Trust Management for Survivable Communication on Self-Organized Wireless Networks

Pedro Velloso and **Guy Pujolle**
UPMC - LIP6 - UNIVERSITAS - France
Emails: {Pedro.Velloso, Guy.Pujolle}@lip6.fr

Eduardo Silva and **Michele Nogueira**
Department of Informatics - UFPR - Brazil
Email: {eduardos, michele}@inf.ufpr.br

***Abstract*—Self-organized wireless networks comprise different devices cooperating among themselves to provide network services. These networks support critical applications in different domains requesting simultaneously robustness, security and availability. Hence, it is essential to develop mechanisms for steering nodes' cooperation in order to preserve essential network operations even in presence of intrusion, attacks or network failures. We propose a trust management framework whose goal is to guide node cooperation towards network survivability. Our framework evaluates node trustworthiness correlating adaptively criteria provided by node behavior, recommendations, security mechanisms and node attributes. Simulation results show improvements on network trustworthiness for different situations.**

I. Introduction

Self-organized wireless networks, such as mobile ad hoc networks, wireless mesh networks and wireless sensor networks, comprise nodes communicating among themselves in a cooperative way [1], [2]. Their nodes perform a set of equivalent functionalities to support network essential services as link-layer connectivity, routing and end-to-end communication. Researchers have envisaged these networks to assist critical applications in domains like medical, commercial and financial ones. These applications request both robustness and security on network services.

Decentralization is a main issue for self-organized wireless networks needing that nodes cooperate to offer network services [2]. Each node possesses autonomy to make its own decision about how to participate in network operations. Nodes' decisions concern primarily their own benefits and, thus, cooperation or fairness cannot be guaranteed. Further, nodes take actions without knowing whether they can trust neighbors with which they are collaborating. Hence, selfish or malicious behaviors can frequently be observed in these networks resulting in inefficiency, low quality and low availability of network services.

Trust management systems (TMSs) have been employed to steer nodes' cooperation in self-organized networks [1]. They evaluate trustworthiness, reliability or competence of nodes making easy nodes decide with whom to collaborate. Different TMSs exist. However, none of them has as goal node's evaluation towards network survivability, i.e., the network capability of limiting damage, recovering and operating robustly even in face of attacks or intrusions. For achieving survivability in self-organized wireless networks, it is essential the development of TMSs that can (i) assist nodes in decisions to improve network security and robustness; (ii) lead network adaptations in face to threats; (iii) detect misbehavior; and (iv) quantitatively assess network survivability.

This paper has as main contribution a trust management framework towards network survivability. Different from existing trust models, our framework focus on evaluating trustworthiness of nodes in order to guide cooperation guaranteeing that network services will be correctly performed even under attacks, intrusions and network failures. Our approach correlates adaptively multiple criteria for trust level calculation provided by security mechanisms and node's attributes, as well as node's behavior and recommendations. Security criteria offer survivability properties as resistance, recognition and recovery. Simulation results show advances on network trustworthiness.

The remainder of this paper is organized as follows. Section II depicts related works. Section III details our trust management framework. Section IV presents simulation results. Finally, Section V concludes the paper and outlines future works.

II. Related Works

Researchers have proposed different TMSs for self-organized wireless networks [3], [4]. We categorize them into three groups: those derived from centralized solutions, intelligent trust models and biologically inspired trust models. The first group comprises trust models that evaluate trust level considering the behavior of nodes and recommendations like [3]. Such models adapt existing solutions for the context of self-organized networks. Pirzada and McDonald [5], e.g., introduced a distributed perspective for computing trust levels of nodes. Nodes individually calculate trust levels of other nodes based upon information gathered in a passive mode related to packet forward. Sun *et. al* [4] proposed also a distributed trust model intending to improve security.

In the second group, intelligent trust models employ artificial intelligent methods or try to adapt to network conditions. Luo *et al.* [6] used fuzzy recommendations for credibility rating of opinions delivered by other nodes, evaluation of recommendations and assessment of past experiences. Boukerche *et al.* [7] proposed an adaptive trust calculation based on past node actions.

Finally, in the third group, biologically inspired trust models took biological phenomena as references in order to improve trustworthiness evaluation. Velloso *et al.* [8] proposed a TMS inspired in the human concept of trust. Trust evaluations consider neighbors' recommendations and the node's experience. Recommendations are pondered by their accuracy and the maturity of the relationship between the evaluating node and

the recommending node. Our trust management framework follows a different perspective. Its goal is to use trust levels for indicating the expectation of a service be provided or a commitment be fulfilled as promoted by Hoffman *et al.* [9].

III. Trust management system

The proposed TMS aims at providing nodes with procedures to evaluate how much a neighbor is trustful to guarantee network service or application requirements. In this work, the trust level of nodes represents how much a node is trustful to provide a network service or to guarantee application requirements. Depending on the application or service, the same neighbor can be more or less trustful.

The trust management system is based on SAMNAR (Survivable Ad Hoc and Mesh Network Architecture), a conceptual architecture for network survivability in face of attacks and intrusions [10]. SAMNAR is inspired on the human body immune system, and it proposes a security management approach lying on the adaptive coordination of preventive, reactive and tolerant defense lines. Preventive defense line consists of security mechanisms, such as cryptography, firewalls and access control techniques. Reactive defenses try to detect and react against intrusions by security mechanisms, such as reputation systems and diagnosis systems. Tolerant defenses aim at mitigating damages caused by attacks or intrusions, and at recovering compromised services.

Fig. 1 illustrates our trust management framework. It proposes that each node individually evaluates trustworthiness of other nodes considering multiple criteria provided by previous experiences, neighbor's recommendations, security mechanisms and nodes' attributes. Each node assigns a value called trust level lying in the range 0 and 1, where 0 means the least reliable node, and 1 means the most reliable node.

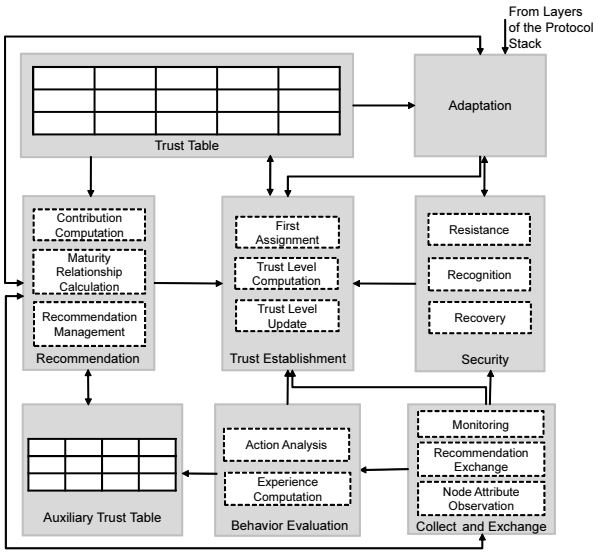

Fig. 1. Trust management framework

We describe each block of our framework in the next subsections. Blocks comprise **TMS operations** running on each node. The **TMS entity** and **managed agents** execute TMS operations. The **TMS entity** represents an application running in the node for controlling the data collection, processing, trust calculation, analysis and decisions. Further, it controls **managed agents**, consisting in a daemon running in background to monitor neighbor actions and collect information.

A. Trust Establishment

The **trust establishment block** assigns trust level for neighbor nodes. This block comprises three main operations as *trust level computation*, *first assignment* and *trust level update*. It receives inputs from different blocks of our framework, and some of its parameters are controlled by the **adaptation block**.

Trust Level Computation: employs criteria to evaluate the trustworthiness of neighbors. Observations of neighbor behavior, neighbor recommendations, security mechanisms and node's attributes supply criteria values. The adaptation block defines how criteria are combined depending on the network situation or application requirements. A given node a computes the trust level, $T_a(b)$, for each node b in its neighborhood following Eq. 1.

$$T_a(b) = \frac{Th_a(b) + Tc_a(b)}{2} \tag{1}$$

In Eq. 1, $Th_a(b)$ denotes trust evaluations performed by node a for node b based on direct observations of a from b's behavior and common neighbors' recommendations. $Tc_a(b)$ represents trust evaluations related to security mechanisms and b's attributes, i.e., $Tc_a(b)$ characterizes the b's level of security and reliability. $Th_a(b)$ is calculated by Eq. 2, being $\alpha avalueintherangebetween0and1$.

$$Th_a(b) = (1-\alpha)Q_a(b) + \alpha C_a(b) \tag{2}$$

The variable $Q_a(b)$ denotes the capability of a node to evaluate the trust level of their neighbors based on its own observations and on neighbor recommendations, called contribution, $C_a(b)$, computed as described in Subsection III-B. The variable α allows nodes to choose the most relevant factor, and the value of $Q_a(b)$ is given following Eq. 3.

$$Q_a(b) = \beta E_T + (1-\beta)T_a(b) \tag{3}$$

The variable E_T denotes the trust value obtained by the judgment of neighbor actions performed by the **behavior evaluation block**, particularly, by the *experience calculator*. Being a value in the range between 0 and 1, the variable β allows to set different weights for the factors of the equation, selecting which factor is the most relevant at a given moment. The variable $T_a(b)$, in this equation offers the last trust level value saved in the **trust table**.

$Tc_a(b)$ is computed by Eq. 4, comprising criteria supplied by security mechanisms and by neighbor attributes. $E(b)$ denotes the current energy level of node b, being an example of criterion provided by neighbor attributes. $N(b)$ designates the number of neighbors of node b. $L(b)$ denotes the probability of node b to be a liar. And $K(b)$ is a normalized value of the

cryptographic key length used by node b. ϵ, λ, δ and γ stand for the weight of each criteria in the equation, and $\epsilon+\lambda+\delta+\gamma = 1$. Every security criterion employed in this equation is detailed in Subsection III-C. Our framework considers the factor $Tc_a(b)$ for the calculation of $T_a(b)$ only if the trust level based on the behavior, $Th_a(b)$, is above a threshold ψ defined by the adaptation block. In this work, we set ψ of 0.5.

$$Tc_a(b) = \epsilon E(b) + \lambda N(b) + \delta L(b) + \gamma K(b) \quad (4)$$

First Assignment: consists in assigning an initial trust level for neighbors. When a node meets for a first time a specific neighbor, it assigns this initial value. Velloso *et. al* [8] propose two strategies for first assignment depending if a node consider the new neighbor as a friend or a stranger.

Trust Level Update: manages entries of the trust table. Since a trust level changes or a node is no longer a neighbor, entries in this table need to be updated. The trust establishment block is aware of neighbor nodes by monitoring procedures in the **collect and exchange block**.

B. Recommendation

The **recommendation block** manages recommendations provided by the neighbors of a given node. Recommendations are obtained using procedures from the collect and exchange block. The block owns three main operations as *recommendation management*, *contribution computation* and *maturity relationship calculation*. Recommendation management generates recommendations to be sent to neighbors, and it controls recommendations received from the collect and exchange block.

The contribution $C_a(b)$ is defined by Eq. 5 and denotes the sum of all recommendations for node b from a subset K_a of b's neighborhood. K_a comprises neighbors of node a that satisfy certain conditions defined by the recommendation management, such as those neighbors owing trust level above a certain threshold. Each recommendation is weighted by the trust level of the node a for each recommender neighbor i.

$$C_a(b) = \frac{\sum_{i \in K_a} T_a(i) M_i(b) X_i(b)}{\sum_{j \in K_a} T_a(j) M_j(b)} \quad (5)$$

The contribution considers not only the trust level of other nodes but also the accuracy and the *relationship maturity*. The accuracy of a trust level is defined by the standard deviation. The value in the trust table of node a regarding node b is associated to a standard deviation, which refers to the variations of the trust level that node a has observed about node b.

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 measures for how long two nodes have known each other. We use the relationship maturity to enhance the confidence in recommendations from nodes that know b for longer time. We assume that the trust level of a neighbor with a more mature relationship has already converged to a common value within the network and therefore its opinion should be more relevant than the opinion of a new neighbors.

C. Security

The **security block** consists of security mechanisms employed by nodes. They follow three defense lines, prevention, reaction and tolerance, providing security criteria used also to evaluate trustworthiness of neighbors. We apply cryptographic key length as a criterion from prevention, cryptography. We assume the existence of a PKI (Public-Key Infrastructure) on the network. Cryptography represents the resistance of the nodes to attacks that can harm the integrity or confidence of the communication. Larger cryptographic keys results in stronger resistance to attacks, minimizing the probability of damages. Hence, nodes using larger cryptographic keys are considered more trustworthy for communication than nodes using shorter ones. For using this criterion in Eq. 4, we normalize cryptographic key length in a value of the interval between 0 and 1.

We assume the existence of a diagnosis system, e.g., a voting scheme [11]. This system detects nodes that lie about recommendations. It returns a value in the range from 0 to 1, representing the probability of a given node not be a liar. We employ the diagnosis system as a reactive defense, improving the recognition property of the TMS by the detection of liars. The probability of lying is employed as another criterion to evaluate the trust level of neighbor nodes. Nodes with higher probability of lying are less trustworthy than node with lower probability.

In order to enhance the recovery property, we apply the number of neighbors as the third security criterion. This criterion represents the tolerance defense. A node owning a larger number of neighbors is more reliable since it increases the probability of having a larger number of trustworthy neighbors. Further, a larger number of neighbors increases the confidence of recommendations and assists in liar inhibition.

We use these three security criteria as complementary needs to enhance the trust level evaluation of a node. We argue that each one can assist in the trustworthiness evaluation of the others. Then, we consider all of them together. Further, they are combined with recommendations, neighbor attributes, and direct neighbor observations about their behavior in order to make more robust the trust level establishment.

D. Collect and Exchange

The collect and exchange block provides procedures for gathering or receiving information from nodes. Information can be recommendations supplied by neighbors of a node, neighbor attributes or security criteria, used to enhance trust level assessment. This block owns mechanisms to monitor nodes. Hence, this block comprises operations as *monitoring*, *node attribute observation* and *recommendation exchange*.

Monitoring intends to observe actions of neighbors, as the forward of data packets. They are employed by the behavior evaluation block, particularly, by the action analysis operation. Observations provide evidences related to the neighbors' behavior to assist trust level evaluation. The security criteria observation collects values of cryptographic and liar criteria. Recommendation exchange performs interactions among nodes to provide recommendations. In [8], a recommendation

exchange protocol (REP) is proposed. It includes three basic messages as Trust Request (TREQ), Trust Reply (TREP) and Trust Advertisement (TA). When two nodes meet for the first time, they broadcast a TREQ to their direct neighbors. Their neighbors receive the TREQ and answer it with a TREP message. The TREP contains the recommendation of a specific node. Finally, TA messages are employed to announce other neighbors about changes in trust evaluation about a specific node. We use these messages of the REP protocol to piggyback criteria values as the remaining energy level of their battery and the number of neighbors. Such values are informed by direct neighbors, differently of other criteria that are observed by the evaluator node or by recommender nodes.

E. Behavior Evaluation

The behavior evaluation block provides procedures for evaluating the behavior of neighbors based on observed actions. Also, it offers procedures to calculate the neighbor experience. In this work, nodes observe if their neighbors have forwarded packets or not. Each time a neighbor correctly forwards a packet the node accounts a positive action; otherwise the node reduces from 1 the number of positive actions. We define a *perception* parameter, τ. This parameter represents the percentage of neighbor actions a node can observe. If the perception equals 100%, the node observes all actions of its neighbors. However, due to resource limitations, nodes may not be able to observe all actions performed by their neighbors, reducing perception probability.

F. Adaptation

Based on inputs provided by layers of the protocol stack and application requirements, our framework proposes that nodes can self-adjust parameters used by other blocks. The adaptation block controls those parameters giving priority for some criteria in the trust level calculation. Depending on the situation, nodes can give priority for evaluations resulted from the neighbor behavior or from the security criteria. For instance, the adaptation block controls priorities changing the values of ϵ, λ, δ, γ, β, α in Eq. 2 and Eq. 3, and the threshold ψ, according to application requirements or service.

Learning algorithms assist this block in adapting parameters or replacing security mechanisms based on previous knowledge. Since adaptations must be executed quickly, learning algorithms gain knowledge of previous decisions and actions, and then change faster configurations. Changes include adjusts on parameter values as well as the replacement of protocols or even security mechanisms.

This block also adapt security criteria applied for trust level calculation or can control how the recommendations are exchanged among nodes. Values in the trust table or those used by security mechanisms can trigger the adaptation block to react, aiming at better evaluating nodes depending on the situation and application requirements. Further, the adaptation block can replace security mechanisms adapting the TMS towards survivability.

G. Trust Table

Each node must keep a trust table containing the trust level for all its neighbors. Each entry on the trust table owns a timeout. Hence, an entry is excised from the trust table whenever the node associated to that entry is no longer a direct neighbor, or when it expires. All recommendations related to that entry are excised as well. The trust establishment block, i.e., the trust level update, manages trust table entries.

H. Auxiliary Trust Table

The auxiliary trust table (ATT) aims to offer additional information to nodes to improve the trust level evaluation. It contains information related to the confidence in each trust level and for how long neighbor nodes have kept that information. Maintaining ATT requires more resources, such as energy and storage. Thus, we define that nodes can maintain this table or not depending on their resource capabilities. In this work, we consider that nodes maintain ATT.

IV. EVALUATIONS

A. Simulation Environment

Evaluations are performed by the network simulator NS-2, version 2.28. We use a scenario of reference comprising of 30 nodes with 250 m transmission range, randomly distributed in an area of 600 by 600 meters. Random CBR sources send packets of 500 bytes at 10 kbps. The simulation time is 600s. Nodes use the DSR routing protocol, forwarding only 70% percent of the received data packets. All results in graphics present 95% of confidence interval.

Nodes monitor the action of forwarding data packets and calculate the trust level based on neighbor behaviors. Nodes-own τ of 50%, meaning that they perceive 50% of the actions of their neighbors. Parameters α and β in Eq. 2 and Eq. 3 are set to 0.5. $Tc_a(b)$ of Eq. 1 is computed depending on the values of the criteria: $E(b)$, $N(b)$, $L(b)$ and $K(b)$.

For evaluations, we defined two situations. In the first one, called *routing situation*, nodes use trust level to evaluate which neighbors can offer more robust routes. In this situation, the remaining energy level of nodes has higher priority than other criteria. Thus, we set ϵ with the highest value, i.e., 0.5. The second situation, called *security situation*, is related to applications that request high level of security. In this situation, security criteria have higher priority than the energy criterion. Hence, the values of λ, δ, and γ have been set to 0.1, 0.3 and 0.5.

B. Results

First, we add to trust level calculation the remaining energy criterion. It is joined to the behavior evaluation and neighbor recommendations used in the TMS of reference called HIT (Human-Inspired Trust) model [8]. Fig. 2 compares trust levels calculated by a node to a specific neighbor following HIT and HIT-e (when we consider the new criterion for trust evaluation). We vary the remaining energy level of this neighbor from 20% to 100%, then we observe the variation resulted from HIT-e. Adding the new criterion, trust levels

highlight the neighbor condition. When the neighbor presents low level of remaining energy, the resulted trust level is low; and when the neighbor presents a high level of remaining energy, the resulted trust level is high.

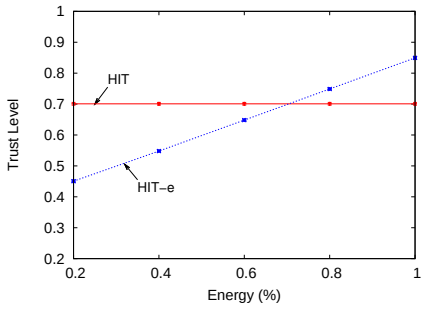

Fig. 2. Comparing trust levels

Fig. 3 compares the error resulted from calculating trust levels by HIT or HIT-e. This error occurs due to the node perception of 50% in neighbor action monitoring. We note that HIT-e reduces in almost the half the error resulted from HIT under the same conditions. For the next simulations, values for each criteria were randomly selected among 25%, 50%, 75%, and 100%.

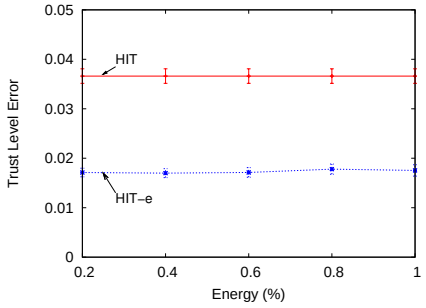

Fig. 3. Evaluating error on trust level calculation

We set the value of each parameter (ϵ, λ, δ, and γ) to 1 and the others to 0. Fig. 4 shows the trust level of six nodes according to each criterion. Each node may present distinct trust levels depending on the considered criterion.

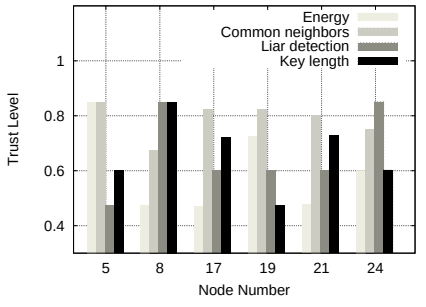

Fig. 4. Trust level for each single criterion

Fig. 5 shows results combining all criteria. We observe both situations following values defined for the parameters that weight the priority of each criterion. We note that a node can present different trust levels depending on the situation. Further, for the same situation, neighbors can be distinguished, assisting decisions about with whom to cooperate.

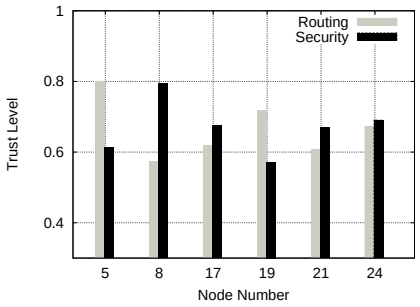

Fig. 5. Comparing trust levels on routing or security situations

V. Conclusion

We propose a trust management framework towards communication survivability on self-organized wireless networks. Our framework is based on the SAMNAR architecture, that proposes the coordinated and adaptive use of preventive, reactive and tolerant defense lines, and on the HIT trust management system. We introduce the use of multiple criteria in the trust level evaluation. Simulations results demonstrated improvements in trust evaluation when employing multiple criteria. As future work, we highlight the improvement of the adaptation block.

References

[1] S. Buchegger, J. Mundinger, and J.-Y. L. Boudec, "Reputation systems for self-organized networks," *IEEE Technology and Society Magazine*, vol. 27, no. 1, pp. 41–47, Spring 2008.

[2] Y. Sun, Z. Han, and K. Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE communications magazine*, vol. 46, no. 2, pp. 112–119, Feb. 2008.

[3] Y. L. 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*, April 2006, pp. 1–13.

[4] Y. Sun and Y. Yang, "Trust establishment in distributed networks: analysis and modeling," in *IEEE ICC*, Jun. 2007, pp. 1266–1273.

[5] A. A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *ACSC*. Darlinghurst, Australia: Australian Computer Society, Inc., 2004, pp. 47–54.

[6] J. Luo, X. Liu, and M. Fan, "A trust model based on fuzzy recommendation for mobile ad-hoc networks," *Comput. Netw.*, vol. 53, no. 14, pp. 2396–2407, 2009.

[7] A. Boukerche, Y. Ren, R. Pazzi, and W. Nelem, "An adaptive computational trust model for mobile ad hoc networks," in *IWCMC*. New York, NY, USA: ACM, 2009, pp. 191–195.

[8] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, "HIT: a human-inspired trust model," in *IFIP/IEEE MWCN*, Aug. 2006.

[9] L. J. Hoffman, K. Lawson-Jenkins, and J. Blum, "Trust beyond security: an expanded trust model," *Communications of the ACM*, vol. 49, no. 7, pp. 94–101, 2006.

[10] M. Lima, H. Silva, A. Santos, and G. Pujolle, "An architecture for survivable mesh networking," in *IEEE GLOBECOM*, Dec. 2008.

[11] A. O. Santin, R. G. Costa, and C. Maziero, "A three-ballot-based secure electronic voting system," *IEEE Security & Privacy*, vol. 6, no. 3, pp. 14–21, 2008.