiple direct interactions. In other words, an agent will learn to trust (or distrust) another agent based on its past interactions with this agent. If the past interactions with a particular agent have been particularly rewarding, the agent would then learn to associate a higher trust value resulting in a higher chance of future interactions with this agent. On the other hand, if a certain agent is known to defect over the past interactions, the other agent will choose not to deal with it in future thus representing a lower (learned) value of trust.

In these learning and evolutionary models, having multiple direct interactions among agents is the key to establishing trust and in learning to evolve strategies over time. However, in highly dynamic and open multi-agent systems such as VANETs, it is not logical to expect that this assumption will hold. Therefore, the trust models whose success depends on a certain minimum number of direct interactions between the agents, fail when directly applied to the domain of VANETs. In our multi-faceted framework, we incorporate the evidence from direct interactions, whenever available. In the trust calculation, the weight for available data can be raised to cope with the data sparsity problem, while it may have a lower default value. We also have role-based trust to distinguish trustworthy agents from untrustworthy ones to some extent.

Many of the learning and evolutionary models of trust also assume complete information about other agents and the system (e.g., strategies, payoff matrix etc.) in order to make their trust learning algorithms work. This assumption might hold in certain restrained scenarios (such as controlled simulations) but is not true in VANETs where agents are inherently limited in their capacity to gather information about other agents or the environment. Though this issue arises in any multi-agent environment where there is some degree of uncertainty about other agents and the environment, we believe that it is of far more concern in the domain of trust for VANETs and we also attribute it to the rapidly changing dynamics of the agents/environment in the context of VANETs.

(B) Reputation (Social) Models

Reputation models allow an agent to evaluate its trust in another agent by asking other agents for their opinions, to get a more reliable evaluation of the trust when direct interactions are not sufficient. Various reputation models presented in literature [15], [27], [18], [16], [23], [22], [25], [26], [24], [10], [20], provide different mechanisms to gather and aggregate opinions from other agents to calculate the trust value. Most of the reputation models assume that agents are related to each other either by the way they are connected or through various roles they play in the network giving rise to what we call a social network. Through these social relationships some agents can act as witnesses of transactions and can share this information with other agents in the form of a performance metric (e.g. good or bad) which ultimately gives rise to the concept of reputation.

Many of the above models assume a static environment (i.e., the number of agents present remains more or less constant) or allow limited dynamism if at all [18], [16], [23], [22], [25], [26], [24]. Certain models [15], [20], [10], [27] have been proposed to deal with this issue to some extent. For example, the Beta Reputation System (BRS) of [7] and the personalized approach of [27] introduce the concept of time window and a forgetting factor to deal with the possible changes of sellers' behavior. Older opinions provided by other agents about one agent will be discounted more and are assigned with less weight when modeling the trustworthiness of that agent. Opinions about the more recent behavior of the agent will be put more weight.

Teacy *et al.* [20] propose the TRAVOS model, which is a trust and reputation model for agent-based virtual organizations. This approach is based on the beta probability density function. It considers reputation advice from other agents by estimating the accuracy of the current reputation advice provided by them. This way of estimation also takes into account the possible changes of the other agents' behavior in providing reputation advice. More specifically, the TRAVOS model divides the interval of [0, 1] into a number of equal bins. It then finds out all the previous advice provided by the advisor that is similar to the advice being currently given by the other agent. The two pieces of advice are similar if they are within the same bin. The accuracy of the current advice will be the expected value of the beta probability density function representing the amount of the successful and unsuccessful interactions between the buyer and the seller when the buyer follows the previous advice.

The Bayesian Network based model of Regan *et al.* [15] considers a particular scenario where buying agents trying to choose selling agents based on the opinions of other buying agents (advisors) that have had past interactions with the selling agents. They propose that the evaluation function used by the advisors in reporting the ratings of the sellers can be learned over time by the buying agent and then can be used to produce a personalized re-interpretation of the ratings reducing the effects of a buyer's subjectivity and deception and the change in buyer and seller behavior. More concretely, they model the properties of sellers and evaluation of advisors as random variable enabling the buyer to learn a probabilistic model that can help to (implicitly) discount the effect of unreliable seller evaluations. Their model achieves higher accuracy compared to the models of BRS and TRAVOS.

However, we believe that these models still fall short in their applicability to VANETs essentially because of the rate at which agents are moving around (an average of 100km/h) and joining or leaving the network is unparalleled to any other setting. Furthermore, none of these models have been explicitly introduced to the VANET setting. We propose that any good trust model for VANETs should introduce certain dynamic trust metrics in order to capture the changes in the environment, allowing an agent to control trust evaluation depending on the situation at hand [4], [14].

(C) Socio-cognitive Models of Trust

The approaches to trust that have been presented in the previous two subsections mainly calculate trust based on the outcomes of interactions between the agents. These approaches deal with mechanisms for quantitatively evaluating trust given an agent's direct interaction with another agent or the aggregated opinion of other agents. In contrast to this quantitative evaluation of trust, socio-cognitive models of trust argue that there are certain subjective perceptions that might be important in providing a comprehensive view of trust in another agent.

For example in judging the capabilities of an opposing agent to do what it says it will one could consider the agent's (a) Competence - the belief that an agent is capable of carrying out a task; (b) Willingness - the belief that an agent has decided or intends to do what it has proposed to do; (c) Persistence - the belief that an agent is stable enough about its intensions. 4) Motivation - the belief that an agent has some motives to do what it says it will.

It should be clear that all the above beliefs are subjective in their nature and when properly weighted and combined with the quantitative metrics of trust, can greatly help to flexibly derive a much more comprehensive view of trust that neither of them may not be able to achieve individually. Castelfranchi and Falcone [1] initiate this line of research.

Socio-cognitive models of trust, because of the inherent subjectivity of the high level perceptions that they consider, are more challenging to implement as compared with other trust models. Therefore, it is not clear how these models could be applied to establish trust in different emerging applications. Although these models might not be employed independently, still they can provide a comprehensive view of trust when combined with other quantitative models of trust. In other words, one might imagine continuing to expand the multi-faceted model of trust that we have currently proposed to also include other dimensions that reflect the inherent satisfaction in the reports that are being received by other agents and to consider to a greater extent subjective differences between the different agents in the environment, regarding their tolerance of inaccurate reports.

(D) Summary of Related Work

We have discussed the challenges in employing existing models of trust and reputation for the application of managing trust in vehicular ad-hoc networks. In the following section, we step back to reflect on our particular design for trust modeling in VANETs. As we outline in greater detail the value of certain elements of our particular trust model, we introduce further brief comparisons with existing models of trust and reputation.

VI. DISCUSSION

We consider the following properties to be important to introduce into any framework for modeling trustworthiness in the environment of vehicular ad-hoc mobile networks: decentralization; coping with scarcity of information; ensuring that trust is task, location and time specific; and scalability.

To explain, in VANET environments, there is a need for prompt evaluation of the trustworthiness of information, in real-time. As such, the processing must be decentralized, performed by each vehicular agent, in order to quickly determine the paths to recommend for the driver.

It should be clear that with a massive number of cars roaming in a dynamically changing environment, opportunities to interact with the same agent may be quite rare. As several trust and reputation modeling methods rely on learning over time how to trust a source, through increased experience with that same agent [21], a distinct approach is necessary for this VANET environment. The large number of vehicles within the network also suggests that any trust modeling method must be scalable.

Finally, it will turn out to be the case that the reliability of the information provided by a source will be quite dependent on the location of the source and the time of request, relative to the expected time of travel. Cars within the immediate vicinity should have more trustworthy information and when a report is needed for the immediate future, agents within vehicles that are able to respond at once will hold more value. Traditionally, trust and reputation models build up their evaluation of an agent, over time and some do discount advice that has been provided in the past [7], but making the determination of trustworthiness both event and time dependent is not a usual feature of the modeling.

Our particular approach serves to accommodate these desirable properties for trust modeling in this environment. We first note that *decentralization* is addressed by the nature of our model (i.e., agents are distributed in the network).

Scarcity is addressed by first of all introducing role-based trust as a central element, allowing an agent to gauge the trustworthiness of a source with which it has had little or no experience, simply making use of the role of the agent and its expected trustworthiness. Another element in our solution is the action of testing the trustworthiness of certain agents whose advice will not be relied on immediately, in order to build up knowledge of these agents, if encountered in the future. A third technique that is introduced is raising the weight of data that might have a lower default value, to term the data more valuable. Note that when direct interaction has taken place, our model does accommodate making use of this information and learning about how to trust these sources, for the future.

Being *event/task* and *location/time specific* is addressed in our model by dynamically adjusting the factors α and β in our experience-based trust and by

introducing the time closeness and location closeness parameters (C_t and C_l) in the majority-based trust; that is, messages from agents closer to the event location or time are given higher weights. In addition, we assign a decay factor to messages, in order to discount older information. Finally, we map specific roles to sets of events, which can potentially be implemented easily in our role-based framework.

To consider the challenge of *scalability*, we first of all allow the user to limit the number of agents being consulted, by setting the parameter n. In addition, within experience-based trust we update agents' trustworthiness by accumulatively aggregating agents' past interactions in a recursive manner. Further, only the most recent trust ratings are stored, to make the process manageable.

We also advocate being sensitive to privacy concerns and propose making use of a public key infrastructure (PKI) to allow agents to authenticate one another.

Our research also contrasts with other artificial intelligence efforts for vehicular ad-hoc networks. For example [8] mostly serves to reinforce the need for trust modeling. In this work, simulations determine that self-interested agents who either want to simply maximize their own utility or seek to behave maliciously, can achieve road congestion.

A number of researchers have proposed trust and reputation models with role-based approach and the notion of confidence [14], [20], [11]. In particular, [19] introduced FIRE, a framework that integrates direct trust and role-based trust, in which the direct trust model of [16] is proposed as the method for capturing this element of the overall calculation, with some adjustment to consider more carefully the decay of trust values over time. In contrast, our model incorporates role-based trust and experience-based trust, which are combined using a priority-based approach, together with majority-based trust to aggregately evaluate the trustworthiness of agents while taking into consideration the important properties specific to VANET environments.

In this paper, we have proposed an expanded trust model for agents in VANET. Initial experimental results indicate that our approach works effectively for the domain of VANET. As part of the future work we plan to expand our experimental evaluation to include more complex scenarios where we test the effectiveness of other components including *event/task* and *location/time* specific components. Furthermore, it is very important to measure scalability of our trust model with increasing number of agents in the system. This is an ongoing work that presents another step towards a robust trust model for agents in intelligent vehicular systems.

NOTES

1. There is no need for each group to have the same number of elements. We provide here only a simplified example.
2. Our experience-based trust may be helpful for role categorization. When agents have sufficient experience-based trust information about each other, they may report this information to a trusted authority (i.e. the transportation department of government). A mapping between agents' real-world profiles and their trustworthiness can then be derived for helping categorize their roles.
3. The value of t may be scaled within the range of [0, 1]. This can be achieved by setting a threshold t_{max} of the maximum time for an agent to totally forget the experience happened at the time that is t_{max} prior to the current time.
4. Note that in Table II, agent A_6's trust has been decreased from -0.5 to -0.57 even though its advice is reliable. This is because the amount of A_6's trust being forgotten exceeds the increased amount of trust for providing the reliable advice. A_6's trust would get improved if the time difference were smaller, say 0.1 as agent A_5, or if a larger value for the forgetting factor were chosen, say 0.9.
5. By setting ε to 0.1, agent A wants at least 90% of the reports to agree with each other.
6. Note that if agent A allowed ε to be more tolerant, say $\varepsilon \geq 0.2$, then majority consensus would also be achieved in this case.

REFERENCES

[1] C. Castelfranchi and R. Falcone, "Principles of Trust for Mas: Cognitive Anatomy, Social Importance and Quantification," in *Proceedings of the International Conference of Multi-Agent Systems*, 1998, 72–79.

[2] D. R. Choffnes and F. E. Bustamante, "An Integrated Mobility and Traffic Model for Vehicular Wireless Networks," in *Proceedings of VANET*, 2005.

[3] Y. S. D. J. Wu, "The Emergence of Trust in Multi-agent Bidding: A Computational Approach," in *Proceedings of the 34th Hawaii International Conference on System Sciences*, **1**, 2001.

[4] F. Dotzer, "Vars: A Vehicle Ad-hoc Network Reputation System," in *Proceedings of WoWMoM*, 2005.

[5] "Global Mobile Information Systems Simulation (GloMoSim) Library," *http://pcl.cs.ucla.edu/projects/glomosim/.*

[6] "JiST/SWANS," *http://jist.ece.cornell.edu/.*

[7] A. Jøsang and R. Ismail, "The Beta Reputation System," in *Proceedings of the Bled E-Commerce Conference*, 2002.

[8] R. Lin, S. Kraus, and Y. Shavitt, "On the Benefit of Cheating by Self-interested Agents in Vehicular Networks," in *Proceedings of AAMAS*, 2007.

[9] S. Marsh, "Formalising Trust as a Computational Concept". Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

[10] Y. Mass and O. Shehory, "Distributed Trust in Open Multiagent Systems," *Trust in Cyber-societies*, Berlin: Springer-Verlag, 159–173, 2001.

[11] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation," in *Proceedings of the Hawaii International Conference on System Science*, 2002.

[12] R. Mukherjee, B. Banerjee, and S. Sen, "Learning Mutual Trust," *Trust in Cyber-societies*, Springer-Verlag, 145–158, 2001.

[13] "The Network Simulator - ns-2," *http://www.isi.edu/nsnam/ns/*.

[14] M. Raya, P. Papadimitratos, V. Gligor, and J. Hubaux, "On Data-centric Trust Establishment in Ephemeral Ad hoc Networks," *Technical Report*, LCA-REPORT-2007-003, 2007.

[15] K. Regan, P. Poupart, and R. Cohen, "Bayesian Reputation Modeling in E-marketplaces Sensitive to Subjectivity, Deception and Change," in Proceedings of the Twenty-First Conference on Artificial Intelligence *(AAAI)*, 2006.

[16] J. Sabater and C. Sierra, "Regret: A Reputation Model for Gregarious Societies," in *Proceedings of AAMAS Workshop on Deception, Fraud and Trust in Agent Societies*, 2001.

[17] S. Sen, "Reciprocity: A Foundational Principle for Promoting Cooperative Behavior among Self-interested Agents," in *Proceedings of the Second International Conference on Multi-Agent Systems*, 1996, 322–329.

[18] T. D. Huynh and N. R. Jennings and N. R. Shadbolt, "Developing an Integrated Trust and Reputation Model for Open Multiagent Systems," in *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Trust in Agent Societies*, 2004.

[19] ——, "An Integrated Trust and Reputation Model for Open Multiagent Systems," *Auton Agent Multi-Agent Syss*, **13**, 119–154, 2006.

[20] W. Teacy, J. Patel, N. R. Jennings, and M. Luck, "Travos: Trust and Reputation in the Context of Inaccurate Information Sources," *Auton Agent Multi-Agent Sys*, **12**, 2006.

[21] T. Tran, "A Reliability Modelling based Strategy to Avoid Infinite Harm from Dishonest Sellers in Electronic Marketplaces," *Journal of Business and Technology Special Issue on Business Agents and the Semantic Web*, **1**(1), 2005.

[22] Y. Wang and J. Vassileva, "Trust-based Community Formation in peer-to-peer File Sharing Networks," in *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, 2004.

[23] ——, "Bayesian Network-based Trust Model," in *Proceedings of the 6th* International Workshop on Trust, Privacy, Deception and Fraud in Agent *Systems*, 2003.

[24] B. Yu and M. Singh, "Distributed Reputation Management for Electronic Commerce," *Computational Intelligence*, **18**(4), 535–549, 2002.

[25] B. Yu and M. P. Singh, "A Social Mechanism of Reputation Management in Electronic Communities," in *Proceedings of the 4th International Workshop on Cooperative Information Agents*, 2000, 154–165.

[26] ——, "An Evidential Model of Distributed Reputation Management," in *Proceedings of International Autonomous Agents and Multi Agent Systems (AAMAS)*, Bologna, Italy, 2002.

[27] J. Zhang and R. Cohen, "Trusting Advice from other Buyers in e-marketplaces: The Problem of Unfair Ratings," in *Proceedings of the Eighth International Conference on Electronic Commerce (ICEC'06)*, 2006.