Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model

Pedro B. Velloso, Rafael P. Laufer, Daniel de O. Cunha, Otto Carlos M. B. Duarte, and Guy Pujolle

***Abstract*—In this paper, we propose a human-based model which builds a trust relationship between nodes in an ad hoc network. The trust is based on previous individual experiences and on the recommendations of others. We present the Recommendation Exchange Protocol (REP) which allows nodes to exchange recommendations about their neighbors. Our proposal does not require disseminating the trust information over the entire network. Instead, nodes only need to keep and exchange trust information about nodes within the radio range. Without the need for a global trust knowledge, our proposal scales well for large networks while still reducing the number of exchanged messages and therefore the energy consumption. In addition, we mitigate the effect of colluding attacks composed of liars in the network. A key concept we introduce is the relationship maturity, which allows nodes to improve the efficiency of the proposed model for mobile scenarios. We show the correctness of our model in a single-hop network through simulations. We also extend the analysis to mobile multihop networks, showing the benefits of the maturity relationship concept. We evaluate the impact of malicious nodes that send false recommendations to degrade the efficiency of the trust model. At last, we analyze the performance of the REP protocol and show its scalability. We show that our implementation of REP can significantly reduce the number messages.**

***Index Terms*—Trust, ad hoc networks, security.**

I. Introduction

Ad hoc networks lack the infrastructure seen in managed wireless networks. As a result, nodes must play the roles of router, server, and client, compelling them to cooperate for the correct operation of the network [1]. Specific protocols have been proposed for ad hoc networks considering not only its peculiar characteristics, but also a perfect cooperation among nodes. In general, it is assumed that all nodes behave according to the application and protocol specifications. This assumption, however, may be false, due to resource restrictions (e.g., low battery power) or malicious behavior. Assuming a perfect behavior can lead to unforeseen pitfalls, such as low network efficiency, high resource consumption, and vulnerability to attacks. Therefore, a mechanism that allows a node to infer the trustworthiness of other nodes becomes necessary [2], [3].

Providing a trust metric to each node is not only useful when nodes misbehave, but also when nodes exchange information. According to the paradigm of autonomic networks [4], a node should be capable of self-configuring, self-managing, and self-learning by means of collecting local information and exchanging information with its neighbors. Thus, it is important to communicate only with trustworthy neighbors, since communicating with misbehaving nodes can compromise the autonomy of ad hoc networks.

We present a flexible trust model based on the concept of human trust and apply this model to ad hoc networks. Our model builds, for each node, a trust relationship to all neighbors. The trust is based on previous individual experiences of the node and on the recommendations of its neighbors. The recommendations improve the trust evaluation process for nodes that do not succeed in observing their neighbors due to resource constraints or link outages. The ability of assessing the trust level of its neighbors brings several advantages. First, a node can detect and isolate malicious behaviors, avoiding relaying packets to malicious neighbors. Secondly, cooperation is stimulated by selecting the neighbors with higher trust levels. Nodes learn based on the information exchanged with trustworthy neighbors to build a knowledge plane [5], [6].

In our model nodes interact only with its neighbors. As a result, nodes do not keep trust information about every node in the network. Keeping neighborhood information implies significant lower energy consumption, less processing for trust level calculation, and less memory space. It also fits well to ad hoc networks, which are usually composed of portable devices with power, processing, and memory restrictions [7]. Moreover, topology changes, due to mobility or battery constraints, make it difficult to maintain information for all nodes [8]. Another result is that recommendations are only exchanged between neighbors, that is, recommendations are not forwarded. This approach also minimizes the probability of false recommendations since the number of received recommendations is significantly smaller and there is no intermediate node to increase the uncertainty of the information. Besides, a node can always balance the received recommendations with its own experiences to calculate the trust level because nodes do not calculate the trust level of nodes that are not neighbors. The decrease in the number of messages sent not only alleviates the network traffic, but also decreases the energy consumption.

We introduce the concept of relationship maturity, which improves the efficiency of the trust evaluation process in the

Manuscript received August 31, 2009; revised February 24, 2010. The associate editor coordinating the review of this paper and approving it for publication was M. Brunner.

M. Velloso was a post-doc researcher at LIP6, UPMC, Paris, France, He is now a research engineer at Bell Labs France.

M. Laufer is a Ph.D. student in the Computer Science Department, UCLA, California, USA.

M. Cunha is a post-doc researcher at France Telecom, Paris, France.

M Duarte is a professor at COPPE/DEL from Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

M. Pujolle is a professor at LIP6, Université Pierre et Marie Curie, Paris, France.

Digital Object Identifier 10.1109/TNSM.2010.1009.I9P0339

presence of mobility. The basic idea is to use the period of time the recommender node knows the target node as a metric to calculate the weight of its recommendation. Humans are able to know each other better as time goes by and the same idea applies here. Nodes increase the weight of the recommendations coming from older neighbors and decrease the weight of recommendations coming from new neighbors. We also propose the Recommendation Exchange Protocol (REP), which enables nodes to send and receive recommendations of their neighbors.

We present the correctness of our model through simulations. An analysis of the impact of the most relevant parameters on the trust level evaluation process is performed [9]. We also present the benefits of the proposed relationship maturity in mobile ad hoc networks [10]. The effect of liars on the trust evaluation process is analyzed [11]. The results show that the relationship maturity parameter decreases the trust level error up to 50%. Moreover, the proposed model is robust, tolerating up to 35% of liars.

In this paper, we present a detailed description of our model, which includes the architecture and its components. The REP protocol scalability is evaluated, taking into account our implementation design and our results show an overhead reduction of almost 60% with roughly no impact at the convergence rate. Finally, we present a brief discussion about the results.

The paper is organized as follows. We present our model in Section II. Details of the implementation of our model are presented in Section III. Section IV shows our simulation results. We expose the related works in Section V. In Section VI we present our conclusions and future work.

II. The trust model

The basic idea is to build a trust model that provides nodes with a mechanism to evaluate the trust of its neighbors. A node assigns a so-called trust level for each neighbor, which represents how trustworthy each neighbor is.

In our work we define trust as the value that reflects the behavior history that a node has about a specific neighbor. This information is used as an expectation of its neighbor future behavior. We extend this definition to include the recommendations of others as well. Therefore, similar to the concept of human trust, the computation of the trust level of a given neighbor is based on previous experiences and also on the opinion of other neighbors. By previous experiences, we mean that a node keeps track of the good and bad actions taken by its neighbors. A bad action is the one that does not correspond to the expected behavior. As a result, previous experiences allow a node to have a personal "opinion" about all its neighbors. Neighbor nodes can further share their own opinions in order to improve the trust level evaluation. The transmission of a personal opinion about a specific node i is defined as a recommendation. Neighbor nodes take into account this recommendation while calculating the trust level for node i. The main goal of the recommendations is to compensate for the lack of monitoring capabilities due to resource constraints. Usually, a node is not able to observe the complete behavior of a given neighbor over time. Recommendations from other neighbors are useful in this case for an accurate trust level assignment. Moreover, the use of recommendations can speed up the convergence of the trust evaluating process, as showed in Section IV. For that purpose, we introduce the concept of relationship maturity, which is based on the age of the relationship between two nodes. This concept allows nodes to give more importance to recommendations sent by long-term neighbors rather than short-term neighbors. Nodes use the Recommendation Exchange Protocol (REP) to send and receive recommendations.

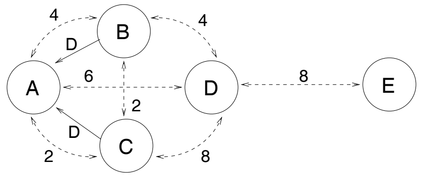


Fig. 1. Example: node A receives recommendations about node D.

Figure 1 illustrates an example of a recommendation. Nodes connected by a dotted arrow are neighbors and the number indicates for how long they know each other, namely, the relationship maturity parameter. A normal arrow represents a recommendation and the letter indicates the target node. First thing to notice is that recommendations concern one common neighbor of different nodes. In that case, node D is a common neighbor of node A, B, and C. Node B and C send their recommendation about node D to node A. Node A will consider the recommendation from node C more important than the one received from node B because node C has a longer relationship with node D. It is worth mentioning that recommendations sent by node D about node E will be ignored by node A, B, and C because node E is not a neighbor of A.

Each node assigns a trust level for each neighbor. We propose a continuous representation for the trust level, ranging from 0 to 1 where 0 means the least reliable node and 1 means the most reliable node.

Our model can be divided in two distinct plans as shown in Fig. 2. The Learning plan is responsible for gathering and converting information into knowledge. For instance, this plan is responsible for monitoring the behavior of each neighbor. The Trust plan defines how to assess the trust level of each neighbor using the knowledge information provided by the Learning plan and the information exchanged with neighbors. Both plans can interact with all layers of the TCP/IP model. Therefore, the learning process considers information from all layers and the trust information generated by the Trust plan is also available for all layers.

Since we take into account not only malicious nodes but also selfish behaviors due to resource constraints, a trust value is associated to a particular scope, like forwarding packets, sending recommendations, and other application-specific scopes. Therefore, we consider that a node might behave differently according to the scope and the resource constraints. Consequently, the type of information to be collected by the Learning plan depends on the defined scopes. For instance, for the routing process, the Learning plan must observe if neighbors respond to route requests, if they send false routes, etc.

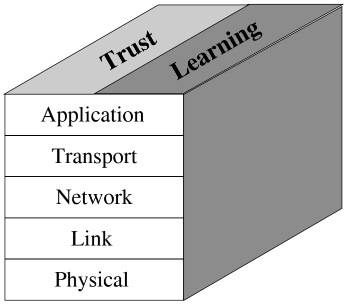


Fig. 2. The proposed trust model architecture.

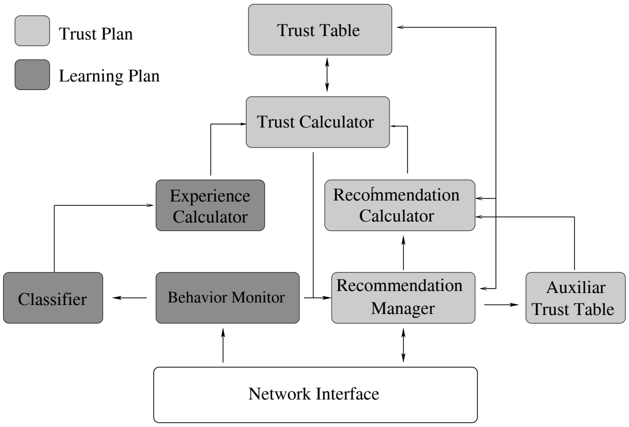


Fig. 3. The proposed trust system components.

The Learning plan relies on three basic components as displayed in Fig. 3. The Behavior Monitor observes neighbors in order to collect information about their behavior. It must be able to notice other nodes' actions and transmit them to the Classifier. The Behavior Monitor also indicates the presence of new neighbors to the Recommendation Manager. The Classifier is the component dedicated to reason about the information collected by the Monitor. The Classifier decides the quality of an action according to a previously defined classification. The Classifier then sends its verdict to the Experience Calculator. Finally, the Experience Calculator estimates a partial trust value for a given node based on the information received by the Classifier.

In this paper, we focus on the Trust plan and we assume an imperfect Learning plan, which only perceives part of the behavior of other nodes. The definition of the Learning plan functionalities is defined in Section III.

The Trust plan is composed of five main components as depicted in Fig. 3. Each node must keep a main Trust Table which contains the trust level for each neighbor. Additionally, a node can also store the opinion of its neighbors about their common neighbors on the Trust Table. Each entry on the Trust Table is associated with a timeout. Therefore, an entry is erased from the Trust Table whenever the node associated to that entry is no longer a neighbor or when it expires. All the recommendations related to that entry are erased as well. In our model, nodes can also keep an additional table that is not mandatory. The Auxiliary Trust Table (ATT) contains the variance of each trust level and for how long they keep that information, which indicates relationship maturity. The goal of the Auxiliary Trust Table is to supply nodes with additional information that improves the trust level evaluation. Nevertheless, this trust evaluation improvement requires more energy consumption and nodes with power or storage constraints can choose not to implement the entire trust system. Thus, in order to cope with the heterogeneity that characterizes ad hoc networks [12], we define three operation modes: simple, intermediate, and advanced. Nodes with low power/storage capacity operate in the simple mode, in which they use just the main Trust Table and the Recommendation Exchange Protocol (REP) protocol is optional. Nodes with a medium capacity operate in the intermediate mode, which also keeps the recommendations of other nodes. In the advanced mode, nodes implement the whole trust system with all features. The amount of saved resource and the accuracy of trust level for each operation mode depends on the monitoring, which is application-specific, and whether the REP protocol is used or not. In the rest of this paper, we consider that nodes operate in the advanced mode.

The Recommendation Manager is responsible for receiving, sending, and storing recommendations. The interactions between the Network Interface and the Recommendation Manager are performed by the Recommendation Exchange Protocol (REP). The reception of a recommendation involves two actions. First, the recommendation is stored in the Auxiliary Trust Table (ATT) and then it is forwarded to the Recommendation Calculator component. The Recommendation Calculator computes all the recommendations for a given neighbor and determines a trust value based on the opinions of other nodes. This value is passed to the Trust Calculator component. The Trust Calculator evaluates the trust level based on the trust values received from the Experience Calculator (individual experiences) and the Recommendation Calculator (neighbor recommendations). The Trust Calculator also notifies the Recommendation Manager the need of sending a trust recommendation advertisement. Our proposition only requires interactions with neighbors and only stores information about neighbors. This is an important feature for mobile ad hoc networks composed by portable devices that have energy, processing, and memory restrictions [13]

A. Trust level evaluation

We define the trust level evaluation from node a about node b, $T_a(b)$, as a weighted sum of its own trust (monitor) and the recommendations of neighbors, similar to Virendra *et al.* [14]. The fundamental equation is

$$T_a(b) = (1-\alpha)Q_a(b) + \alpha R_a(b), \tag{1}$$

where the variable $Q_a(b)$, that ranges from [0,1], represents the trust node a has on node b based only on its own observations and $R_a(b)$, that ranges from [0,1], is the aggregate value of the recommendations from all other neighbors, explained in Section II-B. The variable α, that ranges from $[0,1]$, is a parameter in our model that allows nodes to choose the most relevant factor. In our model, the value of $Q_a(b)$ is given by

$$Q_a(b) = \beta E_a(b) + (1-\beta)T_a(b), \tag{2}$$

where E_a represents the trust value obtained by the judgment of the actions of a neighbor performed by the Classifier component, and the variable $T_a(b)$ gives the last trust level value stored in the Trust Table. The variable β, that ranges from $[0, 1]$, allows different weights for the factors of the equation, selecting which factor is the more relevant at a given moment.

Equations 1 and 2 describe how the Trust Calculator combines the information from the Experience Calculator ($E_a(b)$), the Recommendation Calculator ($R_a(b)$), and the Trust Table ($T_a(b)$) to derive a trust level.

B. Recommendation computation

The trust level calculation considers the recommendations of neighbors obtained by the Recommendation Exchange Protocol (REP) described in Section II-D. In our proposal, $R_a(b)$, in Equation 1, represents the aggregate trust that the neighbors of node a have on node b.

First, node a defines a set K_a which is a subset of the its neighbors comprising all nodes whose trust level is above a certain threshold, to increase the confidence of recommendations. The recommendation, $R_a(b)$, is defined as the weighted average of the recommendations from all nodes $i \in K_a$ about node b. The weight for a recommendation from a neighbor i is the trust level that node a has on node i, as follows:

$$R_a(b) = \frac{\sum_{i \in K_a} T_a(i) M_i(b) X_i(b)}{\sum_{j \in K_a} T_a(j) M_j(b)}. \tag{3}$$

The recommendations considers not only the trust level of other nodes (T_a), but also the accuracy (X_i) and the relationship maturity (M_i). The accuracy of a trust level is based on the standard deviation, similar to Theodorakopoulos and Baras [15]. The value in the Trust Table of node a regarding node b is associated to a standard deviation $\sigma_a(b)$, which refers to the variations of the trust level that node a has observed about node b. We use X as a random variable with a normal distribution to represent the uncertainty of the recommendation. It can be expressed as

$$X_i(b) = N(T_i(b), \sigma_i(b)). \tag{4}$$

The recommendation of node i about node b is weighted by $M_i(b)$, which defines the maturity of the relationship between nodes i and b, measured at node i. The relationship maturity is a measure of the time that two nodes have known each other. We use the relationship maturity to give more relevance to the nodes that know the evaluated neighbor for a longer time. Accordingly, we assume that the trust level of a older neighbor has already converged to a common value within the network and therefore its opinion should be more relevant than the opinion of a new neighbor. It is important to notice that maturity is only considered between the recommender, node i, and the node that is being evaluated, node b, as illustrated in Fig. 1.

Malicious nodes can implement an attack exploiting the concept of relationship maturity by attributing fake trust levels. In order to minimize this effect, each node defines a maximum relationship maturity value M_{max}, which represents an upper bound for the relationship maturity. This value is based on the average maturity relationship value of its most trusted neighbors.

C. The First Trust Assignment

We divide the trust scheme in two distinct phases. In the initial phase, nodes first meet and assign a trust level to each other. The second phase is the trust level update, which assumes that the nodes have already met each other.

When a node first meets a specific neighbor, it assigns an initial level of trust to this neighbor. We classify the first trust assignment strategy as prudent or optimistic. In the prudent strategy the node does not trust strangers and considers that every new neighbor as a possible threat to the network. As a consequence, the node assigns a low value of trust for the new neighbor. On the other hand, the optimistic strategy assumes that every node is reliable until proven otherwise. In such case, the node associates a high level of trust for new neighbors. Right in the middle of these two strategies, one could think of a moderate strategy, in which the node assigns an intermediate level of trust for strangers.

The first trust assignment can also take into account the recommendation of known neighbors weighted by their trust levels. For a node a to calculate the first trust level of a node b, we propose the same approach as Equation 1, but replacing the term that reflects its own experience by the First Trust Value, (F_a), given by:

$$T_a(b) = (1 - \alpha) F_a + \alpha R_a(b), \tag{5}$$

where F_a is the value used by node a according to the adopted strategy, $R_a(b)$ is the aggregate recommendation of neighbors about node b, and α is the weight factor that allows us to give more relevance to the desired parameter.

D. The Recommendation Exchange Protocol

The recommendation from a node $i \in K_a$ includes the trust level $T_i(b)$ of the target node b, its accuracy $\sigma_i(b)$ and for how long they know each other, $M_i(b)$. For a node that does not implement the Auxiliary Trust Table the recommendation includes just the trust level $T_i(b)$.

We propose the Recommendation Exchange Protocol (REP) as a part of the Recommender Manager in Fig. 3. This protocol allows nodes to exchange recommendations among them and only considers interactions with neighbors, which significantly simplifies the protocol. Thus, all messages are transmitted by one hop broadcasts avoiding flooding in multihop communications. When using IP to broadcast the message, the Time to Live (TTL) field is set to 1. The protocol is composed of three messages: Trust Request (TREQ) message, Trust Reply (TREP) message, and Trust Advertisement (TA) message.

When nodes first meet, each one broadcasts a Trust Request (TREQ) message to their neighbors with the IP address of the new neighbor as the target node. All neighbors receive the TREQ message and check if the target node is a neighbor or not. Nodes that have the target node as a neighbor, will answer with a Trust Reply (TREP) message, which contains the recommendation about the target node, after waiting for a random period of time t_{REP} to avoid collisions and to wait for receiving other TREQs. We also define a TREP threshold under which it will not answer the TREQ. The threshold is based on the trust level of the requesting node. This strategy reduces the effect of non trustworthy nodes that repeatedly send TREQ messages. Before sending a TREQ message, a node waits for a specific period of time t_{REQ} trying to gather the maximum number of new neighbors. After t_{REQ}, the node will request the recommendations of all the q new neighbors it has collected. Thus, instead of sending q TREQ messages it sends just one with q node IDs. After sending a TREQ, the trust requesting node will wait for a specific timeout period to receive the TREPs from its neighbors. If a node does not receive any TREP, it ignores the recommendation of its neighbors by choosing $\alpha = 0$ in Equation 5.

During a trust level update, the Trust Level (TL) may change. If the trust level changes significantly, the node sends a Trust Advertisement (TA) message to notify its neighbors about the change. In order to prevent nodes from sending TA messages for every change in the Trust Level, we defined the TA threshold (π) as a minimum difference, between the new TL and the TL in the last recommendation sent, above which nodes must announce the new TL by sending a TA. The reception of a TA message does not imply a recalculation of the trust level to reduce the effect of malicious nodes that send TAs to trigger trust level recalculation in other nodes. The recalculation is triggered by the perception of an action.

E. Authentication mechanism

An authentication mechanism is essential, because malicious nodes may pretend to be another node. Nevertheless, our model does not require a sophisticated authentication mechanism. Nodes do not need to know nor recognize any other node *a priori*, namely, a node does not need to identify a new neighbor when it arrives. In our system, nodes must be able to identify neighbors that they already know. Therefore, there is no need of a certification authority. Hence, nodes must exchange identifiers when they first meet and keep a neighbor identifier during all the period they remain in the radio range of each other. Thus, a pair of public/private key for each node is enough to allow our mechanism to work adequately. It is important to notice that there is no correct identifier and a node might use different identifiers. However, the Sybil attack is not a real problem for the proposed mechanism, because nodes must behave in order to have a high trust level. Therefore, even though a node may have multiple identities, its neighbors will be able to identify the benign ones, and will avoid interacting with the malicious ones Nevertheless, authentication mechanisms are not in the scope of this work.

III. The Trust Model Implementation

We have developed a simulator, which is specifically designed for our model, in order to evaluate and identify the main characteristics of the proposed model.

In ad hoc networks, nodes can perform several actions, like sending packets, forwarding packets, responding to routing messages, sending recommendations, among others. The set of performed actions define the node's behavior. Therefore, the Learning plan monitors the neighbor's actions trying to evaluate their behavior. In our simulator, each node performs good actions and/or bad actions. The time between two consecutive actions performed by a node is exponentially distributed (mean = 5 time units). The kind of action that will be performed depends solely on the nature of the node. A node with a nature equals to 0.8 means that it performs eight good actions out of ten.

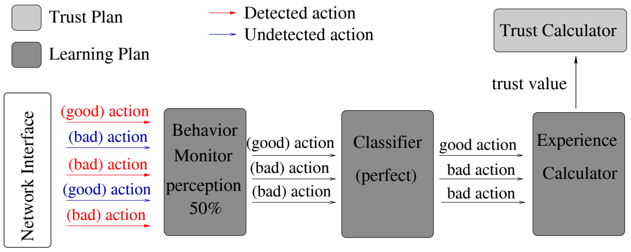


Fig. 4. The Learning plan implementation.

The nature of a node ranges from 0 to 1. Trustworthy nodes have nature equals to 1 while untrustworthy nodes have nature equals to 0. The nature is used as a reference of the ideal global trust level that a node should receive by its neighbors. We use it here as a metric to evaluate how close the measured global trust level of a node actually gets from its nature.

We emulate the Behavior Monitor (Fig. 3) by introducing in our simulator the concept of perception. The perception indicates the probability of noticing a certain action. Each Behavior Monitor presents its own perception. Therefore, a node with a perception of 0.6 is able of noticing 60% of all the actions performed by its neighbors. Figure 4 illustrates the Learning plan components. The Behavior Monitor passes all the perceived actions to the Classifier without knowing its nature. In our simulator, we assume a perfect Classifier, which means that the judgment of an action always matches with the original nature of the action. It is worth to mention that noticing and judging an action does not imply using promiscuous mode. We believe that a node should be able to decide whether it will use promiscuous mode or not based on its own constraints and needs. Thus, nodes may decide not to use promiscuous mode at the expense of having a lower perception. Therefore, the perception parameter can reflect nodes that operate in simple and intermediate modes. Finally, the judgments are transmitted to the Experience Calculator.

For the Experience Calculator, we propose a simple approach which consists of evaluating the trust value based on a set of the last i perceived actions from the same neighbor. This implies the existence of a minimum number of actions i_{min} that a node must notice from each neighbor before having a concrete opinion about them, based on its own experience. It means that during the initial phase of first contact, nodes use just the recommendations of its neighbors to evaluate the trust level of the new one. The minimum number of perceived actions is crucial for the accuracy of the measure. A higher perception allows a more accurate result. At the same time, a large number of necessary initial actions leads to a longer delay for assessing the trust value for new neighbors, leading to a higher convergence time. For the simulations, we assume the Experience Calculator considers the last 10 actions from a neighbor to estimate the trust value.

IV. Results

In this section, we present the results of the experiments. First, we expose the results that demonstrate the correctness of our model and the impact of the main parameters on the trust evaluation process. Then, we evaluate our model in mobile multihop ad hoc networks. We show the effectiveness of the relationship maturity parameter and how the other parameters are tuned to improve the trust evaluation in mobile scenarios. The last results assess the robustness of our model to slander attacks. All nodes operate in the advanced mode, which means that they implement all the features of the proposed system, as described in Section II. The mean value of the time between two actions performed by a node is set to 5 units of time. All results are presented with a confidence interval of 95%.

A. Performance on Small Networks

Our main goal in this experiment is to evaluate and analyze the influence of the number of neighbors, the first trust assignment strategy, and the variation of the parameters α and the perception τ on the trust evaluation process. We assumed in this experiment small ad hoc networks in which all nodes are at most one hop away from each other, which we call single hop networks. The reason for only analyzing single hop networks is to isolate all the problems related to multihop networks and focus strictly on the dynamics of our model. Results for multihop networks are presented in the next section

The simulation scenario consists of nodes with 250 m transmission range, which are randomly placed in a 150 m × 150 m area. We defined three values for the first trust assignment: 0.1 for the prudent, 0.5 for the moderate, and 0.9 for the optimistic strategy. All nodes adopt the same strategy. We also assume $\alpha = \beta = \tau = 0.5$. These are the standard values for the simulations. For each specific configuration, the parameters that differ from these standard values are outlined. At last, in each configuration, all nodes have the same nature. The time unit corresponds to seconds.

Figure 5 presents the time response of the average trust level from all neighbors about a specific node. In this specific scenario, composed of 4 nodes, the nature of nodes is set to 0.2 and the simulation time is 900 units.

We observe in Fig. 5 that the trust level value begins in a certain level but tends to the expected trust level. The expected (correct) level is the nature of the node that is being analyzed, which is set to 0.2. After a specific amount of time $t_1 \approx 350$ units, the curve approximates the correct value. Thus, we verify the existence of a transient period and stationary period. In the transient period, nodes are trying to approximate to the expected value, while in the stationary period the trust level is stable, very close to the correct value.

From now on, instead of presenting the average trust level, we present the average error for the trust value evaluated, that is, the difference between the trust level and the correct value, given by the nature of the node. The ideal result is a curve that reaches the zero value, which means that there is no error between the average trust values calculated by the neighbors and the value of the nature of the node. For the next results, the scale corresponds to a minute (60 time units).

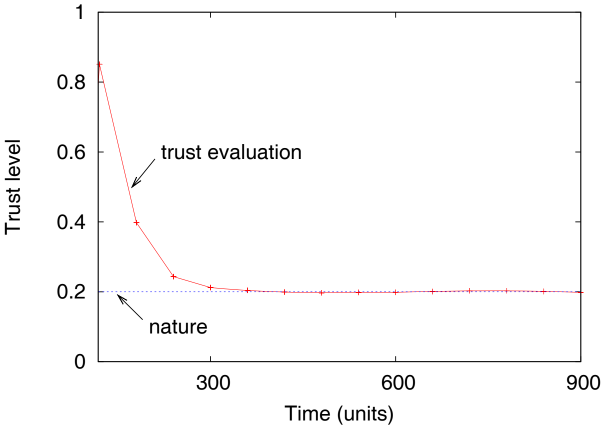


Fig. 5. Behavior of the trust level in both transient and stationary periods, for $\alpha = \tau = 0.5$.

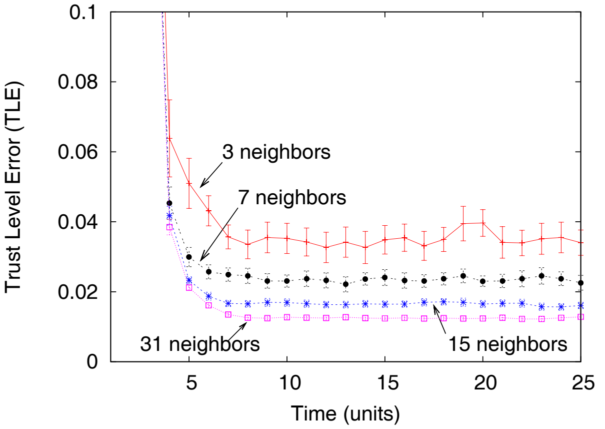


Fig. 6. Influence of the number of neighbors on the Trust Level Error, for $\alpha = \tau = 0.5$.

In Fig. 6, nodes adopt an optimistic strategy. We vary the number of nodes, N, and, consequently, the number of neighbors of each node, which is given by $N - 1$. The nature is set to 0.2. We can notice that the error gets closer to zero as the number of neighbors increases. This behavior occurs because increasing the number of neighbors results in an increase of the number of recommendations, which implies a greater probability of receiving recommendations closer to the correct value.

Figure 7 shows the influence of the parameter α on the trust level evaluation with 15 neighbors. Decreasing α implies that the recommendation of other nodes has a minor effect in the trust level calculation. Although the global opinion about a specific node changes slower when α is larger, the convergence value is closer to the expected one (lower TLE) and presents a smaller variation.

As discussed in the Section III, the perception τ is the fraction of actions a node can notice from its neighbors. Figure 8 shows the impact of the perception on the trust level evaluation. It is clear that the perception is strongly related to the duration of the transient period. It occurs because a node

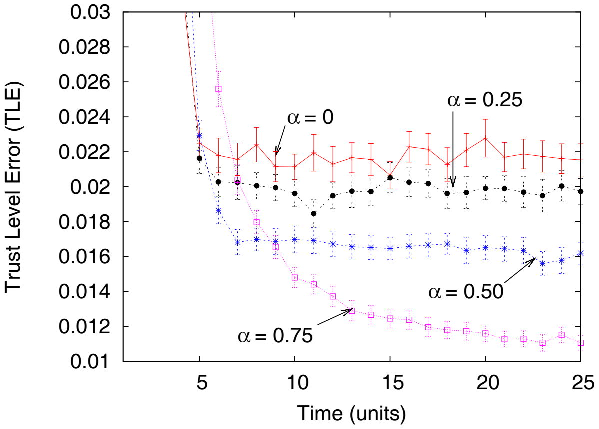


Fig. 7. The influence of α on the transient period and on the TLE, for $\tau = 0.5$.

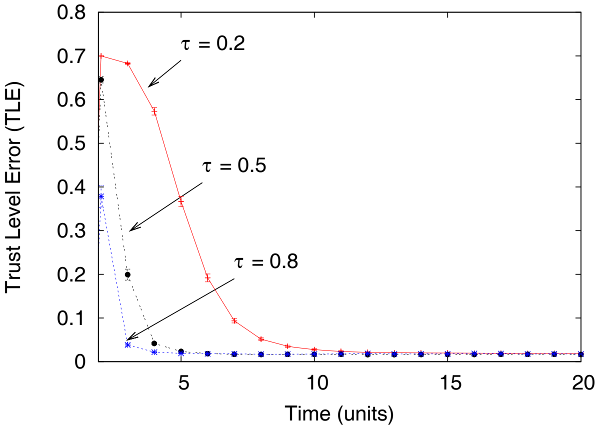


Fig. 8. The influence of τ on the time required to attain the stationary period, for $\alpha = 0.5$.

requires a minimum number of actions from each neighbor to consider its own experiences. If we increase the number of actions a node must notice before judging the nature of a neighbor, it will increase the precision of the judgment, but it will also increase the convergence time.

B. Performance on Multihop Mobile Ad Hoc Networks

Our main goal with this experiment is to evaluate the trust system performance in mobile multihop networks. We are also interested in analyzing the impact of the relationship maturity and the influence of the variation of parameters α and τ. All figures present the trust level error (TLE) as a function of time, as in the previous section. Although we present a simple scenario with a specific mobility pattern, it represents a non-favorable scenario for our model.

The simulation scenario consists of 21 nodes with 250 m transmission range, which are placed in a 1000 m × 400 m area, as shown in Fig. 9. The distance between nodes is 150 m. We defined the first trust assignment equals to 0.9 for every node in the simulation. We also assume $\alpha = \beta = \tau = 0.5$. These are the standard values for the simulations. For each specific configuration, the parameters that differ from its standard values are outlined. At last, for each configuration, all nodes have the same nature equals to 0.2. The mobility model chosen for these experiments represents the worst-case scenario for our trust model. In this case, a node moves to a region with completely new neighbors. As a result, it can not take advantage of maturity and must start new relationships "from scratch".

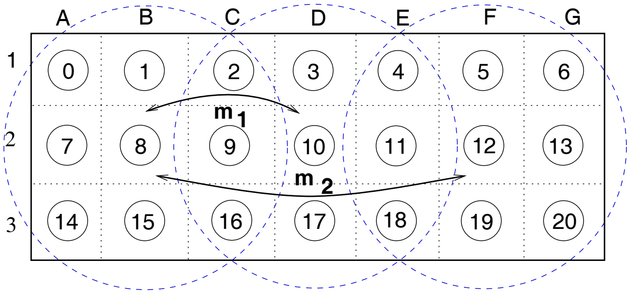


Fig. 9. The multihop experiment scenario.

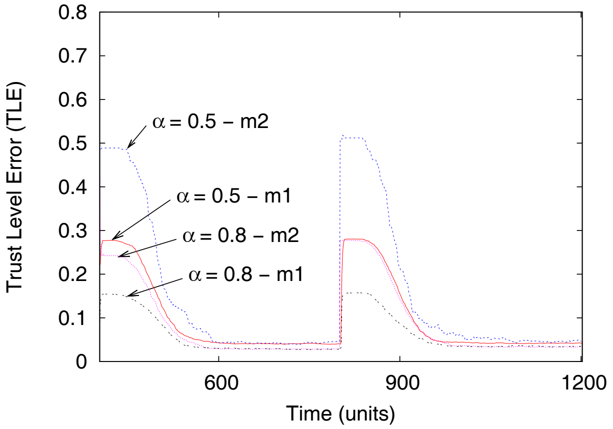


Fig. 10. Trust Level Error in the presence of mobility with different velocities and different α, for $\tau = 0.5$.

In the first configuration, node mobility is represented by m_1, where node 8, initially in zone B2, moves away to a specific place, zone D2, and then, after a pause, comes back to its origin. In the second configuration, mobility is represented by m_2, where node 8 goes to zone F2. Figure 10 presents the average Trust Level Error for all neighbors of node 8. The main difference between the two configurations is the number of new neighbors. In the shorter movement, m_1, node 8 keeps 3 old neighbors while in m_2 all neighbors are new ones. We set the speed equals to 1 m/s and 2 m/s, respectively. Thus, node 8 takes the same amount of time to move to both destinations. We observe in Fig. 10 that the TLE begins in a certain level, tends to zero, but never reaches it.

The first thing we notice is that the transient period begins with a lower value than the first trust assignment value (0.9) because when node 8 arrives the other nodes have already converged to the trust level of their neighbors. Thus, it will receive “correct” recommendations from its neighbors. Therefore, when we use a higher α to increase the importance of the neighbor recommendation, TLE decreases in the transient period, as showed in Fig. 10. It is clear that knowing some neighbors at the destination place decrease the Trust Level

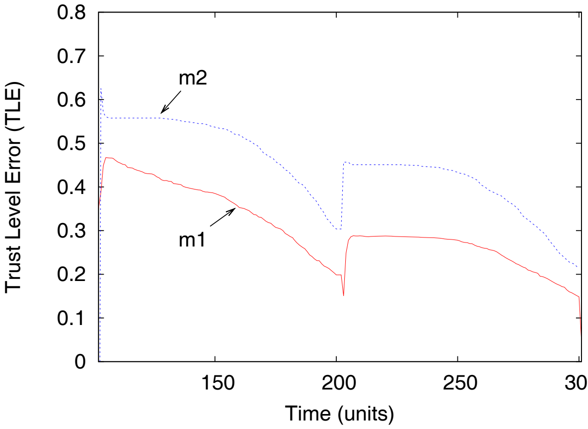


Fig. 11. Trust Level Error in the presence of a higher mobility, for $\alpha = \tau = 0.5$.

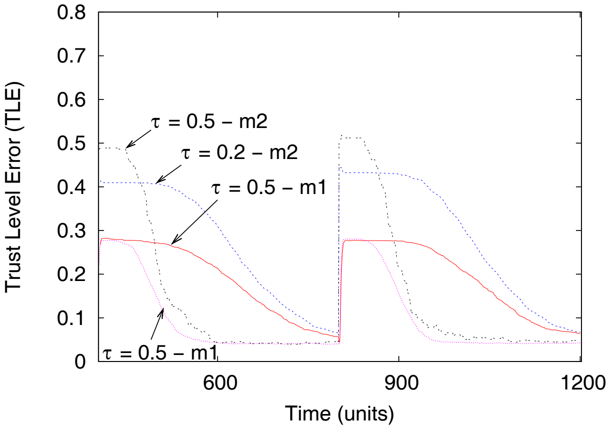


Fig. 12. The effect of a low perception on the TLE in the presence of mobility, for $\alpha = 0.5$.

Error (TLE) in the transient period because the node will consider their recommendations since its arrival. In addition, when nodes increase the α parameter, it is possible to decrease the TLE because it increases the importance of recommendations. Figure 11 shows the same scenario but node 8 moves three times faster. It is clear that in these conditions, node 8 does not stay long enough to evaluate the trust level of its neighbors and TLE remains high.

Figure 12 presents the results from the same movement pattern previously described (speed = 1 m/s and 2 m/s), and we vary the perception of node 8. It shows that if node 8, the one that moves, has a lower perception, it takes longer to correctly evaluate the nature of its neighbors, reducing the accuracy in the node classification.

C. Relationship maturity

Next, we analyze the impact of the relationship maturity in the evaluation of the trust level. For this purpose, we use a new configuration in the same scenario of Fig. 9. In the new configuration, nodes 1, 8, 15 are going to move to zone F2, the same zone as node 12. Instead of monitoring the trust level of all neighbors of node 8, we consider the trust level evaluation of node 8 about node 7 and node 20. Therefore, when node 8 arrives at the destination, zone F2, nodes 1 and 15 have just arrived there. It means that node 20 has 3 new neighbors and 3 old ones. The old ones have a more accurate trust level of node 20 than the new ones. Without the relationship maturity, when node 8 receives the recommendations of its neighbors, it will treat them all the same manner. Using the relationship maturity, node 8 gives more importance to the recommendations of the oldest neighbors of node 20. The result can be seen in Fig. 13.

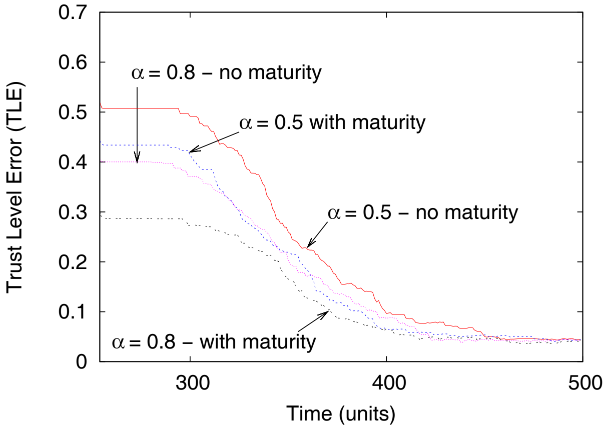


Fig. 13. The impact of the relationship maturity varying α, for $\tau = 0.5$.

It can be noticed in Fig. 13 that the transient is shorter with the relationship maturity. We can have almost the same effect of increasing α just by using the relationship maturity. The figure also shows that with a greater α the impact of the relationship maturity in the transient is more significant. It improves the efficiency of the system due to the fact that node 8 prioritizes the recommendations of its neighbors.

Figure 14 displays the impact of the relationship maturity when node 8 has a lower perception ($\tau = 0.2$). We can observe that it presents a lower peak when node 8 arrives at the destination, but the difference is not significant as in the other figures. In this case, node 8 has a longer transient caused by the lower perception. It happens because trust updates are triggered only by actions, thus a low of perception implies a longer transient. The result when we decrease the perception of node 8 and increase the value of α indicates that this is a good combination for a mobile network. Moreover, the effect of the relationship maturity is more evident. In this case when nodes have some difficulty to notice the actions of its neighbors, expressed by the low perception, the recommendations have greater importance. Therefore, giving more weight to the recommendations from nodes that have a longer relationship with the target node is more effective. Although node 8 is not able to reach the stationary period, it achieves a lower TLE than without using the relationship maturity.

D. Lying Attacks

The objective of this experiment is to evaluate the trust system performance under slander and collusion attacks in single-hop ad hoc networks. Mundinger and Le Boudec [16] perform an analysis of a reputation system for mobile ad hoc networks in the presence of liars. They conclude that there

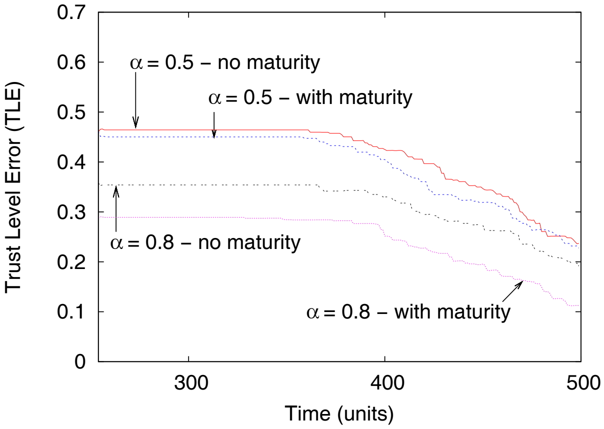


Fig. 14. The impact of the relationship maturity - $\tau = 0.2$.

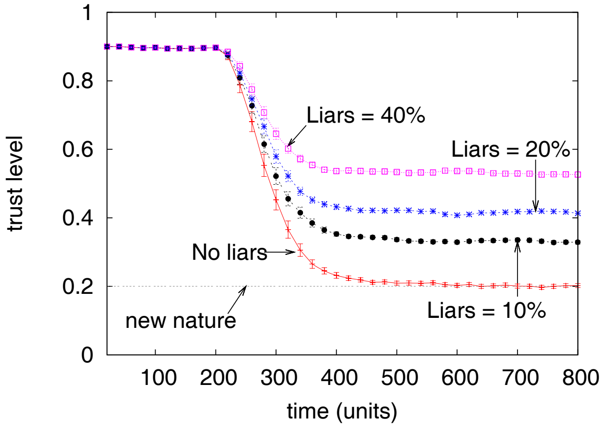


Fig. 15. Nodes trying to cover behavior changes, $\alpha = \tau = 0.5$.

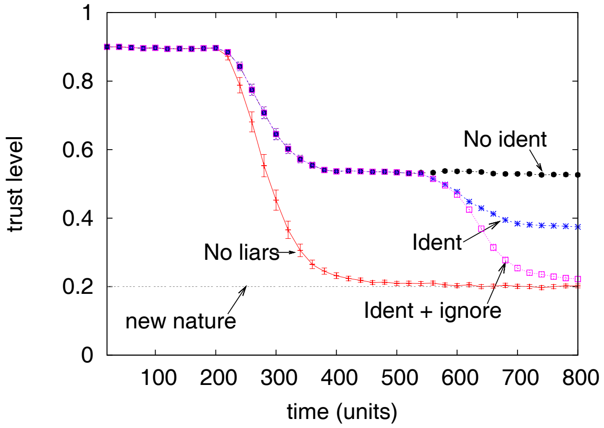


Fig. 16. Detecting liars - nodes trying to cover behavior changes - $\alpha = \tau = 0.5$.

is a threshold proportion of lying nodes above which the reputations system cannot work. Below this threshold liars do not cause a significant impact on the system. Therefore, we aim at finding that threshold below which our trust model fails to work properly. The scenario consists of 20 nodes with 250 m transmission range, which are randomly placed in a 150 m × 150 m area. All figures present the trust evaluation of node 2 about node 1. It means that node 2 is trying to assess the trust level of node 1. We defined the first trust assignment equal to 0.9 and $\alpha = \beta = \tau = 0.5$ for all nodes. These are the standard values for the simulations. For each specific configuration, the parameters that differ from its standard values are outlined. At last, in each configuration, all nodes have nature equals to 0.9, which means one out of ten actions taken is bad on average.

1) Changing behavior: In Section IV-A, we show that nodes are capable of evaluating their neighbor nature using our trust model. However, a node might change its behavior and consequently its nature during its lifetime. The behavior variation of a node occurs due to several reasons. For instance, a node may behave well at first, but after being compromised it starts to misbehave. Another possibility is a good node that experiences an energy consumption problem which causes an anomalous behavior. A third option consists of malicious nodes that continuously change their behavior between good and bad in order to cause damage to the trust system. Therefore, it is important for a trust model to provide nodes with the capability of identifying such behavior variations as quick as possible. Thus, in the first set of simulations we analyze the trust evaluation of a node that changes its behavior during the simulation. In this first scenario, node 1 changes its nature and malicious nodes might try to cover the behavior variations of each other in order to keep a good reputation even though they have a bad behavior.

Figures 15 and 16 show a scenario where node 1 changes its nature from 0.9 to 0.2 and malicious nodes lie about node 1 trying to convince the other nodes that node 1 still has a trust level equals to 0.9. The first thing we can notice is that node 2 perfectly succeeds in remarking a change in node 1 behavior when there is no liar. Figure 15 also reveals the effect of a collusion attack varying the percentage of malicious nodes participating in the attack. We observe that malicious nodes can deteriorate the trust evaluation. However, it shows that node 2 manages to identify node 1 as a bad node, namely trust level less than 0.5, if the percentage of malicious nodes is smaller than 35%.

Next, we propose a scenario similar to the last one, but we fixed the percentage of malicious nodes in 40%. In this scenario, we consider that nodes are capable of identifying when a neighbor lies about its recommendations after a certain amount of time. After detecting a neighbor as a liar, the node can degrade the trust level of its neighbor. The results show that detecting liars can significantly improve the trust evaluation performance, as we see in curve “ident” in Fig. 16, in the presence of liars. An even better solution is to detect and then to completely ignore the recommendations of malicious nodes, as shown by curve “dent + ignore” in Fig. 16. Ignoring liars is a simple task. Node can simply ignore all recommendations of neighbors with a trust level under a certain threshold. We observe that ignoring liars can neutralize a lying collusion attack. The only damage is during the process of liar detection.

2) Slander attack: The slander attack consists of sending false recommendations to injure the reputation of a node. Malicious nodes can collude to improve the effect of the

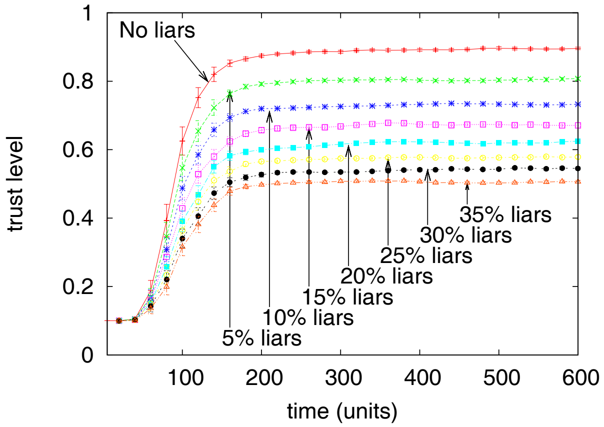


Fig. 17. Slander attack - varying the proportion of liars, for $\alpha = \tau = 0.5$.

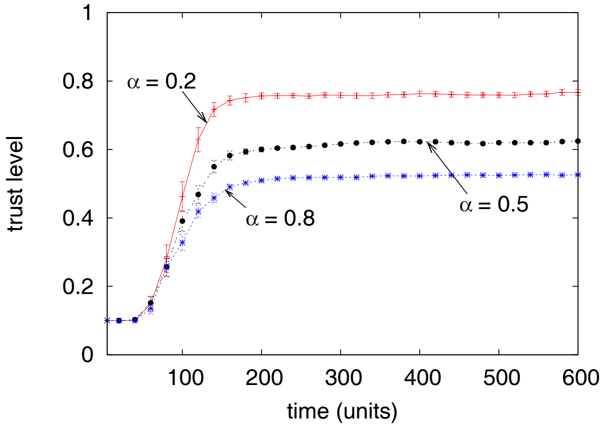


Fig. 18. Slander attack - varying α parameter, for $\tau = 0.5$.

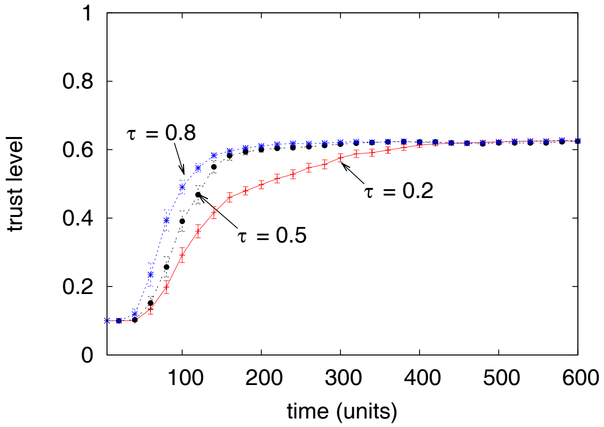


Fig. 19. Slander attack - varying perception τ, for $\alpha = 0.5$.

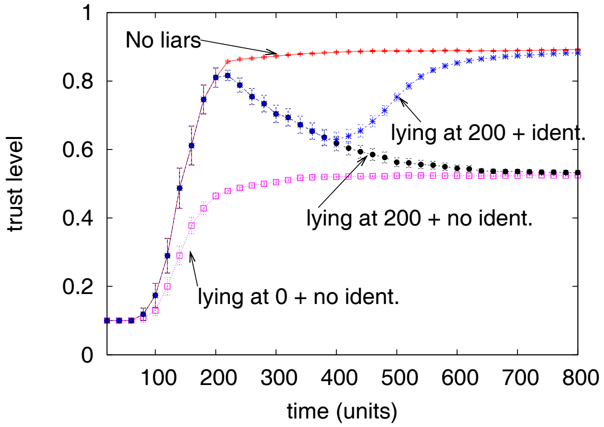


Fig. 20. Convenient scenario for slander attacks, for $\alpha = 0.8$ and $\tau = 0.2$.

attack. In this experiment, node 2 tries to evaluate the trust level of node 1, whose nature is equal to 0.9. Malicious nodes send false recommendations saying that node 1 has a trust level equals to 0.2. Node 2 adopts a pessimistic strategy, which means it assigns a low trust level (0.1) for new neighbors. In Fig. 17, we vary the percentage of liars to show that node 2 can succeed in identifying node 1 as a good node, assuming that a good node has a Trust Level > 0.5, for a percentage of liars smaller than 35%.

Figures 18 and 19 present the result for the variation of two important parameters in our model. First, in Fig. 18, we vary α. The parameter α is the one that controls the weight of recommendations and own experiences in the calculation of the trust level in Equation 1. With a higher α the recommendations of other nodes has a higher weight on the trust level evaluation. It is clear that the more a node considers the recommendations of other nodes, the more it is vulnerable to lying attacks. Therefore, a node might have a low value for α (e.g.,$\alpha < 0.5$) in order to be more resistant to liars.

Figure 19 displays the impact of the perception on the slander attack. The first remark is that the perception does not impact the trust level evaluation under a slander attack. The perception has strong influence only in the duration of the transient period and has no influence on the level achieved after convergence, in the stationary period. In the transient period, nodes are trying to approximate to the expected value, while in the stationary period, the trust level is stable, very close to the correct value.

We changed the perception of node 2 to 0.2 and the parameter α to 0.8 to evaluate a more convenient scenario for a slander attack. Figure 20 presents the results when malicious nodes begin to lie after 200 time units so they already have a good reputation. We observe that if node 2 detects the misbehavior of the malicious nodes and ignore their recommendations (curve “lying at 200 + ident.”) there is no damage to the trust evaluation process, except for the period during which node 2 has not yet notice the liars. This period depends solely on the capacity of the node in detecting a lie.

In Fig. 21 we vary the duration of the detection of liars. The results show that identifying liars is an important task to avoid damage to the trust system. A fast liar detection mechanism can offer a robust trust system against slander attacks. It can be noticed that the recovery delay, namely, the time a node take to achieve the correct trust value after identifying all liars in the neighborhood, remains the same regardless of the detection delay.

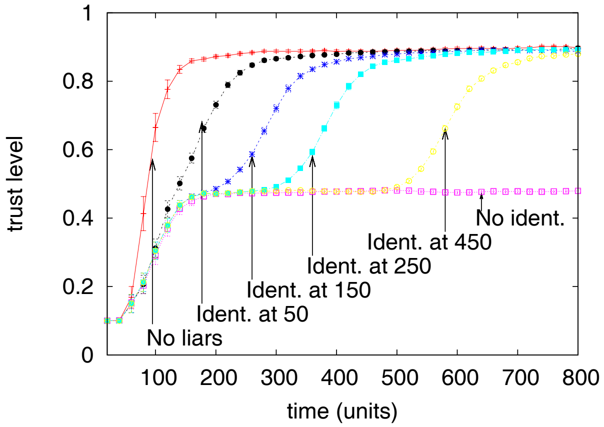


Fig. 21. Detection delay of slanderer nodes, for $\alpha = \tau = 0.5$.

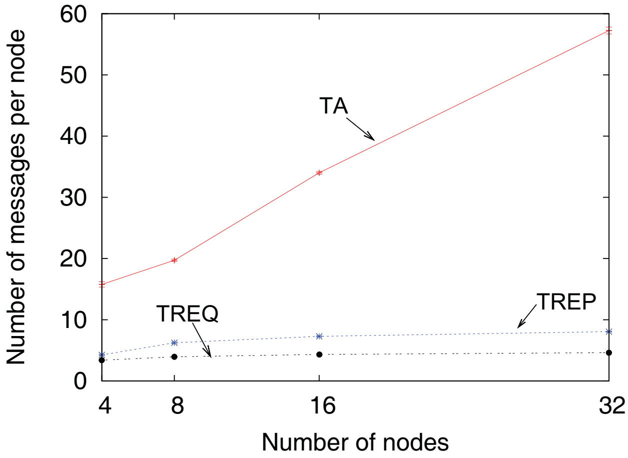


Fig. 22. The influence of the number of neighbors on the number of messages.

E. Recommendation Exchange Protocol

The Recommendation Exchange Protocol (REP) (Section II-D) is an important feature in our trust model. In order to evaluate the performance of the REP protocol we use a single-hop network because it is a "local" protocol, that is, the interactions are limited to neighbors, and thus mobility does not have a real impact on the performance of REP. The scenario consists of n nodes randomly placed in a 150 $m \times 150$ m area, which means that each node has $n-1$ neighbors. The first trust value is 0.9 and node 2 has a nature equals 2. All nodes arrive at the same time and try to evaluate the trust level of their neighbors. We believe that this is a representative scenario, since in this scenario all types of messages are used. The first set of simulations aims at evaluating the impact of the number of neighbors on the performance of the REP protocol, more specifically on the number of sent messages. Therefore, we vary the number of nodes n from 4 to 32.

Figure 22 presents the result of the number of messages sent per node in the scenario described above. The TREQ message is sent just once when two nodes first meet. Thus, each node should send at most $n-1$ TREQs. However, we implemented a timer before sending a TREQ message that is used to collect the maximum number of TREQs in one single message. The timer also permits the TREQ suppression when the node receives a TREP during the timer period. This approach allows reducing significantly the number of TREQs when the neighborhood changes in short-term period, as in the case of a network in which nodes start simultaneously. Results show the effectiveness of our approach. In this scenario we reach more than 85% of reduction (the case with 32 nodes). The TREP message is sent just once per TREP request, which means that the expected number of TREPs $(n-1)(n-2)$ messages. First, we implement the TREP as a broadcast message which is only considered by nodes that have sent a TREQ recently. Thus, the number of expected messages drop to $(n-1)$. Finally, we implement the same timer approach for the TREP. Figure 22 shows that for the TREP, these two approaches are can reduce the number of TREPs by more than 99%.

We notice from the previous result (Fig. 22) that the TA message is more sensitive to the increase of the number of neighbors. However, we observe that there is no exponential increase (mostly $\frac{3n}{2} \to O(n)$) and if we consider that these messages are sent at each transient period, we have less than one TA message per unit of time during the transient period.

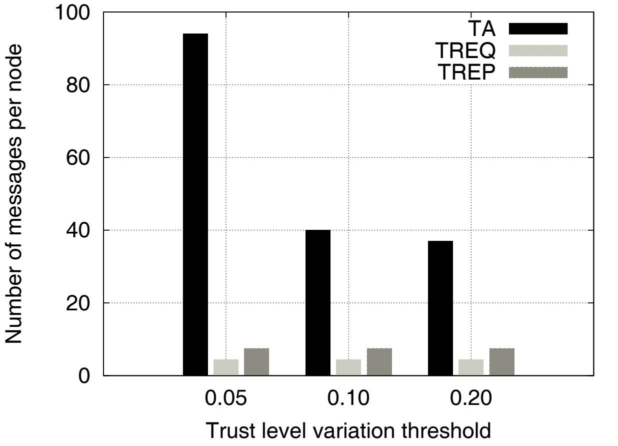
We can try to optimize the number of TA messages sent during the transient period. TA messages are sent by nodes whenever the trust level of a given neighbor has varied more than a certain threshold (π). This approach avoids sending trust level information after every change in the trust level of a neighbor, instead, we advertise the trust level information just after a significant change compared to the last advertised value.

In Fig. 23 we use the same scenario but with 20 nodes. Figure 23(a) shows the impact of the value of π on the number of messages and Fig. 23(b) shows the impact of π on the trust evaluation process. The first important observation is that, as expected, TREQ and TREP messages are not influenced by the value of π. Second, the lower is the value of π, the larger is the number of TA messages and the faster is the transient period. An interesting result is that setting $\pi = 0.2$ does not reduce significantly the number of messages, comparing to $\pi = 0.1$, because the trust level variation is smoother which leads to a longer transient period. Moreover, for $\pi = 0.2$ the trust evaluation process does not converge to the correct value (0.2). Therefore, there is an optimum value for π that reduces the number of TA messages and provides a fast and correct convergence.

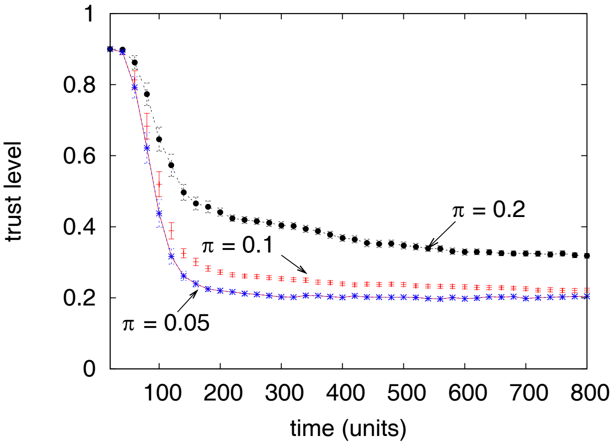
F. Discussion

In our simulations, we exploited the advantages of our model that limits the interactions to the neighbors. Therefore, we are able to evaluate the performance of our model in a single hop network, instead of using a more complex scenario, without loss of generality.

The simple mobility model in this paper is used to demonstrate the basic characteristics of our model. For instance, we show that in a scenario where the topology changes faster than the convergence of the proposed model, the node will not be



(a) The number of messages per node.



(b) Trust level evaluation.

Fig. 23. The impact of the trust level variation threshold (π) on the number of messages per node.

capable of evaluating the trust level of its neighbors. We show also that we can increase α to mitigate this problem. These two results are independent of the mobility model and can be applied in different scenarios.

The simulation results show that our implementation using smart timers to suppress redundant messages scales well to larger networks, reducing the trust management overhead by 85-99%. We also show that using a threshold heuristic is useful to reduce the overhead since we only send updates if a significant change in the trust level occurs. Our results show an overhead reduction of almost 60% (Fig. 23(a)) with roughly no impact at the convergence rate (Fig. 23(b)).

We also shows that increasing the value of α is a good strategy to improve the trust model efficiency, since we give more weight to the neighbor recommendations. Nevertheless, α plays a key role in reducing the influence of liars. Nodes with a large α are more vulnerable to false recommendations. Therefore, we observe the existence of an important trade-off between mobility and vulnerability to slander attacks. A possible solution to overcome the trade-off problem consists of implementing a liar detection mechanism. A feasible approach for liar detection is to compare the recommendations of all neighbors. Considering that the percentage of malicious nodes is smaller than 50%, a node might assume as a liar every node that keeps sending conflicting recommendations.

Another important aspect is that our model provides nodes with a mechanism to assess the trust level of its neighbors. Therefore we consider that each service/application must define and implement how the information will be used and disseminated, if necessary, instead of adding complexity to REP protocol.

V. Related work

Although researchers usually assume that nodes collaborate in ad hoc networks, it is not so obvious that this collaboration exists in practical networks. Each node must forward packets for other nodes and spend its energy without receiving any direct gain for this act. There is no real incentive for nodes to participate in the routing and forwarding process. Yu and Liu [17] state that before ad hoc networks can be successfully deployed in autonomous ways, the issues of cooperation stimulation and security must be resolved first. Several works propose mechanisms to stimulate the cooperation among nodes. Their goal is to avoid selfish and malicious behavior to guarantee the right implementation of routing and forwarding tasks by all nodes of the network [17]–[24]. Nevertheless, all these works are restricted to stimulate the collaboration of nodes to relay traffic for other nodes. We are concerned with all kinds of distributed mechanisms and applications, such as authentication, key distribution, access control, and management.

In general, the trust models in ad hoc networks try to protect or enforce the two basic functions of the network layer: routing and packet forwarding [25]. Sun *et al.* [26] investigate the benefits of using trust models in distributed networks, the vulnerabilities in trust establishment methods, and the defense mechanisms.

Several works propose monitoring schemes to generate trust values describing the trustworthiness, reliability, or competence of individual nodes. Theodorakopoulos and Baras [27] analyze the issue of evaluating the trust level as a generalization of the shortest-path problem in an oriented graph, where the edges correspond to the opinion that a node has about other node. They consider that nodes use just their own information to establish their opinions. The opinion of each node includes the trust level and its precision. The main goal is to enable nodes to indirectly build trust relationships using exclusively monitored information.

Sun *et al.* [28] have developed a framework capable of measuring the trust level and propagating it through the network in order to make routing more secure and to assist intrusion detection systems. The framework includes a defense mechanism against malicious nodes. The authors use a probabilistic model based on the uncertainty of a neighbor to execute one specific action and consider only the monitoring information.

He *et al.* [29] propose an architecture for stimulating the collaboration based on the reputation of nodes. The system is based only on the monitored information to evaluate the reputation of nodes. The goal is to detect and to punish nodes that do not participate in the routing process.

The main difference of these works and our trust model is that they use only the node own experience, namely, the monitored information on the trust evaluation process. Our trust model considers the monitored information and the recommendations of neighbors to achieve a faster convergence time and an accurate trust level for each neighbor.

In probabilistic-based models, a common approach consists of using Bayesian networks, which is a probabilistic tool that provides a flexible means of dealing with probabilistic problems involving causality [30]. Buchegger and Le Boudec [31] investigate the trade-off between robustness and efficiency of reputation systems in mobile ad hoc networks. A mechanism based on Bayesian statistics is used to filter slanderer nodes. The proposed system considers the monitored information and the recommendation of other nodes to compute the reputation of a specific node. They show that taking into account the recommendations of other nodes can speed up the process of discovery of malicious nodes. Chinni *et al.* [32] offer a distributed trust model for certificate revocation in ad hoc networks. This model allows trust to be built based on the interactions between nodes, using monitored information. Furthermore, trust in a node is defined not only in terms of its potential for maliciousness, but also in terms of the quality of the service it provides. The trust level of nodes where there is little or no history of interactions is determined by recommendations from other nodes. If the nodes in the network are selfish, trust is obtained by an exchange of portfolios. Bayesian networks form the underlying basis for this model.

Another approach consists of using linear functions to infer trust. Pirzada and McDonald [33] propose another trust model for ad hoc networks to compute the trustworthiness of different routes. Nodes can use this information as an additional metric on routing algorithms. Although the authors present an interesting approach, the model presents disadvantages. For instance, it is currently restricted to Dynamic Source Routing (DSR) protocol. It also relies on using the promiscuous mode, ignoring the energy constraints of mobile nodes. Finally, it requires each node to store information for all other nodes in the network, which is not scalable.

Virendra *et al.* [14] present a trust-based architecture that allows nodes to make decisions on establishing keys with other nodes and forming groups of trust. Their scheme considers trust self-evaluation and recommendation of other nodes to compute trust. Their trust self-evaluation is based on monitoring nodes and a challenge-response system. Some authors present trust models specifically designed to work with a particular routing protocol. Komathy and Narayanasamy [34] add a trust-based evolutionary game model to the AODV routing protocol in order to cope with selfish nodes.

Kostoulas *et al.* [35] propose a decentralized trust model to improve reliable information dissemination in large-scale disasters. The proposed model includes a distributed recommendation scheme, incorporated into an existing membership maintenance service for ad hoc networks. In addition, trust-based information is propagated through a nature-inspired activation spreading mechanism.

The main differences of our work from all the related work are that nodes interact only with neighbors. Neighborhood interactions imply low resource consumption and minimize the effect of false recommendations. Another important issue is the introduction of the concept of relationship maturity in our model which improves the efficiency of the trust model in MANETS. At last, only a few works analyze the robustness of the trust model against liars and its scalability, as we do in Sections IV-D and IV-E.

VI. Conclusion

This paper addresses the problem of trust evaluation and management in ad hoc networks. We propose a flexible trust model based on the concept of human trust, which provides nodes with a mechanism to evaluate the trust level of its neighbors. The basic idea consists of using previous experiences and recommendations of other neighbors to appraise the trust level of other nodes. We introduce the concept of relationship maturity, which allows nodes to attribute more relevance to the recommendations issued by nodes that know the evaluated neighbor for a long time. We also propose the Recommendation Exchange Protocol (REP) which enables nodes to send and receive recommendations.

In our model, the interactions among nodes are confined to neighbors. Such approach implies lower resource consumption and a lower vulnerability to false recommendations attack. Another important quality is the flexibility due to the possibility of operating in three different modes, depending on the node resource restrictions. Thus, our model is suitable for heterogeneous network, where nodes present distinct constraints. Besides, the presence of nodes that do not implement at all our trust system do not disturb the other nodes that are using the system, since nodes are capable of evaluating the trust level even in the presence of a few cooperating neighbors.

We perform a number of simulations to evaluate the performance of the Recommendation Exchange Protocol and show its scalability. We show that our implementation of the REP protocol can significantly reduce the number messages. We also present other results that indicate that our model detects behavior changes of nodes and is robust to slander and colluding attacks. The results reveal that the proposed model tolerates up to 35% of liars. We also evaluate our model in mobile multihop ad hoc networks. We show the effectiveness of the relationship maturity parameter, which reduces the trust level error by almost 50%, in certain scenarios.

Future work includes defining and implementing a monitoring scheme for a specific application and applying our model to improve the service/application performance, as for instance, an authentication protocol.

References

[1] C. E. Perkins, *Ad Hoc Networking*, 1st edition. Addison-Wesley Professional, 2001.

[2] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web," *Web Semantics: Science, Services Agents World Wide Web*, vol. 5, no. 2, pp. 58-71, June 2007.

[3] A. Josang, "Trust and reputation systems," in *Foundations Security Analysis Design IV, FOSAD 2006/2007 - Tutorial Lectures*, (Bertinoro, Italy), Springer LNCS 4677, Sep. 2007.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *IEEE Computer*, vol. 36, no. 1, pp. 41-52, Jan. 2003.

[5] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *ACM SIGCOMM'03*, Aug. 2003.

[6] D. F. Macedo, A. L. Santos, J. M. S. Nogueira, and G. Pujolle, “A distributed information repository for autonomic context-aware manets," *IEEE Trans. Netw. Service Management*, vol. 6, no. 1, pp. 45-55, Mar. 2009.

[7] N. C. Fernandes, M. D. D. Moreira, and O. C. M. B. Duarte, “An efficient filter-based addressing protocol for autoconfiguration of mobile ad hoc networks," in *IEEE INFOCOM'09*, Apr. 2009.

[8] B. Ishibashi and R. Boutaba, “Topology and mobility considerations in mobile ad hoc networks," *Ad Hoc Netw. J.*, vol. 3, no. 6, pp. 762-776, Nov. 2005.

[9] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, “HIT: a human-inspired trust model," in *8th IFIP IEEE International Conf. Mobile Wireless Commun. Netw.*, Santiago, Chile, Aug. 2006.

[10] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, “Analyzing a human-based trust model for mobile ad hoc networks," in *IEEE Symp. Comput. Commun.*, Marrakech, Morocco, July 2008.

[11] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, “A trust model robust to slander attacks in ad hoc networks," in *IEEE International Conf. Comput. Commun. Netw. ANC workshop*, Virgin Islands, USA, Aug. 2008.

[12] A. Malatras, G. Pavlou, and S. Sivavakeesar, “A programmable framework for the deployment of services and protocols in mobile ad hoc networks," *IEEE Trans. Netw. Service Management*, vol. 4, no. 3, pp. 12-24, Dec. 2007.

[13] P. B. Velloso, M. G. Rubinstein, and O. C. M. B. Duarte, “Evaluating voice traffic requirements on IEEE 802.11 ad hoc networks," *Annals Telecommun.*, vol. 63, no. 5-6, pp. 321-329, June 2008.

[14] M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya, “Quantifying trust in mobile ad-hoc networks," in *Proc. IEEE International Conf. Integration Knowledge Intensive Multi-Agent Syst.*, Waltham, USA, Apr. 2005.

[15] G. Theodorakopoulos and J. S. Baras, “Trust evaluation in ad-hoc networks," in *ACM Workshop Wireless Security*, Oct. 2004.

[16] J. Mundinger and J.-Y. Le Boudec, “Analysis of a reputation system for mobile ad-hoc networks with liars," *Performance Evaluation*, vol. 65, no. 3-4, no. 3-4, pp. 212-226, 2008.

[17] W. Yu and K. J. R. Liu, “Attack-resistant cooperation stimulation in autonomous ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 12, pp. 2260-2271, Dec. 2005.

[18] S. Zhong, J. Chen, and Y. R. Yang, “Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *IEEE INFOCOM'03*, San Francisco, USA, Apr. 2003.

[19] L. Buttyan and J. P. Hubaux, “Enforcing service availability in mobile ad-hoc wans," in *IEEE/ACM MobiHoc'00*, Aug. 2000.

[20] L. Buttyan and J. P. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM/Kluwer Mobile Netw. Appl. (MONET)*, vol. 8, no. 5, pp. 579-592, Oct. 2003.

[21] J. Crowcroft, R. Gibbens, F. Kelly, and S. Östring, “Modelling incentives for collaboration in mobile ad hoc networks," *Performance Evaluation*, vol. 57, no. 4, pp. 427-439, Aug. 2004.

[22] J. Pan, L. Cai, X. S. Shen, and J. W. Mark, “Identity-based secure collaboration in wireless ad hoc networks," *Comput. Netw.*, vol. 51, no. 3, pp. 853-865, Feb. 2007.

[23] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks," in *ACM MobiCom'00*, Aug. 2000.

[24] J. N. Al-Karaki and A. E. Kamal, “Stimulating node cooperation in mobile ad hoc networks," *Wireless Personal Commun.*, vol. 44, no. 2, pp. 219-239, Jan. 2008.

[25] A. Adnane, R. T. de Sousa Jr., C. Bidan, and L. Mé, “Autonomic trust reasoning enables misbehavior detection in OLSR," in *ACM Symp. Appl. Comput. (SAC'08)*, Ceará, Brazil, Mar. 2008.

[26] Y. L. Sun, Z. Han, and K. J. R. Liu, “Defense of trust management vulnerabilities in distributed networks," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 112-119, Feb. 2008.

[27] G. Theodorakopoulos and J. S. Baras, “On trust models and trust evaluation metrics for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 318-328, Feb. 2006.

[28] Y. Sun, Z. Han, W. Yu, and K. J. R. Liu, “A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks," in *IEEE INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[29] Q. He, D. Wu, and P. Khosla, “A secure incentive architecture for ad hoc networks," *Wireless Commun. Mobile Comput.*, vol. 6, no. 3, pp. 333-346, May 2006.

[30] J. Li, R. Li, and J. Kato, “Future trust management framework for mobile ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 108-114, Apr. 2008.

[31] S. Buchegger and J.-Y. Le Boudec, “The effect of rumor spreading in reputation systems for mobile ad-hoc networks," in *Modeling Optimization Mobile, Ad Hoc Wireless Netw.*, Sophia-Antipolis, France, Mar. 2003.

[32] S. Chinni, J. Thomas, G. Ghinea, and Z. Shen, “Trust model for certificate revocation in ad hoc networks," *Ad Hoc Netw.*, vol. 6, no. 3, pp. 441-457, May 2008.

[33] A. A. Pirzada and C. McDonald, “Trust establishment in pure ad-hoc networks," *Wireless Personal Commun.: An International J.*, vol. 37, no. 1-2, pp. 139-168, Apr. 2006.

[34] K. Komathy and P. Narayanasamy, “Trust-based evolutionary game model assisting AODV routing against selfishness," *J. Netw. Comput. Appl.* (available online), Feb. 2008.

[35] D. Kostoulas, R. Aldunate, F. P. Mora, and S. Lakhera, “A nature-inspired decentralized trust model to reduce information unreliability in complex disaster relief operations," *Adv. Eng. Informat.*, vol. 22, no. 1, pp. 45-58, Jan. 2008.

Pedro B. Velloso received the B.Sc. and M.Sc. degrees in Electrical Engineering from the Universidade Federal do Rio de Janeiro, Brazil, in 2001 and 2003, respectively. He received the PhD degree from the Université Pierre et Marie Curie (Paris 6) in 2008. He spent one year as a post-doc researcher at Laboratoire d'Informatique de Paris 6 in 2008/2009. Currently, he is a research engineer at Bell Labs France. His interests are in autonomic networks, distributed applications, wireless communications, and security.

Rafael P. Laufer received the B.Sc. and the M.Sc. degrees in Electrical Engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, in 2003 and 2005, respectively. He is now a Ph.D. student in Computer Science at the University of California, Los Angeles (UCLA). He received the 2008 Young Scholar Award from the Marconi Society. His current research interests are distributed systems, wireless networking, and security.

Daniel de O. Cunha received B.Sc. And M.Sc. degrees in electrical engineering from UFRJ in 2002 and 2004, respectively. He received the Ph.D. degree from UFRJ/UPMC. His major research interests are in the area of wireless networks, especially ad hoc and sensor networks, energy conservation, and cooperative communications.

Otto Carlos M. B. Duarte received electronic engineer and M.Sc. degrees from UFRJ in 1976 and 1981, respectively, and a Dr.Ing. degree from ENST/Paris, France, in 1985. Since 1978 he has been a professor with UFRJ. Between January 1992 and June 1993 he worked as a researcher at the MASI laboratory at the University of Paris 6. In 1995 he spent three months with the International Computer Science Institute (ICSI), University of California, Berkeley. In 1999, 2001, and 2006 he was an invited professor at Université Paris 6. His major research interests are in multicast, QoS guarantees, security, and mobile communications.

Guy Pujolle is currently a Professor at the Pierre et Marie Curie University (Paris 6), a member of the Institut Universitaire de France, and a member of the Scientific Advisory Board of Orange/France Telecom. Dr. Pujolle is the French representative at the Technical Committee on Networking at IFIP. He is an editor for International Journal of Network Management, WINET, Telecommunication Systems and Editor in Chief of the indexed Journal "Annals of Telecommunications". He was an editor for Computer Networks, Operations Research, Editor-In-Chief of *Networking and Information Systems Journal*, *Ad Hoc Journal* and several other journals. Guy Pujolle is a pioneer in high-speed networking having led the development of the first Gbps network to be tested in 1980. Guy Pujolle is co-founder of QoSMOS (www.qosmos.fr), Ucopia Communications (www.ucopia.com), Ginkgo-Networks (www.ginkgo-networks.com), EtherTrust (www.ethertrust.com), and Virtuor (www.VirtuOR.com).