HIT: A Human-Inspired Trust Model *

Pedro B. Velloso[1], Rafael P. Laufer[2], Otto C. M. B. Duarte[3], and Guy Pujolle[1]

[1] Laboratoire d'Informatique de Paris 6 (LIP6) - Paris VI
Paris, France
[2] Computer Science Department - UCLA
California, USA
[3] Grupo de Teleinformática e Automação (GTA) - UFRJ
Rio de Janeiro, RJ, Brazil

Abstract. This paper presents a new approach to assign trust levels in ad hoc networks. Our system is inspired by the human concept of trust. The trust level considers the recommendation of trustworthy neighbors and their own experience. For the recommendation computation, we take into account not only the trust level, but also its accuracy and the relationship maturity. We also propose the Recommendation Exchange Protocol (REP), which minimizes the number of exchanged messages. The results show the efficacy of the system and the influence of main parameters.

1 Introduction

Ad hoc networks rely on collaborative behavior of nodes to work properly. Therefore, nodes must trust each other at some level to allow distributed applications, including routing and admission control. A naive trust model might lead to low efficiency, high energy consumption, and network attacks. The behavior of the nodes is dynamic and depends on their goals and constraints, which might lead to distinct behaviors. Nodes must decide what is best for themselves but in a context of minimum collaboration, like in a society. We believe that the first step towards a self-learning and collaborative system is defining whether neighbors are reliable or not, because trust allows the information exchange and stimulates cooperation among nodes. Moreover, trust can also be used to minimize the effect of malicious nodes

Our work aims at building a trust relationship inspired on the human concept of trust among nodes of an ad hoc network. Each node must assign trust levels to other nodes based on the recommendation of trustworthy neighbors and its own experiences. There are already some effort in bringing trust to ad hoc networks [1–8], but most of them are concerned solely about routing aspects.

Liu *et al.* [1] propose a trust model to ad hoc networks based on the distribution of threat reports to interested nodes. The goal is to make security-aware routing decisions, where nodes use the trust level as an additional metric for

* This work has been supported by CNPq.

Please use the following format when citing this chapter:

Velloso, P.B., Laufer, R.P., Duarte, O.C.M.B., Pujolle, G., 2006, in IFIP International Federation for Information Processing, Volume 211, ed. Pujolle, G., Mobile and Wireless Communication Networks, (Boston: Springer), pp. 35–46.

routing packets. The authors present different approaches for the trust level calculation. Nevertheless, they assume that nodes cooperate with each other which is not always the case. They also assume that all nodes are capable of detecting malicious behavior by means of Intrusion Detection Systems (IDS). This assumption leads to high energy consumption, which is clearly not an appropriate option for ad hoc networks. All the trust level dynamics is based on the reports provided by the IDS.

Yan *et al.* [2, 3] propose a security solution for ad hoc networks based on a trust model. They suggest using a linear function to calculate the trust according to a particular action. The function considers different factors that can affect the trust level, including intrusion black lists, previous experience statistics, and recommendations. Nonetheless, the influence of such factors on the trust evaluation is not defined. Although mentioning general trust concepts, the work focus on specific routing issues.

Pirzada and McDonald [4] propose another trust model for ad hoc networks to compute the reliability of different routes. Nodes can use this information as an additional metric on routing algorithms. The authors propose an extension to DSR protocol which applies their trust model in order to find trustworthy routes. Although the authors present an interesting approach, the model presents several disadvantages. For instance, it is restricted to DSR so far, it relies on using promiscuous mode ignoring the energy constrains of mobile nodes, and it stores a significant amount of information, since it keeps information for all nodes in the network.

Virendra *et al.* [5] present an architecture based on trust 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. Although we have a similar approach, our trust model differs in the following way. Their trust self-evaluation is based on monitoring nodes and a challenge-response system. We propose a self-learning and context-based approach in which nodes evaluate their neighbors based on their own goals, current state, present location, and network conditions.

We focus on providing nodes with a trust level regarding their direct neighbors. The goal is to make nodes capable of gathering information to reason, learn, and make their own decisions. Different from most related works, our work improves scalability by restricting nodes to keep and exchange trust information solely with direct neighbors, that is, neighbors within the radio range.

The contributions of this paper are twofold. First, we propose a trust model that takes into account time-space parameters of each node, such as its current state and its location, the network conditions, and mobility parameters to compute the trust level. Accordingly, nodes rely on a self-learning mechanism to set some parameters in our model. Finally, we propose the Recommendation Exchange Protocol (REP), which allows nodes to build and update their trust table.

The paper is organized as follows. Our self-learning approach to build trust is presented in Section 2. Section 3 shows our simulation results. In Section 4 we present our conclusions.

2 Trust model

The basic idea consists of building a trust information system that allows nodes to learn based on the information exchanged with other nodes. The main goal is to make nodes self-configuring, self-adaptive, self-optimized. As a result, nodes are capable of making their own decisions. Moreover, nodes might use the trust information to detect and isolate malicious nodes.

The proposed model can be divided in two distinct layers. The Learning layer is responsible for gathering and converting information into knowledge. The Trust layer defines how to assess the trust level using knowledge information provided by the Learning layer. Both layers can interact with all other layers. In this paper, we only describe the Trust layer.

The level of trust is based on both the previous experiences of nodes and the recommendation of others. Previous experiences allow nodes to judge the actions performed by other nodes. These actions can lead to three types of verdict. An action affects negatively, positively, or does not affect other nodes at all. The first two types of effect will generate a reaction that begins with a trust level update, but can also change the node behavior. The Learning layer is the responsible for the evaluation of other nodes actions and for choosing the appropriate reaction.

The recommendation of other nodes can be taken into account while calculating the trust level. For that, 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 than recommendations sent by new neighbors. Nodes willing to consider the recommendation of other nodes can use the proposed Recommendation Exchange Protocol (REP) to keep updated the trust level of each neighbor.

Each node is responsible for computing and storing the trust level of each neighbor. For that purpose, nodes keep a so-called trust table which contains the trust level of all direct neighbors. Additionally, a node might also store the trust table of its neighbors whenever it is possible.

In our model, nodes can also keep an additional table that is not mandatory. The Auxiliary Trust Table (ATT) contains the confidence of the trust level and the so-called relationship maturity for each neighbor. The confidence of the trust level represents the accuracy of this measure whereas the relationship maturity represents the time that the node has met this specific neighbor. The goal of ATT is to supply nodes with additional information that can improve the trust-level evaluation.

Nodes with power or storage constrains can choose not to implement the entire trust system. We define three operation modes. Nodes with low power/storage capacity can operate in the simple mode, where they only use the main trust table. Nodes with a medium capacity operate in the intermediate mode, where nodes exchange trust information using the REP protocol and store the trust table of neighbor nodes. In the advanced mode, nodes implement the same features used in intermediate mode and also use the ATT to keep track of additional parameters.

We divide the trust scheme in two distinct phases. An initial phase is used when nodes first meet. At this phase, nodes assign an initial trust level to each other. The second phase is triggered by trust level updates, which assumes that the nodes have already met. 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.

2.1 First Trust Assignment

When a node first meets a specific neighbor, it must assign an initial level of trust for this neighbor. The first trust assignment depends on several network parameters, such as mobility, location of nodes, and its current state. We can classify the first trust assignment strategy as prudent or friendly/naive. In the prudent strategy the node does not trust strangers and considers that every new neighbor might be a threat to the network. As a consequence, the node assigns a low value of trust for the new neighbor. Following the friendly/naive strategy means that every node is considered reliable until proven otherwise. In such case, the node associates a high level of trust for new neighbors. When a node adopts this strategy based on previous experience, we consider it friendly and if the node chooses this strategy due to lack of options it is considered naive. 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.

Different situations might demand distinct strategies. For example, if a node has already a significant number of reliable neighbors it can adopt a prudent strategy because it does not need new reliable neighbors. Further, the addition of a new neighbor might not significantly increase the probability of augmenting its satisfaction level. On the other hand, in a network where topology periodically changes and neighbor relationships are ephemeral, a node can opt for the naive strategy. In hostile environments, nodes might want to adopt the prudent strategy whereas in well-known cordial environments nodes can select the friendly strategy.

The first trust assignment occurs during the initial phase. The first trust level can also take into account the recommendation of known neighbors weighted by their trust levels. In order to a node a calculate the first trust level of node b, we propose the following equation

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

where F_a is the value used by node a according to the chosen strategy, $C_a(b)$ is the contribution of the trust level of other nodes about node b, and and α is the weight factor that allows us to give more relevance to the desired parameter. The group K_a defines the nodes from which recommendations will be considered. It is a subset of the neighbors of node a comprising all nodes that satisfy certain conditions. We consider two basic conditions for selecting K_a. The first one selects the nodes whose trust level is above a certain threshold (T_{th}). Let N_a be the set of neighbors of node a that includes all nodes known for a period of time longer than the relationship maturity threshold (M_{th}). The subset K_a can be defined as follows

$$K_a = \{\forall i \in N_a | T_a(i) \geq T_{th}\}. \tag{2}$$

Another option would be selecting r nodes in N_a with the highest trust levels.

Deciding the best strategy to derive F_a is not a simple task. For instance, F_a must take into account the level of mobility, the current satisfaction, the number of reliable neighbors. As choosing the best strategy evolves several parameters, we suggest a learning approach to select the strategy. This means that the Learning layer is responsible for selecting the best strategy.

2.2 Recommendation Computation

All nodes are qualified to contribute in the trust assignment. Therefore, the trust level evaluation might consider the recommendations of other nodes. The variable $C_a(b)$ is the contribution of all nodes $i \in K_a$ about node b weighted by the trust level of node a about node i, as follows

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

The relevance of the recommendation of other nodes is strongly related to the selection of K_n. The more trustworthy K_n is the more useful the recommendation of others is. The contribution considers not only the trust level of others but also the accuracy and the relationship maturity. The accuracy of a trust level is defined by the standard deviation, similar to Theodorakopoulos and Baras [8]. The relationship maturity is defined by $M_a(b)$ and it is expressed in hours by a continuous variable and X is a random variable with a normal distribution, which can be expressed as

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

The value in the trust level 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. After a trust level update of node a about node b, node a must update $\sigma_a(b)$. The vaule of $\sigma_a(b)$ is defined as

$$\sigma_a(b) = \sqrt{\frac{\sum_{j=1}^{k} (\overline{S} - s_j)^2}{k-1}}, \tag{5}$$

where S represents the set of the k last trust level samples about a specific node, for $2 \leq k \leq 10$. The value of $\overline{S}$ represents the average of these k samples. The parameter $\sigma_a(b)$ tells us the confidence of the trust level. A high value fot $\sigma_a(b)$ has two meanings. Either the node is not able to assess the trust value with accuracy or the node whose trust level is being estimated is unstable.

Malicious nodes might try to fake trust levels for several reasons. One can try to slander a trustworthy node, to make other nodes believe that a specific malicious node can be trusted, or just to confuse other nodes. In order to minimize this effect, each node must define a maximum relationship maturity value M_{max}, which represents an upper bound for the relationship maturity. This value is based on the average time for which a node knows its neighbors. Accordingly, we can express $M_i(b)$ as

$$M_i(b) = \begin{cases} M_i(b), \text{ if } M_i(b) < M_{max} \\ M_{max}, \text{ if } M_i(b) \geq M_{max}. \end{cases} \tag{6}$$

2.3 Trust Level Updating

After assigning a trust level to a specific neighbor a node must be able to change it whenever an event triggers this change. Updating the trust level imply two different steps. First, a node must know when to change a certain trust level. Second, a node must define how to calculate the new value of trust level.

We consider that every update is triggered by an event, but the occurrence of an event does not imply an automatic trust level update. The definition of event consists of the reception of a new recommendation or an action performed by a neighbor. The second type of event, which is related to the actions performed by a neighbor, is the most difficult to evaluate.

We first define the trust level update as a sum of its own trust and the contribution of other nodes, in the same way as defined by Virendra *et al.* [5]. The fundamental equation is

$$T_a(b) = (1 - \alpha)Q_a(b) + \alpha C_a(b), \tag{7}$$

where α permits choosing the most relevant factor. The variable $Q_a(b)$ represents the capability of a node to evaluate the trust level of their neighbors based on its own information. In order to obtain $Q_a(b)$, we propose the following equation

$$Q_a(b) = \beta E_T + (1 - \beta)T_a(b), \tag{8}$$

where E_T represent the value obtained by the judgment of a neighbor actions, β allows choosing which is the factor more relevant at a given moment. It means that α depends on which event has triggered the trust level update. For example, supposing node a starts a trust level update about node b, triggered by a new recommendation from neighbor c, but node a has noticed nothing strange in the behavior of node b. Thus, node a can ignore the first factor of Equation 8.

2.4 Recommendation exchange protocol

The Recommendation Exchange Protocol (REP) includes three basic messages and is not mandatory for every node. Nodes can choose weather to use it or not according to their current goals and constrains. When two nodes first meet they can broadcast a Trust Request (TREQ) with TTL equals to 1. Accordingly, the TREQ will not be forwarded by its neighbors. The other nodes must answer TREQ with a Trust Reply (TREP) message after waiting for a random period of time t_{REP} to avoid collisions and to wait for receiving other TREQs. The TREP message contains the recommendation of a specific node. If the replying node has received more than one TREQ, it might choose between sending different unicast messages and sending a broadcast message with all the requested recommendations. A node might set a TREP threshold under which it will not answer the TREQ. The threshold is based on the trust level of the requesting node. Before sending a TREQ message, a node might wait 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 will send 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 $\alpha = 1$ in Equation 7. The Trust Advertisement (TA) message is an unsolicited recommendation. A Node only send a TA message when the recommendation about a particular neighbor changes due to a reaction. Receiving a TA does not necessarily mean a recalculation of the trust level.

The recommendation includes the trust level for a particular node, its accuracy and for how long they know each other. For a node that does not implement the auxiliary trust table the recommendation includes just the trust level.

3 Results

This section presents the results and the main characteristics of the simulator we have implemented to evaluate the proposed scheme. In the simulator, each node has a particular nature which defines its behavior. The nature of a node ranges from 0 to 1. Most reliable nodes have nature equals to 1 while nodes not reliable have nature equals to 0.

All events that might happen with a node, like a route request not answered, a packet correctly forward, a useful information received, among others, are represented by "actions". Therefore, each node performs good actions or bad actions. Bad actions are represented by the value -1 and good actions by 1. Nodes perform actions according to an exponential distributed variable. 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.

Another important characteristic introduced in our simulator is the perception of a node. The perception indicates the probability of noticing a certain action. Therefore, a node with 0.4 of perception is able of noticing 40% of all the actions performed by its neighbors. This parameter simulates an interaction between the learning layer and the trust layer, since the perception and the judgment of an action is the responsibility of the Learning layer.

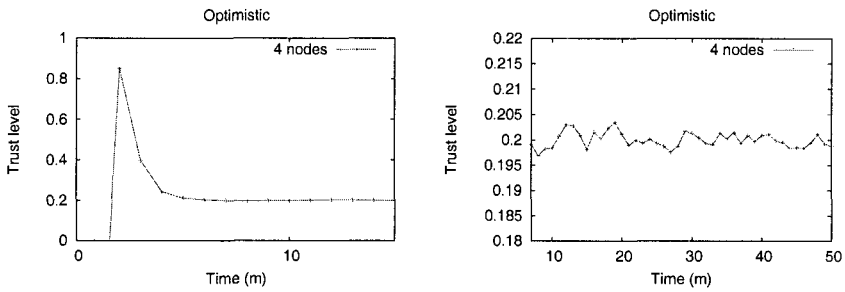
The term that considers the experiences of the own node in Equation 8 is calculated using the last i perceived actions. It implies the existence of a minimum number of actions i that a node must notice from each neighbor to be able of having an opinion about them, based on its own experience. This means 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.

Each neighbor might assume three different conditions. When nodes have not yet identified the existence of each other, they consider each other as an "unknown" neighbor. Nodes sense the presence of each other upon the reception of a message or the perception of an action. From this moment on, neighbors are considered "acquaintance" until the first trust level is assigned. Meanwhile, the trust level of an acquaintance is set to -1.

Our main goal in this paper is to evaluate and analyze the influence of the number of neighbors, the first trust assignment strategy, and the variation of parameters α and perception. All results are presented with a confidence interval of 95%.

The simulation scenario consists of 16 nodes with 250 m transmission range, which are randomly placed in a 150 m $\times$ 150 m area. Under these circumstances, all nodes can communicate directly to each other, characterizing a single hop ad hoc network. We chose this scenario to make easier the evaluation of the effect of the basic parameters already mentioned. All nodes operate in the advanced mode, which means that they implement all the features of the proposed system. We defined three values for the first trust assignment: 0.1 for the prudent, 0.5 for the moderate, and 0.9 for the friendly/naive strategy, also called optimistic strategy. All nodes adopt the same strategy. We also chose $\alpha = \beta = perception = 0.5$. These are the standard values for the simulations. For each specific configuration, the parameters that differ from its standard values are outlined. At last, in each configuration, all nodes have the same nature.

Figure 1 presents the time response of the average trust level from all neighbors about a specific node. In this specific scenario, the simulation time is 6,000 seconds. We observe in Figure 1(a) 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. After a specific amount of time $t_1 \approx 5$min, the curve oscillates around the correct value. Thus, we verify the existence of a transient period and stationary period. In the transient period (Figure 1(a)), nodes are trying to approximate to the expected value, while in the stationary period, the trust level is almost stable, very close to the correct value, varying like a smoothed sine function.

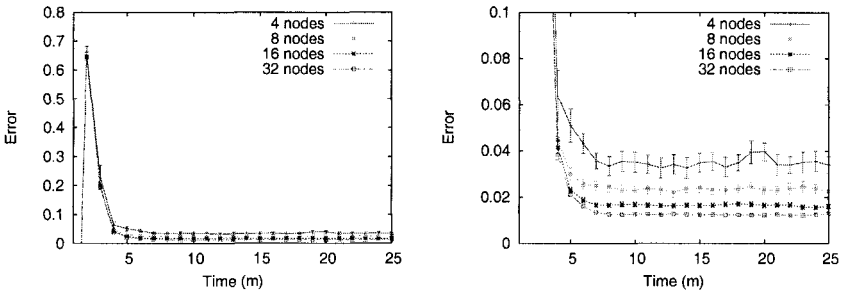


(a) Transient period (b) Stationary period

Fig. 1. Variation of trust level during time.

In the other figures, instead of presenting the average trust level, we present the average error of the trust value evaluated, that is, the difference between the trust level and the correct value. At the end, the ideal result is a curve that reaches the value zero, 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.

In Figure 2, nodes adopt an optimistic strategy and we vary the number of neighbors. The nature is set to 0,2. We can notice that the greater is the number of neighbors the closer to zero is the error. It occurs due to the fact that augmenting the number of neighbors means increasing the number of recommendations, which implies a greater probability of receiving recommendations closer to the correct value.



(a) (b)

Fig. 2. Influence of the number of neighbors.

Figure 3 shows the influence of the parameter *alpha* on the trust level evaluation. Decreasing *alpha* implies that the contribution of other nodes has a minor effect in the trust level calculation. The first observation from Figure 3(a) is that the convergence to the correct value is faster with a higher *alpha*, namely, the transient is longer. Therefore, although the global opinion about a specific node changes slower when *alpha* is larger, the convergence value is closer to the expected one and presents a smaller variation, as shown by Figure 3(b).

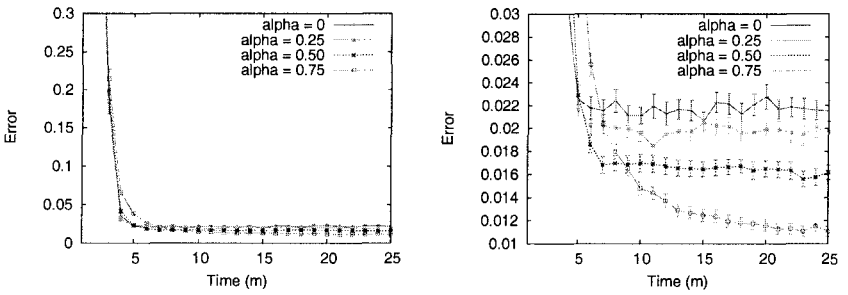


Fig. 3. The influence of *alpha*.

The perception is the fraction of actions a node can notice from its neighbors. Figure 4 shows the impact of the perception on the trust level evaluation. It is clear that the perception is strong related to the duration of the transient period. It occurs due to the existence of a minimum number of actions from each neighbor for nodes to consider its own experiences. If we augment 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 transient.

Afterwards, the perception is set to 0.2, varying the number of nodes. Figure 5 reveals that with a low perception the importance of the number of neighbors to reach closer to the expected value is clearer. It means that the lowest is the perception, the lowest is the probability of noticing the real nature of a neighbor by the judgment of its actions. On the other hand, a low perception can be compensated by a larger number of neighbors.

At last, Figure 6 presents the influence of the nature on the trust level evaluation. For this purpose, we set the strategy to optimistic (Figure 6(a)) and moderate (Figura 6(b)), varying the nature. We can observe, by Figure 6(a), that the nature does not affect significantly the duration of the transient, only the peak, according to the chosen strategy. On the other hand, Figure 6(b) shows that it is easier for nodes to find the correct value when the nature is in

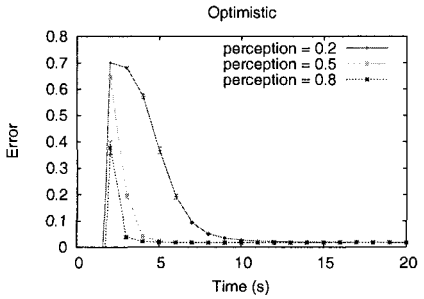


Fig. 4. The influence of perception.

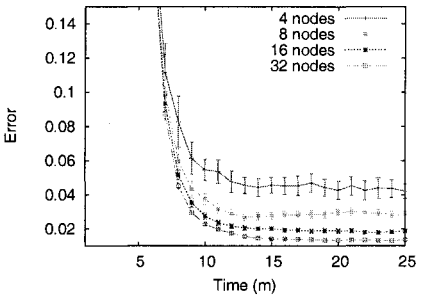
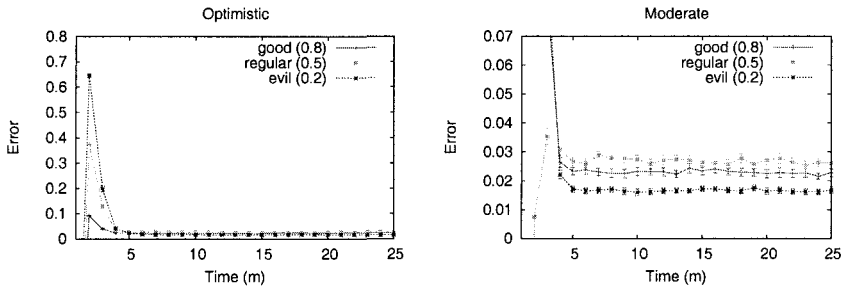


Fig. 5. Number of neighbors with a low perception.

the extremities. It happens because when a node produces the same amount of good and bad actions, the probability of sensing the exact proportion of good and bad actions decreases, considering that perception is less than 1.0.

4 Conclusion

In this paper, we propose a trust assignment model for ad hoc networks. We aim at building a trust relationship among nodes inspired by the human concept of trust. Our concern is different from other works that focus strictly on security issues. We focus on providing nodes a way of having an opinion about their neighbors. This opinion governs the interaction among nodes. The goal is to make nodes capable of making their own decisions based on the autonomic paradigm. The proposed model results in a utterly distributed trust system for ad hoc networks based on the recommendation of other nodes and on the own experiences of the nodes. Our approach considers not only the trust level but also its accuracy and the relationship maturity. We also define the Recommendation Exchange Protocol (REP) that allows nodes to exchange recommendations in an efficient way. The system performance is analyzed through



(a) Transient period (b) Stationary period

Fig. 6. The influence of the nature.

simulations The results reveal the efficacy of the proposed system and show the influence of the main parameters.

References

1. Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," in *IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, (Suzhou, Chine), May 2004.
2. Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solution in ad hoc networks," in *Proceedings of the Seventh Nordic Workshop on Secure IT Systems, (NordSec'03)*, (GjÃ¸vik, Norway), Oct. 2003.
3. H. Deng, W. Li, and D. P. Agrawal, "Routing security in wireless ad hoc networks," *IEEE Communiations Magazine*, pp. 70–75, Oct. 2002.
4. A. A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *Proceedings of 27th Australasian Computer Science Conference (ACSC'04)*, (Dunedin, New Zealand), Oct. 2004.
5. M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya, "Quantifying trust in mobile ad-hoc networks," in *Proceedings of IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'05)*, (Waltham, USA), Apr. 2005.
6. S. Buchegger and J.-Y. Le Boudec, "The effect of rumor spreading in reputation systems for mobile ad-hoc networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt03)*, (Sophia-Antipolis, France), Mar. 2003.
7. K. Ren, T. Li, Z. Wan, F. Bao, D. Robert H, and K. Kim, "Highly reliable trust establishment scheme in ad hoc networks," *Computer Networks*, vol. 45, no. 6, pp. 687–699, Aug. 2004.
8. G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSE'04)*, (Philadelphia, USA), Oct. 2004.