University of Murcia
Faculty of Computer Science

Department of Information
and Communications Engineering

Trust and Reputation Management in Distributed and Heterogeneous Systems

PhD Thesis

Author
Félix Gómez Mármol

Thesis advisor
Dr. Gregorio Martínez Pérez

Murcia, April 2010

The following Thesis is a compilation of the next published articles, being the PhD student the main author in all of them:

1. Félix Gómez Mármol, Gregorio Martínez Pérez, "State of the art in trust and reputation models in P2P networks", Handbook of Peer-to-Peer Networking, Eds: X. Shen, H. Yu, J. Buford, M. Akon, Publisher: Springer, ISBN: 978-0-387-09750-3, pp 761-784, 2010
   http://dx.doi.org/10.1007/978-0-387-09751-0_26

2. Félix Gómez Mármol, Gregorio Martínez Pérez, Antonio F. Gómez Skarmeta, "TACS, a Trust Model for P2P Networks", Wireless Personal Communications, vol. 51, no. 1, pp 153-164, 2009
   http://dx.doi.org/10.1007/s11277-008-9596-9

3. Félix Gómez Mármol, Gregorio Martínez Pérez, Javier Gómez Marín-Blázquez, "META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm", Intelligent Automation and Soft Computing (Autosoft) Journal, 2010 (in press)

4. Félix Gómez Mármol, Gregorio Martínez Pérez, "Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique", Telecommunication Systems Journal, vol. 46, no. 2, 2010 (in press)
   http://dx.doi.org/10.1007/s11235-010-9281-7

5. Félix Gómez Mármol, Gregorio Martínez Pérez, "Towards Pre-Standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems", Computer Standards & Interfaces, Special Issue on Information and Communications Security, Privacy and Trust: Standards and Regulations, vol. 32, no. 4, pp. 185-196, 2010
   http://dx.doi.org/10.1016/j.csi.2010.01.003

6. Félix Gómez Mármol, Gregorio Martínez Pérez, "Security Threats Scenarios in Trust and Reputation Models for Distributed Systems", Elsevier Computers & Security, vol. 28, no. 7, pp. 545-556, 2009
   http://dx.doi.org/10.1016/j.cose.2009.05.005

7. Félix Gómez Mármol, Gregorio Martínez Pérez, "TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks", IEEE International Conference on Communications (IEEE ICC 2009), Communication and Information Systems Security Symposium, Dresden, Germany, 14-18 June 2009
   http://dx.doi.org/10.1109/ICC.2009.5199545

Universidad de Murcia

D. Gregorio Martínez Pérez, Profesor Titular de Universidad del Área de Ciencia de la Computación e Inteligencia Artificial en el Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia, AUTORIZA:

La presentación de la Tesis Doctoral titulada "Gestión de la Confianza y Reputación en Sistemas Distribuidos y Heterogéneos", realizada por D. Félix Gómez Mármol, bajo mi inmediata dirección y supervisión, en el Departamento de Ingeniería de la Información y las Comunicaciones, y que presenta para la obtención del grado de Doctor Europeo por la Universidad de Murcia.

En Murcia, a 19 de Mayo de 2010

D. Gregorio Martínez Pérez

Universidad de Murcia

D. Antonio Fernando Gómez Skarmeta, Catedrático de Universidad del Área de Ingeniería Telemática y presidente de la Comisión Académica del Postgrado de la Facultad de Informática de la Universidad de Murcia, INFORMA:

Que la Tesis Doctoral titulada "Gestión de la Confianza y Reputación en Sistemas Distribuidos y Heterogéneos", ha sido realizada por D. Félix Gómez Mármol, bajo la inmediata dirección y supervisión de D. Gregorio Martínez Pérez, y que la Comisión ha dado su conformidad para que sea presentada ante la Comisión General de Doctorado.

En Murcia, a 19 de Mayo de 2010

D. Antonio Fernando Gómez Skarmeta

Table of Contents

Acknowledgements

Gracias a Dios, ya he hecho mi tesis.

En primer lugar no puedo sino dar las gracias a mi familia. A mi hermano Alberto, que desde que lo conozco no ha hecho otra cosa que hacerme feliz. Muchas gracias Alberto. A mis padres, por supuesto, porque siempre me han animado y apoyado en mis decisiones, siempre han estado a mi lado en los momentos difíciles y siempre me han demostrado lo mucho que me quieren. Y al resto de mi familia, por todo su cariño.

También quisiera dar las gracias a todos los amigos que a lo largo de mi vida me han acompañado. A mis compañeros del colegio (por los buenos recuerdos), a los del instituto (por la amistad que perdura y los buenos momentos), a los de la universidad (por su ayuda y apoyo), a mis amigos de ÁTICA (porque siempre me resuelven muchos problemas), de CYUM (por tantas buenas experiencias), a mis compañeros y amigos de despacho (por sus sabios consejos y el buen clima de trabajo), a mis amigos de San Benito (porque construyeron una parte determinante de mi vida) y de Caravaca (por acogerme como uno más de ellos desde el principio), a los compañeros y amigos que hice en Heidelberg (por todos los buenos ratos que compartimos) y a todos aquellos que, de una forma u otra, quisieron compartir parte de sus vidas conmigo.

No puedo olvidarme de los que desde pequeño me enseñaron a aprender, a querer aprender y a querer enseñar a aprender. A mis profesores del colegio, a mis profesores del instituto y a mis profesores de la universidad. A muchos de ellos los recuerdo con mucho cariño.

Por último, pero no por ello menos importante, merece una mención especial mi director de tesis, Gregorio. Los que me han oído hablar de él (y los que lo conocen personalmente) sabrán si leen estas líneas, que seguramente me quedaré corto en lo que diga. Ahora vuelvo la vista atrás y no sé exactamente en qué momento sucedió que dejó de ser "sólo" mi director de tesis para convertirse, también, en mi amigo. No sé cuándo fue, pero me alegro profundamente de que ocurriera. Él me ha enseñado muchas cosas, tanto a nivel profesional como personal.

A nivel profesional le tengo que agradecer que me enseñara a ser más riguroso, más diligente, a trabajar duro y con perseverancia, a no tener miedo por apostar alto; en definitiva, a tratar de mejorar día a día en lo profesional, pero sin descuidar nunca el plano personal.

En este último aspecto, le agradezco sus constantes lecciones de humildad, de sencillez, de amabilidad y buen trato con todo el mundo, de atención por el estudiante y sus preocupaciones y de trabajar siempre con el objetivo de mejorar el entorno que nos rodea.

Tratando mi tesis sobre la gestión de la confianza y la reputación, debo decir que Gregorio se ha ganado, con creces, mi total confianza y que para mí goza de una reputación intachable e inmejorable.

Muchas gracias por todo Gregorio.

Thank God, I have already finished my Thesis.

Firstly, I cannot but thank my family. To my brother Alberto, who has not done anything but make me happy, ever since I have known him. To my parents, of course, because they have always encouraged and supported me in my decisions, they have been by my side in the difficult moments and they have always showed how much they love me. And to the rest of my family, for all their affection.

I would also like to thank all the friends who have accompanied me throughout my life. To my primary school mates (for the good memories), to my secondary school ones (for the friendship that still remains and the good moments), to my university mates (for their help and support), to my friends from ÁTICA (because they always solve many of my problems), from CYUM (for so many good experiences), to my friends and office mates (for their wise advice and for the good atmosphere at work), to my friends from San Benito (because they built a determining part of my life) and from Caravaca (for taking me in as one of them from the beginning), to the friends and workmates I met in Heidelberg (for all the good moments we shared), and to all of those that, in one way or another, were willing to share part of their lives with me.

I cannot forget those that, since I was a child, taught me to learn, to want to learn, and to want to teach to learn. To my primary school teachers, my secondary school ones, and my university professors. I remember many of them with great affection.

Finally, my Thesis advisor, Gregorio, deserves a special mention. Those who have heard me talking about him (and those who know him personally) will know if they read these lines, that I will probably fall short in my words. Now I look back and I do not know when it happened exactly that he stopped being "only" my Thesis advisor in order to become also my friend. I do not know when it was, but I am profoundly glad that it happened. He has taught me many things, both at a professional and a personal level.

At a professional level I must thank him for teaching me to be more rigorous, more diligent, to work hard and with perseverance, not to be frightened for betting high; in other words, to try to improve day by day in the professional aspect, without overlooking the personal one.

In this last respect, I am grateful for his being a constant model of humility and modesty, of kindness and pleasant manner with everybody, of attention to students and their worries, and of working always with the aim of improving our surrounding environment.

Since my Thesis deals with trust and reputation management, I must say that Gregorio has fully gained my absolute confidence and that, in my opinion, he has an irreproachable and unbeatable reputation.

Thank you very much for everything, Gregorio.

Abstract

I Motivation and Goals

≪In the first fifteen years of its existence, the World Wide Web has had a profound and transformative impact on all facets of our society. While the Internet has been with us for 40 years, the Web has caused an exponential growth of its use; with up to 1.5 billion users worldwide now accessing more than 22 billion web pages. 'Social Networks' are attracting more and diverse users. With 4 billion subscribers to mobile telephony across the globe (there are almost 7 billion people on earth) and mobile phones being increasingly used to connect to the Internet, mobile web applications and services are developing fast.

And there is much more to come, which will go well beyond information processing and data exchange. The 'Internet of Things', the Semantic Web and Cloud Computing are all evolving fast, reflecting the dynamism of the technology developments that are related to the digitization of the world around us and our relationship with it. They in turn raise issues of e-Identity and Trust in the digital interactions they enable.

However, while we are staring at this amazing new world and getting excited by the use of previously unimagined devices, we are also perplexed and concerned by the ease with which our data can be stolen, our profiles used for commercial purposes without our consent, or our identity purloined. We get more and more alarmed by the loss of our privacy; often justified by unseen security requirements, or by the risks of failures in and deliberate attacks on our critical infrastructures. The trustworthiness of our increasingly digitized world is at stake.

[...] We may be scared with the idea that we will have to live with a "digital shadow" that does not forget possible past little misdemeanours or indiscretions, and which can then be accessed by future employers or partners. The idea of being robbed or cheated by somebody at the other end of the world whom you have never met, without understanding how it happened and with little chance for legal redress, seems intolerable for European citizens.

[...] Uncontrolled technology development and innovation can lead the Internet and the Web to become a jungle; where trust is lost, crime and malfeasance rise and each individual is forced to defend themselves with limited tools. At the same time, policy development without awareness of technology development and trends will choke innovation and economic growth. Most importantly, if citizens feel threatened, mistrustful and increasingly hesitant towards innovative applications and services, our whole society may end up being the loser.≫

Viviane Reding,
Member of the European Commission
Responsible for Information Society and Media
In "Trust in the Information Society"
A Report of the Advisory Board RISEPTIS, Feb 2010 [64]

≪This paper investigates the notion and role of trust in modern societies as a first step towards the construction of indicators that could better inform our understanding of societal progress. Trust is commonly viewed as a proxy indicator of social capital, and a high level of trust is considered a factor that can enhance economic growth and social well-being. Indicators of trust inform about the quality of people's interactions with others, hence on their assessment of the extent to which other people in the community are perceived as potential partners rather than as rivals.

The paper, starting from the various notions and theories of trust provided in literature, discusses different definitions of trust, its various dimensions (i.e. interpersonal and institutional trust), their relation to the broader notion of social capital, and the different factors that affect it. It then overviews the measures currently used to assess trust, discussing their advantages and disadvantages. Questions assessing the degree of trust of respondents towards other people and institutions have been asked in dozens of large-scale surveys worldwide, and these data highlight systematic relations between trust and various dimensions of economic and social well-being. The paper concludes by noting the limits of available evidence and the scope for improvements through better survey design and more comparable survey questions.≫

Adolfo Morrone, Giulia Ranuzzi and Noemi Tontoranelli,
In "How Good is Trust?: Measuring Trust and its Role for the Progress of Societies"
An OECD Statistics Working Paper, Oct 2009 [57]

The aforementioned quotations are just two examples of the importance of an accurate trust and reputation management in order to build up a trustworthy information society. To achieve a real and widespread deployment of information technologies, we all have to overcome the barriers and obstacles that hinder us from fully trusting these systems. Electronic communications, eCommerce transactions and many other tasks involving computer networks cannot completely succeed without the guarantee that we will not be cheated or defrauded by malicious users. And if such behavior occurs, there should exist mechanisms to detect and, when possible, punish it.

Security in computer networks has been studied for years, developing robust and accurate solutions accomplishing high rates of confidence in the scenarios where they are applied. Traditional schemes such as IPSec, Authentication, Authorization and Accounting (AAA) [120], Public Key Infrastructures (PKI) [121], etc., have been proved to be really effective and useful when dealing with certain threats and risks in distributed and heterogeneous networks.

Nevertheless, the rapid development of the technology in the last years has led to a situation where the number of devices connected and permanently joining the Internet is amazingly increasing day by day. Thus, we have been witnesses of a quick growth of mobile and ad-hoc networks, wireless sensors and actuators ones, vehicular networks... In short, highly distributed environments where traditional solutions are not always easy to deploy.

Additionally, there are other trading and business networks, usually composed by lots of users, where most of the times service providers or sellers and service consumers or customers/clients, are absolute strangers, being eBay one of the most common examples of this kind of electronic markets. Therefore such entities need a precise mechanism assisting them when making the decision of who to interact with, or who to have a transaction with.

Recently, trust and reputation management has arisen as a novel and powerful way of dealing with this topic, drawing the attention of a number of research groups in both Industry and Academia. It is a promising field where big amounts of money have been and are being invested. SWIFT [106, 83], SENSEI [81], GRIDTRUST [75], ANTIPHISH [74], S3MS [80], OPENTC [77], PRIME [78], HUMABIO [76], REPUTATION [79], SOCIALREP [82] and TRUSTREP [84] are just some examples of EU projects that have incorporated such trust and reputation management within their proposed solutions. These collaboration efforts constitute a proof of the relevance and interest that this issue is reaching for many institutions.

This PhD Thesis, in particular, explores the current state of the art in the literature, proposing new and novel bio-inspired models as well as a comprehensive analysis of both the taxonomy of these concrete systems and the security threats that are specifically applicable to them. It also provides a generic framework for easily developing and comparing new approaches of trust and reputation management.

Thus, the specific goals we pursue with the work done in this PhD Thesis are the following:

- Analyze the current state of the art in this field.
- Identify possible deficiencies in the studied trust and reputation management proposals.
- Design and suggest innovative and original alternatives overcoming such drawbacks.
- Make a profound analysis of the intrinsic nature of these mechanisms, proposing some generic steps to be followed by every trust and/or reputation model.
- Study those threats that are specifically applicable in these environments, providing solutions in order to defeat them
- Develop a generic tool that may help other researchers to implement new trust and reputation management approaches, and easily perform a comparison amongst them.
- Liken some of our original alternatives to some of the current most representative models found in the literature.
- Survey some real and final scenarios where to deploy and apply trust and reputation management, contributing with our own proposals.

II Methodology

This PhD Thesis was conceived from its very beginning as a publications compilation one. So all the work carried out throughout it was directed towards this target.

The first step carried out as part if this PhD Thesis was to survey the current state of the art in the field of trust and reputation management in distributed systems. Such study evidenced the importance that this topic has achieved in the last few years, with a recent growth of works related to this field.

Besides, we realized that the problem had been tackled from many perspectives, ranging from fuzzy logic, to bayesian networks, or even just analytic expressions. Nevertheless, there were only very few works applying a bio-inspired approach in order to handle concepts such as trust or reputation in computer networks.

Hence, we designed and developed our first trust and reputation model, called TACS (Trust Ant Colony System), aimed to work in P2P networks. It was built upon an ant colony system algorithm, since we considered that the properties of this particular bio-inspired mechanism could fit in very well within our problem requirements.

However, TACS ended up having many input parameters, which could lead to the wrong idea that it was too complex and dependent on a concrete values configuration of those parameters in order to obtain a good performance. That was the reason why we developed META-TACS, which consisted of a genetic algorithm that allowed us to successfully verify the underlying trust model TACS using several parameters values sets.

Next we decided to enhance and adapt our model in order to be suitable for wireless sensor networks, developing this way the so-called BTRM-WSN model (Bio-inspired Trust and Reputation Model for Wireless Sensor Networks). This adaptation took into consideration the

constrained capabilities of the devices composing such networks, in terms of energy consumption, for instance. Actually two versions of the model were proposed depending on the resources restrictions of the environments where to apply the model.

After a second revision of the state of the art in order to be updated with the latest works done in this area, and once we had already developed two own proposals, we noticed that most of those models followed a similar pattern. Then we suggested a pre-standardization scheme posing several generic steps that, in our opinion, every trust and reputation model should accomplish.

Simultaneously we also realized that many authors did not take into consideration some of the most important threats that could compromise their trust and reputation models. Therefore, we analyzed which were those main risks and proposed some recommendations in order to face them.

Finally, having such security threats compilation, together with the previously standardization approach, an adequate next step was to design and develop a generic tool implementing the aforementioned steps, as well as some of those threats. This framework, called TRMSim-WSN (Trust and Reputation Models Simulator for Wireless Sensor Networks), helped us to conduct a comparison between our model BTRM-WSN and some of the current most representative trust and reputation models. This experiment allowed us to check the goodness of our proposal, not alone but in contrast to other well known solutions.

III Results

The results of this PhD Thesis are all exposed in the articles that compose it. Thus for instance, the initial literature survey has been presented in the book chapter entitled 'State of the Art in Trust and Reputation Models in P2P networks' [61], published in the Springer Handbook of Peer-to-Peer Networking.

In this work we have described some of the most representative trust and reputation models for P2P networks, explaining how each of them works, how they manage the concepts of trust and reputation or how they gather information about other peers in the network.

But we have also discussed the main common characteristics of all the exposed models, as well as the most important differences amongst them. We have analyzed their weaknesses and deficiencies and shown some real examples of systems using trust and or reputation schemes.

Regarding our first developed model TACS [7], described in the article entitled 'TACS, a Trust Model for P2P Networks' [11] and published in Elsevier Wireless Personal Communications Journal, it presents one of the first P2P network trust models based on ant colony system, where the traces of pheromone are identified with the trust that a client has on finding a benevolent server offering a certain service.

The experiments performed showed its high accuracy when selecting the most trustworthy peer in the community, having no previous information about its participants. It is able to select a suitable server in a high percentage of the cases on static networks regardless of its size, and on dynamic networks (with peers continuously entering and leaving the system) regardless of its dynamism. Oscillating networks, where the goodness of a peer might change along the time, have also been solved with TACS.

Additionally, the parameters optimization of such model has been carried out through the CHC genetic algorithm [85], demonstrating the robustness of TACS model against a wide range of working parameter values. That is, TACS seems to work reasonably well regardless the values given for its parameters (within a certain interval). This work has been accepted for publication in the Intelligent Automation and Soft Computing Journal, under the title 'META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm' [10].

Next, an enhancement of TACS model specifically oriented to wireless sensor networks (WSN), called BTRM-WSN has been published in the Springer Telecommunications System Journal with the title 'Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique' [9]. We have proposed two versions of such new model, depending on the capabilities of the WSN we are dealing with and on the security restrictions we would like to apply. Thus, the first version is lighter and more scalable while the second is more resilient and accurate.

The experiments conducted have proved that BTRM-WSN is highly scalable, accurate, light and robust. Its main deficiencies come when the percentage of malicious servers is greater than or equal to 90%. So the key factor that makes our model fail when searching the most trustworthy server through the most reputable path is that proportion of fraudulent servers.

As for the trust and reputation models pre-standardization proposal, it has been published under the title 'Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems' [62] in Elsevier Computer Standards & Interfaces, Special Issue on Information and Communications Security, Privacy and Trust: Standards and Regulations.

In this work we have given some classification recommendations of several trust and/or reputation models found in the literature according to the type of model (trust, reputation or hybrid), its scope (multi-agents systems, P2P networks, ad-hoc networks, Wireless Sensor Networks or a combination of these) and the technique used to compute trust and/or reputation values (fuzzy logic, Bayesian networks, bio-inspired algorithms, social networks or analytic expressions).

We have also described the components we consider a future trust and/or reputation standard model should have: gathering behavioral information, scoring and ranking entities, selecting the entity to interact with, performing a transaction with the selected entity and rewarding or punishing that entity.

Furthermore, we have proposed an interface aimed to supply a common layer when designing a trust and/or reputation model for distributed environments, describing its main parameters and outcomes. And finally some designing advices that we think trust and/or reputation models should include have been given. Thus, someone interested in developing such a model, will have a starting point to begin his/her design, with possible functions or steps provided with different input and output parameters and parameter values.

In turn, in the article entitled 'Security Threats Scenarios in Trust and Reputation Models for Distributed Systems' [63] and published in Elsevier Computers & Security, we have analyzed the main security threats that can be applied in most of trust and reputation schemes. Moreover, we have discussed them and suggested a possible way of tackling each one of those risks.

A complete taxonomy of those threats or attacks has been developed as well, describing several possible dimensions of an attack over trust and reputation systems, and categorizing the exposed threats according to these dimensions or properties.

Additionally, we have presented some representative trust and reputation models and shown how they deal with those threats that can be applied to them, revealing that not all the threats are paid the same attention and none of them is categorically solved.

Finally, in order to provide a generic way to implement new models and to ease the comparison with some of the most relevant ones, we developed a simulator called TRMSim-WSN [50], presented in 'TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks' [52], published in the IEEE International Conference on Communications 2009.

In this paper we have presented TRMSim-WSN, a novel trust and reputation models simulator for wireless sensor networks. As far as we know, this is one of the first simulators of these characteristics for WSNs. We have shown the generic trust and reputation models interface we have designed and developed, and explained how a new trust and reputation model can be easily added to the simulator. We have also described the main features and possibilities that TRMSim-WSN offers, and how to configure it in order to carry out customized simulations.

IV Conclusions and Future work

Internet and the World Wide Web are continuously changing our lives. Today's capability of instantaneously communicating with somebody else at the other end of the world, as well as sharing information, or even doing business through electronic transactions, was certainly unbelievable only some years ago.

Nevertheless, besides the wide range of opportunities that are offered to us and that promote the economic growth of the nations, there are unfortunately several threats that hinder such development as well.

And while traditional network security solutions have been proved to be quite effective in certain scenarios, the astonishing fast expansion of current technologies and the emergence of new ones make them not that suitable for every environment.

Trust and reputation management has been recently proposed as a novel and accurate alternative of dealing with some specific situations where a lack of information about the rest of the members composing the community could lead to a set of highly harmful attacks.

By gathering behavioral information from other participants, a user can generate its own opinion about how trustworthy or reputable another user is, when performing a specific action or task. Such mechanism is very adequate, for instance, in those scenarios where a public key infrastructure is not feasible.

Hence, in our opinion, an appropriate trust and reputation management in distributed systems might be definitely helpful in increasing the security and confidence of end users, supporting this way a prosperous information society development.

It is indeed an appealing research field, though there is still too much work to do, with very promising outcomes. We consider it a profitable topic where to invest money and resources is surely a wise move.

Thus, the work carried out as part of this PhD Thesis helped us to achieve a better knowledge about this kind of mechanisms, due to the continuous and comprehensive survey of the state of the art done throughout it. The effectiveness of the application of bio-inspired algorithms in order to manage trust and reputation has been demonstrated as well, with the development of two new and novel models in this sense: TACS and BTRM-WSN.

The pre-standardization of trust and reputation models proposal, as well as the analysis of specifically applicable security threats are two of our most relevant contributions. We expect them to serve as reference guidelines for current and future researches who want to focus in this topic.

Equally useful is the trust and reputation models simulator. In fact, it allowed us to conduct a comparison between our model BTRM-WSN and some of the current most representative ones, concluding that none of them is perfectly suitable for any situation.

Regarding our particular ongoing and future work, we are studying two concrete and real scenarios where to apply and deploy a trust and reputation mechanism, showing this way the utility of these approaches for final users. Those environments are: Identity Management Systems and Wireless Sensor and Actuator Networks. For both cases we have designed and implemented different trust and reputation models aimed to improve the security of such systems, as well as the user's impression about them.

Besides, we are currently benefiting from the advantages of fuzzy logic and fuzzy set representation in order to provide a more human-interpretable trust model, while preserving the accuracy of our already developed solutions.

We are also planning to enhance our simulator TRMSim-WSN, incorporating new trust and reputation models as well as a improving its security threats module. Our goal is for it to become one of the reference tools within this area of knowledge.

Lastly, we consider two final real scenarios that would benefit from the application of trust and reputation management. Those are Vehicular-to-Vehicular (V2V) or Car-to-Car (C2C) networks and the so called 'Internet of Things'. In fact, there is too few work done in this direction, so it constitutes an excellent starting point for a new research line.

Resumen

I Motivación y Objetivos

≪En sus primeros quince años de existencia, la World Wide Web ha tenido un profundo y transformador impacto en todas las facetas de nuestra sociedad. Mientras que Internet ya lleva 40 años con nosotros, la Web ha provocado un crecimiento exponencial de su uso, con más de 1500 millones de usuarios en todo el mundo accediendo a más de 2200 millones de páginas web. Las 'Redes Sociales' están atrayendo cada vez a más y más usuarios diferentes. Con 4000 millones de suscriptores de telefonía móvil a lo largo del planeta (hay casi 7000 millones de personas en la Tierra) y con el aumento del uso de los teléfonos móviles para conectarse a Internet, las aplicaciones y servicios web móviles se están desarrollando con rapidez.

Y aún queda mucho por venir, y que sin duda superará el mero procesamiento de información e intercambio de datos. La 'Internet de las Cosas', la Web Semántica o la Computación en la Nube están avanzando rápidamente, reflejando el dinamismo de los desarrollos de la tecnología relacionados con la digitalización del mundo que nos rodea y nuestra relación con éste. Éstos a su vez plantean cuestiones de e-Identidad y Confianza en las interacciones digitales que posibilitan.

Sin embargo, mientras observamos con asombro este nuevo mundo y nos entusiasma el uso de dispositivos previamente inimaginables, también nos sentimos perplejos y preocupados por la facilidad con la que nuestros datos pueden ser robados, nuestro perfiles usados con propósitos comerciales sin nuestro consentimiento, o nuestra identidad suplantada. Nos alarma cada vez más la pérdida de privacidad, a menudo justificada por requisitos de seguridad ocultos, o por el riesgo de fallos y ataques deliberados sobre nuestras infraestructuras críticas. La fiabilidad de nuestro mundo altamente digitalizado está en juego.

[...] Podemos temer a la idea de tener que vivir con una "huella digital" que no olvida posibles delitos menores o indiscreciones del pasado, que por tanto puede ser consultada por futuros compañeros o superiores. La idea de ser robado o estafado por alguien en el otro extremo del mundo a quien nunca hemos conocido, sin comprender cómo ha sucedido y con pocas opciones de redención legal, resulta intolerable para los ciudadanos europeos.

[...] Una innovación y desarrollo tecnológico descontrolados pueden convertir a Internet y la Web en una jungla, donde la confianza está perdida, el crimen y las malas conductas en general aumentan y cada individuo se ve obligado a defenderse a sí mismo con herramientas limitadas. Al mismo tiempo, el desarrollo de políticas desligadas de las tendencias y desarrollos de la tecnología obstruirán la innovación y el crecimiento económico. Aún más importante, si los ciudadanos se sienten amenazados, desconfiados y cada vez más dudosos hacia los servicios y las aplicaciones innovadoras, nuestra sociedad por completo puede acabar siendo la perdedora.≫

Viviane Reding,
Miembro de la Comisión Europea
Responsable para la Sociedad de la Información y los Medios de Comunicación
En "Trust in the Information Society"
Un informe del panel asesor RISEPTIS, Feb 2010 [64](traducción al castellano)

≪Este artículo investiga la noción y el papel de la confianza en las sociedades modernas como un primer paso hacia la construcción de indicadores que pudieran mejorar nuestro conocimiento acerca del progreso de la sociedad. La confianza se ve comúnmente como un indicador del poder del capital social, y un alto nivel de confianza se considera como un factor que puede mejorar el crecimiento económico y el bienestar social. Los indicadores de confianza informan acerca de la calidad de las interacciones de las personas entre ellas, y por tanto de su percepción de hasta qué punto otras personas en la comunidad son percibidas como potenciales compañeros en vez de rivales.

El artículo, comenzando por las distintas nociones y teorías de la confianza proporcionadas por la literatura, trata de las diferentes definiciones de confianza, sus variadas dimensiones (por ejemplo, confianza interpersonal e institucional), su relación con la noción más extensa de capital social, y los diferentes factores que le afectan. A continuación repasa las medidas usadas actualmente para evaluar la confianza, debatiendo acerca de sus ventajas e inconvenientes. Docenas de encuestas a gran escala a lo largo de todo el planeta han preguntado cuestiones acerca del grado de confianza de los encuestados en otras personas e instituciones, y estos datos ponen de manifiesto las relaciones sistemáticas entre la confianza y varias dimensiones del bienestar social y económico. El artículo concluye remarcando los límites de las evidencias disponibles así como el alcance de mejoras a través de un mejor diseño de las encuestas y preguntas más fácilmente comparables.≫

Adolfo Morrone, Giulia Ranuzzi y Noemi Tontoranelli,
En "How Good is Trust?: Measuring Trust and its Role for the Progress of Societies"
Un artículo de trabajo estadístico de la OCDE, Oct 2009 [57] (traducción al castellano)

Las citas anteriormente mencionadas son tan sólo un par de ejemplos de la importancia que tiene la gestión precisa de la confianza y la reputación con el objetivo de construir una sociedad de la información fiable. Si queremos conseguir un despliegue real y extendido de las tecnologías de la información, todos debemos superar las barreras y obstáculos que nos impiden confiar plenamente en dichos sistemas. Las comunicaciones electrónicas, las transacciones de comercio electrónico y muchas otras tareas que conlleven el uso de redes de ordenadores no pueden tener éxito completamente sin la garantía de que no seremos estafados ni defraudados por usuarios maliciosos. Y si tal comportamiento ocurriera, deberían existir los mecanismos necesarios para detectarlos y, si fuera posible, penalizarlos.

Durante años se ha estudiado la seguridad en redes de ordenadores, desarrollando soluciones robustas y precisas alcanzando altas cotas de confianza en aquellos escenarios en los que eran aplicadas. Esquemas tradicionales tales como IPSec, Autenticación, Autorización y Auditoría (AAA) [120], Infraestructuras de clave pública (PKI) [121], etc. han sido utilizadas como herramientas realmente eficientes y útiles a la hora de abordar ciertos riesgos y amenazas en redes distribuidas y heterogéneas.

Sin embargo, el rápido desarrollo de la tecnología en los últimos años nos ha conducido a una situación donde el número de dispositivos conectados y continuamente uniéndose a Internet está creciendo asombrosamente día a día. Así hemos sido testigos de un rápido crecimiento de las redes móviles y ad-hoc, las redes de sensores y actuadores, las redes vehiculares... En resumen, entornos altamente distribuidos donde las soluciones tradicionales no son siempre fáciles de aplicar.

Además, existen otras redes comerciales, normalmente compuestas por multitud de usuarios, donde la mayoría de las veces los proveedores de servicios o vendedores y los clientes son completos desconocidos los unos para los otros, siendo eBay uno de los ejemplos más comunes de este tipo de mercados electrónicos. Por lo tanto, dichas entidades necesitan un mecanismo preciso que los ayuda a la hora de tomar la decision sobre con quién interactuar, o con quién llevar a cabo una transacción.

Recientemente la gestión de la confianza y la reputación ha surgido como una alternativa novedosa capaz de tratar con este tema, captando la atención de numerosos grupos de investigación tanto desde la Universidad como desde la empresa privada. Se trata de un campo prometedor donde se han invertido y se siguen invirtiendo grandes sumas de dinero. SWIFT [106, 83], SENSEI [81], GRIDTRUST [75], ANTIPHISH [74], S3MS [80], OPENTC [77], PRIME [78], HUMABIO [76], REPUTATION [79], SOCIALREP [82] y TRUSTREP [84] son sólo algunos ejemplos de proyectos europeos que han incorporado dicha gestión de la confianza y la reputación dentro de las soluciones propuestas. Estos esfuerzos de colaboración constituyen una prueba de la relevancia y el interés que este campo de investigación está alcanzando para muchas instituciones.

Esta Tesis Doctoral en particular explora el estado del arte actual, proponiendo nuevos y novedosos modelos bio-inspirados así como un análisis exhaustivo tanto de la taxonomía de estos sistemas en concreto, como de las amenazas de seguridad que les son específicamente aplicables. También proporciona un marco de trabajo genérico para desarrollar y comparar fácilmente nuevas propuestas de gestión de la confianza y la reputación.

Así, los objetivos específicos que perseguimos con el trabajo realizado en esta tesis doctoral son los siguientes:

- Analizar el estado del arte actual en este campo.
- Identificar posibles deficiencias en las propuestas de gestión de confianza y reputación estudiadas.
- Diseñar y sugerir alternativas innovadoras y originales que den respuesta a las carencias previamente encontradas.
- Realizar un profundo análisis de la naturaleza intrínseca de estos mecanismos, proponiendo unos pasos genéricos a seguir por cualquier modelo de confianza y/o reputación.
- Estudiar aquellas amenazas que son específicamente aplicables en estos entornos, proporcionando soluciones para superarlas.
- Desarrollar una herramienta genérica que pueda servir a otros investigadores para implementar nuevas propuestas de gestión de la confianza y la reputación, así como realizar de forma sencilla comparaciones entre éstas.
- Comparar algunas de nuestras alternativas originales con algunos de los modelos actuales más representativos encontrados en la literatura.
- Sondear algunos escenarios finales y reales donde poder aplicar la gestión de la confianza y la reputación, aportando nuestras propias propuestas.

II Metodología

Esta Tesis Doctoral fue concebida desde el principio como una tesis basada en compilación de artículos. Por lo tanto, todo el trabajo realizado a lo largo de la misma ha sido orientado en esta dirección.

El primer paso llevado a cabo como parte de esta Tesis Doctoral consistió en sondear el estado del arte en el campo de la gestión de la confianza y la reputación en sistemas distribuidos. Dicho estudio evidenció la importancia que este tema ha alcanzado en los últimos años, con un crecimiento reciente de trabajos relacionados con este campo.

Además, nos dimos cuenta que el problema había sido abordado desde distintas perspectivas, desde lógica difusa hasta redes bayesianas, o incluso sencillamente expresiones analíticas. Sin embargo, sólo existían unos pocos trabajos aplicando un método bio-inspirado para manejar conceptos tales como confianza o reputación en redes de ordenadores.

Por lo tanto, diseñamos y desarrollamos nuestro primer modelo de confianza y reputación, llamado TACS (*Trust Ant Colony System*), orientado a redes P2P. Fue construido sobre la base de un algoritmo de colonia de hormigas, ya que consideramos que las propiedades de este mecanismo bio-inspirado en particular se podían ajustar muy bien a los requisitos de nuestro problema.

Sin embargo, TACS acabó teniendo muchos parámetros de entrada, lo que podría conducir a la idea equivocada de que era muy complejo y dependiente de una configuración específica de valores de dichos parámetros para poder conseguir un buen rendimiento. Ésa fue la razón por la que desarrollamos META-TACS, que consistió en un algoritmo genético que nos permitió probar con éxito el modelo de confianza subyacente, TACS, usando diferentes conjuntos de valores de parámetros.

A continuación decidimos mejorar y adaptar nuestro modelo para adecuarlo a las redes de sensores inalámbricas, desarrollando de esta manera el modelo conocido como BTRM-WSN (*Bio-inspired Trust and Reputation Model for Wireless Sensor Networks*). Dicha adaptación tuvo en cuenta las capacidades limitadas de los dispositivos que componen dichas redes, en términos de consumo de energía, por ejemplo. En realidad se propusieron dos versiones del modelo dependiendo de las restricciones de recursos que tuviera el entorno donde se fuera a aplicar el mismo.

Tras una segunda revisión del estado del arte para mantenernos actualizados con los últimos trabajos realizados en el área, y una vez que ya habíamos desarrollado dos modelos propios, nos dimos cuenta de que la mayoría de dichos modelos seguían un patrón similar. Entonces propusimos un esquema de pre-estandarización planteando varios pasos genéricos que, en nuestra opinión, todo modelo de confianza y reputación debería cumplir.

Al mismo tiempo también observamos que muchos autores no habían considerado algunas de las amenazas más importantes que podían comprometer sus modelos de confianza y reputación. Por lo tanto, analizamos cuáles podían ser esos riesgos principales y propusimos algunas recomendaciones para hacerles frente.

Por último, teniendo dicha recopilación de amenazas de seguridad, junto con la propuesta anterior de pre-estandarización, un siguiente paso que consideramos adecuado fue el diseño y desarrollo de una herramienta genérica que implementara los pasos anteriormente mencionados, así como alguna de las amenazas. Este entorno, llamado TRMSim-WSN (*Trust and Reputation Models Simulator for Wireless Sensor Networks*), nos ayudó a llevar a cabo una comparación entre nuestro modelo BTRM-WSN y algunos de los modelos de confianza y reputación actuales más representativos. Este experimento nos permitió comprobar la bondad de nuestra propuesta, no en solitario, sino en contrate con otras soluciones bien conocidas.

III Resultados

Todos los resultados de esta Tesis Doctoral se encuentran expuestos en los artículos que la componen. Así por ejemplo, el sondeo inicial de la literatura se ha presentado en el capítulo de libro titulado '*State of the Art in Trust and Reputation Models in P2P networks*' [61], publicado en el *Springer Handbook of Peer-to-Peer Networking*.

En este trabajo se han descrito algunos de los modelos de confianza y reputación más representativos para redes P2P, explicando cómo funciona cada uno de ellos, cómo gestionan la confianza y reputación o cómo recopilan información acerca de otros miembros de la red.

Pero también se han tratado las principales características comunes a todos los modelos expuestos, así como las diferencias más significativas entre ellos. Hemos analizado sus debilidades y deficiencias y hemos mostrado algunos ejemplos reales de sistemas que utilizan esquemas de confianza y reputación.

Sobre nuestro primer modelo desarrollado TACS [7], descrito en el artículo titulado '*TACS, a Trust Model for P2P Networks*' [11] y publicado en *Elsevier Wireless Personal Communications Journal*, éste presenta uno de los primeros modelos de confianza para redes P2P basado en sistema de colonia de hormigas, donde los rastros de feromona se identifican con la confianza que un cliente tiene en encontrar un servidor benévolo que ofrezca un determinado servicio.

Los experimentos realizados muestran su alta precisión a la hora de seleccionar el usuario más confiable de la comunidad, sin tener ninguna información previa acerca de sus participantes. Es capaz de seleccionar un servidor adecuado en un alto porcentaje de los casos para redes estáticas, independientemente de su tamaño, y en redes dinámicas (con usuarios continuamente entrando y saliendo del sistema) independientemente de su dinamismo. Las redes oscilantes, donde la bondad de un nodo puede cambiar repentinamente a lo largo del tiempo, también han sido resueltas con TACS.

Además, se ha realizado una optimización de los parámetros de dicho modelo mediante el algoritmo genético CHC [85], demostrando la robustez del modelo TACS frente a un amplio rango de valores de los parámetros de trabajo. Esto es, TACS parece funcionar razonablemente bien independientemente de los valores dados a sus parámetros (dentro de un cierto intervalo). Este trabajo ha sido aceptado para su publicación en el *Intelligent Automation and Soft Computing Journal*, bajo el título '*META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm*' [10].

A continuación, se ha publicado una mejora de TACS específicamente orientada a redes de sensores inalámbricas, denominada BTRM-WSN, en el *Springer Telecommunications System Journal* con el título de '*Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique*' [9]. Se han propuesto dos versiones de dicho modelo, dependiendo de las capacidades de la red con la que estemos tratando y de las restricciones de seguridad que queramos aplicar. Así, la primera versión es más ligera y escalable mientras que la segunda es más precisa y robusta.

Los experimentos realizados han demostrado que BTRM-WSN es altamente escalable, preciso, ligero y robusto. Sus principales deficiencias aparecen cuando el porcentaje de servidores maliciosos es mayor o igual que el 90%, por lo que el factor clave que hace que nuestro modelo falle a la hora de seleccionar el servidor más confiable a través del camino más seguro es la proporción de servidores fraudulentos.

En cuanto a la propuesta de pre-estandarización de modelos de confianza y reputación, ésta fue publicada bajo el título de '*Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems*' [62] en el *Computer Standards & Interfaces, Special Issue on Information and Communications Security, Privacy and Trust: Standards and Regulations*.

En este trabajo se han dado algunas recomendaciones de clasificación de varios modelos de confianza y/o reputación encontrados en la literatura de acuerdo con el tipo de modelo (confianza, reputación o híbrido), su ámbito (sistemas multi-agentes, redes P2P, redes ad-hoc, redes de sensores inalámbricas, o una combinación de éstas), así como la técnica usada para calcular los valores de confianza/reputación (lógica difusa, redes bayesianas, algoritmos bio-inspirados, redes sociales o expresiones analíticas).

También se han descrito los componentes que nosotros consideramos que un futuro modelo estándar de confianza y reputación debería tener: recolección de información del comportamiento de otros usuarios, valoración y ordenamiento de dichas entidades, selección del usuario con el que finalmente interactuar, llevar a cabo la transacción con la entidad seleccionada y penalizar o recompensar a dicho usuario.

Más aún, se ha propuesto una interfaz orientada a proporcionar una capa común a la hora de diseñar un modelo de confianza y/o reputación para entornos distribuidos, describiendo sus principales parámetros de entrada y valores de salida. Finalmente también se han aportado algunas sugerencias de diseño que los modelos de confianza y reputación deberían incluir en nuestra opinión. Así, alguien interesado en desarrollar un modelo de este tipo contará con un punto de partida para comenzar su diseño, con pasos y funciones junto con diferentes parámetros de entrada y salida, así como diferentes valores para dichos parámetros.

Por su parte, en el artículo titulado '*Security Threats Scenarios in Trust and Reputation Models for Distributed Systems*' [63] y publicado en el *Elsevier Computers & Security*, se han analizado las principales amenazas de seguridad que se pueden aplicar en la mayoría de los esquemas de confianza y reputación. Además, los hemos estudiado en profundidad y hemos propuesto una posible vía para abordar cada uno de estos riesgos.

También se ha desarrollado una taxonomía completa de estas amenazas o ataques, describiendo varias dimensiones posibles de u ataque a sistemas de confianza y reputación, categorizando las amenazas descritas de acuerdo con estas dimensiones o propiedades.

Adicionalmente se han presentado algunos de los modelos más representativos de confianza y reputación y se ha mostrado cómo manejan aquellas amenazas que les son aplicables, revelando que no a todas las amenazas se les presta igual atención y que ninguna de ellas está completamente resuelta.

Finalmente, y con el objetivo de proporcionar una forma genérica de implementar nuevos modelos y de facilitar la comparación con algunos de los más relevantes, se desarrolló el simulador llamado TRMSim-WSN [60], presentado en el artículo titulado '*TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks*' [62], publicado en la *IEEE International Conference on Communications 2009*.

En este artículo se ha presentado TRMSim-WSN, un simulador novedoso de modelos de confianza y reputación para redes de sensores inalámbricas. Hasta donde nosotros sabemos, éste es uno de los primeros simuladores de estas características para redes de sensores inalámbricas. Se ha mostrado la interfaz genérica de modelos de confianza y reputación que hemos diseñado y desarrollado, y se ha explicado cómo se puede incluir fácilmente un nuevo modelo en el simulador. También hemos descrito las principales características y posibilidades que TRMSim-WSN ofrece, y cómo configurarlo para llevar a cabo simulaciones personalizadas.

IV Conclusiones y Trabajo futuro

Internet y la World Wide Web están continuamente cambiando nuestras vidas. La capacidad actual de comunicación instantánea con alguien en el otro extremo del planeta, compartir información fácilmente, o incluso hacer negocios a través de transacciones electrónicas, era ciertamente impensable hace tan sólo unos pocos años.

Sin embargo, además del amplio rango de oportunidades que se nos ofrecen y que impulsan el crecimiento económico de las naciones, desafortunadamente existen también varias amenazas que frenan dicho desarrollo.

Y mientras que se ha demostrado que las soluciones tradicionales de seguridad en redes son bastante eficaces en determinados escenarios, la asombrosamente rápida expansión de las tecnologías actuales, así como la aparición de otras nuevas, hace que estas soluciones no siempre sean adecuadas en cualquier entorno.

La gestión de la confianza y la reputación ha sido propuesta recientemente como una alternativa novedosa y precisa para tratar con ciertas situaciones específicas en las que la falta de información acerca del resto de miembros que componen la comunidad puede conducir a un conjunto de ataques altamente dañinos.

Mediante la recopilación de la información del comportamiento de otros participantes, un usuario puede generarse su propia opinión acerca de cuán confiable es otro miembro a la hora de realizar una acción o tarea en concreto. Dicho mecanismo es muy adecuado, por ejemplo, en aquellos escenarios donde una infraestructura de clave pública no es viable.

Por tanto, en nuestra opinión, una gestión apropiada de la confianza y la reputación en sistemas distribuidos puede resultar definitivamente útil para aumentar la seguridad y confianza de los usuarios finales, apoyando así un próspero desarrollo de la sociedad de la información.

Verdaderamente se trata de un campo de investigación muy interesante, aunque aún queda mucho trabajo por hacer, con resultados muy prometedores. Consideramos que se trata de un asunto rentable donde la inversión de dinero y recursos es sin duda una sabia decisión.

Así, el trabajo realizado como parte de esta Tesis Doctoral ha permitido profundizar en el conocimiento de este tipo de mecanismos, debido al continuo y exhaustivo sondeo del estado del arte realizado a lo largo de la misma. También se ha demostrado la efectividad de la aplicación de algoritmos bio-inspirados para gestionar la confianza y la reputación, con el desarrollo de dos nuevos modelos novedosos en este sentido: TACS y BTRM-WSN.

La propuesta de pre-estandarización de modelos de confianza y reputación, así como el análisis de las amenazas de seguridad específicamente aplicables son dos de nuestras contribuciones más relevantes. Esperamos que puedan servir de guías de referencia para investigadores actuales y futuros que deseen centrarse en este tema.

Igualmente útil es el simulador de modelos de confianza y reputación. De hecho, nos permitió realizar una comparación del modelo BTRM-WSN y algunos de los modelos actuales más representativos, concluyendo que ninguno es perfectamente apropiado para cualquier situación.

Con respecto a nuestro trabajo actual y futuro, estamos estudiando dos escenarios reales concretos donde aplicar y desplegar un mecanismo de confianza y reputación, mostrando así la utilidad de estas alternativas para los usuarios finales. Dichos entornos son: sistemas de gestión de identidades y redes inalámbricas de sensores y actuadores. Para ambos casos hemos diseñado e implementado sendos modelos de confianza y reputación encaminados a mejorar la seguridad de dichos sistemas, así como la percepción del usuario acerca de éstos.

Además, actualmente nos estamos beneficiando de las ventajas de la lógica difusa y la representación de los conjuntos difusos con el objetivo de proporcionar un modelo de confianza que sea más fácilmente interpretable por los humanos, al mismo tiempo que se preserva la precisión de las soluciones que ya hemos desarrollado.

También estamos planeando mejorar nuestro simulador TRMSim-WSN, incorporando nuevos modelos de confianza y reputación, así como mejorando el módulo referente a las amenazas de seguridad. Nuestro objetivo es que se convierta en una de las herramientas de referencia dentro de este área de conocimiento.

Por último, consideramos otros dos escenarios finales reales que se podrían beneficiar de la aplicación de una gestión de la confianza y reputación. Éstos son las redes Car-to-Car (C2C) o Vehicular-to-Vehicular (V2V), y el conocido como 'Internet de las Cosas' (*Internet of Things*). De hecho, hay muy poco trabajo hecho en esta dirección, por lo que constituye un excelente punto de partida para una nueva línea de investigación.

Publications composing

the PhD Thesis

1

State of the Art in Trust and Reputation Models in P2P networks

Title:	State of the Art in Trust and Reputation Models in P2P networks
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Book chapter
Series:	Handbook of Peer-to-Peer Networking
Publisher:	Springer
Editors:	Xuemin Shen, Heather Yu, John Buford, Mursalin Akon
Pages:	761-784
Year:	2010
Month:	March
ISBN:	978-0-387-09750-3
DOI:	http://dx.doi.org/10.1007/978-0-387-09751-0_26
State:	Published

Table 1: State of the Art in Trust and Reputation Models in P2P networks

State of the Art in Trust and Reputation Models in P2P networks

Félix Gómez Mármol and Gregorio Martínez Pérez

Abstract Ensuring security in a distributed environment such as P2P networks is a critical issue nowadays. Nevertheless, it is in those kind of scenarios in which entities can enter or leave the community whenever they want, where traditional security schemes can not always be applied. Specifically, the use of a PKI (Public Key Infrastructure) may be unacceptable within highly distributed systems. Therefore, modeling concepts like trust and reputation may result very helpful and useful when trying to gain a certain level of security and confidence among inter-operating entities. Thus, this chapter presents a review of some of the most representative trust and reputation models for P2P networks, discussing their main characteristics and also their weaknesses and deficiencies. Open issues and challenges associated with them will be also covered.

1 Introduction

P2P networks have been widely spread in the recent years. We find them in many scenarios and applications, from file sharing systems to even military environments. They have helped to improve and increase the accessibility to the information as well as the opportunity of communication or the performance of electronic transactions, for instance.

Consequently, many research works have been done and are still in progress in order to improve their robustness, applications, scalability or security among other features.

And it is just security one of the most critical issues when dealing with this kind of networks. Due to its intrinsic nature, where every peer can enter and leave the network whenever it wants and where most of the times there is no central entity controlling the community, many security threats have arisen during the development of P2P networks, since malicious behaviors must be managed by all the participants themselves.

X. Shen et al. (eds.), *Handbook of Peer-to-Peer Networking*, 761
DOI 10.1007/978-0-387-09751-0_26, © Springer Science+Business Media, LLC 2010

Some of these threats have their origin in the lack of knowledge a peer has (specially newcomers, who join the network for the first time) about the other peers belonging to the community, and their current and past behavior.

Without loss of generality we can assume a P2P network where some nodes offer certain services or carry out certain tasks and other peers apply for those services or tasks. In such a situation when a newcomer enters the network she will probably not know any or most of the rest of peers already joined, and vice versa, i.e., they will recognize her as a stranger.

In the last few years trust and reputation management has been proposed as a novel approach in order to overcome this problem. In fact, by counting with a system where every node could ask for the reputation hold by a peer in the community, more accurate and intelligent choices could be done when deciding which peer to interact with since, in many cases, fraudulent interactions could be avoided.

And this reputation values come from the trust relationships established among peers who have had interactions in the past and have evaluated and rated each others. However, the application of these trust and reputation schemes also has its security threats. For instance, a collusion could be formed among a set of peers who rated themselves with the maximum value, increasing this way their reputation in the community.

Therefore, in this work we are going to present and describe a set of some of the most representative trust and reputation models for P2P networks currently published in the literature and we will describe how each one of them deals with these and other problems.

The rest of the paper is organized as follows. The trust and reputation models review will be done in Section 2, while Section 3 will show a global analysis of all the previously described models, talking about common features and deficiencies detected in most of them, extracting the main steps followed by the majority of those models and presenting common issues related to trust and reputation management and how the current proposals tackle such problems. Finally Section 4 will present some conclusions of our analysis, as well as some future research directions.

2 Trust and Reputation Models

CuboidTrust

CuboidTrust [1] is a global reputation-based trust model for peer to peer networks which builds four relations among three trust factors including contribution of the peer to the system, peer's trustworthiness (in reporting feedbacks) and quality of resource. It applies power iteration in order to compute the global trust value of each peer.

A cuboid is built where each small cube with coordinates (x, y, z), denoted by $P_{x,y,z}$, represents the quality of resource z stored at peer y rated by peer x. Once peer

x has downloaded resource z from peer y, it may rate it as positive ($P_{x,y,z} = 1$) if the downloaded resource z is considered authentic, or negative ($P_{x,y,z} = -1$) if the downloaded resource z is considered inauthentic or the download is interrupted for any reason.

Two coefficient matrixes are defined, E and D, whose elements are:

$$D_{ij} = avg(P_{j,i,\cdot}) \in [-1,1] \qquad 1 \leq i \leq M,\ 1 \leq j \leq M$$

$$E_{ij} = avg(P_{i,\cdot,j}) \in [-1,1] \qquad 1 \leq i \leq M,\ 1 \leq j \leq N$$

where M is the number of peers, N is the number of distinct resources, $P_{j,i,\cdot}$ represents the vector with $X = j$ and $Y = i$ in the cuboid, and $P_{i,\cdot,j}$ represents the vector with $X = i$ and $Z = j$ in the cuboid. Therefore, each element D_{ij} stores the average score of peer i rated by peer j while each element E_{ij} stores the average score of resource j rated by peer i.

The first relation among the three mentioned factors effectively combines two of them (the trustworthiness score and the contribution score) as it is shown in the following equation:

$$C_i = \sum_{j=1}^{M} (D_{ij} \times T_j) \qquad 1 \leq i \leq M$$

where T_j represents the trustworthiness of peer j, and C_i reflects the contribution of peer i to the system by considering the experiences of all peers belonging to the network.

The second relation combines the quality of a resource with the trustworthiness of a peer as follows:

$$T_i = \sum_{j=1}^{N} (E_{ij} \times Q_j) \qquad 1 \leq i \leq M$$

where Q_j is the quality score of resource j, and T_i actually represents the trustworthiness of peer i.

The third relation takes into consideration the trustworthiness of the peers in the system as well as the quality of the resources being exchanged in the following manner:

$$Q_i = \sum_{j=1}^{N} (E_{ij}^T \times T_j) \qquad 1 \leq i \leq N$$

where E_{ij}^T is the element with coordinates (i,j) of the transposition of matrix E, denoted as E^T, and represents the average score of resource i rated by peer j. T_j indicates the trustworthiness of peer j and Q_i is the quality of resource i in the system.

And the last relation combines the contribution and the trustworthiness as follows:

$$T_i = \sum_{j=1}^{M} (D_{ij}^T \times C_j) \qquad 1 \leq i \leq M$$

where D_{ij}^T represents the average score of peer j rated by peer i, C_j indicates the contribution score of peer j, and T_i reflects the trustworthiness of peer i.

Combining now these four relations we can obtain:

$$\begin{array}{l} C = D \times T = D \times E \times Q = D \times E \times E^T \times T = \\ D \times E \times E^T \times D^T \times C = (D \times E) \times (D \times E)^T \times C \end{array} \tag{1}$$

And applying power iteration over equation (1), we get

$$C^{(k)} = R \times C^{(k\ \ 1)} = R^{(1)} \times C^{(k\ \ 2)} = \cdots = R^{(k\ \ 1)} \times C^{(1)} = R^{(k)} \times C^{(0)}$$

where $R = (D \times E) \times (D \times E)^T$, and $C^{(k)}$ represents the global contribution score of every peer in the system, after k iterations.

Another combination of the four relations can give us the following expression:

$$\begin{array}{l} Q = E^T \times T = E^T \times D^T \times C = E^T \times D^T \times D \times T = \\ E^T \times D^T \times D \times E \times Q = (D \times E)^T \times (D \times E) \times Q \end{array} \tag{2}$$

And applying again power iteration over equation (2), we get

$$Q^{(k)} = S \times Q^{(k\ \ 1)} = S^{(1)} \times Q^{(k\ \ 2)} = \cdots = S^{(k\ \ 1)} \times Q^{(1)} = S^{(k)} \times Q^{(0)}$$

where $S = (D \times E)^T \times (D \times E)$, and $Q^{(k)}$ represents the global quality score of every resource in the system, after k iterations.

In CuboidTrust the global contribution score of a peer in the system represents the global trust value of that peer.

EigenTrust

One of the most cited and compared trust models for P2P networks is EigenTrust [9]. It assigns each peer a unique global trust value in a P2P file-sharing network, based on the peer's history of uploads, achieving thus a decreasing in the number of downloads of inauthentic files.

The local trust value s_{ij} is defined as follows:

$$s_{ij} = sat(i, j) - unsat(i, j)$$

where $sat(i, j)$ is the number of satisfactory transactions peer i has had with peer j (equally, $unsat(i, j)$ is the number of unsatisfactory transactions).

A probability distribution $\mathbf{p}$ (with $p_i \in [0, 1]$) is defined over pre-trusted peers. For instance, if some set of peers P are previously known to be trusted, then $p_i = 1/|P|$ if $i \in P$, and $p_i = 0$ otherwise. With a definition like this, a normalized local trust

value $c_{ij} \in [0, 1]$ can be defined as:

$$c_{ij} = \begin{cases} \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)} & \text{if } \sum_j \max(s_{ij},0) \neq 0 \\ p_j & \text{otherwise} \end{cases}$$

Therefore, if a peer does not trust anybody or does not know anybody, she will choose to trust the pre-trusted peers.

The global reputation of peer i is defined in EigenTrust in terms of the local trust values assigned by other peers to peer i, weighted by the global reputation of the assigning peers. So the aggregation of normalized local trust values is computed as:

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

being t_{ik} the amount of trust that peer i places in peer k based on asking his friends. Let C be defined as the matrix $[c_{ij}]$ and $\mathbf{t_i}$ as the vector containing the values t_{ik}, then we have that $\mathbf{t_i} = C^T \mathbf{c_i}$.

Peer i may wish to ask her friends' friends in order to get a wider view. In such a situation we would have that $\mathbf{t_i}^{(2)} = (C^T)^2 \mathbf{c_i}$. If she continues in this way (i.e., $\mathbf{t_i}^{(n)} = (C^T)^n \mathbf{c_i}$), she will achieve a complete view of the network after n iterations.

The trust vector $\mathbf{t_i}$ will converge to the same vector for every peer i, if n is large enough. In other words, it will converge to the left principal eigenvector of C. Namely, $\mathbf{t}$ is a global trust vector in this model whose elements, t_j, quantify how much trust the system as a whole places in peer j.

Finally, in order to avoid malicious collectives in P2P networks, the global trust value is re-defined as:

$$\mathbf{t}^{(k+1)} = (1-a) C^T \mathbf{t}^{(k)} + a\mathbf{p}$$

where a is some constant less than 1 and $\mathbf{t}^{(0)} = \mathbf{p}$.

BNBTM

In BNBTM [20] multidimensional application specific trust values are used and each dimension is evaluated using a single Bayesian network.

Beta probability distribution functions are used in order to represent the distribution of trust values according to interaction history as follows:

$$\tau_i = \frac{\alpha_i}{\alpha_i + \beta_i}, \quad (i \in \{G, L, C\}, \bar{i} \in \{\bar{G}, \bar{L}, \bar{C}\})$$

where $\alpha_i = r_i + 1$ and $\beta_i = s_i + 1$, r_i and s_i are the number of interactions with outcome i and $\bar{i}$, respectively, and G means shipping goods as described, L means shipping lower quality goods, C means not shipping any good, and $\bar{G}$, $\bar{L}$ and $\bar{C}$ the opposite.

Since $\beta_G = \alpha_L + \alpha_C - 1$, then

$$\tau_i = \frac{\alpha_i}{\sum_{j\epsilon\{G,L,C\}} \alpha_j - 1} = \frac{r_i + 1}{\sum_{j\epsilon\{G,L,C\}} r_j + 2}$$

The trust value is then obtained by normalizing this beta function:

$$P_i = \frac{\tau_i}{\sum_{j\epsilon\{G,L,C\}} \tau_j + 2}$$

This trust value P_i is considered to be reliable if its corresponding confidence γ_i is greater than a certain threshold θ_γ.

An entity can estimate another's reputation according to the received ratings about the latter. A Bayesian network is constructed to perform the estimation for each dimension of trust (G, L, C). This eases both to extend the model to involve more dimensions of trust and to combine Bayesian networks to form an opinion about the overall trustworthiness of an entity. Each entity can evaluate its peers according to its own criteria, and the dynamic characteristics of criteria and of peer behavior can be captured by updating Bayesian networks.

After obtaining reputation values for a seller agent, the buyer normalizes them to get (P'_G, P'_L, P'_C) which are then used to calculate the utility of dealing with the seller. Buyers can select a utility function $U_R(x)$ according to their attitude to risk, where R is the risk tolerance. Suppose the price of an item is q, its intrinsic value is v, and the intrinsic value of a lower quality item is v'. Then, the expected utility can be calculated as:

$$EU = P'_G \times U_R(v-q) + P'_L \times U_R(v'-q) + P'_C \times U_R(-q)$$

$EU > 0$ means that dealing with the seller is worthy, otherwise is too risky.

GroupRep

GroupRep [18] is a model where trust relationships are classified in three levels: trust relationships between groups, between groups and peers and only between peers.

Group i trust in group j, $Tr_{G_iG_j}$, is computed as follows:

$$Tr_{G_iG_j} = \begin{cases} \frac{u_{G_iG_j} \quad c_{G_iG_j}}{u_{G_iG_j} + c_{G_iG_j}} & \text{if } u_{G_iG_j} + c_{G_iG_j} \neq 0 \\ Tr_{G_iG_j}^{reference} & \text{if } u_{G_iG_j} + c_{G_iG_j} = 0 \text{ and } \exists\, Trust_{G_iG_j}^{path} \\ Tr_{G_iG_{strange}} & \text{otherwise} \end{cases}$$

Where $u_{G_iG_j} \geq 0$ and $c_{G_iG_j} \geq 0$ are the utility and the cost, respectively, that nodes of group j have given to nodes in group i.

Given a set of reference paths between G_i and G_j, the most trustworthy reference path is that one including the most trustworthy group. Therefore, $Tr_{G_iG_j}^{reference}$ is defined as the minimum trust value along the most trustworthy reference path.

The trust value for strange groups (groups that have not had transactions with G_i) is calculated with this expression:

$$Tr_{G_iG_{strange}} = \begin{cases} \frac{u_{G_iG_{strange}} \; c_{G_iG_{strange}}}{u_{G_iG_{strange}} + c_{G_iG_{strange}}} & \text{if } u_{G_iG_{strange}} + c_{G_iG_{strange}} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Trust value of group G_i about peer j, $Tr_j^{G_i}$, is defined as follows:

$$Tr_j^{G_i} = \begin{cases} \frac{u_j^{G(j)} \; c_j^{G(j)}}{u_j^{G(j)} + c_j^{G(j)}} & \text{if } u_j^{G(j)} + c_j^{G(j)} \neq 0 \wedge j \in G_i = G(j) \\ Tr_{strange}^{G(j)} & \text{if } u_j^{G(j)} + c_j^{G(j)} = 0 \wedge j \in G_i = G(j) \\ \min\{Tr_{G_iG(j)}, Tr_j^{G(j)}\} & \text{if } j \notin G_i \end{cases}$$

$$Tr_{strange}^{G(j)} = \begin{cases} \frac{u_{strange}^{G(j)} \; c_{strange}^{G(j)}}{u_{strange}^{G(j)} + c_{strange}^{G(j)}} & \text{if } u_{strange}^{G(j)} + c_{strange}^{G(j)} \neq 0 \\ 0 & \text{if } u_{strange}^{G(j)} + c_{strange}^{G(j)} = 0 \end{cases}$$

Where $G(j)$ is the group peer j belongs to, and $u_j^{G(j)} \geq 0$ and $c_j^{G(j)} \geq 0$ are the utility and cost, respectively, that peer j gives to other peers in group $G(j)$.

Finally, the trust value between node i and j is expressed as:

$$Tr_{ij} = \begin{cases} \frac{u_{ij} \; c_{ij}}{u_{ij} + c_{ij}} & \text{if } u_{ij} + c_{ij} \neq 0 \\ Tr_j^{G(i)} & \text{if } u_{ij} + c_{ij} = 0 \end{cases}$$

AntRep

AntRep [19] is a novel model where reputation evidences are distributed over a P2P network, based on the swarm intelligence paradigm [10]. Specifically, authors propose the use of an ant system [2, 3] for building trust relationships in P2P networks efficiently.

In AntRep each peer has a Reputation Table (RT) which is very similar with the distance-vector routing table [15], but differs from: (i) each peer in the RT corresponds to one reputation content; (ii) the metric is the probability of choosing each neighbor as the next hop instead of the hop count to destinations.

There are two kinds of forward ants sent out for a particular reputation:

1. Unicast ants are sent out to the neighbor with the highest probability in the reputation table.
2. Broadcast ants are sent out when there is no preference to neighbors. This happens either when no path to the reputation has been explored or the information the node has is outdated.

Once forward ants find the required evidence (reputation information), a backward ant is generated. When the backward ant visits each node i, it updates the reputation table at the same time. This updating is carried out due to the next reinforcement rule:

$$P_i(t) = \frac{[\tau_i(t)]^\alpha [\eta_i(t)]^\beta}{\sum_{j \in N} [\tau_j(t)]^\alpha [\eta_j(t)]^\beta}$$

where η_i is the goodness value of the link between current node and its neighbor node i. τ_i is the pheromone deposit, which is defined as follows: if at time $t + \Delta t$, current node receives a backward ant from node i, then

$$\tau_i(t + \Delta t) = f(\tau_i(t), \Delta t) + \Delta p$$
$$\tau_j(t + \Delta t) = f(\tau_j(t), \Delta t), \ j \in N, \ j \neq i$$

where $\Delta p = \frac{k}{f(c)}$, being $k > 0$ a constant, $f(c)$ a nondecreasing function of cost c and c could be any parameter revealing the information of evidence or the scenario of current network. $f(\tau_i(t), \Delta t)$ is the pheromone evaporation function:

$$f(\tau_i(t), \Delta t) = \frac{\tau_i(t)}{e^{\Delta t / k}}$$

Finally, α and β are constants varied in different network environments.

Another function of the pheromone is to decide when to send out broadcast forward ants. When node k receives a request at time t, it first searches if there is an entry for the desired evidence. If no such entry exists, it simply sends out broadcast ants. Otherwise, it finds the one with the highest probability.

Semantic Web

A model is presented in [24] where the trustworthiness between two agents is computed by searching all the paths that connect themselves; next, for each path the ratings associated with each edge are multiplied and finally all the paths are added (normalizing that aggregation).

Let N be the number of paths from agent P to agent Q. D_i denotes the number of steps between P and Q on the i_{th} path. The set of Q's friends or neighbors is called M. m_i denotes Q's immediate friend or neighbor on the i_{th} path. w_i denotes weight of the i_{th} path. The weight of each path is calculated as follows (giving a higher weight to shorter paths):

$$w_i = \frac{\frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}}$$

If agent P and agent Q are friends, $P \rightarrowtail Q$, or neighbors, $P \leftrightarrow Q$, then P's trust in Q, $T_{P \to Q}$, can be got directly. Otherwise

$$T_{P\to Q} = \sum_{i=1}^{N} \frac{T_{m_i\to Q} \times \prod_{i\rightarrowtail j \cup i\leftrightarrow j} R_{i\to j} \times \frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}} = \sum_{i=1}^{N} T_{m_i\to Q} \times \prod_{i\rightarrowtail j \cup i\leftrightarrow j} R_{i\to j} \times w_i$$

Where the reliable factor $R_{i\to j}$ denotes to which degree i believes in j's words or opinions.

Global Trust

In [23] a trust vector is created for each node i, $\overrightarrow{TT_i} = \alpha \cdot \overrightarrow{DT_i} + (1-\alpha) \cdot \overrightarrow{Rep_i}$, being $\overrightarrow{Rep_i}$ the addition of indirect trust $\overrightarrow{Rep_i} = \overrightarrow{IDT_i^1} + \overrightarrow{IDT_i^2} + \cdots + \overrightarrow{IDT_i^m}$ and where IDT_{ij}^m represents the path of m steps between i and j with the greatest indirect trust value, and $\overrightarrow{DT_i}$ is i's vector of direct trust values.

Therefore, IDT_{ij}^m is computed as follows:

$$IDT_{ij}^m = (R_{ik_1} \otimes R_{k_1k_2} \otimes \cdots \otimes R_{k_{m-1}k_m}) \cdot DT_{k_mj}$$

Since $A \otimes B = \min(A,B)$, in order to get the indirect trust IDT_{ij}^m what is done is to keep the minimum recommendation $R_{k_pk_q}$ along the path of m steps between i and j and multiply it by the direct trust value of the last node of the path about j, i.e. DT_{k_mj}.

The reputation of j from i's point of view is computed as:

$$Rep_{ij} = IDT_{i1j} \oplus IDT_{i2j} \oplus \cdots \oplus IDT_{irj}$$

And since $A \oplus B = \max(A,B)$, i obtains the reputation of j by keeping the maximum indirect trust value of its r acquaintances about j.

PeerTrust

PeerTrust [22] is a reputation-based trust supporting framework, which includes a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system. It has two main features: on the one hand it introduces three basic trust parameters and two adaptive factors in computing trustworthiness of peers, namely, feedback a peer receives from other peers, the total number of transactions a peer performs, the credibility of the feedback sources, transaction context factor and the community context factor. On the other hand, it defines a general trust metric to combine these parameters.

Let $I(u,v)$ denote the total number of transactions performed by peer u with peer v, $I(u)$ denote the total number of transactions performed by peer u with all other peers, $p(u,i)$ denote the other participating peer in peer u's i_{th} transaction, $S(u,i)$ denote the normalized amount of satisfaction peer u receives from $p(u,i)$ in its i_{th}

transaction, $Cr(v)$ denote the credibility of the feedback submitted by v, $TF(u,i)$ denote the adaptive transaction context factor for peer u's i_{th} transaction, and $CF(u)$ denote the adaptive community context factor for peer u. The trust value of peer u denoted by $T(u)$, is defined in equation (3).

$$T(u) = \alpha \cdot \sum_{i=1}^{I(u)} S(u,i) \cdot Cr(p(u,i)) \cdot TF(u,i) + \beta \cdot CF(u) \tag{3}$$

where α and β denote the normalized weight factors for the collective evaluation and the community context factor, respectively.

PeerTrust proposes two different credibility measures. The first one is to use a function of the trust value of a peer as its credibility factor so feedback from trustworthy peers are considered more credible and, thus, weighted more than those from untrustworthy peers. This credibility measurement is defined in equation (4).

$$Cr(p(u,i)) = \frac{T(p(u,i))}{\sum_{j=1}^{I(u)} T(p(u,j))} \tag{4}$$

The second credibility measure is for a peer w to use a personalized similarity measure between itself and another peer v to weight the feedback by v on any other peers. Let $IS(v)$ denote the set of peers that have interacted with peer v, the common set of peers that have interacted with both peer v and w, denoted by $IJS(v,w)$, is $IS(v) \cap IS(w)$. This measure is defined in equation (5).

$$Cr(p(u,i)) = \frac{Sim(p(u,i),w)}{\sum_{j=1}^{I(u)} Sim(p(u,j),w)} \tag{5}$$

where

$$Sim(v,w) = 1 - \sqrt{\frac{\sum_{x \in IJS(v,w)} \left(\frac{\sum_{i=1}^{I(x,v)} S(x,i)}{I(x,v)} - \frac{\sum_{i=1}^{I(x,w)} S(x,i)}{I(x,w)} \right)^2}{|IJS(v,w)|}}$$

About the transaction context factor $TF(u,i)$, several transaction contexts such as the size, the category, or time stamp of the transaction, can be incorporated in the metric so that the feedback for larger, more important, and more recent transactions can be assigned a higher weight than those for other transactions.

The incentive problem of reputation systems such as those where users will not receive rating information without paying or providing ratings, can be addressed in PeerTrust by building incentives or rewards into the trust metric through community context factor for peers who provide feedback to others. For example, let $F(u)$ denote the total number of feedback peer u gives to others. Then, a community context factor measure could be defined as follows:

$$CF(u) = \frac{F(u)}{I(u)}$$

PATROL-F

In the case of *PATROL-F* [17] the model incorporates many important concepts in order to compute a peer reputation, such as: direct experiences and reputation values, the node credibility to give recommendations, the decay of information with time based on a decay factor, first impressions and a node system hierarchy. It uses three fuzzy subsystems.

First one is used to set the importance factor of an interaction and related decisions. To decide and choose which data is critical or indispensable, or which data is needed more quickly, is a concept close to humans that fuzzy logic can model.

Moreover, there is the region of uncertainty where an entity is not sure whether to trust or not (when the reputation of a host is greater than the absolute mistrust level ϕ, but less than the absolute trust level θ). It is in this region where the fuzzy techniques are effectively applied.

Finally, for the result of interaction (RI) value, fuzzy logic can be used to capture the subjective and humanistic concept of "good" or "better" and "bad" or "worse" interaction. RI becomes the result of several concepts effectively combined to produce a more representative value. The decay factor τ is calculated based on the difference of a host's values of RIs between successive interactions.

Trust Evolution

Authors of [21] present a trust evolution model for P2P networks where trust relationships among peers are automatically built, in which two critical dimensions, experience and context, are taken into account.

The model applies the real numbers within the interval $[0, 1]$ in order to quantify two kinds of trust: direct trust and recommendation trust.

Direct trust (DT) between two peers is computed using the last n interactions between those entities. An experience vector is used, assigning a weight $W_i > 0$ to each interaction, where $\forall\, i, j \quad i < j \Rightarrow W_i > W_j$, that is, the effect of an experience on DT is regressive with the time.

The content of each cell of the experience vector, d_i is computed as follows. On the one hand, a quality tuple is used in order to represent the quality standards of the interactions, $< q_1, q_2, \ldots, q_n >$. On the other hand we have the results of the interaction got by the truster, $< r_1, r_2, \ldots, r_n >$. Thus, the following rules are applied in order to compare the results and the standards.

If q_i is quantitative and better with the bigger value, then

$$d_i = \begin{cases} \frac{r_i}{q_i} & \text{if } r_i \leq q_i \\ 1 & \text{if } r_i > q_i \end{cases}$$

If q_i is quantitative and better with the smaller value, then

$$d_i = \begin{cases} \frac{r_i}{q_i} \text{ if } r_i \geq q_i \\ 1 \text{ if } r_i < q_i \end{cases}$$

If q_i is boolean, then $d_i = 1 - (r_i \oplus q_i)$, where $\oplus$ denotes the exclusive-OR relationship.

If q_i is an application-dependable type, the application should provide a method to map the comparison of r_i and q_i to the value in $[0,1]$.

Therefore, the direct trust of the truster A towards the trustee B can be calculated as follows:

$$DT_B^A = \frac{\sum_{i=1}^{n} W_i \cdot d_i}{\sum_{i=1}^{n} W_i}$$

On the other hand, for a particular recommendation from peer B toward C, peer A evaluates it as follows. It first calculates Q:

$$Q = \frac{M \cdot |DT_C^A - DT_C^B|}{\sqrt{M \cdot DT_C^B \cdot (1 - DT_C^B)}}$$

where M is the number of A's experiences about C. Then it calculates k_1 and k_2 according to $k_i = \int_{-\infty}^{\frac{\alpha_i}{2}} \frac{e^{-x^2/2}}{\sqrt{2\pi}}$, $\alpha_0 > \alpha_1 > 0$. Next, it compares Q with k_i to get the result e, where $e = satisfied$ means that A's experience is consistent with the recommendation from B.

$$e = \begin{cases} satisfied \text{ or } + & \text{if } Q \leq k_0 \\ uncertain \text{ or } / & \text{if } k_0 < Q < k_1 \\ unsatisfied \text{ or } - & \text{if } Q \geq k_1 \end{cases}$$

Each peer holds a recommendation vector for a particular recommender to record the recommenders' behaviors. Each cell of that vector, C_i, has a weight w_i, equal to the weight used in the experience vector. Finally, the recommendation trust of B from A's point of view, RT_B^A is computed as follows:

$$RT_B^A = \begin{cases} RT_B^A + \frac{\ln(RT_B^A)}{\ln(W) \ \ln(RT_B^A)} \cdot \left(\frac{\sum_{C_i=+} w_i}{\sum_{i=1}^{n} w_i} \right) \cdot (1 - RT_B^A) & \text{if } e = satisfied \\ RT_B^A & \text{if } e = uncertain \\ RT_B^A - \frac{\ln(1 \ \ RT_B^A)}{\ln(W) \ \ln(1 \ \ RT_B^A)} \cdot \left(\frac{\sum_{C_i=-} w_i}{\sum_{i=1}^{n} w_i} \right) \cdot RT_B^A & \text{if } e = unsatisfied \end{cases}$$

where $W = \alpha_0 \cdot \left(\sum_{C_i \neq /} w_i \right) + \alpha_1 \cdot \left(\sum_{C_i = /} w_i \right)$.

Finally, the combination trust of peer A toward peer C is computed as:

$$CT_C^A = \frac{DT_C^A + \sum_{i=1}^{n} RT_{R_i}^A \cdot RT_{R_i}^A \cdot DT_C^{R_i}}{1 + \sum_{i=1}^{n} RT_{R_i}^A}$$

TDTM

In TDTM [25] the ant colony system is also applied. On the one hand it identifies the pheromone and the trust, and on the other hand it identifies the heuristic and the distance between two nodes. In this model the level of trust increases when an interaction takes place and it slows down when there are no interactions for a long time.

Trust-pheromone between node i and j at time $t+1$ is defined as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \sigma\tau_{ij}(t)$$

Where ρ is the trust dilution factor and $\sigma\tau_{ij}(t)$ is the additional intensity at each inter-operation between entities, defined as:

$$\sigma\tau_{ij}(t) \begin{cases} \frac{1}{\frac{1}{1-\tau_{ij}(t)}+1} & \text{if } i \text{ and } j \text{ interact at time } t \\ 0 & \text{otherwise} \end{cases}$$

Given a certain threshold R, and being $p_{ij}(t)$ the trust-degree between i and j at time t, if $p_{ij}(t) > R$, then entities i and j have enough trust-degree, so they can validate their certificate each other, not otherwise. Thus, in this model every entity has a local trust vector $p_i(t) = (p_{i1}(t), p_{i2}(t), \ldots, p_{in}(t))$ (reflecting the trust degree of peer i with every other one in the network) and the existence of a PKI is assumed.

TACS

TACS [5, 6] (Trust Ant Colony System) is a Trust model for P2P networks based on the bio-inspired algorithm of Ant Colony System (ACS), where pheromone traces are identified with the amount of trust a peer has on its neighbors when supplying a specific service.

It has the particularity that it not only gets the most trustworthy node to interact with, but it also obtains the most trustworthy path leading to the most reputable peer. In summary, the steps that compose this model are the next ones:

1. Client C executes TACS in order to find the "optimum" server S offering the service s
2. TACS launches the ACS algorithm and ants modify the pheromone traces of the network
3. TACS finishes, having selected the "optimum" path to server S'
4. TACS informs the client C that the "optimum" server found is S'
5. Client C requests service s to the server S'
6. Server S' provides service s' to the client C
7. Client C evaluates his satisfaction with the received service s'

8. If client C is not satisfied with the received service s', she punishes the server S' evaporating the pheromone of the links leading from C to S'

Each peer has its own pheromone traces for every link in the whole network, so it needs to know at any time the current topology of that network. When a client launches a set of ants and these ants are travelling and building the most trustworthy path leading to the most reputable server they have to decide at each peer whether to stop or keep looking for that path.

They will stop if they find a node offering the service requested by the client and whose pheromone traces belonging to the current path leading to it are high enough (over a certain threshold). Otherwise ants will select one of current node's neighbors who has not been visited yet. Which of those not visited neighbors to move towards is decided using the probability

$$p_k(c,s) = \frac{\tau_{cs}}{\sum_{u \in J_k(c)} \tau_{cu}} \tag{6}$$

where τ_{cs} is the pheromone trace of the link connecting nodes c and s and $J_k(c)$ is the set of neighbors of node c not visited yet by ant k. Actually, if $q \leq q_0$ the neighbor with the maximum pheromone trace is selected, otherwise equation (6) is used, where q is a random number within the interval $[0,1]$, and q_0 is a constant within the same interval.

Every time an ant crosses a link it modifies its pheromone trace in the following way:

$$\tau_{cs} = \tau_{cs} + (1-\varphi) \cdot \varphi \cdot (1-\tau_{cs}) \cdot \tau_{cs}$$

where $\varphi \in [0,1]$ is a constant used to control the pheromone local updating.

Once every ant returns to the client having found a path, the client has to decide which of those paths is better. In order to measure the quality of each path, the next expression is used:

$$Q(S_k) = \frac{\%A_k}{\sqrt{Length(S_k)}} \cdot \overline{\tau}^k$$

where S_k is the solution found by ant k, $\%A_k$ is the percentage of ants that have selected the same path as ant k, $Length(S_k)$ is the length of the path S_k, and $\overline{\tau}^k$ is the average pheromone of that solution.

Finally, an additional pheromone updating is carried out over the links belonging to the best path found by all ants in the following way:

$$\tau_{cs} = \tau_{cs} + \rho \cdot \tau_{cs}^2 \cdot Q(S_{Best})$$

where $\rho \in [0,1]$ is a constant used to control the pheromone global updating.

This process of launching ants, modifying pheromone traces and selecting the most trustworthy path is repeated a number of times (number of iterations) which depends on the size of the network.

Once the algorithm has finished, the client applies for the requested service to the node selected by TACS and assesses its satisfaction with the received service by measuring its similarity with the originally requested one. If the client is not fully satisfied, a punishment in terms of pheromone evaporation is carried out all along the path connecting the client and the server.

Therefore, if the satisfaction (which is a number within the interval $[0,1]$) is greater than or equal to 0.5, then the punishment is:

$$\tau_{cs} \leftarrow \tau_{cs} - \varphi(1 - Sat) \cdot 2 \cdot df_{cs}$$

Otherwise, if satisfaction is less than 0.5 the punishment we have is

$$\tau_{cs} \leftarrow \left(\frac{\tau_{cs}}{df_{cs}} - \varphi \right) \cdot Sat$$

where df_{cs} is a function that assigns values in $[0,1]$ to every link of the path and it is defined as follows:

$$df_{cs} = \sqrt{\frac{d_{cs}}{L \cdot (L - d_{cs} + 1)}}, \quad d_{cs} \in \{1, 2, \ldots, L\}$$

being L the actual length of the whole path and d_{cs} the distance of link e_{cs} from the client (number of hops). Therefore, the last link (the one reaching the server) takes a value of 1, and the rest take smaller values. The nearer to the client a link is, the closer to 0 is given.

Finally, all the links falling into the malicious server are also punished, but only if the satisfaction is under 0.5.

3 Discussion

3.1 Trust and Reputation Management – What for?

P2P networks have provided us with many solutions and advantages in human relationships, communications and e-businesses, among other fields. However, in most of the cases, a user belonging to a P2P network will have to interact with many other unknown or stranger participants, in a human-to-machine (instead of a human-to-human) manner. This intrinsic feature of P2P networks can lead to the possibility of being easily defrauded or disappointed.

In the same way we ask our friends or acquaintances their opinion about a good or a service (a film, a book, etc.) before buying that good or asking for that service, it would be equally efficient and beneficial to be able to perform the same survey in virtual communities such as P2P networks.

Having such a useful mechanism in order to assess the global trust or reputation of a peer in the system, by collecting the recommendations of other users in the network who have had some previous interactions with that peer, the probability of being cheated by a malicious entity would be significantly decreased.

In the following subsections we will describe the general steps followed by most of the trust and reputation models and present some threats directly related with these specific approaches. Finally, we will analyze the strong and weak points of every trust and reputation model presented in Section 2, and conclude with some real systems applying a trust and reputation scheme.

3.2 Trust and Reputation Models Steps

We have seen that the main target followed by every trust and reputation model is, in summary, to identify those peers who are most reliable supplying a certain service or more trustworthy carrying out a certain task.

How those peers are selected differs from one model to other but, for instance, in most of them we can observe more or less the same generic steps [12], as depicted in Fig. 1. First of all an entity checks its previous experiences with a given peer in order to form what is usually called direct trust.

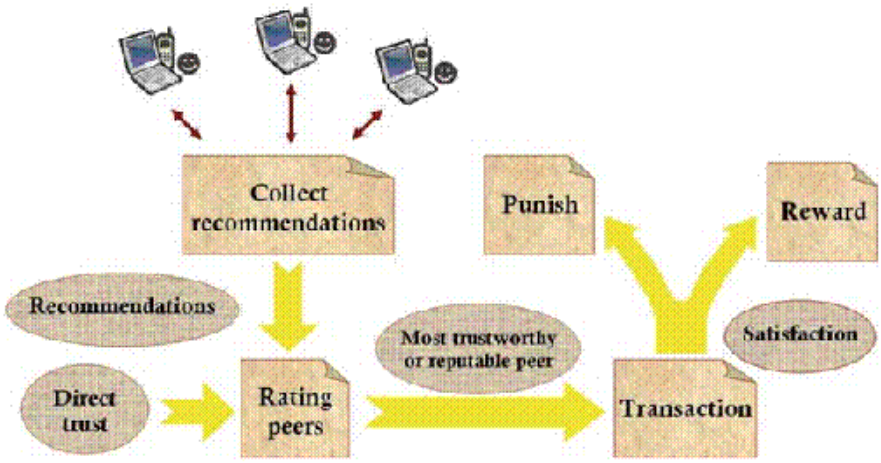


Fig. 1 Trust and reputation models steps

This direct trust can be assessed using complex expressions which usually take into account the number of previous transactions, the importance given to each transaction, the satisfaction obtained in each one, the time when it was performed,

etc. Or it could even be computed as the difference between the number of satisfactory transactions and unsatisfactory ones, like in Eigentrust [9].

Additionally the indirect experiences (or experiences of other peers) are taken into account as well, obtaining what is commonly known as the reputation of a peer. At this point, some models (like [21, 23, 24]) even distinguish between the trust given to a peer as a service provider, and as a recommender, filtering out this way unfair ratings coming from malicious users, since their recommendations will be discarded.

How this reputation value is obtained is also very specific for each model, but the main idea is to collect information about the behavior of the target peer from other peers who have had previous interactions with it. This information or recommendations are influenced in some models by the reliability of the recommender, as we mentioned. Otherwise, a collusion could be established where a set of malicious peers rated each other with the maximum value.

Therefore, an aggregation between the direct trust or direct experiences and the reputation or indirect experiences, weighted by the reliability of each recommender is performed in order to obtain a unique global trust value for a certain peer. Most of the models do not specify which peer is finally selected. It could be just the one with highest score, but not necessarily. And only TACS [5], from the studied models, provides not only the most reputable peer, but the most trustworthy path leading to it as well.

Once the peer to interact with has been selected, the transaction is effectively carried out. Then, the user who applied for a service or a task assesses her satisfaction with the received service or performed task. According to this satisfaction, a last step of punishing or rewarding the entity the transaction was done with, is performed.

However, not many models apply a specific and independent step of punish and reward, but they rather implicitly incorporate it in the rating step. Only TACS [5] and TDTM [25], from the analyzed ones, do it.

3.3 Common Challenges and Solutions in Trust and Reputation Management Over P2P Networks

As we have already seen, trust and reputation management in P2P networks provides several benefits to electronic interactions between users, like a minimum guarantee of benevolent behavior of another interacting peer.

Nevertheless, this kind of systems also have several common issues and challenges that need to be addressed when developing such mechanisms. Next we are going to discuss some of them.

3.3.1 Modeling Trust and Reputation

One of the first things to face when developing a new trust and reputation model, or when analyzing an existing one is the way of modeling (and consequently managing) precisely that: trust and reputation.

Thus, some models use bayesian networks (like BNBTM [20]), while others use fuzzy logic (like PATROL-F [17]), or even bio-inspired algorithms (as is the case of TACS [5], TDTM [25] and AntRep [19]). Other models, however, just give an analytic expression to compute trust (for example, GroupRep [18], and other models such as [23, 24] or [21]).

Each way of modeling trust and reputation has its own advantages and drawbacks. For instance, fuzzy logic allows us to model concepts like trust, reputation or recommendations in a manner closer to the way humans understand them. However, fuzzy logic will be difficult to scale to larger problems because there exist important limitations with conditional possibility, the fuzzy set theory equivalent of conditional probability [7].

Bio-inspired mechanisms have demonstrated a high adaptability and scalability in dynamic scenarios such as P2P networks. However, in some cases, their indeterminism and approximation technics can lead to choose a malicious peer as the most trustworthy one, discarding another clearly benevolent who could be selected.

Analytic expressions are most of the times easy to read and understand, but they may not take into consideration all the possible factors involved in the evaluation of trust and reputation for a certain participant in a P2P network since they need to manage those factors explicitly, while other approaches effectively deal with them in an implicit way.

Finally, Bayesian networks provide a flexible mechanism to represent multifaceted trust in many contexts of each others' capabilities (providing different services or carrying out several tasks, for instance). It also allows to efficiently combine different aspects of trust. One drawback is, however, that the approach can be computationally intensive, especially when the variables being studied are not conditionally independent of one another.

3.3.2 Contextualized Trust and Reputation

Another important concept managed in many trust and reputation models is what is commonly known as the context. Since a peer can be very trustworthy and benevolent when supplying a service or performing a task but, at the same time, very fraudulent or malicious when dealing with another service or task, it is not fair to identify it as fully trustworthy or untrustworthy.

That is why several models like CuboidTrust [1], BNBTM [20], PeerTrust [22], PATROL-F [17] or TACS [5] include a context factor or distinguish in one or another way the trust placed on a peer depending on the task or service it is requested to supply or perform.

3.3.3 P2P Networks Dynamism

Not many models take into account the intrinsic dynamic nature of P2P networks (i.e. nodes entering and leaving the community whenever they want) when modeling trust relationships.

Furthermore, only a few ones among the studied models (like CuboidTrust [1], EigenTrust [9], TACS [6] and PeerTrust [11]) present experiments dealing with this issue as well as with the fact that also the behavior of peers may be dynamic, i.e., peers are not always benevolent or malicious. And how fast and accurate a model can react against this behavioral changes is an important feature that every trust and reputation model should consider.

Therefore, in our opinion every trust and or reputation model for P2P networks should consider three basic scenarios:

1. A static one, where the topology of the network does not change along the time. This is the simplest scenario where trust and reputation models should work efficiently. As we said, many authors just consider a situation like this when developing their models. It is a good starting point but, however, it is not realistic.
2. A dynamic one, where the topology changes along the time, with nodes joining and leaving the network.
   This scenario could be used in order to test the reaction of a trust and reputation model against changes in the size and topology of the network, and the specific nodes composing it. For instance, it could be checked the reaction of the model if a very reputable (or, equally, a very fraudulent) node enters or leaves the system.
   This kind of experiment also allows to test how the model faces the topic of newcomers and deals with some threats like the Sybil attack [4].
3. An oscillating one, where the behavior of the nodes changes along the time, so they can be benevolent and become malicious and vice versa.
   Finally, this scenario would show if the model has a quick and accurate response or not against sudden behavioral changes of nodes trying to cheat. A good trust and reputation model should identify immediately these fluctuations and react consequently.

3.3.4 Collusion

There are also some security threats related to trust and reputation systems which are not completely considered in every model. For instance, only PeerTrust [22], CuboidTrust [1] and EigenTrust [9] explicitly treat the problem of collusion among malicious nodes.

A collusion consists of several malicious nodes (providing a bad service or performing tasks inadequately) joining in order to increase their reputation values by fake rating themselves and, on the other hand, decrease the reputation of current benevolent peers by giving negative recommendations about the latter, as it can be observed in Fig. 2.

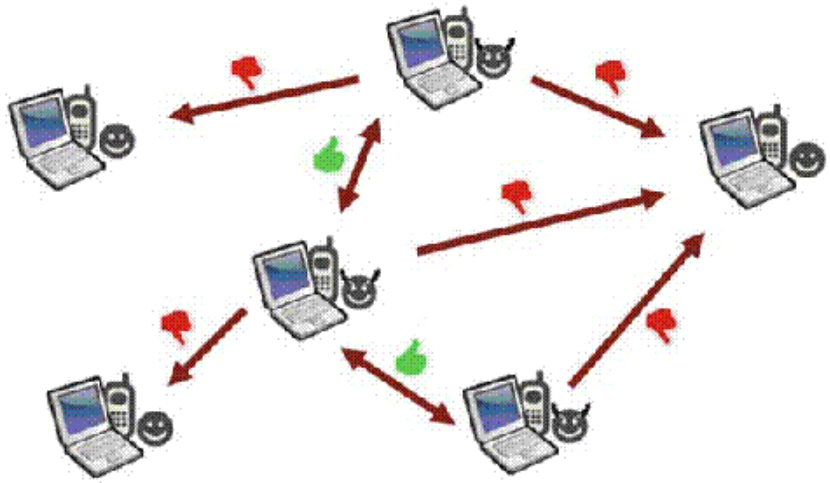


Fig. 2 Malicious peers forming a collusion

It is not easy to overcome this problem, specially when the percentage of nodes forming the collusion is quite high. Actually, any of the studied models is completely resilient against this kind of attack. What authors do is to try to minimize its global impact by punishing every node in the network which is known to belong to the collusion.

There are even variants of this attack, like a set of nodes providing good services but rating positively other malicious peers and negatively other benevolent ones. Those models which do not distinguish between the reliability of a user when providing a service or carrying out a task, and when supplying recommendations cannot effectively tackle this attack.

3.3.5 Identity Management – Sybil Attack

The last challenge we will discuss has to do with the identity management in virtual communities, specifically in P2P networks. It is a fact that cannot be obviated when designing and developing a new trust and reputation model since many deficiencies and weaknesses can emerge from an inaccurate management of this topic.

One of the most common problems related to identity management in trust and reputation schemes is what is known as Sybil attack. In a Sybil attack the reputation system of a P2P network is subverted by creating a large number of pseudonymous entities, using them to gain a disproportionately large influence. A reputation system's vulnerability to a Sybil attack depends on the cost of generating new identities, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

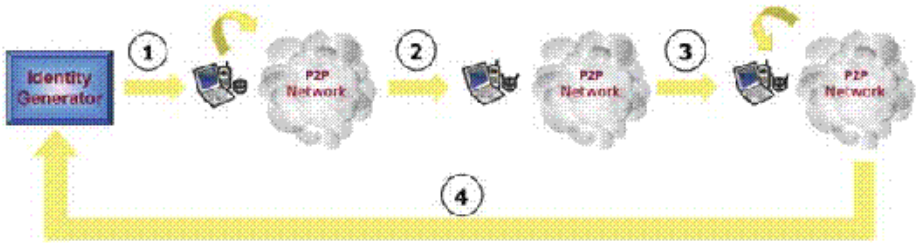


Fig. 3 Sybil attack

Figure 3 shows the steps followed in a Sybil attack and some other attacks related to identity management in trust and reputation models. These steps are:

1. An entity joins the community with a new identity, looking like being a trustworthy peer.
2. At a certain moment (probably after gaining some reputation in the system), this entity swaps her goodness and becomes malicious, obtaining thus a greater self-profit.
3. Once the system has detected her behavioral change and has identified her as a malicious participant, she leaves the network.
4. Finally, she generates a new identity and enters again the system, repeating the process indefinitely.

None of the studied models (except for EigenTrust [9]) considers explicitly the problem of the Sybil attack.

3.4 Strengths and Weaknesses of the Described Models

Once we have shown and described a number of trust and reputation models over P2P networks it is time to analyze in depth several important characteristics and features that define those models as well as some detected weaknesses and deficiencies.

Thus for instance, CuboidTrust [1] implements the testing scenarios described before, showing good outcomes and it also considers the problem of collusion. On the other hand, direct trust or direct experiences are not given a differentiated treatment, and the score takes discrete values in the set $\{-1, 1\}$ instead of continuous ones in the interval $[-1, 1]$.

EigenTrust [9] also implements the three mentioned scenarios as well as the collusion one. Moreover, it also takes into account the Sybil attack. It introduces the concept of pre-trusted peers, which is very useful in the model, but it is no applicable in all the cases, since there is not always a set of peers that can be trusted by default, prior to the establishment of the community.

One positive point of BNBTM [20] is its management of trust in different contexts, which can be combined to form an overall opinion of the trustworthiness of a peer. A weak point, in our opinion, is that it only deals with three discrete valuations for a transaction.

The distinction among trust between groups of peers, between groups and peers, and only between peers is the strong point of GroupRep [18]. However, it is missing a global trust value for a peer as a result of the combination of the three previous ones.

AntRep [19] has the ability to easily adapt to the dynamic topologies of P2P networks thanks to the use of ant colony systems. As a disadvantage, it just provides a mechanism to distribute reputation evidences, not to assess those evidences.

Searching all the paths connecting two agents may lead to some scalability problems in [23, 24]. On the other hand, both models clearly distinguish between direct trust and indirect trust or reputation, and also between the trust given to a peer as a service provider and as a recommender.

PeerTrust's [22] strengths are its good outcomes for the three previously proposed scenarios, as well as for a collusion one. It also introduces a context factor to distinguish the trust given to a peer for different transactions. However, the way it measures the credibility of a peer in equation (4) does not distinguish between the confidence placed on a peer when supplying a service or carrying out a task, and when giving recommendations about other peers.

This distinction is effectively done, however, in PATROL-F [17] and Trust Evolution [21]. Moreover, PATROL-F allows to model concepts like the result of an interaction in a similar way humans do thanks to the use of fuzzy logic.

The assumption of the existence of a PKI in a P2P network can be seen as a drawback of TDTM [25]. The use of an ant colony system allows it to adapt accurately to sudden changes in the topology of the network, as it is also the case of TACS [5].

TACS implements the three testing scenarios too, and it has a good performance in all of them as well. However, it needs every node to know the topology of the network at every moment, and that is not always feasible in a P2P community.

3.5 Real Scenarios

Finally, let us show some real trust and reputation systems [8] where most of the concepts explained here are employed. For instance, eBay auction market has a feedback scheme where every buyer and seller rates each other after a transaction between them is carried out. These feedbacks are centrally aggregated in order to get a reputation value for each role. Some studies [14, 16] reveal that buyers provide ratings about sellers 51.7% of the time, and sellers provide ratings about buyers 60.6% of the time. Of all ratings provided, less than 1% is negative, less than 0.5% is neutral and about 99% is positive. It was also found that there is a high correlation between buyer and seller ratings, suggesting that there is a degree of reciprocation of positive ratings and retaliation of negative ratings.

Another distributed system modeling reputation is PageRank [13], the algorithm which the search engine of Google is based on. It represents a way of ranking the search results based on a page's reputation, which is mainly obtained by the number of links pointing to it, since the higher is the number of incoming links, the better content that page is supposed to have. PageRank applies the principle of trust

transitivity to the extreme since rank values can flow through looped or arbitrarily long hyperlink chains. Amazon, BizRate or Advogato are other examples of systems where a trust and or reputation scheme is applied in many different environments.

4 Conclusions and Future Work

P2P networks have been rapidly spread in the last few years. Nevertheless, together with its fast development, many security threats have also appeared, compromising sensitive information and promoting frauds in electronic transactions.

Trust and reputation management has arisen as one of the most innovative and accurate solutions to most of these threats. By using a trust and reputation system a peer who wants to interact with another peer in the community has more information and, therefore, more opportunities to select the right partner to have a transaction with, rather than with a fraudulent one.

In this work we have presented a description of some of the most representative trust and reputation models for P2P networks. We have explained how each of them works, how they manage the concepts of trust and reputation or how they gather information about other peers in the network.

But we have also discussed the main common characteristics of all the exposed models, as well as the most important differences among them. We have analyzed their weaknesses and deficiencies and shown some real examples of systems using trust and or reputation schemes.

However, there is still some work to do, for instance in the proposal of some standard patterns to lead a proper and accurate development of trust and reputation models for P2P networks, as well as validation tools allowing to homogeneously test every designed trust and reputation model.

Acknowledgements

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2007. Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

References

1. Chen, R., Chao, X., Tang, L., Hu, J., Chen, Z.: CuboidTrust: A Global Reputation-Based Trust Model in Peer-to-Peer Networks. In: Autonomic and Trusted Computing, no. 4610 in LNCS, pp. 203–215. 4th International Conference, ATC 2007, Springer, Hong Kong, China (2007)
2. Cordón, O., Herrera, F., Stützle, T.: A review on the ant colony optimization metaheuristic: Basis, models and new trends. Mathware and Soft Computing **9**(2–3), 141–175 (2002)
3. Dorigo, M., Maniezzo, V., Colomi, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics **26**(1), 29–41 (1996)

4. Douceur, J.R., Donath, J.S.: The sybil attack. In: Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS 02), pp. 251–260 (2002)
5. Gómez, F., Martínez, G., Skarmeta, A.F.: TACS, a Trust model for P2P networks. Wireless Personal Communications, in press (2008)
6. Gómez Mármol, Félix: TACS – Trust Ant Colony System. URL http://ants.dif.um.es/~felixgm/research/tacs
7. Halpern, J.Y.: Reasoning about Uncertainty. The MIT Press (2003)
8. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems **43**(2), 618–644 (2007)
9. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: Proc. of the International World Wide Web Conference (WWW). Budapest, Hungary (2003)
10. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
11. L3S Research Center: PEERTRUST – L3S Research Center. URL http://www.l3s.de/peertrust
12. Marti, S., Garcia-Molina, H.: Taxonomy of trust: Categorizing P2P reputation systems. Computer Networks **50**(4), 472–484 (2006)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1998)
14. Resnick, P., Zeckhauser, R.: Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In: The Economics of the Internet and E-Commerce, vol. 11, pp. 127–157. Elsevier Science (2002)
15. Stallings, W.: Data and Computer Communications (7th ed.). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2004)
16. Sundaresan, N.: Online trust and reputation systems. In: Proceedings of the 8th ACM conference on Electronic Commerce, pp. 366–367 (2007)
17. Tajeddine, A., Kayssi, A., Chehab, A., Artail, H.: PATROL-F – a comprehensive reputation-based trust model with fuzzy subsystems. In: Autonomic and Trusted Computing, no. 4158 in LNCS, pp. 205–217. Third International Conference, ATC 2006, Springer, Wuhan, China (2006)
18. Tian, H., Zou, S., Wang, W., Cheng, S.: A group based reputation system for P2P networks. In: Autonomic and Trusted Computing, no. 4158 in LNCS, pp. 342–351. Third International Conference, ATC 2006, Springer, Wuhan, China (2006)
19. Wang, W., Zeng, G., Yuan, L.: Ant-based reputation evidence distribution in P2P networks. In: GCC, pp. 129–132. Fifth International Conference on Grid and Cooperative Computing, IEEE Computer Society, Changsha, Hunan, China (2006)
20. Wang, Y., Cahill, V., Gray, E., Harris, C., Liao, L.: Bayesian network based trust management. In: Autonomic and Trusted Computing, no. 4158 in LNCS, pp. 246–257. Third International Conference, ATC 2006, Springer, Wuhan, China (2006)
21. Wang, Y., Tao, Y., Yu, P., Xu, F., Lu, J.: A Trust Evolution Model for P2P Networks. In: Autonomic and Trusted Computing, no. 4610 in LNCS, pp. 216–225. 4th International Conference, ATC 2007, Springer, Hong Kong, China (2007)
22. Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities. IEEE Transactions on Knowledge and Data Engineering **16**(7), 843–857 (2004)
23. Yu, F., Zhang, H., Yan, F., Gao, S.: An improved global trust value computing method in P2P system. In: Autonomic and Trusted Computing, no. 4158 in LNCS, pp. 258–267. Third International Conference, ATC 2006, Springer, Wuhan, China (2006)
24. Zhang, Y., Chen, H., Wu, Z.: A social network-based trust model for the semantic web. In: Autonomic and Trusted Computing, no. 4158 in LNCS, pp. 183–192. Third International Conference, ATC 2006, Springer, Wuhan, China (2006)
25. Zhuo, T., Zhengding, L., Kai, L.: Time-based dynamic trust model using ant colony algorithm. Wuhan University Journal of Natural Sciences **11**(6), 1462–1466 (2006)

2

TACS, a Trust Model for P2P Networks

Title:	TACS, a Trust Model for P2P Networks
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez, Antonio F. Gómez Skarmeta
Type:	Journal
Journal:	Wireless Personal Communications, Special Issue on "Information Security and data protection in Future Generation Communication and Networking"
Impact factor (2008):	0.331 (position 53/67)
Publisher:	Springer
Volume:	51
Number:	1
Pages:	153-164
Year:	2009
Month:	October
DOI:	http://dx.doi.org/10.1007/s11277-008-9596-9
State:	Published

Table 2: TACS, a Trust Model for P2P Networks

Wireless Pers Commun (2009) 51:153–164
DOI 10.1007/s11277-008-9596-9

TACS, a Trust Model for P2P Networks

Félix Gómez Mármol · Gregorio Martínez Pérez · Antonio F. Gómez Skarmeta

Published online: 10 October 2008
© Springer Science+Business Media, LLC. 2008

Abstract Many researchers are working hard in order to improve the capabilities of P2P networks. In this way, one of the main factors slowing down the improvement and development of P2P networks is the need to guarantee a minimum level of trust between two interacting nodes. This paper presents an innovative trust model providing a solution to this problem. Our trust model, which is based on the ant colony system, is able to select the best server to interact with, in terms of being the most trustworthy, in a very high percentage. Here, the traces of pheromone left by the ants correspond to the amount of trust that a node has on its neighbors.

Keywords Trust management model · P2P networks · Soft-computing

1 Introduction

The recent and fast development of P2P networks has caught the attention of many researchers because of their numerous advantages. However, they also suffer from certain deficiencies that have not yet been solved completely. One of these shortcomings is the efficient and effective management of trust among the nodes composing the network.

How to distinguish between potential benevolent nodes and potential malicious nodes is one of the main research lines on P2P networks nowadays. The amount of P2P network applications would be greatly increased by solving these kind of problems. So far, the most

F. Gómez Mármol (✉) · G. Martínez Pérez · A. F. Gómez Skarmeta
Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia,
30071 Murcia, Spain
e-mail: felixgm@um.es

G. Martínez Pérez
e-mail: gregorio@um.es

A. F. Gómez Skarmeta
e-mail: skarmeta@um.es

extended method to overcome this problem relies on computing a trust and a reputation value for each node on a system.

The innovation of our model lies on the application of an evolutionary bio-inspired algorithm based on the ant colony system [4]. This algorithm is widely accepted as one of the best soft-computing algorithms when solving some kind of optimization problems [5].

Here the ants look for the most trustworthy server all over the network and they are guided by the traces of pheromone they leave while building of a route. Thus, the more visited a path is, the greater amount of pheromone it will have and consequently, more ants will choose to follow that route. Finally, the algorithm will converge with the majority of ants following the best path.

The choice of using ant colony systems to solve the problem of computing trust values in a P2P network was taken, among other factors, because of the ability of this algorithm to adapt itself to dynamic environments.

Following this model, we can reach, therefore, a greater level of confidence in those P2P network nodes to interact with. We can also reach more suitable decisions despite of uncertainty and a more accurate perception of the real behavior of each node on the network, since the pheromone traces over the network tell us how trustworthy each peer is.

The rest of the paper is organized as follows. First, in Sect. 2 we present a summary of some of the related work on this topic. Section 3 describes the ant colony system as it is, while Sect. 4 presents our P2P trust model, called trust ant colony system (TACS). Experiments and results are described in Sect. 5 and finally Sect. 6 provides conclusions and lines of future work.

2 Related Work

Many research groups have focused their efforts on modelling trust in P2P systems. The difference comes in how they approach the problem.

For instance, authors of *Sporas* [18], *Regret* [10,11], *AFRAS* [3], PATROL-F [13], AntRep [16] and [19], all of them use a multiagent system. Sporas, Regret and AFRAS only deal with reputation, while PATROL-F and AntRep model both trust and reputation concepts [19] only manages trust.

AFRAS and PATROL-F represent trust and/or reputation by means of fuzzy sets and they use fuzzy logic to manage them. Sporas and Regret just give some analytic expressions to compute reputation. AntRep is the only one using a bioinspired algorithm of ant colony system [4,5], like us. And [19] makes use of a social network.

Some other trust and reputation models more specific for P2P are [17], DWTrust [7], TPOD [14], GroupRep [15], PTM [1,2] and TDTM [20].

An ant colony system is used as well in TDTM in order to manage trust. But authors of this model assume the existence of a public key infrastructure (PKI) in a P2P scenario, which we consider is not always possible.

Research work shown in [8] and [9] has also helped to design our model, its components and its properties.

3 Ant Colony System

Ant Colony Optimization [4] (from now on, ACO) is a bio-inspired meta-heuristic that allows the design of algorithms easy to understand, fast and with high performance, to solve optimization problems that can be represented as a weighted graph.

Thus, the artificial ant is an agent remembering visited nodes using a list L. At the end that list contains the solution built by the ant. Moreover, on each step, being at node c it chooses which node s to move towards between the neighbors of c that had not been visited ($J(c) = \{s \mid \exists\, e_{cs} \wedge s \notin L\}$), using a transition rule and leaving on that edge a certain trace of pheromone, τ_{cs}.

Therefore, the probability of the ant k located at node c of moving towards node s is defined by the following transition rule

$$p_k(c,s) = \begin{cases} \frac{[\tau_{cs}]^\alpha \cdot [\eta_{cs}]^\beta}{\sum_{u \in J_k(c)} [\tau_{cu}]^\alpha \cdot [\eta_{cu}]^\beta} & \text{if } s \in J_k(c) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where τ_{cs} is the pheromone of the edge e_{cs}, η_{cs} is the heuristic information of the edge e_{cs}, $J_k(c)$ is the set of reachable nodes from c not visited yet by the ant k and α and β are two weights that establish a balance between memoristic information and heuristic one, respectively.

Specifically, the Ant Colony System [5] (ACS) incorporates a proportional pseudo-aleatory transition rule as follows:

$$s = \begin{cases} \arg\max_{u \in J_k(c)} [\tau_{cu}]^\alpha \cdot [\eta_{cu}]^\beta & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \tag{2}$$

where s is the chosen node by ant k on its next movement, q is an uniform in $[0,1]$, q_0 is the probability of choosing deterministically the most promising edge and S is a random node selected using Eq. 1.

The algorithm of the Ant Colony System is as follows:

```
1  for It = 1 to Number_of_iterations do
2    for k = 1 to Number_of_ants do
3      S_k ← initial node
4    for i = 2 to Number_of_nodes do
5      for k = 1 to Number_of_ants do
6        S_k ← S_k ∪ Transition_Rule(S_k, τ, η, α, β, q_0)
7        Pheromone_local_updating(S_k, φ, τ_0)
8    for k = 1 to Number_of_ants do
9      if (Q(S_k) > Q(S_Current Best)) then
10       S_Current Best ← S_k
11   if (Q(S_Current Best) > Q(S_Global Best)) then
12     S_Global Best ← S_Current Best
13   for i = 1 to Number_of_nodes do
14     Pheromone_global_updating(S_Global Best, Q(S_Global Best), ρ)
```

Algorithm 1 ACS, Ant Colony System

The pheromone local updating has the following appearance:

$$\tau_{cs}(t) = (1-\varphi) \cdot \tau_{cs}(t-1) + \varphi \cdot \tau_0 \tag{3}$$

With this operation, the pheromone associated with an edge slows down on each time an ant visits it and thus visited edges are less promising since more ants travel across them in the current iteration, which encourages the exploration of edges not visited yet. This avoids the premature convergence of the ants.

Additionally, pheromone global updating is only applied on those nodes who constitute the best global solution and it depends on the quality of the latter. It has the following appearance:

$$\tau_{cs}(t) = (1-\rho)\cdot\tau_{cs}(t-1) + \rho\cdot Q(S_{\text{Global Best}}) \quad (4)$$

4 TACS, Trust Ant Colony System

The approach of this model is as follows: any P2P network node requests a service to the community and acts as a client. It executes the TACS algorithm and retrieves the path that leads to the "optimum" node that offers that requested service. That node then acts as a server.

But, what does being the "optimum" server mean in this context? It means being the most trustworthy server in the network offering one specific required service.

In our model, a P2P network is represented as a weighted graph where each edge e_{cs} has two weights: τ_{cs} and η_{cs}. τ_{cs}, the pheromone, represents the trust that node c has on reaching the optimum server, for a certain service, travelling through node s. And η_{cs}, the heuristic, means the similarity between the service requested by the client c and the service offered by the server s. Those weights τ_{cs} and η_{cs} take values within the interval $[0, 1]$.

It is important to have in mind that the service offered by a server and the service finally given by this one are not necessarily the same. It will depend on the goodness of the server. If the server is benevolent, it will provide exactly the same service it promised. But if we have a malicious server, it will probably supply a different (worse) service than the one he initially offered.

If a node s does not offer the requested service then $\eta_s = 0$ and any ant will choose it to travel across, according to the transition rule of the Eq. 2. Equally, if a node of the path leaves the P2P network (or another one joins the network), ants are able to find dynamically alternative routes.

Once the algorithm has selected the "optimum" server to interact with, the client interacts with it and requests the desired service. When the server supplies that service to the client, this one evaluates its satisfaction with the received service. If the client is fully satisfied nothing happens. If not, a punishment is carried out all along the path that connects the client and the server (and also over all the edges that fall into the malicious server, as we will see later).

In summary, the steps that compose this model are the following:

1. Client C executes TACS in order to find the "optimum" server S offering the service s
2. TACS launches the ACS algorithm and ants modify the pheromone traces of the network
3. TACS finishes, having selected the "optimum" path to server S'
4. TACS informs the client C that the "optimum" server found is S'
5. Client C request service s to the server S'
6. Server S' provides service s' to the client C
7. Client C evaluates his satisfaction with the received service s'
8. If client C is not satisfied with the received service s', he punishes the server S' evaporating the pheromone of the edges leading from C to S'

4.1 Pheromone Updating

Let z_1 be

$$z_1 = 1 + (1-\varphi)\cdot(1 - \tau_{cs}(t-1)\eta_{cs}(t-1)) \quad (5)$$

Then, in our model, pheromone local updating is defined as follows:

$$\tau_{cs}(t) = (1 - \varphi) \cdot \tau_{cs}(t - 1) + \varphi \cdot z_1 \cdot \tau_{cs}(t - 1) \tag{6}$$

Since $\tau, \eta, \varphi \in [0, 1]$, it is possible to demonstrate that

$$\lim_{t \to \infty} \tau_{cs}(t) = z_1 \cdot \tau_{cs}(t - 1) \tag{7}$$

What implies an increase of pheromone above the previous value, but never higher than a 100%. Moreover, with the term $(1 - \tau_{cs}\eta_{cs})$ we achieve that edges with lower values of pheromone can recover faster (increasing more its traces) and those who have higher values increase themselves slower.

On the other hand, about pheromone global updating, let z_2 be

$$z_2 = 1 + \tau_{cs}(t - 1)\eta_{cs}(t - 1) \cdot Q(S_{\text{Global Best}}) \tag{8}$$

So pheromone global updating takes here the following appearance:

$$\tau_{cs}(t) = (1 - \rho) \cdot \tau_{cs}(t - 1) + \rho \cdot z_2 \cdot \tau_{cs}(t - 1) \tag{9}$$

Therefore now those edges with a higher value of τ and η are more rewarded than those with lower values. By doing like this, we achieve a certain balance which gives quite good results.

4.2 Quality of a Path

Once every ant has built a path leading to a server, one of them has to be selected as the best of all. To do this we need to measure the quality of each route. Thus, the quality of the path found by ant k, S_k, is:

$$Q(S_k) = \frac{\%A_k}{\sqrt{\text{Length}(S_k)}} \cdot \overline{\tau}^k \tag{10}$$

where $\%A_k$ is the percentage of ants that have selected the same path as ant k, $\text{Length}(S_k)$ is the length of the path chosen by ant k, and $\overline{\tau}^k$ is the average pheromone of that solution.

Since $\%A_k, \overline{\tau}^k \in [0, 1]$ and $\text{Length}(S_k) \geq 1$, then $Q(S_k) \in [0, 1]$.

With this definition of quality, the higher are the percentage of ants that select a certain path and the average pheromone of that route, the higher is the quality of that solution. And on equal conditions, a shorter path is preferred.

4.3 Ants Transition and Stop Condition

How the ants build the solutions in our proposed trust model is different from the manner shown in Algorithm 1, because a P2P network can have any topology, and its dynamism can make any node to leave or enter the network at any time. These are factors that lead us to make certain important adjustments in the behaviour of the basic ACS algorithm.

Now each ant will probably find a different length path, so the stopcondition has to change. Ants should stop looking for a trustworthy server when they find a server that is supposed to be benevolent or when they explore the whole net and there is not any server offering the requested service.

Hence, let ant k be at node s in a certain instant and planning to move towards node r. Algorithm 2 shows the behavior of ant k in that situation.

This algorithm can also be explained as it is described next:

1. If node s has no neighbors or all its neighbors have been visited by ant k:
   (a) If s offers the requested service, ant k stops and returns current path
   (b) Otherwise ant k moves back until finding a node with alternative not visited paths, or a node that offers the requested service, or until the whole net has been explored with no server offering the desired service
2. Otherwise (r has been selected as next node among the neighbors of s):
   (a) If r offers the required service, then ant k has to decide whether to stop and return the current path, or going on searching for a better server.
      i. If $\overline{\tau}^k \leq 0.5$ the trustworthiness of node r will not probably be very high, so ant k keeps on searching a better server beyond node r.
      ii. Otherwise ant k stops and returns the current solution with a probability given by $\overline{\tau}^k$
   (b) Otherwise ant k continues building its solution

```
1  r ← Transition_Rule(s, τ, η, α, β, q0)
2  if ((s has no neighbors) or
3     (all its neighbors have been visited)) then
4     if (s offers the requested service) then
5        Stop and return current solution
6     else
7        while (solution is not empty) do
8           s ← Previous_Node(s)
9           if (s has more neighbors not visited) then
10             Keep trying those alternative paths
11          else if (s offers the requested service) then
12                Stop and return current solution
13       if (solution is empty) then
14          The whole net has been explored and there is no server offering
15                the desired service
16 else
17    if (r offers the requested service) then
18       if ((τ̄^k > 0.5) and (τ̄^k > random(0,1))) then
19          Stop and return current solution
```

Algorithm 2 Ants transition and stop condition

So when ant k discovers a node that offers the required service, that ant decides to stop or not depending on the average pheromone $\overline{\tau}^k$ of the current path S_k. The higher $\overline{\tau}^k$ is, the higher the probability of stopping is. Note that an ant can not visit the same node twice because it will result in a loop.

It is quite important to realize that the transition rule together with the punishment method are the core and the key of this model. A good combination of both can result in very good outcomes. But a bad use or design of them will surely lead to very bad results.

4.4 Satisfaction Computation

As commented before, once the ants have selected the "optimum" server to interact with, the client indeed interacts with it and requests the desired service. When the server supplies that service to the client, this one evaluates its satisfaction with the received service (which may be different from the one initially offered by that server).

If the client is fully satisfied nothing happens. Otherwise a punishment, in terms of pheromone evaporation, is carried out all along the path that connects the client and the server.

The satisfaction of a client is computed measuring the similarity between the requested service and the finally received one, taking into account several characteristics, weighting each of them. That similarity (equally, the satisfaction) takes values within the interval $[0, 1]$.

It is important to mention that each client defines subjectively those involving weights, so it can give more importance to some service characteristics rather than others.

4.5 Punishment

Many punishment methods were tested and all of them had one thing in common: the pheromone of those all edges that fall into the malicious server should be evaporated. Other conclusion was obtained from those several trials: the closer is the edge to the malicious server the stronger should be its punishment. So the evaporation is not the same all along the edges of the path. Therefore there is a distance factor in the punishment of each edge e_{cs} as follows:

$$df_{cs} = \sqrt{\frac{d_{cs}}{L \cdot (L - d_{cs} + 1)}}, \quad d_{cs} \in \{1, 2, \ldots, L\} \tag{11}$$

where L is the actual length of the whole path and d_{cs} is the distance of edge e_{cs} from the client (number of hops).

Once we have this, there are two ways of carrying out the punishment depending on the satisfaction. If the satisfaction is greater or equal to 0.5, then the punishment is:

$$\tau_{cs} \leftarrow \tau_{cs} - \varphi(1 - \text{Sat}) \cdot 2 \cdot df_{cs} \tag{12}$$

Otherwise, if satisfaction is less than 0.5 the punishment we have is

$$\tau_{cs} \leftarrow \left(\frac{\tau_{cs}}{df_{cs}} - \varphi\right) \cdot \text{Sat} \tag{13}$$

But it is important to note that the punishment is only executed on those edges falling into relay nodes (not on those leading to servers offering the requested service).

And finally, all the edges that fall into the malicious server have the following punishment, but only if satisfaction is under 0.5.

$$\tau_{cs} \leftarrow (\tau_{cs} - \varphi) \cdot \text{Sat} \tag{14}$$

5 Experiments and Results

In order to demonstrate the behavior of the proposed P2P trust model at work, some experiments have been performed in two type of networks: static and dynamic ones.

The parameter values for all the experiments are the following ones:

$$\begin{array}{llll} \eta_{cs} = 1, \ \forall c, s & \varphi = 0.1 & \alpha = 1.0 & N_{\mathrm{I}} = \lceil \ln(N_N) \rceil \\ q_0 = 0.98 & \rho = 0.1 & \beta = 1.0 & N_{\mathrm{A}} = \lceil \ln(N_N) \rceil \end{array} \quad (15)$$

where N_{I} is the number of iterations, N_A is the number of ants and N_N the number of nodes belonging to the network.

5.1 Experiments and Results on Static Networks

We tested our model on static networks, that is, networks that do not change its size along the time. We tried 5,000 networks with a random size between 2 and 10 nodes, 10 and 20 nodes, 20 and 30 and so on. And on each of those networks there were 100 interactions.

The experiment carried out on this kind of networks was a comparison between our model and other Ant Colony Optimization (ACO) algorithms such as Ant Colony System [5] (as it is), Ant System [6], Elitist Ant System [6] and Max-Min Ant System [12]. Outcomes are presented in Fig. 1.

As it can be clearly observed in the graphics, the benevolent servers selection percentage is, in the case of TACS, much greater than the one got by the rest of algorithms. And although all of them share the same punishment mechanism, the pheromone updating and transition rule differs on each one.

On the other hand, the solution quality in all the compared algorithms does not take into account neither the percentage of ants that have selected the same solution, nor the average pheromone of such solution, as it indeed is done in TACS (see formula (10)).

These algorithms only measure the length of the path, and consequently they get a poorer information measurement and therefore, less effective.

Furthermore, the transition rule in all those algorithms does not take into account all the cases considered in TACS (see Algorithm 2), so they cannot even distinguish a relay node from another one offering the required service.

For all these exposed reasons, all the compared algorithms have worse behavior (regardless of the size of the network) than the one shown by our model.

Finally, centering on TACS, it is shown that the selection percentage always remains above the 96.5%, with a standard deviation that never exceeds the 3.6%. It can be also seen that the performance gets worse, although very slightly, as the network size grows.

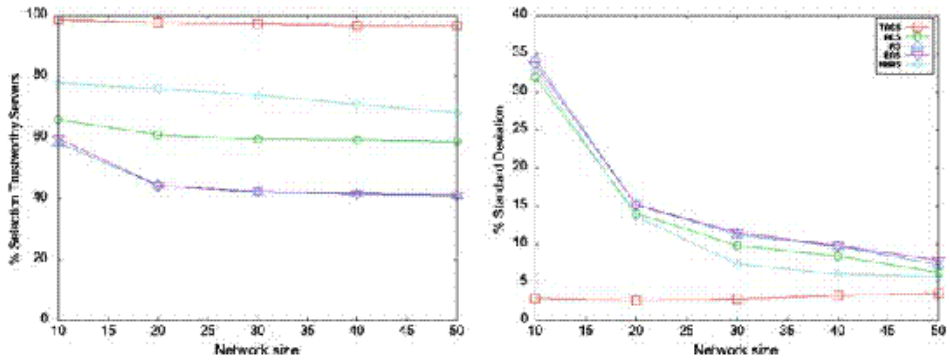


Fig. 1 Static networks. ACO algorithms comparison

5.2 Experiments and Results on Dynamic Networks

Once we made the specified experiments on static networks we thought that the actual strong point of our model has to be its behavior in dynamic networks, as one of the main characteristics of P2P networks is its dynamism.

Therefore the following experiments were focused on networks that changed their size along the time, dynamically. Thus, we defined a "probability of dynamism" which meant the probability that a certain network has of changing its size (and, consequently, its topology) after each interaction.

If the network changes its size, it can be by one of two ways: adding a new relay node (linking it randomly to the rest of nodes of the network), or removing a random one.

In these circumstances, we tested 5,000 networks with 0% of probability of dynamism, 5,000 with 20%, 40%, 60%, 80% and 100%. Each network had a random size between 2 and 50 nodes and 100 interactions were taken on each of them. Outcomes are shown in Fig. 2.

The explanation of why our model behaves better than any of the compared algorithms fits in with the one given for the equivalent experiment made on static networks. Even more, this difference of behaviour is now greater just due to the networks dynamism and to the inability of the rest of algorithms to adapt themselves quickly to changes in the topology of these networks, modifying properly the pheromone traces.

If we center now specifically in the outcomes given by our model TACS we can check that the selection percentage hardly varies with the networks dynamism and it always remains above the 98.6% with a standard deviation that does not exceed the 1.7% at any case.

These results show the high adaption capacity that TACS model has on highly changing environments, as actual P2P networks usually are.

5.3 Oscillating Networks

Now we present a new experiment where the trustworthiness of a server may change along the time. In fact, we have tried 5,000 networks with sizes between 2 and 10 nodes, 10 and 20, 20 and 30 and so on, with 1,000 interactions for each network, where after every 50 interactions the trustworthiness of the current selected server was changed and a random benevolent node was turned into a malicious one and a random malicious node was turned into a benevolent one.

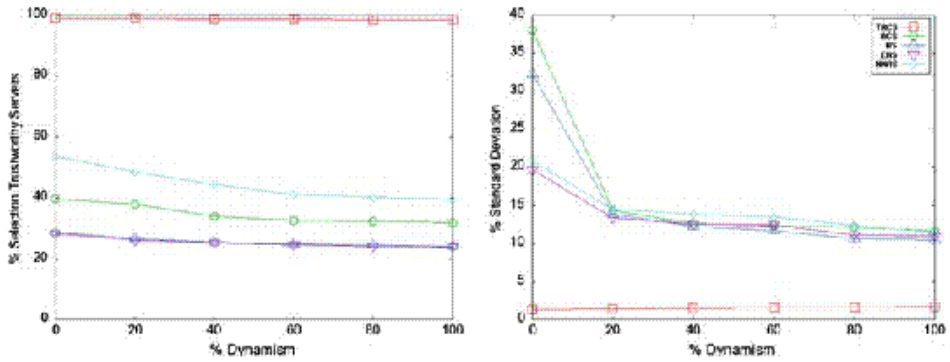


Fig. 2 Dynamic networks. ACO algorithms comparison

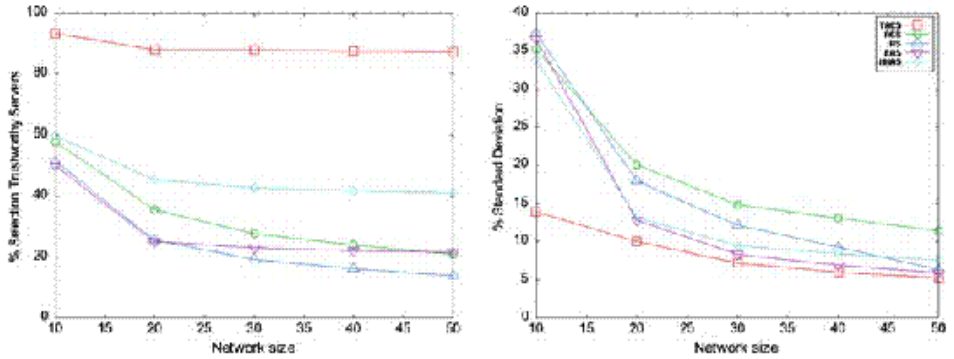


Fig. 3 Oscillating networks. ACO algorithms comparison

Therefore, ants may had to recalculate new routes leading to a benevolent server every 50 interactions. Figure 3 shows the outcomes.

As it can be observed, the behavior of TACS gets worse than the one presented in previous experiments. However, it is still the best of the compared algorithms with a selection percentage average always above the 87%. And it can be also observed that from 10 nodes and beyond, the behavior nearly changes, so we can say that TACS is nearly independent of the network size.

5.4 Multi-Service Networks

Real computer networks rarely offer just one single service, but they often provide several ones at the same time. Our model is able to deal with this kind of situations. And it is done by having different pheromone traces for each service provided by the network.

Thus, a server can be very benevolent delivering a certain service and very malicious with other different one. In such state ants will tend to select it in order to ask it for the first service, but will scarcely choose it to supply the second one.

So the behavior of the model is exactly the same regardless the service requested by the client, as each service has its own pheromone traces over the network. In this way our model is not only able to distinguish good servers from bad ones just for one service, but it can also differentiate within a particular server the trustworthiness it has delivering each one of the services it offers.

6 Conclusions and Future Work

In this paper an innovative P2P network trust model is presented based on the Ant Colony System where the traces of pheromone are identified with the trust that a client has on finding a benevolent server offering a certain service.

Our model is able to select a suitable server more than the 96% of the cases on static networks regardless of its size, and on dynamic networks regardless of its dynamism. Oscillating networks have also been solved with TACS.

Comparing our model with other classic Ant Colony Optimization algorithms we observe that the best outcomes are reached with TACS, which with its particular pheromone updating, transition rule and punishment method performs better than any other.

As far as we know this is the first work that uses a modified version of ACS in this way obtaining such good results. However, many technics can be incorporated and tested such as fuzzy logic or linguistic labels (making use of an ontology in order to represent concepts such as service, or satisfaction), and the model can be made as complex as desired, for example taking into account the possibility of having malicious relay nodes.

Acknowledgements This research work has been partially funded by secure widespread identities for federated telecommunications (SWIFT), FP7-ICT-2007-1, Grant No.: 215832) EU IST project, and by a Séneca Foundation grant within the Human Resources Researching Training Program 2007 (06826/FPI/07). Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

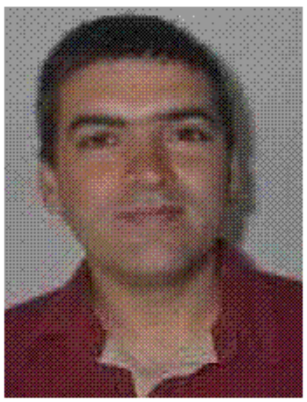
References

1. Almenárez, F., Marín, A., Campo, C., & García, C. (2004). PTM: A pervasive trust management model for dynamic open environments. In *Privacy and trust*. In *Proceedings of the First Workshop on Pervasive Security, Privacy and Trust, PSPT'04*. Boston, MA, USA.
2. Almenárez, F., Marín, A., Díaz, D., & Sánchez, J. (2006). Developing a model for trust management in pervasive devices. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*.
3. Carbó, J., Molina, J. M., & Dávila, D. (2003). Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems, 12*, 135–155.
4. Cordón, O., Herrera, F., & Stützle, T. (2002). A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware and Soft Computing, 9*, 2–3.
5. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach in the traveling salesman problem. *IEEE Transaction on Evolutionary Computing, 1*(1), 53–66.
6. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, 26*(1), 29–41.
7. Huang, C., Hu, H., & Wang, Z. (2006). A dynamic trust model based on feedback control mechanism for P2P applications. In *Autonomic and trusted computing* (pp. 312–321). Berlin: Springer.
8. Marsh, S. P. (1994). Formalising trust as a computational concept. PhD thesis, Department of Computing Science and Mathematics, University of Stirling.
9. Marti, S., & Garcia-Molina, H. (2006). Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks, 50*(4), 472–484.
10. Sabater, J., & Sierra, C. (2001). REGRET: Reputation in gregarious societies. In *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 194–195).
11. Sabater, J., & Sierra, C. (2002). Social ReGreT, a reputation model based on social relations. *SIGecom Exch, 3*(1), 44–56.
12. Stützle, T., & Hoos, H. (2000). MAX-MIN ant system. *Future Generation Computer Systems, 16*(8), 889–914.
13. Tajeddine, A., Kayssi, A., Chehab, A., & Artail, H. (2006). PATROL-F – A comprehensive reputation-based trust model with fuzzy subsystems. In *Autonomic and trusted computing* (pp. 205–217). Berlin: Springer.
14. Tang, Y., Sun, L., Luo, J., Yang, S., & Zhong, Y. (2006). TPOD: 2006A trust-based incentive mechanism for peer-to-peer live broadcasting. In *Autonomic and trusted computing* (pp. 332–341). Berlin: Springer.
15. Tian, H., Zou, H., Wang, W., & Cheng, S. (2006). A group based reputation system for P2P networks. In *Autonomic and trusted computing* (pp. 342–351). Berlin: Springer.
16. Wang, W., Zeng, G., & Yuan, L. (2006). Ant-based reputation evidence distribution in P2P networks. In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing* (pp. 129–132).
17. Yu, F., Zhang, H., Yan, F., & Gao S. (2006). An improved global trust value computing method in P2P system. In *Autonomic and trusted computing* (pp. 258–267). Berlin: Springer.
18. Zacharia, G., & Maes, P. (2000). Trust management through reputation mechanisms. *Applied Artificial Intelligence, 14*, 881–907.
19. Zhang, Y., Chen, H., & Wu Z. (2006). A social network-based trust model for the semantic web. In *Autonomic and trusted computing* (pp. 183–192). Berlin: Springer.
20. Zhuo, T., Zhengding, L., & Kai, L. (2006). Time-based dynamic trust model using ant colony algorithm. *Wuhan University Journal of Natural Sciences, 11*(6), 1462–1466.

Author Biographies

Félix Gómez Mármol is a Ph.D. student in the Department of Information and Communications Engineering of the University of Murcia. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received an M.Sc. in Computer Engineering from the University of Murcia.

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security and management of IP communication networks. He received an M.Sc. and Ph.D. in Computer Engineering from the University of Murcia.



Antonio F. Gómez Skarmeta is an associate professor at the University of Murcia, Spain. His research interests include advanced networking services and applications over IP networks, network security and mobility. He received an M.Sc. in Computer Science from the University of Granada and a Ph.D. in Computer Science from the University of Murcia.

3

META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm

Title:	META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez, Javier Gómez Marín-Blázquez
Type:	Journal
Journal:	Intelligent Automation and Soft Computing Journal
Impact factor (2008):	0.224 (position 51/53)
Publisher:	World Automation Congress
Volume:	
Number:	
Pages:	
Year:	2010
Month:	
DOI:	
State:	In press

Table 3: META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm

Intelligent Automation and Soft Computing, Vol. 16, No. X, pp. 1-19, 2009
Copyright © 2009, TSI® Press
Printed in the USA. All rights reserved

META-TACS: A TRUST MODEL DEMONSTRATION OF ROBUSTNESS THROUGH A GENETIC ALGORITHM

FÉLIX GÓMEZ MÁRMOL [*], **GREGORIO MARTÍNEZ PÉREZ AND JAVIER G. MARÍN-BLÁZQUEZ**
Departamento de Ingeniería de la Información y las Comunicaciones
Facultad de Informática, Campus de Espinardo, s/n
University of Murcia, 30.071 Murcia, Spain
{felixgm, gregorio, jgmarin}@um.es

ABSTRACT Ensuring trust and confidence in virtual communities' transactions is a critical issue nowadays. But even more important can become the use of robust and accurate trust models allowing an entity to decide which other entity to interact with. This paper aims to study the robustness of TACS (Trust Ant Colony System), a previously proposed bio inspired P2P trust model, when applying a genetic algorithm in order to find the range of values of its working parameters that provides the best TACS performance. The optimization of those parameters has been carried out using the CHC genetic algorithm. Experiments seems to demonstrate that TACS can achieve high performance ratios due to the enhancement provided by META TACS, and to achieve them for a wide range of working parameters, hence showing a remarkable robustness.

Key Words: Robust Trust Model, CHC Genetic Algorithm, TACS, Trust Ant Colony System, Trust management, P2P Networks, Meta heuristics

1. INTRODUCTION

Different research works have been done so far in order to improve security in P2P networks by ensuring a minimum level of confidence between every two interacting nodes. However, it is not definitively solved since none of the studied models has become a de facto standard in this field.

Moreover, there is a wide variety of trust and reputation models depending on their scope (P2P, Ad-hoc, Wireless Sensor Networks, multi-agent systems, etc.), their type (trust model, reputation one or even both), and the technique they use (fuzzy logic, Bayesian networks, etc).

Nevertheless, they are mainly focused on the way the trust and/or reputation values are computed and they do not manage neither how the node to have a transaction with is finally selected, nor which is the path leading to that certain node, nor how robust the model is.

In this paper we will first review a resilient trust model for P2P networks [24, 25] where some nodes offer some services or goods and other ones are requesting those services. The former will be always looking for the best self profit, while the latter will be demanding the best services with respect to some quality characteristics, such as the prize or delay, for instance.

[*] Corresponding author; phone number: +34 968 367866; fax number: +34 968 364151

The main feature of our model is the use of a bio-inspired Ant Colony System (ACS) algorithm [1] allowing to determine the route (sequence of nodes) leading to the most trustworthy server offering a particular service all over the network.

Ant Colony Optimization (ACO) [1, 2, 22, 23] is widely accepted as one of the most promising soft-computing algorithms when solving some kinds of optimization problems such as the traveling salesman problem.

In this type of optimization algorithm ants travel along the network searching the optimum path fulfilling certain conditions (for instance, the optimum path leading to a node offering certain service). While they are traveling they leave a trace of pheromone, called τ, which is used to guide other ants (the more pheromone trace a path has, the more probability it has to be chosen). Finally, the path with the strongest trace of pheromone will be selected as the optimum one.

But the main objective of this work is to propose and study the behavior of META-TACS, an optimizer of the working parameters of TACS via evolutionary computation. In previous work [24,25] TACS was proposed as a bio-inspired P2P trust model including some preliminary results. In the present work, a CHC (Cross generational elitist selection, Heterogeneous recombination, Cataclysmic mutation) genetic algorithm [26] was employed to implement META-TACS. META-TACS allows studying the performance and the robustness of TACS against a wide range of values for its working parameters. One of the results of the study was that TACS remains obtaining good outcomes almost regardless the combination of values of its working parameters, which suggests that it is both resilient and easily configurable. CHC was chosen because of its rather small population size (since evaluating a set of parameters, that is, an individual, is comparatively costly) while keeping a good balance between exploration and convergence. As we will see later, META-TACS provided us with a bounded range of possible values for each parameter of TACS. The long term objective was to provide TACS with mechanisms to self-adapt to the current network conditions, although the results show a remarkable resilience to diverse typical situations.

The rest of the paper is organized as follows: section 2 presents a set of related works in this area. In section 3 our model TACS is described. Then, section 4 presents a genetic algorithm designed to optimize TACS parameters, called META-TACS, whose experiments and results are described and analyzed in section 5. Finally, section 6 exposes some conclusions and future work.

2. RELATED WORK

A number of models have arisen in the field of trust and reputation recently. That is the case, for instance, of [3-8], where several trust and reputation models are proposed for a multi-agent system [9] in which agents interact (competition, coordination, cooperation...) in order to get the greatest self profit. In [10-16, 31], however, authors develop trust and reputation models mainly for P2P networks, although some of them could be used in a multi-agent system or even in an ad-hoc network. Ad-hoc and Wireless Sensor Network trust and reputation models are exposed, for example, in [17-19, 32].

Some of these models are based on fuzzy logic in order to represent trust and reputation values. Others rely on Bayesian networks and a posteriori probabilities. There are also models based on social networks. And some others just give analytic expressions to compute trust and reputation.

Among all the studied and analyzed works, we have just found two models where the bio-inspired ant colony optimization is used. They are [20, 21], but none of them face the problem as we do. AntRep [20], for instance, uses the ACS in order to distribute reputation information, while TDTM [21] requires the existence of a Public Key Infrastructure in the network. In our

opinion this can become a strong restriction since not all the P2P networks will be composed of devices with enough capabilities to support such cryptographic infrastructure.

All the cited models just give a trust and/or reputation value for each entity in the network, but any of them tells how to reach that entity. That is, any of them provide the path to follow in order to go to that certain node.

Many works have been done in order to formalize trust and categorize trust models [30] but, as far as we know, this is one of the first papers where a trust model optimization is presented. In this way, we decided to use a genetic algorithm (in particular, the CHC one, given its specific GA features) [26, 27] since it has been proved that evolutionary computation (EC) seems an intelligent option when dealing with optimization problems where there is not much knowledge available about the search space structure while, at the same time, EC is still able to obtain high quality solutions.

3. TACS OVERVIEW

TACS (Trust Ant Colony System) [24, 25] is a Trust model for P2P networks based on the bio-inspired algorithm ACS (Ant Colony System [1,2]) where the pheromone traces, $\tau \in [0,1]$, are identified with the confidence a client requesting a certain service has on finding a trustworthy server through a specific route. The heuristic component $\eta \in [0,1]$ (also involved in the decision of which path to choose) is identified with the similarity between the service requested by the client and the service actually offered by a certain server. And if a server does not offer the requested service then η is defined as the goodness of that server acting as a relay node.

In summary, the steps that compose this model are the following, as it is shown in Figure 1:

1. Client C executes TACS in order to find the "optimum" server S all over the network offering the desired service s
2. TACS launches the ACS algorithm and ants modify the pheromone traces of the network
3. TACS finishes, having selected the "optimum" path to server S'
4. TACS informs the client C that the "optimum" server found is S'
5. Client C requests the desired service s to the server S'

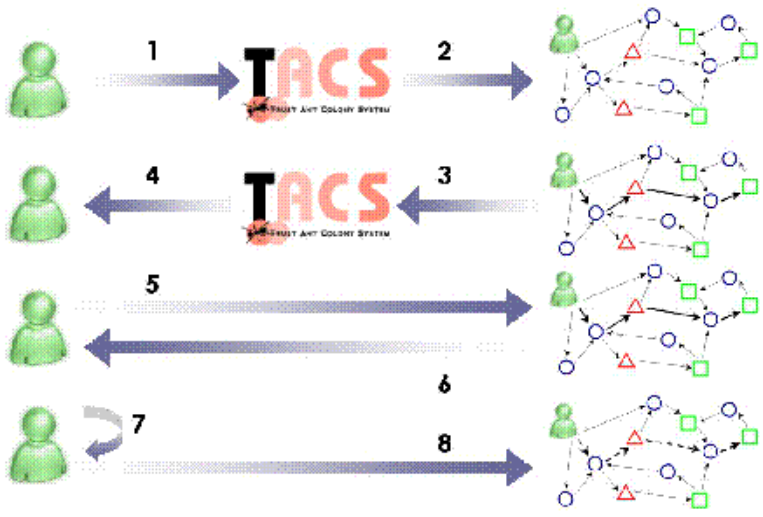


Figure 1. TACS Model Steps.

6. Server S' provides service s' to the client C
7. Client C evaluates his satisfaction with the received service s'
8. If client C is not satisfied with the received service s', he punishes the server S' evaporating the pheromone of the path that leads from C to S'. Otherwise no punishment is carried out

At the beginning of the process all the pheromone traces are initiated according to this formula:

$$\tau(0) = IniPh + (2 \cdot r - 1.0) \cdot IniPh \cdot (1.0 - IniPh) \quad (1)$$

where $IniPh \in [0,1]$ is a parameter indicating the initial pheromone value desired and $r \sim [0,1]$ is a random number within the interval $[0,1]$.

While an ant builds the path leading to the most trustworthy server, it modifies the pheromone traces along it. This pheromone local updating is carried out through the following expression:

$$\tau_{cs}(t) = (1-\varphi) \cdot \tau_{cs}(t-1) + \varphi \cdot z_1 \cdot \tau_{cs}(t-1) \quad (2)$$

where $\tau_{cs}(t)$ is the pheromone value of the edge e_{cs} (linking node c and s) at time t, $\varphi \in [0,1]$ is a constant called *phi* and z_1 is defined as follows:

$$z_1 = 1 + (1-\varphi) \cdot (1 - \tau_{cs}(t-1) \cdot \eta_{cs}(t-1))$$

What implies an increase of pheromone above the previous value, but never higher than a 100%. Moreover, with the term $(1 - \tau_{cs}(t-1) \cdot \eta_{cs}(t-1))$ we achieve that edges with lower values of pheromone can recover faster (increasing more its traces) and those who have higher values increase themselves slower.

Moreover, when all the ants (which number depends on the size of the network) have built a path, and the best of these has been chosen, an additional pheromone global updating is done along that selected route as follows:

$$\tau_{cs}(t) = (1-\rho) \cdot \tau_{cs}(t-1) + \rho \cdot z_2 \cdot \tau_{cs}(t-1) \quad (3)$$

where $\rho \in [0,1]$ is a constant called *rho*, and z_2 is defined as:

$$z_2 = 1 + \tau_{cs}(t-1) \cdot \eta_{cs}(t-1) \cdot Q(S_{Better\_Global})$$

where $Q(S_{Better\_global})$ is the quality of the best path that all the ants have found in one single iteration of the algorithm. Therefore now those edges with a higher value of τ and η are more rewarded than those with lower values. Thus, both expressions (2) and (3) have been designed in order to establish a good balance between exploration and convergence, when searching the most trustworthy server.

The quality of the path built by ant k, $Q(S_k) \in [0,1]$, is computed as follows:

$$Q(S_k) = \frac{A_k}{Length(S_k)^{PLF}} \cdot \bar{\tau}_k \quad (4)$$

where $A_k \in [0,1]$ is the ratio of ants that have selected the same path as the ant k, $Length(S_k)$ is the length of the solution S_k, $\overline{\tau}_k$ is the average pheromone of that path and $PLF \in [0,1]$ is a constant called *Path Length Factor*.

We think this is a reasonable good way of measuring the quality of a path (and the experiments have demonstrated it) since it takes into account three main factors: the ratio of ants who have chosen the same path, the average pheromone of that path and the length of the route. Having that expression we give a higher quality value to those paths which have been chosen by a larger number of ants, which have a greater average pheromone value and which are shorter in length.

When an ant k discovers a server offering the required service, it has to decide whether to stop and return the current path leading to that server, or keep on trying to find a better (more trustworthy) one. This decision could be expressed as the transition rule: if $\overline{\tau}_k > TraTh$ and $\overline{\tau}_k > r \sim [0,1]$ then ant k stops and returns current solution, where $\overline{\tau}_k$ is the average pheromone of the current path, $TraTh \in [0,1]$ is a constant called Transition Threshold and $r \sim [0,1]$ is a random number within the interval $[0,1]$.

And when an ant is currently in a server who does not offer the desired service, it has to move one step forward, choosing among the current node's neighbors. Let ant k be at node r, then the probability of choosing neighbor s as the next node in the path is computed as:

$$p_k(r,s) = \frac{[\tau_{rs}]^{\alpha} \cdot [\eta_{rs}]^{\beta}}{\sum_{u \in J_k(r)} [\tau_{ru}]^{\alpha} \cdot [\eta_{ru}]^{\beta}} \tag{5}$$

where $J_k(r)$ is the set of reachable nodes from *r* not visited yet by ant k and α, β are two weights establishing a balance between pheromone traces and heuristic values, respectively.
But specifically, the ACS adds a proportional probabilistic transition rule as follows:

$$p'_k(r,s) = \begin{cases} \arg\max_{u \in J_k(r)} [\tau_{ru}]^{\alpha} \cdot [\eta_{ru}]^{\beta} & \text{if } q \le q_0 \\ p_k(r,s) & \text{otherwise} \end{cases} \tag{6}$$

where $q \sim [0,1]$ is a random number within the interval $[0,1]$ and q_0 is the probability of choosing deterministically the most promising next node *u*.

As we have seen, at step 7 of TACS client C evaluates his satisfaction, $Sat \in [0,1]$, with the received service s'. To do so, he assesses the similarity between that service and the one initially requested: s. This assessment may be different for each client, since it depends on some weights (meaning client's preferences) that the own client can define by himself. It is modeled as follows:

$$Sat = Sim(s,s') = f(w_{prize}, w_{quality}, w_{delivery}) \tag{7}$$

Finally, the last step of the model consists of punishing a server in case he has provided a worse service than the one he initially offered. If $Sat \geq PunTh$, where $PunTh \in [0,1]$ is a constant called *Punishment Threshold*, then the punishment carried out is:

$$\tau_{cs} = \tau_{cs} - \varphi \cdot (1 - Sat) \cdot 2df_{cs} \tag{8}$$

where $df_{cs} \in [0,1]$ is a distance factor for edge e_{cs} that implies a stronger punishment for those edges closer to the malicious server and which is computed as follows:

$$df_{cs} = \sqrt{\frac{d_{cs}}{L \cdot (L - d_{cs} + 1)}}, \qquad d_{cs} = 1,2,\ldots,L-1$$

being L the actual length of the whole path and d_{cs} the distance of edge e_{cs} from the client.

Otherwise, if $Sat < PunTh$, the stronger punishment for e_{cs} is:

$$\tau_{cs} = \left(\frac{\tau_{cs}}{df_{cs}} - \varphi \right) \cdot Sat \tag{9}$$

The number of ants and iterations depends on the number of nodes of the network according to the expressions $N_A = N_N^{N_{ants}}$ and $N_I = N_N^{N_{iter}}$, respectively, where $N_{ants}, N_{iter} \in [0,1]$.

3.1 Advanced Features

Some features that can be derived from the mathematical model of TACS are now presented. This model allows the anonymity of the entities participating in the network since it does not need to associate an entity with its actual identity. Every entity is just given a pseudo-identifier per session not associated with its real identity. However, newcomers do not have more opportunities than non malicious remaining entities in the network. Otherwise an entity could achieve enough reputation to interact with other ones, then keep cheating until its level of reputation did not allow him to interact again (at least for awhile), and then leave and re-enter the network as a newcomer and start again. This is achieved in TACS by evaporating pheromone traces only when an unsuccessful interaction has been carried out.

Nonetheless, benevolent newcomers indeed have the ability to participate although there was a very trustworthy entity in the network, because those newcomers will receive gradually more ants depositing pheromone traces until they reach a certain level that allows them (in terms of probability) to be selected. Likewise, redemption of past malicious entities that has become benevolent is accepted. And an exploitation of a good built up reputation is avoided since the punishment for a high deception in a transaction is even higher (see equation (9)).

When evaluating the satisfaction perceived by a certain transaction, a subjectivity assessment is allowed since each client may define his own weights in order to compute the similarity between the requested service and the one actually received (as shown in equation (7)).

3.2 TACS Performance

The overhead added in this kind of networks is most of the times a critical issue since the interacting devices in a P2P, Ad-hoc or even Wireless Sensor Network usually have great constraints about memory, processing and communication capacity.

Therefore, we made some tests in order to measure the performance or throughput of our model. Our library [25] size is close to 31 Kbytes. Moreover, Figure 2 depicts the average time in milliseconds needed to perform one transaction (without taking into account transmission delays).

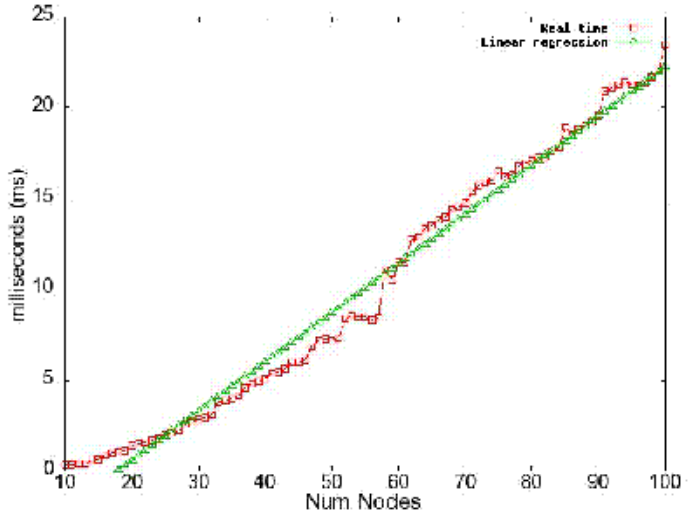


Figure 2. TACS Execution Time.

This graphic indicates the overhead introduced by TACS. It needs less than 25 milliseconds to be executed over a 100 nodes network and up to 0.33 ms when the network is only composed by 10 nodes. It can be checked that the performance nearly follows a linear function.

4. META-TACS

As we have seen, TACS includes several parameters involved in the model. And with the values shown in Table I we got good outcomes in comparison with other bio-inspired methods [24].

However, we thought that it would be interesting to have an optimization of these working parameters. Even more, it would be necessary to obtain knowledge about the behavior of TACS regarding the specific working parameters and trying to provide a more solid support for using a set of particular values, instead of only trying a few empirical values (as the ones shown in Table I, for instance).

That is the reason why we applied a genetic algorithm in order to optimize TACS parameters and give a formal support to those values. Our pursued intention was to check the behaviour of TACS model against a relatively wide range of values for each parameter (having in mind that every optimized parameter can take values within the interval $[0,1]$).

Genetic algorithms have been proved to be a good option when optimizing continuous variables [26], as it is our case. Specifically, the CHC algorithm [27], that stands for Cross generational elitist selection, Heterogeneous recombination, Cataclysmic mutation, was chosen. The most important reason as to choose CHC in particular in order to implement META-TACS lies in that it works with a small population size. For the present problem small population size is a convenient feature because evaluating an individual requires the execution of an instance of TACS. Although TACS is a remarkably fast algorithm such execution is comparatively costly compared with the other components of the CHC genetic engine. The other features found in CHC

Table I. TACS Parameters.

Parameter name	Value	Range	Meaning
φ	0.1	[0,1]	Pheromone local updating and punish and reward parameter
ρ	0.1	[0,1]	Pheromone global updating parameter
α	1.0	[0,1]	Learning weight in the transition rule
β	1.0	[0,1]	Heuristic weight in the transition rule
N_{ants}	0.35	[0,1]	Exponent to determine the number of ants
N_{iter}	0.35	[0,1]	Exponent to determine the number of iterations
q_0	0.98	[0,1]	Probability of choosing deterministically the most promising next node
$IniPh$	0.5	[0,1]	Initial pheromone trace
$TraTh$	0.5	[0,1]	Transition threshold, to determine if an ant must stop when it finds a node offering the requested service or not
$PunTh$	0.5	[0,1]	Punishment threshold, to determine if an edge must be punished or not
PLF	0.5	[0,1]	Path length factor, to determine the importance of the length of a path when measuring its quality

were designed to counterweight the weakness that small populations present. Elitist selection allows a monotonic improvement of the solution. A specially designed heterogeneous (and uniform) recombination, termed HUX, provides better sampling coverage when few individuals are available, aided by the incest prevention policy followed in CHC. With such aggressive crossover operator the typical mutation operator is not required. The cataclysmic mutation is performed when the expected premature convergence of small populations finally arrives. In all CHC is a well documented, good performance general purpose optimizer, with a small population size, and therefore seemed an adequate choice for META-TACS. Algorithm 1 shows its design.

In such algorithm L represents the size of an individual, M the number of individuals in a population and $r \in [0,1]$ a constant indicating the percentage of variation of the best individual when a re-initialization is carried out.

The hamming distance between two individuals t^1 and t^2 is computed as follows:

$$Hamming\_distance(t^1, t^2) = \sum_{i=1}^{L} hamming(t_i^1, t_i^2)$$

$$hamming(t_i^1, t_i^2) = \begin{cases} 0 & \text{if } |t_i^1 - t_i^2| < \delta \\ 1 & \text{otherwise} \end{cases}$$

where δ is a similarity threshold between the components of two individuals. Thus, if two individuals are very similar, they are not crossed, preventing this way the incest.
Moreover, instead of using the HUX cross, we implemented the BLX-α cross. That is, given two individuals $t^1 = (t_1^1, t_2^1, \ldots, t_L^1)$ and $t^2 = (t_1^2, t_2^2, \ldots, t_L^2)$, BLX-$\alpha$ cross generates another two individuals $t^k = (t_1^k, t_2^k, \ldots, t_L^k)$, where t_i^k is randomly generated within the interval

$$[t_{\min} - I\alpha, t_{\max} + I\alpha]$$

being $t_{\min} = \min\{t_i^1, t_i^2\}$, $t_{\max} = \max\{t_i^1, t_i^2\}$, $\alpha \in [0,1]$ and $I = t_{\max} - t_{\min}$. The bigger α is the greater is the diversity when searching the best individual.

```
d = L/4
initialize population P(t)
evaluate individuals in P(t)
while stop condition not satisfied do
  t = t + 1
  Copy all members of P(t-1) to C(t) at random
  /* HUX cross and incest prevention */
  for each of the M/2 pairs of members in C(t)
    if (hamming_distance(t^i, t^j)/2) > d
      swap half the differing bits at random
    else
      delete the pair of individuals from C(t)
  evaluate individuals in C'(t)
  /* Elitist selection */
  form P(t) from P(t-1) by replacing the worst members of P(t-1)
with the best members of C'(t)
  if P(t) equals P(t-1)
    d = d - 1
  /* Re-initialization */
  if d < 0
    replace P(t) with M copies of the best member of P(t-1)
    for all but one member of P(t)
      flip r × L bits at random
      evaluate individual
    d = r × (1 - r) × L
```

Algorithm 1. CHC algorithm.

In the re-initialization step, we did not use the method explained in algorithm 1 (i.e. flip $r \times L$ bits at random), since our individuals representation is not a bit string, but real parameters, that is $t_i \in \mathbf{R}$ (in fact $t_i \in [0,1]$). We used, therefore, a normal distribution with mean 0 and

standard deviation σ, $\mathcal{N}(0,\sigma)$, in order to diverge and reinitialize the individuals of a population. Specifically, the divergence is carried out as follows:

$$t_i = t_i + \mathcal{N}(0,\sigma)$$

The bigger $\sigma \in [0,1]$ is, the greater the divergence of a member when re-initialization is done. We set σ to 0.1, which means that almost all possible divergence is inside the ±0.3 range.

Since we developed our customized genetic algorithm in order to optimize the parameters shown in Table I (except for alpha and beta), each individual is represented as a tuple:

$$t = (\varphi, \rho, q_0, IniPh, N_{ants}, N_{iter}, TraTh, PunTh, PLF)$$

Thus, evaluating a member means executing TACS with the parameters values specified by that certain member. And its fitness is the average satisfaction of the client with the received service minus half the standard deviation ($\mu_{Sat} - \sigma_{Sat}/2$).

Finally, the stop condition for our CHC algorithm was:

$$fitness_{Best\_individual} > 1.0 - 10^{-6}$$

That value was considered that provided an acceptable while close-to-perfect result.

5. META-TACS EXPERIMENTS AND RESULTS

This section presents the whole set of experiments with their corresponding outcomes carried out in order to optimize TACS parameters through our customized CHC genetic algorithm.

As we have seen before, CHC algorithm has its own setting parameters. The values used for those parameters in all the experiments done can be observed in Table II.

Table II. META-TACS Parameters.

Parameter name	Value	Range	Meaning
M	40	[0,∞]	Number of individuals in a population
L	9	[0,∞]	Size of an individual
r	0.35	[0,1]	Percentage of variation of the best individual when a re-initialization is carried out
δ	0.01	[0,1]	Similarity threshold between the components of two individuals
α	0.3	[0,1]	Determines the diversity when crossing two individuals
σ	0.1	[0,1]	Divergence of individuals when a re-initialization is carried out

Here we defined the three same scenarios that were designed to test TACS alone [24, 25], that is, static networks, dynamic networks and oscillating ones. In fact, each individual fitness assessment requires an execution of TACS over 5000 random networks with 100 interactions for each of them (except for the oscillating scenario, where 1000 transactions are done, as we will explain later) using the parameters values contained in that specific individual.

5.1 Static Networks

Under these conditions, we launched one CHC genetic algorithm for each case of the first scenario. In other words, we launched a CHC algorithm for static networks with 10 to 20 nodes, one CHC algorithm for static networks with 20 to 30 nodes and so on (30 to 40 and 40 to 50).

Table III and Figure 3 show the mean, μ, the standard deviation, σ, the minimum and the maximum values for each one of the optimized parameters obtained within the first scenario.

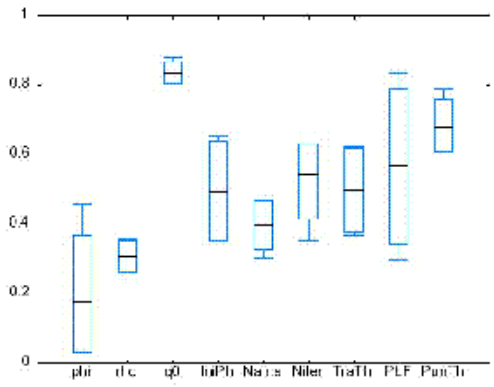
The first appreciation that can be done is that in a static environment φ can take a wide range of values, while ρ 's values fall into a smaller interval. φ is involved in the pheromone local updating (formulae (2)) and in the punishment method (formulae (8) and (9)), so the bigger φ is, the stronger the punishment is (and vice versa). However, the closer to 0.5 φ is, the greater is the pheromone local updating, and the closer to the extremes 0 or 1 φ is, the weaker is the pheromone local updating carried out.

ρ is only involved in the pheromone global updating, and a greater value of ρ implies a bigger updating (and vice versa). As it can be seen ρ is never greater than 0.5, its range of values is relatively small and on average, it is greater than φ.

About q_0, if it was equal to 0, every ant would use the transition rule described in equation (5) (direct exploration), otherwise, if $q_0 = 1$, the transition rule always used would be the one shown in equation (6) (probabilistic transition). In a static scenario, q_0 takes higher values with a low standard deviation.

Table III and Figure 3. META-TACS Outcomes for Static Networks.

	μ	σ	Max	Min
phi	0.1784	0.1903	0.4545	0.0319
rho	0.3045	0.0482	0.3576	0.2602
q0	0.8332	0.0321	0.8788	0.8057
IniPh	0.4929	0.1459	0.6514	0.3511
N_{ants}	0.3973	0.0689	0.4691	0.3032
N_{iter}	0.5404	0.1279	0.6301	0.3533
TraTh	0.4972	0.1214	0.6218	0.3689
PLF	0.5652	0.2226	0.8331	0.2961
PunTh	0.6806	0.0781	0.7896	0.6088
Fitness	0.9998	$9.7 \cdot 10^{-5}$	0.9999	0.9997



The most interesting thing to say about the $IniPh$ parameter is that its mean is close to 0.5 with a considerable standard deviation. This means that the best initial conditions for a client is neither to be very trustful, nor to be very untrustful, but a mean term with certain divergence.

In order to help understanding the meaning of the values of N_{ants} and N_{iter} (and even PLF), Figure 4 depicts the set of functions x^a where $a \in \{0.1, 0.2, \ldots, 0.9, 1.0\}$. As it can be observed, when the size of the network is less than 50 nodes, the difference between $N_N^{0.1}$ and $N_N^{0.5}$ is minimal. The difference begins to be relevant when N_{ants} or N_{iter} is greater than 0.5 (and even greater when the number of nodes increases).

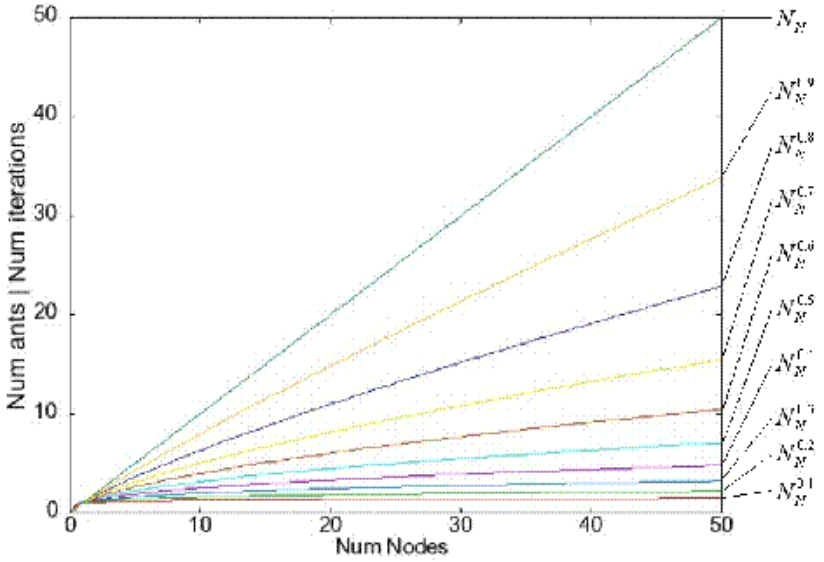


Figure 4. N_{ants} **and** N_{iter} **vs** N_N

Therefore, if N_{ants} and N_{iter} remain close to or less than 0.5, the difference between one value or another is not significant (maybe one or two ants/iterations of difference, which has no real strong effect in the results).

If $TraTh = 0$ then the probability that an ant *k* stops when it discovers a server offering the requested service is determined only by the average pheromone of the current path built by ant *k*, $\overline{\tau}_k$. That is, if $\overline{\tau}_k = 0.9$, for instance, ant *k* has a 90% of probability of stopping when discovering such a server. If $TraTh \to 1$, then $\overline{\tau}_k$ has to be close to 1 (a very trustworthy path) in order to let the ant to stop; otherwise it will keep on trying whenever it finds a server offering the requested service, until it reaches one of these servers without a neighbor to move forward.

PLF is used when assessing the quality of a path (see equation (4)). The bigger PLF is the more influence the path length has on its quality. The average length of the solutions depends on the number of nodes, but it is not usually very high [29]. Thus, there is no much difference between the values of *PLF* accepted in a static scenario. And since its mean value is near to 0.5, equation (4) may be viewed as:

$$Q(S_k) \approx \frac{A_k}{\sqrt{Length(S_k)}} \cdot \overline{\tau}_k \tag{10}$$

Finally, if $PunTh = 0$, every edge of the path leading to the most trustworthy server would be punished according to formula (8); otherwise, if $PunTh = 1$, the punishment applied to all those edges would be the one shown in formula (9). In this first scenario $PunTh$ takes higher values, thus meaning that a stronger punishment method is more suitable.

5.2 Dynamic Networks

The second tested scenario was similar to the first, but including dynamism. Note that P2P networks are strongly characterized by their high dynamism, where every node can enter or leave

the network at any moment, therefore the need of this type of scenario. In this dynamic scenario the topology of the network may change along the time. Table IV and Figure 5 show the outcomes of META-TACS for this scenario.

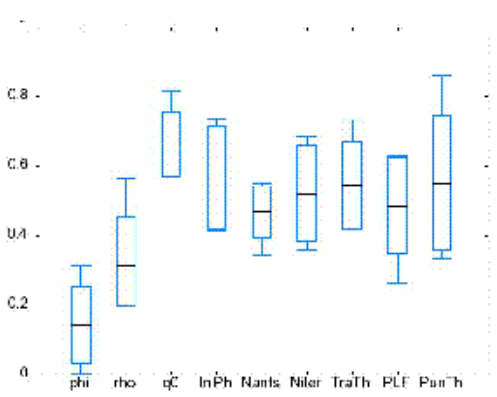
In this scenario φ parameter mean takes a very similar value than in the static scenario, but with a lower deviation, so it has a weaker pheromone local updating and punishment. ρ, however, also has a similar mean but this time the deviation is greater than in the first scenario. This means that greater and lower pheromone global updates are allowed in the dynamic scenario. It is deduced that this exchange of deviation sizes implies that in a dynamic P2P network a better balance is achieved by contributing with less pheromone in the local updating but with a greater trace in the global one, while the punishment is also low. As it can be observed, φ and ρ remain under 0.5 and ρ mean is still greater than φ's one.

As q_0 takes a lower value than in the first scenario, it seems that in a dynamic one is more suitable for ants to choose the next node to move forward using more often the expression (5). And this has sense since with equation (6) an ant always chooses the most promising node to move forward, and this might be good for a static environment. But in a dynamic scenario is better to sometimes use the probability defined in formula (5) to randomly select the next node of the path. Otherwise, following the most promising path could lead us to a dead end, since nodes can leave the network whenever they want.

IniPh has nearly the same standard deviation than in the first scenario, but the mean is higher. This can be interpreted as that in a dynamic environment is better to initially be a bit more confident.

Table IV and Figure 5. META-TACS Outcomes for Dynamic Networks.

	μ	σ	Max	Min
phi	0.1411	0.1096	0.3138	0.0000
rho	0.3116	0.1397	0.5642	0.1950
q0	0.6619	0.0924	0.8156	0.5707
IniPh	0.5655	0.1471	0.7344	0.4132
N_{ants}	0.4661	0.0748	0.5498	0.3435
N_{iter}	0.5208	0.1377	0.6839	0.3567
TraTh	0.5425	0.1264	0.7329	0.4165
PLF	0.4845	0.1395	0.6272	0.2608
PunTh	0.5501	0.1933	0.8613	0.3351
Fitness	0.9996	$3.5 \cdot 10^{-4}$	0.9999	0.9991



N_{iter} remains greater than N_{ants} in this second scenario, and the values of both parameters, as we explained before with Figure 4, are quite similar to those obtained for the first scenario.

The standard deviation of *TraTh* is also very similar in this environment than in the first one but, once again, here the mean is a bit greater than in the static scenario. This situation implies that ants do not stop when they find a server offering the requested service as often as they would do in a static network. In a dynamic network nobody guarantees that a server will stay in the network for a long time, so it is better that ants explore the network as much as they can.

PLF has here a range of values with lower bounds than in the first scenario, but its mean is close to 0.5 too, so the rewriting of formula (4) can be also applied here.

And the behavior of the $PunTh$ parameter is interesting. Here it has a lower mean, closer to 0.5, and a greater interval of values, which means that in a dynamic network the punishment scheme can vary from a quite hard and strict one until a soft and relaxed one. Or it can also be seen as that a good balance between strong and weak punishment is more suitable for dynamic networks.

5.3 Oscillating Networks

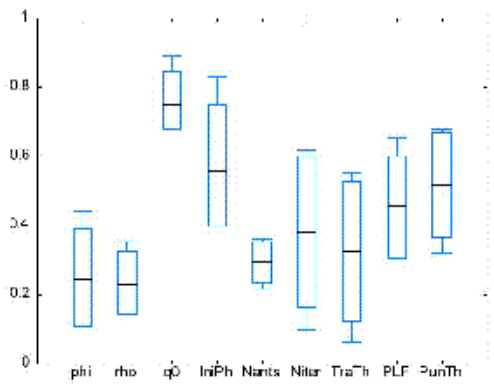
Finally, the third and last scenario consisted of a set of static networks where after every a certain number of transactions (50 in our case) the goodness of the currently selected most trustworthy server swapped, so it turned into a malicious one. In a P2P network any peer can suddenly change its goodness in order to cheat and try to get a greater self-profit. Thus, this kind of scenario aims to study this other type of dynamism, but this time focused on the behaviour of the nodes instead of the variability of the topology of the network In this oscillating scenario, in order to evaluate the fitness of each individual we executed TACS model over 5000 random networks with 1000 transactions per network with sizes of 10 to 20 nodes, 20 to 30, and so on. Outcomes of the genetic algorithm are shown in Table V and Figure 6.

A very significant issue derived from the experiments is that ρ parameter is now very close to φ. In fact it is a bit lower. But it can be because φ takes here the highest value and ρ its smallest one among the three tested scenarios. Remember that this meant that the pheromone local updating and the punishment are both greater, and the pheromone global updating is weaker.

Parameter q_0 has here a greater value than in the dynamic scenario, but lesser than in the static one, so in an oscillating network it is also better for an ant to choose the best path to follow by using equation (6) (probabilistic transition rule) rather than equation (5) (direct exploration).

Table V and Figure 6. META-TACS Outcomes for Oscillating Networks.

	μ	σ	Max	Min
phi	0.2460	0.1451	0.4417	0.1092
rho	0.2294	0.0947	0.3531	0.1429
q0	0.7496	0.0964	0.8888	0.6793
IniPh	0.5558	0.1939	0.8307	0.3987
N_{ants}	0.2926	0.0605	0.3617	0.3435
N_{iter}	0.3812	0.2198	0.6148	0.0985
TraTh	0.3243	0.2031	0.5497	0.0637
PLF	0.4543	0.1468	0.6546	0.3023
PunTh	0.5181	0.1528	0.6771	0.3175
Fitness	0.9995	$7.3 \cdot 10^{-4}$	0.9999	0.9983



The highest standard deviation for $IniPh$ is reached here, but its range of values is quite similar to those obtained in previous scenarios, that is, the most profitable initial conditions consist of not being too confident, neither too unconfident.

N_{ants} takes here its lowest values for both the mean and the standard deviation, while N_{iter} takes the lowest mean, but the highest standard deviation. This implies that, on the one hand a balance between these two parameters is necessary, and on the other hand, N_{iter} is in average term higher than N_{ants}.

$TraTh$ also has its lowest mean and highest standard deviation in the oscillating scenario, which means that the probability of an ant of stopping when it discovers a server offering the desired service is mainly determined by the average pheromone of the path currently built.

The lowest mean of PLF parameter is obtained in this scenario, but its range of values is very similar to the previous ones, and the rewriting of formula (4) done in formula (10) is also valid here.

Again, another lowest mean is reached here. And it is the $PunTh$ parameter. But its values interval is quite similar than the one achieved in the dynamic scenario, so it has the same impact it had there.

In summary, this scenario has the most extreme values among the three tested environments for many parameters. And the reason for this is that an oscillating scenario is the most changing and aggressive for our model (in fact the "worst" fitness values are also achieved here).

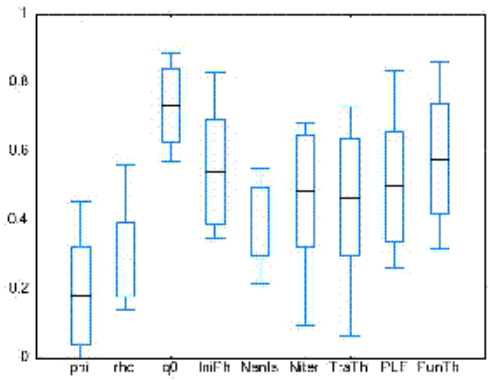
5.4 Global Outcomes

Finally, Table VI and Figure 7 summarize the global outcomes (average of the three tested scenarios) for META-TACS model. These outcomes can give us an interesting perspective of the performance of our model when there is no information about the behaviour of the network nodes or the stability of the topography of the network where it is going to be deployed.

As it can be seen, the standard deviation of every parameter goes from 0.1 to 0.16, approximately, which are high values since all the optimized parameters take values within the interval $[0,1]$. That is, all of them have a standard deviation between the 10% and the 16%, approximately.

Table VI and Figure 7. Global META-TACS Outcomes.

	μ	σ	Max	Min
phi	0.1817	0.1409	0.4545	0.0000
rho	0.2861	0.1073	0.3531	0.1429
q0	0.7359	0.1057	0.8888	0.5707
IniPh	0.5420	0.1516	0.8307	0.3510
N_{ants}	0.3969	0.0982	0.5498	0.2162
N_{iter}	0.4865	0.1645	0.6840	0.0985
TraTh	0.4672	0.1680	0.7329	0.0637
PLF	0.4989	0.1611	0.8330	0.2608
PunTh	0.5782	0.1608	0.8613	0.3175
Fitness	0.9996	$4.4 \cdot 10^{-4}$	0.9999	0.9983



We did not expect with these experiments to obtain a very accurate value for each one of the TACS parameters, but to demonstrate the robustness of the model against a certain range of input values for those parameters. As we can see, standard deviations take most of the times high values, which means a wider interval where to take values from for a certain parameter, obtaining however good outcomes (the worst one is a 99.83%, as it can be seen in Table V).

Some general relations between TACS parameters can be derived from the experiments carried out with the CHC genetic algorithm. These relations are described in Table VII.

These relations are just rough general approximations. In future work some rules relating several parameters could be extracted through a data mining process, for instance.

Table VII. TACS Parameters Relations.

$\rho \geq \varphi$
$N_{iter} > N_{ants}$
$q_0 > 0.5$
$IniPh \rightarrow 0.5$
$PLF \rightarrow 0.5$

6. CONCLUSIONS AND FUTURE WORK

This paper has studied the robustness of TACS (Trust Ant Colony System), a novel trust model for P2P networks using the bio-inspired algorithm of Ant Colony System. It has described its main characteristics and features and shown its performance.

A genetic algorithm based on the CHC algorithm has been applied in order to optimize the working parameters of TACS model and obtaining better knowledge about the behaviour of TACS with regards to them. The performance optimizations of this genetic algorithm have demonstrated the robustness of TACS model against a wide range of working parameter values. That is, TACS seems to work reasonably well regardless the values given for its parameters (within a certain interval).

Besides, this work opens some future ways of research. For instance, some rules relating TACS parameters could be obtained through a data mining process as to better adapt to the particular scenario found when applied in real networks. Although TACS has been shown as a robust system within wide parameter ranges it is always a recommendable safeguard to set the parameters in the best parameter setting areas depending on the type of network at hand. Therefore a self-adapting system based on meta-knowledge rules or metaheuristics may prove a useful addition to META-TACS in the future.

We have focused our model on P2P networks; nevertheless, other environments such as wireless sensor networks, ad-hoc, or even V2V (Vehicular-to-Vehicular) networks are also firm candidates for applying our model on them.

We are also planning to implement and test our model over a simulation environment such as OMNeT++ [28] and even participate in workbenches competitions such as ART Testbed [29].

Finally we are thinking of introducing ontologies, as well as fuzzy sets, in our model in order to provide it with some semantics. For instance, concepts as the similarity between two services could be better modelled in this way, in our opinion.

ACKNOWLEDGEMENTS

This work has been supported by a Seneca Foundation grant within the Human Resources Researching Training Program 2007. Thanks also to the Funding Program for Research Groups of Excellence with code 04552/GERM/06 also granted by the Seneca Foundation.

REFERENCES

1. M. Dorigo and T. Stüztle, "Ant Colony Optimization," *Bradford Book*. 2004.
2. O. Cordón, F. Herrera, and T. Stüztle, "A review on the ant colony optimization metaheuristic: Basis, models and new trends," *Mathware and Soft Computing* 9 (2-3), pp 141-175, 2002.

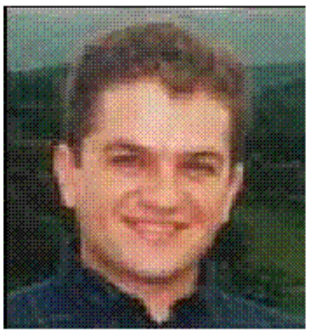
3. G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intelligence* (14), pp 881-907, 2000.
4. J. Sabater and C. Sierra, "Regret: reputation in gregarious societies," *Fifth International Conference on Autonomous Agents*, Montreal, Canada, pp 194-195, 2001.
5. J. Sabater and C. Sierra, "Social ReGreT, a reputation model based on social relations," *SIGecom Exch* 3 (1), pp 44-56, 2002.
6. J. Carbó, J. Molina, and J. Dávila, "Trust management through fuzzy reputation," *International Journal of Cooperative Information Systems* 12, pp 135-155, 2003.
7. S. Songsiri, "Mtrust: A reputation-based trust model for a mobile agent system," *Autonomic and Trusted Computing*, Third International Conference, ATC 2006, Wuhan, China, pp. 374-385.
8. L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation," *35th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington DC, USA, 2002.
9. Y. Shoham, "Agent-Oriented Programming," *Artificial Intelligence* 60 (1), pp 51-92, 1993.
10. F. Almenárez, A. Marín, C. Campo, and C. García, "PTM: A pervasive trust management model for dynamic open environments," *Privacy and Trust*, First Workshop on Pervasive Security and Trust, Boston, USA, 2004.
11. Y. Wang, V. Cahill, E. Gray, C. Harris, and L. Liao, "Bayesian network based trust management," *Autonomic and Trusted Computing*, no. 4158 in LNCS, Third International Conference, ATC 2006, Springer, Wuhan, China, pp 246-257, 2006.
12. A. Tajeddine, A. Kayssi, A. Chehab, and H. Artail, "PATROL-F – a comprehensive reputation-based trust model with fuzzy subsystems" , *Autonomic and Trusted Computing*, no. 4158 in LNCS, Third International Conference, ATC 2006, Springer, Wuhan, China, pp 205-217, 2006.
13. S. Kamvar, M. Schlosser, and H. García-Molina, "The EigenTrust Algorithm for Reputation Management," *P2P Networks*, Budapest, Hungary, 2003.
14. C. Huang, H. Hu, and Z. Wang, "A dynamic trust model based on feedback control mechanism for P2P applications," *Autonomic and Trusted Computing*, no. 4158 in LNCS, Third International Conference, ATC 2006, Springer, Wuhan, China, pp 312-321, 2006.
15. K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," *10th International Conference on Information and Knowledge Management*, 2001.
16. L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities," *IEEE Transactions on Knowledge and Data Engineering* 16 (7), pp 843-857, 2004.
17. S. Buchegger and J. Y. Le Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks," *Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge MA, USA, 2004.
18. A. Abul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities," *33rd Hawaii International Conference on System Sciences*, Hawaii, USA, 2000.
19. A. Boukerche, L. Xu, and K. El-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications* 30 (11-12), pp 2413-2427, 2007.
20. W. Wang, G. Zeng, and L. Yuan, "Ant-based reputation evidence distribution in P2P networks," *Fifth International Conference on Grid and Cooperative Computing*, IEEE Computer Society, Changsha, Hunan, China, pp 129-132, 2006.
21. T. Zhuo, L. Zhengding, and L. Kai, "Time-based dynamic trust model using ant colony algorithm," *Wuhan University Journal of Natural Sciences* 11 (6), pp 1462-1466, 2006.

22. M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, "Ant Colony Optimization and Swarm Intelligence," *5th International Workshop*, ANTS 2006, Vol. 4150 of LNCS, Springer, Brussels, Belgium, 2006.
23. T. Stützle and H.H. Hoos, "MAX-MIN Ant System," *Future Generation Computer Systems* 16 (8), pp 889-914, 2000.
24. F. Gómez Mármol, G. Martínez Pérez, and A.F. Gómez Skarmeta, "TACS, a Trust Model for P2P Networks," *Wireless Personal Communications*, Special Issue on "Information Security and data protection in Future Generation Communication and Networking'," to appear. Camera-ready version available on-line at: http://ants.dif.um.es/staff/felixgm/pub/GomezMarmol-TACS.pdf
25. TACS – Trust Ant Colony System, http://ants.dif.um.es/staff/felixgm/research/tacs
26. D.E. Goldberg, "Genetic Algorithms in Search," *Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston MA, USA, 1989.
27. L. Eshelman, "The CHC Adaptive Search Algorithm: How to Have a Safe Search When Engaging in Non-traditional Genetic Recombination," *Foundations of Genetic Algorithms*, Morgan Kaufman, San Mateo CA, USA, pp 265-283, 1991.
28. OMNeT++ Site, "OMNeT++ Discrete Event Simulation System," http://www.omnetpp.org
29. A. Vázquez, R. Pastor-Satorras, and A. Vespignani, "Internet topology at the router and autonomous system level," 2002, e-print cond-mat /0206084.
30. S. Marti and H. García-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Computer Networks* 50, pp 472-484, 2006.
31. X. Chen, K. Zhao, and X. Chu, "SepRep: A Novel Reputation Evaluation Model in Peer-to-Peer Networks," *Autonomic and Trusted Computing*, no. 5060 in LNCS, 5th International Conference, ATC 2008, Springer, Oslo, Norway, pp 86-99, 2008.
32. Z. Yan, V. Niemi, and G. Yu, "A User Behavior Based Trust Model for Mobile Applications," *Autonomic and Trusted Computing*, no. 5060 in LNCS, 5th International Conference, ATC 2008, Springer, Oslo, Norway, pp 455-469, 2008.

ABOUT THE AUTHORS

F. Gomez Marmol is a PhD student in the Department of Information and Communications Engineering of the University of Murcia. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received an MSc in computer engineering from the University of Murcia.

G. Martinez Perez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security and management of IPv4/IPv6 communication networks. He received an MSc and PhD in computer engineering from the University of Murcia.



J. G. Marín-Blázquez received his B.Sc. (Honors) and M. Sc. degrees in Computer Science from the University of Murcia, Spain, in 1992 and 1994. He was a Research Assistant at the University of Murcia from 1994 to 1997. He received a M. Sc. and a Ph. D. in Artificial Intelligence from The University of Edinburgh, UK. in 1998 and 2002 respectively. He was a Junior lecturer at The University of Edinburgh and did his postdoc at Napier University, UK until 2004.
He is currently a "Ramon y Cajal" program researcher in the Department of Information and Communications Engineering in the Faculty of Computer Science, University of Murcia, Spain. His research interest includes several branches of Computational Intelligence, including Evolutionary Computation, Linguistic Fuzzy Logic, Artificial Intelligence for Videogames, and Meta and Hyperheuristics.

4

Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique

Title:	Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Telecommunication Systems Journal
Impact factor (2008):	0.396 (position 47/67)
Publisher:	Springer
Volume:	46
Number:	2
Pages:	
Year:	2010
Month:	February
DOI:	http://dx.doi.org/10.1007/s11235-010-9281-7
State:	In press

Table 4: Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique

Telecommun Syst
DOI 10.1007/s11235-010-9281-7

Providing trust in wireless sensor networks using a bio-inspired technique

Félix Gómez Mármol · Gregorio Martínez Pérez

© Springer Science+Business Media, LLC 2010

Abstract Wireless Sensor Networks (WSNs) are becoming more and more spread and both industry and academia are focusing their research efforts in order to improve their applications. One of the first issues to solve in order to achieve that expected improvement is to assure a minimum level of security in such a restrictive environment. Even more, ensuring confidence between every pair of interacting nodes is a critical issue in this kind of networks. Under these conditions we present in this paper a bio inspired trust and reputation model, called BTRM WSN, based on ant colony systems aiming at providing trust and reputation in WSNs. Experiments and results demonstrate the accuracy, robustness and lightness of the proposed model in a wide set of situations.

Keywords Trust & reputation management · Wireless sensor networks · Bio inspired algorithms

1 Introduction

WSNs [1] are networks based on small size nodes cooperation. Those nodes are mainly characterized by their low energy consumption, their low cost and, of course, their wireless communication. They can be used to make measurements of temperature, pressure, humidity, lightness, etc, but currently they often have certain probabilities of failure, as well as high restrictions of computing, memory and energy capabilities.

WSNs are usually composed of a large number of these nodes which, together with their highly dynamic topology, may lead to some scalability problems.

A number of research groups are working on them since this kind of networks has several interesting applications ranging from military ones to environmental ones, passing through sanitary applications, domotics, Intelligent Transportation Systems (ITS) [2], etc.

However, due to their important restrictions, they usually suffer from many security weaknesses, which make them often vulnerable to certain threats. Hardware failures could be a source of wrong critical information spreading, for instance. But even more, nodes belonging to a WSN could misbehave when they are asked for a measurement, or some data.

Without loss of generality, we will adopt the scheme where some nodes of the network request some services (and act, therefore, as clients) and some others provide those services (thus acting as servers or services providers). In such a scenario, a node could provide a fraudulent service when this is requested.

In addition, since we have supposed one of the most restrictive cases, where every sensor is only able to communicate with its direct neighbors (that is, it cannot establish a direct communication with a node more than one hop ahead), a malicious node could avoid reaching its benevolent neighbors, or leading always to other malicious nodes, forming thus a collusion.

It is therefore necessary to accurately distinguish trustworthy nodes from fraudulent ones. This trustworthy nodes identification can be achieved through a trust and reputation model [3, 4].

F. Gómez Mármol (✉) · G. Martínez Pérez
Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30.071 Murcia, Spain
e-mail: felixgm@um.es

G. Martínez Pérez
e-mail: gregorio@um.es

Published online: 18 February 2010

In this paper we specifically present a trust and reputation model for WSNs, called BTRM WSN (Bio inspired Trust and Reputation Model for Wireless Sensor Networks) in order to carry out the selection of the most trustworthy node through the most reputable path offering a certain service.

Our proposed model is based on a bio inspired algorithm called ant colony system (ACS) [5 8], where ants build paths fulfilling certain conditions in a graph. These ants leave some pheromone traces that help next ants to find and follow those routes.

Although ACS was initially mainly designed for static networks, experiments demonstrate that the adaptations done to make it suitable for WSNs lead to an accurate performance of the model. As we will see later, it allows a client to interact most of the times with a trustworthy server, rather than with a misbehaving one.

The rest of the paper is organized as follows: Sect. 2 presents a review of a number of trust and reputation models and works oriented to WSNs. In Sect. 3 we present our trust model proposal, describing its main features and characteristics. An interesting analysis of some derived security threats that could be applied in our model is shown in Sect. 4. Experiments and results are exposed in Sect. 5 and, finally, Sect. 6 shows our conclusions and future work.

2 Background and related work

In this section we will present a review of some of the most relevant and novel trust and reputation models over Wireless Sensor Networks.

2.1 ATRM

ATRM [9] is an agent based trust and reputation management scheme for WSNs where trust and reputation management is carried out locally with minimal overhead in terms of extra messages and time delay.

It is based on a clustered WSN with backbone, and its core is a mobile agent system. It requires a node's trust and reputation information to be stored in the forms of t instrument and r certificate by the node itself. In addition, ATRM requires that every node locally hold a mobile agent that is in charge of administrating the trust and reputation of its hosting node.

Considering any two nodes n_i and n_j, the t instrument issued by n_i to n_j under context $C_\star$ is defined as:

$$TI(n_i, n_j, C_\star) = E_{AK}(D, H(D))$$

where $E_{AK}(M)$ is an encryption function using n_i's symmetric key, $H(M)$ is a hash digest function, $D = (ID(n_i), ID(n_j), C_\star, T, t_{i,j})$, T is a time stamp implying the time when the t instrument is issued and $t_{i,j}$ is the trust evaluation made by n_i on n_j.

If there are k concerned contexts, for any node n_i, its r certificate is defined as:

$$RC(n_i) = E_{AK}(R, H(R))$$

where $R = (ID(n_i), T, ((r_1, C_1), (r_2, C_2), \ldots, (r_k, C_k)))$, which means that n_i's reputation is r_1 under context C_1, r_2 under context $C_2, \ldots, r_k$ under context C_k at time point T.

Before starting any transaction between n_i and n_j, the former asks its local mobile agent to obtain the r certificate of the latter by directly querying n_j's local mobile agent. Based on n_j's r certificate, n_i decides whether or not to start the transaction.

After the transaction is finished, n_i makes a trust evaluation on n_j based on the quality of the service it gets, and then submits this evaluation to its local mobile agent which then accordingly generates a t instrument for n_j and sends it to n_j's local mobile agent.

Based on the collected t instruments, a mobile agent periodically issues its hosting node updated r certificates. But since mobile agents are designed to travel over the entire network and run on remote nodes, they must be lunched by trusted entities.

Therefore, in ATRM it is assumed that (1) there is a trusted authority that is responsible for generating and launching mobile agents, and (2) mobile agents are resilient against the unauthorized analysis and modification of their computation logic.

2.2 QDV

Authors of [10] present an Ant Colony Optimization approach for reputation and quality of service based security in WSNs. They specifically propose a quality based distance vector protocol known as QDV, where the more reputation a node has, the more reliable it is for communication purposes.

QDV is able to protect the network against packet injection by those malicious nodes which have been detected. This protection is made by identifying those nodes who drop the packets forwarded to them.

In this model reputation is based on pheromone content of a path for communication. Thus, a path having more deposits of pheromone, τ_{ij}, is considered more secure. On the other hand, QoS considers the distance between two communicating nodes, η_{ij}. Therefore:

$$\phi_{ij}(t) = \frac{\sum_{k=1}^{n_i} \tau_{kj}}{n_i}$$

where τ_{kj} is the pheromone trace between nodes k and j, n_i is the number of i's neighbors, and if $\phi_{ij}(t) < \tau_{min}$, misbehavior or security violation is detected, which means node i has less forwarding capabilities.

In the same direction, QoS is defined as the percentage of exposed traffic according to:

$$\theta_{ij}(t) = \frac{\sum M_g(t) + \sum M_r(t) - \sum M_d(t)}{\sum M_g(t) + \sum M_r(t)}$$

being $\sum M_g(t)$, $\sum M_r(t)$ and $\sum M_d(t)$ the total number of generated, received and dropped packets.

Finally, the quality of security, QSec, depends on the two previous parameters and defines the communication and transfer between two nodes. It is the deciding factor as to which node needs to be selected as the next node in the path and is computed as the weighted sum of reputation and QoS:

$$W_n(t) = w_1\phi_{ij}(t) + w_2\theta_{ij}(t)$$

2.3 ATSN

An agent based trust model for WSN is presented in [11] using a watchdog scheme to observe the behavior of nodes and broadcast their trust ratings. The sensor nodes receive the trust ratings from the agent nodes, which are responsible for monitoring the former and computing and broadcasting those trust ratings. According to the received information, sensor nodes will make the decision about cooperate with their neighbors or not.

In ATSN the reputation space is defined as $RS = \{(p,n) \mid p,n \in \mathbb{N}\}$, where p is the number of positive outcomes and n is the number of negative ones. Given (p,n) the probability x of obtaining a positive outcome is computed as follows:

$$P_{(p,n)}(x) = P(x \mid (p,n)) = \frac{x^p(1-x)^n}{\int_0^1 x^p(1-x)^n dx}$$

Additionally, the certainty of event (p,n) is calculated with the next expression:

$$c(p,n) = \frac{1}{2}\int_0^1 \left| \frac{x^p(1-x)^n}{\int_0^1 x^p(1-x)^n dx} - 1 \right| dx$$

Moreover, the trust space is defined as a triple $TS = \{(pt,nt,ut)\}$, satisfying the following conditions:

$$\begin{cases} pt, nt, ut \in [0,1] \\ pt + nt = c \\ pt + nt + ut = 1 \end{cases}$$

where pt, nt and ut refer to positive trust, negative trust and uncertainty, respectively.

Let now $T = (pt,nt,ut)$ be the transformation from reputation space to trust space, where pt, nt and ut are computed according to the next formula:

$$\begin{cases} pt = c\frac{p+1}{p+n+2} \\ nt = c\frac{n+1}{p+n+2} \\ ut = 1 - pt - nt \end{cases}$$

2.4 RFSN

RFSN [12] is a framework where sensor nodes maintain reputation for other nodes in the network. A node monitors through a watchdog mechanism the behavior of other nodes, based on which it builds up their reputation over time. It uses this reputation to evaluate trustworthiness and in predicting their future behavior. At the time of collaboration, a node only cooperates with those nodes that it trust.

Thus, a data structure termed reputation table RT_i is defined where reputations maintained by node i are stored.

$$RT_i = \{R_{ij}\}$$

being R_{ij} the reputation of node j maintained by node i. A node builds each of these entries in the reputation table over time through the watchdog mechanism as follows

$$R_{ij} = f(D_{ij}, R_{ij})$$

where the output of the watchdog mechanism, D_{ij}, is used to recursively update the reputation of node j at node i. D_{ij} represents the rating that is allocated to the latest action of node j by node i.

Moreover, in RFSN the reputation of a node is a made up of two subcomponents, $(R_{ij})_D$ and $(R_{ij})_{ID}$, as shown next

$$R_{ij} = (R_{ij})_D + (R_{ij})_{ID}$$

Direct reputation $(R_{ij})_D$ is build up using direct observations through the watchdog mechanism and indirect reputation $(R_{ij})_{ID}$ is build up using second hand information. But node i should give more weight to the second hand information received from a highly reputed node and vice versa. Therefore, $(R_{ij})_D$ and $(R_{ij})_{ID}$ are computed as follows

$$(R_{ij})_D = f(D_{ij}, (R_{ij})_D), \quad \forall j \in N_i$$

$$(R_{ij})_{ID} = (R_{ij})_{ID} + w_{ik} \times R_{kj}, \quad \forall k \in N_i$$

where $w_{ik} = g(R_{ik})$ represents the weight that is derived based on the reputation between the two nodes i and k, R_{ik}.

Trust is obtained in RFSN by taking the statistical expectation of the probability distribution representing the reputation between those nodes, i.e., $T_{ij} = E(R_{ij})$.

Finally, when faced with the question of cooperating with a node j in the network, the behavior of node i, B_{ij}, is derived from the trust metric of the two nodes. B_{ij} is a binary variable $\{cooperate, don't\ cooperate\}$ and a simple threshold based policy is used to decide the value of B_{ij}.

2.5 CORE

CORE [13] is a generic mechanism based on reputation to enforce cooperation among nodes in a MANET in order to prevent selfish behavior. All members of a community have to contribute to the community life in order to be entitled to use its resources. In CORE reputation is a measure of someone's contribution to common operations and it is defined as compositional.

That is, the overall opinion on an entity that belongs to the community is obtained as a result of the combination of different types of evaluations. Authors of CORE define a subjective reputation, an indirect reputation and a functional reputation.

The first one is the reputation calculated directly from a subject's observation as follows:

$$r^f_{s_i}(s_j\ f) = \sum \rho(t, t_k) \cdot \sigma_k$$

where $r^f_{s_i}(s_j\ f) \in [-1, 1]$ stands for the subjective reputation value calculated at time t by subject s_i on subject s_j with respect to the function f; $\rho(t, t_k)$ is a time dependent function that gives higher relevance to past values of σ_k and $\sigma_k \in [-1, 1]$ represents the rating factor given to the k th observation.

The indirect reputation of subject s_j collected by s_i at time t for the function f is denoted as $ir^f_{s_i}(s_j\ f)$, and can take only positive values, preventing thus denial of service attacks based on malicious broadcasting of negative ratings for legitimate nodes.

Finally, the functional reputation refers to the subjective and indirect reputation calculated with respect to different functions f. All these types of reputation are combined to assess a global value of a subject's reputation, using the following formula:

$$r^f_{s_i}(s_j) = \sum w_k \cdot (r^f_{s_i}(s_j\ f_k) + ir^f_{s_i}(s_j\ f_k))$$

where w_k represents the weight associated to the functional reputation value and $r^f_{s_i}(s_j)$ is the global reputation value that is evaluated in every node. The choice of the weights w_k used to evaluate the global reputation has to be accurate because it can affect the overall system robustness.

Each entity s_i in CORE is enriched with a set of reputation tables (RT) and a watchdog mechanism (WD). Each row in the RT consists of four entries: the unique identifier of the entity, a collection of recent subjective observations made on that entity's behavior, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated for a predefined function. Each network entity has one RT for each function that has to be monitored. The RT and the WD together constitute the basis of the collaborative reputation mechanism presented in this model.

2.6 DRBTS

DRBTS [14] is a distributed security protocol aimed at providing a method by which beacon nodes (nodes that assist other sensor nodes to determine their location), BN, can monitor each other and provide information so that sensor nodes, SN, can choose who to trust, based on a quorum voting approach. In order to trust a BN's information, a sensor must get votes for its trustworthiness from at least half of their common neighbors.

Let's consider a WSN consisting of n SN, $s_1, s_2, \ldots, s_n$ and m BN, $b_1, b_2, \ldots, b_m$. If a BN reports a trust value over a SN's threshold for another BN, the sensor counts that as a positive vote from the first BN to the second.

There are two classifications of information available for the reputation system. On the one hand, the first hand information is the location information transmitted by a BN, overheard by another BN in its communication range. On the other hand, the second hand information is the reputation information gathered by a BN and published while responding to a request for location information. Both these types of information are used by the BN to update the reputation of their neighbors.

The reputation of b_i from b_k point of view, $R_{k,i}$ is updated as follows:

$$R_{k,i} = \mu_1 \times R_{k,i} + (1 - \mu) \times \tau$$

If b_k believes that the location broadcasted by b_i is truthful, $\tau = 1$, otherwise $\tau = 0$. $\mu_1 \in [0, 1]$ is a factor to weight previous experience against current information.

When a node requests location information, every beacon neighbor of the requesting node will publish its Neighbor Reputation Table (NRT) along with its own location. Let's assume b_k is the publishing node and b_j receives $R_{k,i}$. Before incorporating $R_{k,i}$, b_j first performs a simple deviation test as follows:

$$R_{j,i} - R_{k,i} \leq d$$

If the above deviation test is positive, then the information is considered compatible with b_j's first hand experience, and is accepted. b_j then updates $R_{j,i}$ in NRT_{b_j} as follows:

$$R_{j,i} = \mu_2 \times R_{j,i} + (1 - \mu_2) \times R_{k,i}$$

However, if the deviation test is negative, then the published information is considered to deviate too much from its own first han experience, and is disregarded as incompatible information. In order to discourage nodes from publishing false information, the lying node's reputation is decreased as follows:

$$R_{j,k} = \mu_3 \times R_{j,k}$$

2.7 Discussion and motivation

In this section we have reviewed a number of works, projects and models related to the management of trust and reputation concepts in WSN. Some of them have even become one of the most known in this field [9, 13, 15, 16].

Nevertheless, not all of them take into account the strong restrictions about processing, storage or communication capabilities, in the same way. Even more, some of them just present a formal model without showing any set of experiments demonstrating the accuracy, robustness, scalability and overload introduced by their models in such a sentient environment.

Some of them rely on a watchdog mechanism with or without using a multi agent system [11, 12]. Others take advantage of Bayes theorem [17] and a posteriori probabilities, or just use a Beta distribution [15] in order to represent ratings.

As far as we know our model is on of the first ones (together with [10]) in applying a bio inspired technique such as ant colony system (ACS) to develop a trust and reputation model for WSN.

Likewise, we have taken into consideration the important limitations found in WSN, so we have tried to design a model as much lightweight, efficient, robust and scalable as possible. In fact we present two versions of our model, depending on the features of the WSN where it is to be deployed.

If we are facing a very restrictive network, a simpler model is proposed. This simpler and less resource consuming scheme is, however, more vulnerable to some security threats as we will see later. On the other hand, if we are dealing with a WSN whose nodes are devices with more capabilities and security is a very important issue, then we bet on another more sophisticated model with a small overload on the network.

3 Bio-inspired trust model for WSN

3.1 Assumptions/scenario description

Several types of wireless sensor networks can be found depending on what kind of nodes they are composed of. You can meet from a static WSN where nodes have a certain location, to a highly mobile one where nodes move everywhere (like in a VANET [2]). You can also find from a very restrictive WSN where all nodes remain most of the time asleep in an idle state, to another one comprising nodes provided with high performance features capable of processing many requests per second and that are nearly always active.

Throughout this paper we will assume a scenario where a WSN is composed of nodes with relatively high sensor activity. Without loss of generality, we will consider some nodes requesting generic services and some nodes providing them. In the future these services can be specified in detail. How this definition is carried out is out of the scope of this paper.

We will also assume that every node will only know its neighbors (that is, those nodes within its wireless range), and nothing else about the whole topology of the network (at least at the early stages).

Additionally, this topology is considered to be relatively highly dynamic, with many nodes entering or leaving the community. If this frequent logging in and out of nodes is due to the mobility of these nodes or because they switch on and off, is out of the scope of this paper, as well.

Our model is aimed to help a node requesting a certain service to the network to find the most trustworthy route leading to a node providing the right requested service. A node (equally a path) can be considered untrustworthy either because it intentionally provides a fraudulent service or because it provides a wrong one due to hardware failures or performance deterioration.

As we mentioned above, we are considering dynamic topologies, so we needed to use a technic capable of dealing efficiently with this issue. And in our opinion, one mechanism that fulfills quite well this matter is the ant colony system (ACS) [5, 8].

3.2 BTRM-WSN, a bio inspired approach

BTRM-WSN is a bio inspired trust and reputation model for Wireless Sensor Networks aimed to achieve to most trustworthy path leading to the most reputable node in a WSN offering a certain service.

It is based on the bio inspired algorithm of ant colony system but, due to the specific restrictions and limitations found in WSNs, the ACS cannot be directly applied there. Some adaptations, therefore, have to be made.

In our model, for instance, every node maintains a pheromone trace for each of its neighbors. This pheromone traces $\tau \in [0, 1]$ will determine the probability of ants choosing a certain route or another, and can be seen as the amount of trust given by a node to other one.

The heuristic values $\eta \in [0, 1]$, however, are defined as the inverse of the delay transmission time between two nodes (or the inverse of the distance between them).

The fact that every node controls its own pheromone traces and heuristic values, and no one else but it can modify them can become an important security threat.

Other issue that avoids the direct application of the ACS in this environment is the fact that while an ant is searching for the most reputable server providing a requested service, it could happen that some of the nodes that form the path followed by that ant become inaccessible (either because they

switch off or because they move out of the range of their previous sensor in the path).

In that situation, the ant would be unable to come back to the client and it would get lost. In other words, when a client launches a set of ants, it has no guarantee at all that all of them are going to return and, of course, it cannot wait until all the launched ants came back in one iteration of the algorithm.

Therefore, the algorithmic scheme presented in ACS [8, 18] has to be redefined as shown in Algorithm 1.

Algorithm 1 BTRM WSN

```
 1: while (condition) do
 2:    for k = 1 to Number_of_ants do
 3:       S_k ← initial sensor (client)
 4:       Launch ant k
 5:
 6: do
 7:    for every returned ant k do
 8:       if (Q(S_k) > Q(Current_Best)) then
 9:          Current_Best ← S_k
10: while (timeout does not expire) and
11: Num_returned_ants < %Number_of_ants
12:
13: if (Q(Current_Best) > Q(Global_Best)) then
14:    Global_Best ← Current_Best
15: Pheromone_global_updating
16:    (Global_Best, Q(Global_Best), ρ)
17:
18: return Global_Best
```

The first change we can appreciate is that the main loop is now defined by a generic condition, which may be a certain number of iterations (like in the original algorithm) or it can even be a certain timeout. This definition will depend on the specific WSN this model is going to be applied to.

On the other hand, this algorithm consists of the following steps:

1. Every ant adds the first sensor to its solution, which is always the client they are departing from. Then each ant decides which next sensor to move to according to the transition rule and it is sent there (lines 2–4).
2. Once every ant has left the client, this one waits until they come back. For every returned ant, the client compares its solution and keeps the best one. As explained before, in a WSN the client has no guarantee that all the ants that were launched are going to come back, so it just waits until a timeout expires or a certain percentage of all the ants has returned (lines 6–11).
3. The best solution found by all or some of the ants issued in the current iteration is compared with the global best solution and swapped if it is appropriate (lines 13–14).
4. A pheromone global updating is performed over the links belonging to the global best path (line 16).

As explained before, the definition of the condition shown in line 1 of Algorithm 1, as well as the ones defined in lines 10 and 11, depend directly on the specific features (bandwidth, transmission delay, etc.) of the sensors that compose the WSN we are dealing with.

Next we will describe in detail some features of our trust and reputation model for WSN, such as how to measure the quality of a path, how an ant decides which next sensor to travel towards, or when it should stop and return the current path. We will also explain how the pheromone updating is carried out while ants are building their routes as well as how a punishment is performed (in terms of pheromone evaporation) when the client interacts with a fraudulent server.

Additionally, the differences between the two proposed versions of our model will be explained and some final remarks about the scalability and lightness of BTRM WSN will be shown.

3.2.1 Path quality

Each time a launched ant returns to its client carrying a solution with it, that client has to assess the quality of that solution. Specifically the ant keeps a list of all the sensors belonging to the selected path, together with the pheromone traces of the links that join them.

According to this, the path quality computation can be done in the following way:

$$Q(S_k) = \frac{\overline{\tau}_k}{Length(S_k)^{PLF}} \cdot \%A_k$$

where $\overline{\tau}_k$ is the average pheromone of the path found by ant k, $PLF \in [0, 1]$ and $\%A_k$ represents the percentage of ants that have selected the same solution as ant k.

On one hand, the amount of ants that in one iteration has selected the same path as ant k, and the reputation of that path, represented by its average pheromone, contribute to have a qualified solution. On the other hand, on equal conditions, a shorter path is preferred.

With a definition like this we achieve that our model tends to preferably select those paths which are as short as possible and which have been selected as many times as possible.

3.2.2 Ants transition and stop condition

When an ant is travelling along the WSN searching for the most trustworthy route leading to the most reputable server it has to decide at each sensor which of its neighbors it has to move to. Every ant has also to decide whether to stop when

it finds a server offering the requested service or if it should keep trying to find a more reputable one.

So let ant k be at sensor s in a certain moment of its searching. Several options can happen:

1. Sensor s offers the requested service.
   (a) Sensor s has more neighbors not visited yet by ant k.

   The average pheromone of the path followed by ant k from the client until the sensor s is computed, $\overline{\tau}_k \in [0, 1]$. If $\overline{\tau}_k$ is greater than a certain transition threshold, $TraTh \in [0, 1]$, then ant k stops and returns current solution with a probability equal to $\overline{\tau}_k$. Otherwise, if $\overline{\tau}_k \leq TraTh$, ant k considers sensor s not enough reputable and keeps trying to find a better one.

   (b) Sensor s has no more neighbors or all of them have been already visited by ant k.

   Ant k stops and returns current path.

2. Sensor s does not offer the requested service.
   (a) Sensor s has more neighbors not visited yet by ant k.

   Ant k decides which next sensor to move to according to the traditional transition rule defined in ACS.

   (b) Sensor s has no more neighbors or all of them have been already visited by ant k.

   In this situation ant k has reached a dead end and has no more options than backtracking. That is, it has to follow the inverse route it has currently built until it arrives at a sensor which offers the requested service (and then stops and returns that new path) or until it reaches a sensor not offering the requested service but with more alternative paths not explored yet by ant k (and then keeps trying those routes).

   It could even happen that, while backtracking, ant k reached the client it belonged to. In that situation the whole WSN would have been explored but any server offering the requested service would have been found.

However, in order to prevent some security threats a client cannot interact again with the same malicious server in the next transaction, so ants will not stop when they find it and consequently the will not choose it.

Another important issue to take care about is the number of launched ants, $N_{ants} \in \mathbb{N}$, which depends on the number of sensors that form the WSN and the dynamism of the WSN itself. It is sensible to think that the greater and the more dynamic a WSN is, the greater has to be the number of launched ants (because some of them can be lost), and vice versa.

But if the number of ants is relatively high, maybe the condition defined in line 1 of Algorithm 1 should not lead to a too big number of iterations or a too large timeout. Otherwise, each execution of BTRM WSN would require an amount of time and resources consumption that may not be acceptable in certain WSNs. Therefore, an accurate balance between the number of iterations (or timeout) and the number of ants is necessary in order to achieve reasonably good outcomes.

3.2.3 *Pheromone updating*

While ants are travelling across the WSN searching the most reputable server, they modify the pheromone traces they find. This modification helps next ants to decide which path is better to follow.

Actually, there are two kind of updatings: a local and a global one. The pheromone local updating is carried out by every ant each time it decides to move from one sensor to the next. Let ant k be at sensor s_1. Then, applying the transition scheme explained in the previous section, it decides to move towards sensor s_2 (which is a s_1's neighbor). So, before being actually transmitted, it indicates sensor s_1 that it has to modify its pheromone trace associated with sensor s_2 in the following way:

$$\tau_{s_1 s_2} = (1 - \varphi) \cdot \tau_{s_1 s_2} + \varphi \cdot \Omega \tag{1}$$

where $\Omega = (1 + (1 - \varphi) \cdot (1 - \tau_{s_1 s_2}) \cdot \eta_{s_1 s_2}) \cdot \tau_{s_1 s_2}$ is the convergence value of $\tau_{s_1 s_2}$ when time $t \to \infty$ (given that $\tau, \eta, \varphi \in [0, 1]$), that is, is the pheromone trace value that would have that link after a lot of time if no other modification was carried out over it (notice that $\Omega \in [\tau_{s_1 s_2}, 2 \cdot \tau_{s_1 s_2}]$).

On the other hand, a pheromone global updating is performed over the best path found by all ants in each iteration of Algorithm 1 (see line 16). This is done by sending an extra ant just to modify the pheromone traces of that route. And that modification is carried out using the next expression:

$$\tau_{rs} = (1 - \rho)\tau_{rs} + \rho(1 + \tau_{rs}\eta_{rs}Q(S_{Global\ Best}))\tau_{rs} \tag{2}$$

Therefore, the higher are the pheromone trace, the heuristic value, the quality of the path and $\rho \in [0, 1]$, the higher is the additional pheromone contribution over the best route.

Finally, it is worth to mention how to initialize the pheromone traces. Their initial value $IniPh \in [0, 1]$ will condition some aspects of the model. Thus, if $IniPh \to 0$, for instance, everybody would mistrust everyone at the beginning and it would be difficult to distinguish trustworthy sensors from malicious ones. However, if $IniPh \to 1$ then everybody would trust everyone at the beginning and it would also be difficult to distinguish benevolent sensors from fraudulent ones.

3.2.4 Punish & reward

Once BTRM WSN has selected what it thinks is the most trustworthy path leading to the most reputable server, the client actually requests the desired service to that server. Then, depending on the goodness of the server, it will provide the same service it was offering, or another worse.

In this first stage we will consider only two possibilities. The server can be totally benevolent and provide the same service it was offering (so the client is fully satisfied), or it can be totally fraudulent and provide a completely different service than the one that was offered (having thus a fully unsatisfied client).

If the client is satisfied, a reward by means of additional pheromone contribution is done all along the selected path. The same expression used for pheromone global updating (2) can be applied here as well.

Nonetheless, if the client is not satisfied, a punishment, i.e., an evaporation of pheromone traces of the links belonging to the selected path, is carried out. And this punishment uses the following expression, if satisfaction is less than a certain punishment threshold $PunTh \in [0, 1]$:

$$\tau_{rs} = (\tau_{rs} - \varphi \cdot df_{rs}) \cdot Sat \cdot (1 - df_{rs}) \tag{3}$$

where $Sat \in [0, 1]$ represents the satisfaction of the client with the received service and $df_{rs} \in (0, 1]$ is a distance factor of link e_{rs} computed as follows:

$$df_{rs} = \sqrt{\frac{d_{rs}}{L(S_k) \cdot (L(S_k) - d_{rs} + 1)}} \tag{4}$$

being $d_{rs} \in \{1, 2, \ldots, L(S_k)\}$ the actual distance (number of hops) between sensor r and s, and $L(S_k)$ the length of the path found by ant k.

Otherwise, if $Sat \geq PunTh$ then:

$$\tau_{rs} = \tau_{rs} - \varphi \cdot (1 - Sat) \cdot 2df_{rs} \tag{5}$$

As it can be checked, having a punishing scheme like this, those edges which are closer to the client have a slighter pheromone evaporation, and vice versa.

Furthermore, all the links that fall into the malicious server are also punished. Otherwise ants could select it again through an alternative path, thinking it has become a benevolent sensor (which may not happen most of the times). Therefore, those edges have to be punished according to the next formula:

$$\tau_{rs} = (\tau_{rs} - \varphi) \cdot Sat \tag{6}$$

3.3 Two proposed models

As we have mentioned before, we have actually developed two versions of our model BTRM WSN. The first one is the one we have been showing until now, where pheromone traces are shared for every service offered by a sensor. This allows us to achieve a lighter model (very low overload is added to the network).

However, it also has some drawbacks. For instance, with a model like this, a client could not distinguish a sensor which is very good (benevolent) when supplying a certain service, but very bad (fraudulent) providing another one. It will consider that sensor as very trustworthy or untrustworthy for all the services provided.

If we have a WSN where we are only interested on monitoring the behavior of sensors about just one service (or even if the WSN only provides one service), we could use this model without the problem of distinguishing a sensor's particular behavior for each provided service.

And if our WSN is composed of very restricted sensors, we could adopt a dynamic scenario where some of them switched off for awhile if they did not have any transaction along a timeout or if they were very active (providing too many requests) during another timeout.

But if we have a low constraint WSN (equally, a high performance WSN) and we need a more resilient model, capable of dealing with multiple services, we could adopt the second version of BTRM WSN. In this one, every sensor has a pheromone trace for each one of its neighbors, and for each one of the services provided by the WSN. Likewise, sensors will remain always awake.

Let be m the number of services available in the WSN, and let be n_s the number of neighbors of sensor s. Then, s should manage and store $m \times n_s$ different pheromone traces. Obviously, this decision implies a bigger amount of stored information on each sensor but, on the other hand, it provides a more resilient trust and reputation model, since this is now able to distinguish each sensor as trustworthy or not, for each one of the services it offers.

If we are dealing with a WSN with high resources sensors and where the security is a critical issue when applying for a service, we could make use of this second version of the model.

Additionally, in this second version the client gathers all the paths found by all the ants that visit it (not only its own ants) and join them in order to achieve a local view of the topology of the network (which will probably be an instantaneous view, due to the high dynamism that this kind of networks can reach).

This local view can help the client to take more accurate decisions, since it knows (through the pheromone traces) which servers are more reputable and which not.

3.4 Scalability and lightness

One of the strong points of our trust and reputation model is its scalability. In this kind of networks, whose size can vary

from a handful of nodes until thousands of them, developing a scalable model is a critical issue.

Since in our model every sensor manages and controls its own pheromone traces and there is not any central entity (like a watchdog) gathering ratings or supervising all or a subset of the sensors, we can state that BTRM WSN is scalable.

Even more, if needed, every ant could be provided with a **TTL** (Time To Live), i.e., a maximum number of hops it is able to travel. Notice that this **TTL** would also limit the maximum length of any solution. Even so, if a client launched a set of ants with a **TTL** which did not allow them to reach any server (or all the reached servers were malicious), the client could increase that **TTL** and launch a new set of ants.

About the lightness of the model, we have seen in the previous section that we have two versions of the model. But even the second one, where every sensor s has $m \times n_s$ pheromone traces, does not add too much overload to the network. Moreover, each transmitted ant carries a list of sensors' identifications (which can be just a number) with their corresponding pheromone traces. And since the solutions average length rarely exceeds 5 or 6 hops, the information transmitted with every ant does not involve a big overload.

Of course, the overload introduced will also depend on the number of ants travelling through the WSN. As we explained in Sect. 3.2.2, the number of ants depends on the number of sensors composing the network. Thus, we defined the number of ants launched by every client as $N_{ants} = \lceil N_s^{0.35} \rceil$, where N_s is the number of sensors belonging to the same WSN. With a definition like this we achieve quite good outcomes with a small overload.

The accuracy and robustness of BTRM WSN will be demonstrated in Sect. 5 where experiments and results will be shown.

4 Security threats

The fact that every node maintains the pheromone traces of its neighbors and it is the only one who can manage, control and modify them, can lead to some security threats.

But the only security threats related to this matter can appear if a malicious server colludes with other malicious servers, because a sensor is only able to manage the pheromone traces of its neighbors, but by the same reason it cannot control the pheromone traces that its neighbors have associated with it.

Nevertheless, it is important to notice that a collusion is only possible if the malicious sensors know each other and also know who the benevolent sensors are. And this assumption is not always feasible in every Wireless Sensor Network.

Therefore, two types of security threats may happen if a collusion among malicious sensors can be created. Malicious sensors can praise their malicious neighbors by assigning them the maximum level of pheromone. Equally they can slander their benevolent neighbors by giving them the minimum value of pheromone. We will discuss in detail both situations next.

4.1 Praising malicious sensors

A set of malicious sensors can form a collusion in order to increase their self profit and interests. Each of them manage the pheromone traces of its neighbors, so what they can do is to praise those neighbors belonging to the collusion by giving them the maximum level of pheromone. And, of course, they will not decrease those traces although a client asked for it.

In this situation the malicious node who modifies the pheromone traces of its neighbors can act as a malicious service provider or could behave properly and supply the right requested service. If the second thing occurs ants will choose it as the service provider and its collusion will have no sense.

But if it behaves in a fraudulent way as well and a client selects it to have an interaction with it, all the links falling into it will be punished as explained before and ants will not select it again (or will select it with a very low probability), so its false praising would be useless.

4.2 Slandering benevolent sensors

Another possible security threat would consist in slandering benevolent nodes. This is achieved by assigning the minimum level of pheromone to those benevolent neighbors of a malicious one.

Again the malicious node can actually provide fraudulent services or right ones. In the first case, if there are alternative paths leading to the slandered benevolent sensor, ants should be able to discover them; otherwise, ants would select another different benevolent node.

And if the malicious server acts properly and provides the right service, ants will select it and its collusion will not have sense neither.

It is important to have in mind that there must be at least one accessible benevolent server in the WSN and the key consists of finding it. It actually does not matter which specific sensor is selected to interact with, the important thing is to select a trustworthy one.

5 Experiments and results

Once we have shown in detail the description of our bio inspired trust and reputation model over Wireless Sensor

Table 1 BTRM-WSN parameters

phi	0.01	**alpha**	1.0	**Nants**	0.35
rho	0.87	**beta**	1.0	**Niter**	0.59
q0	0.45	**TraTh**	0.66	**PunTh**	0.48
IniPh	0.85	**PLF**	0.71		

Networks and have described some related security threats, it is time to demonstrate its accuracy, scalability and robustness.

To do so, we have developed a whole testbed focused on three main targets. First, we are interested in finding out how many times our model is able to select the right benevolent server to interact with. In other words, we would like to know the selection percentage of trustworthy servers.

Since our model has a strong basis on random or probabilistic decisions, we considered that it would be also quite interesting to take care about the standard deviation of that selection percentage of trustworthy servers.

Finally, as a possible measure of the adaptability of our model specifically to WSNs, we gathered as well the average path length of the solutions found by our model. As we mentioned before, in a environment with a lot of restrictions like WSNs, the shorter path is always preferred since it supposes less consumption of sensors' resources.

The experiments we carried out had the following structure. We launched our model 100 times (i.e. each client applied for a service 100 times) over 200 WSNs randomly generated, each one composed of 100 sensors. On each network, the percentage of sensors acting as clients was always a 15%. The 85% left were, therefore, sensors acting as servers.

We tried with 200 random WSNs having a 10% (over the 85% left) of malicious servers. 200 with 20%, other 200 with 30%, and so on until a 90% of malicious servers (the worst simulated situation).

But even more, we repeated those experiments over WSNs composed of 200, 300, 400 and 500 sensors (with the same percentages of clients, servers and malicious servers).

We have defined the main condition of our algorithm (line 1 of Algorithm 1) as a certain number of iterations. And that number is defined as $N_S^{N_{iter}}$, (similar to the number of ants definition) where N_S is the number of sensors belonging to the WSN and $N_{iter} \in [0, 1]$.

The same set of parameter values (shown in Table 1) is used for all the experiments and environments.

We have configured four different scenarios: static WSNs, dynamic WSNs, oscillating WSNs and static WSNs with collusion among malicious servers, as we will explain next.

5.1 Experiments and results over static WSNs

The first tested scenario consisted of static Wireless Sensor Networks, that is, networks where their sensors do not switch off and do not move, maintaining thus always the same topology.

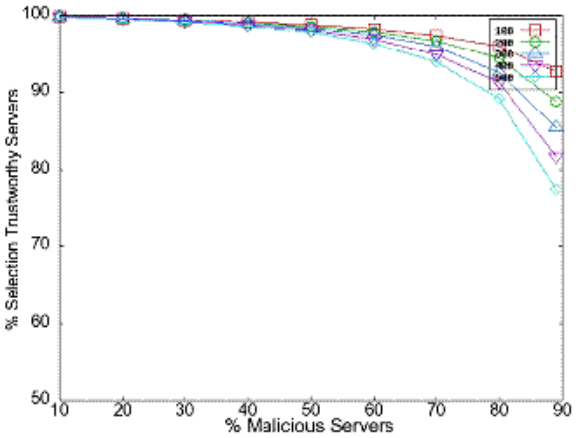


Fig. 1 Static WSNs. Selection percentage of trustworthy servers

5.1.1 Selection percentage of trustworthy servers

So the first and main focus was to evaluate the selection percentage of trustworthy servers achieved with BTRM WSN. The outcomes corresponding to this experiment are shown in Fig. 1.

The very first appreciation that can be done is the similarity of the selection percentages regardless the size of the network, which constitutes a demonstration of the scalability of our model. Outcomes slightly differ from one set of random WSNs to another when we fix the percentage of malicious servers and vary the number of sensors belonging to the same WSN.

Another conclusion that can be obtained is that the selection percentage is quite high (above the 90%) when the percentage of malicious servers is less than or equal to 80%, in every case.

In order to consider a trust and reputation model as acceptable (with a minimum quality level), in our opinion, the selection percentage of trustworthy servers should be greater or at least equal to 70%. A smaller percentage would result in a model with certain security deficiencies. And what is clear is that a selection percentage below the 50% means that the model is not useful at all.

Our experiments have shown that BTRM WSN remains resilient to a high percentage of malicious servers when this percentage is less than or equal to 90%. Its performance gets worse when the percentage of malicious servers in the WSN increases, and the problem intensifies when the size of the WSN grows.

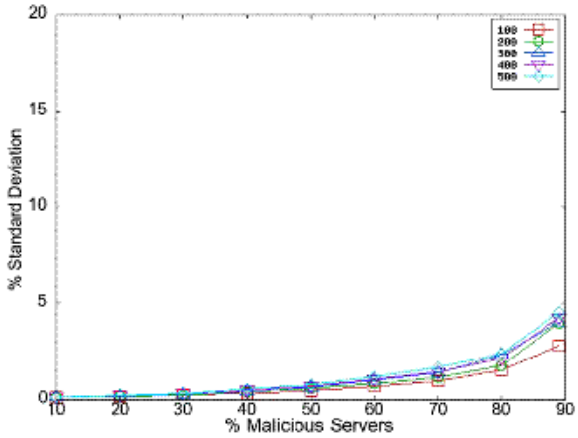


Fig. 2 Static WSNs. Standard deviation of the selection percentage of trustworthy servers

5.1.2 Standard deviation of the selection percentage of trustworthy servers

It is also important to realize that by testing our model against a number of random WSNs, each of them has a random topology, so it could happen that our model was tested against networks where benevolent servers were very near to the clients (maybe one hop forward and, consequently, very easy to solve) or quite the contrary, that is, WSNs where benevolent sensors were quite far from the clients.

Figure 1 actually depicts the average selection percentage of trustworthy servers. But an average selection percentage of 80%, for instance, could be reached because the model always found a trustworthy server the 80% of the times, or just because it found it the 100% of the times in half the tested wireless sensor networks and the 60%, in the other half, for example.

That is the reason why we decided to measure and show the standard deviation related to that average as well. And the outcomes can be checked in Fig. 2.

Again, the first observation that can be done has to do with the similarity between the five graphics corresponding to the five tested sizes for WSNs. And here the standard deviation also remains quite low and nearly undistinguishable among the five tested sizes where the percentage of malicious servers is less than or equal to 90%. Furthermore, this standard deviation remains below a 5% in every case.

This means that when there are less than or equal to 90% of malicious servers in the network, regardless its size, BTRM WSN is able to select a high percentage of trustworthy servers (as shown in Fig. 1) with a quite high accuracy, regardless the topology of the WSN.

In fact the highest value among all the experiments carried out is obtained when we tried our model over 200 random WSNs (100 times on each one), composed of 500 sensors, with a 15% of clients and a 90% of malicious servers (a 90% of the 85% left). In that experiment our model was able to reach a trustworthy server in the 77.35% of the times, with a standard deviation of 4.55%.

So if the percentage of malicious servers is high (greater than or equal to 90%, for instance), and the number of sensors composing the networks is also high, then the percentage selection of trustworthy servers is lower but, however, still accurate (i.e., BTRM WSN is independent of the topology).

This means that if the random tested WSNs size is too high, those networks topology can vary from ones where BTRM WSN works quite fine to others where it is hardly able to find the most trustworthy server. Nevertheless, if the size of the random tested networks is high, their topologies drive the model behaving most of the times in the same way (most of the times well, or most of the times not).

5.1.3 Average path length leading to trustworthy servers

Finally, the last developed experiment consisted of measuring the length (number of hops) of those paths found by BTRM WSN leading to trustworthy servers. That is, when the model fails and selects an untrustworthy server, that path is discarded and not taken into account.

Doing this way we are able to estimate the average path length of those paths found by our model when it successfully reaches a benevolent server.

Our model is aimed to find the closest benevolent servers to the client requesting the service. On the one hand we think that the lesser number of intermediaries present in a transaction, the more secure and robust it can be performed. On the other hand, due to the specific restrictions related to wireless sensor networks, the resources consumption saving is a critical issue. Therefore, a shorter path leading to the final trustworthy server implies less involved sensors and, consequently, less global utilization of resources such as energy or bandwidth.

The outcomes of this experiment are presented in Fig. 3.

As it can be observed, any trustworthy server is never reached (on average terms) at more than 4 hops. In fact, the highest average path length is achieved with 100 sensors WSNs with a 90% of malicious servers. In that situation, the average path length takes the value 3.844.

One more time, differences between the several sizes tested for WSNs become distinguishable when the percentage of malicious servers is greater than or equal to 90%. Under this percentage, the average number of hops is quite low (near to 2), as it can be checked in the figure.

Therefore, our model is able to reach nearby trustworthy servers regardless the size of the network and the percentage of malicious servers. Although the smaller is the former and the greater is the latter, a larger path is found.

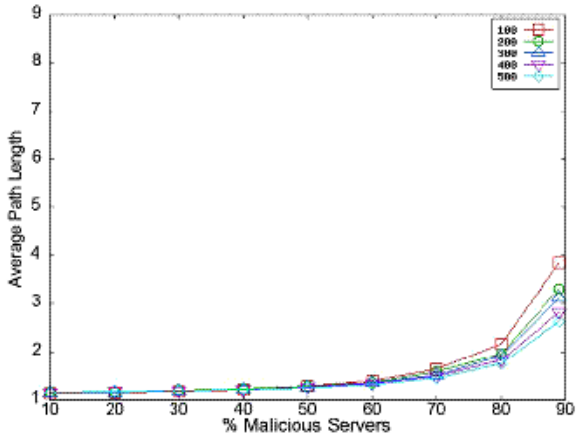


Fig. 3 Static WSNs. Average path length leading to trustworthy servers

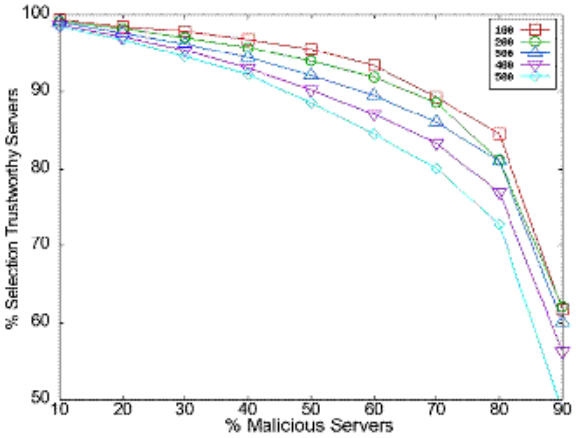


Fig. 4 Dynamic WSNs. Selection percentage of trustworthy servers

5.2 Experiments and results over dynamic WSNs

As we have already mentioned, the first of the two proposed versions of our model is aimed to deal with WSNs composed of sensors with quite high restrictions in energy consumption, bandwidth, storage capacity, etc.

That is the reason why we decided to develop this set of experiments. Here some nodes switch off for awhile some times saving thus an important amount of energy consumption.

The decision scheme of when to switch off and on is as follows: when a server receives and supplies 20 requests it automatically switches off during a certain timeout. On the other hand, if a server does not receive at least 20 requests within a time interval, it also switches off during another timeout.

Once we defined our dynamic scenario in the manner explained above, we carried out the same experiments that were done for static networks, i.e., we measured the percentage selection of trustworthy servers, the standard deviation of this selection percentage, and the average path length of the routes found leading to trustworthy servers.

5.2.1 Selection percentage of trustworthy servers

Figure 4 shows the selection percentage of trustworthy servers achieved with BTRM WSN over WSNs composed of 100 to 500 sensors with a percentage of malicious servers from 10% to 90%.

As it is observed, the selection percentage is nearly greater than 90% when the percentage of malicious servers is less than or equal to 50%, regardless the size of the WSN. And it remains obtaining qualified outcomes (above the 70%) when the proportion of malicious servers is less than or equal to 80%.

Selection percentage get worse when the percentage of malicious servers increases and even worse if the size of the Wireless Sensor Network is greater.

Nevertheless, we can state that BTRM WSN is resilient to a dynamic behavior of the sensors composing the WSNs it is running, if the percentage of fraudulent sensors is less than 80%. And the worsening is not too high when the number of sensors increases.

5.2.2 Standard deviation of the selection percentage of trustworthy servers

In Fig. 5 we can observe the standard deviation of the selection percentage of trustworthy servers achieved in this experiment, using dynamic Wireless Sensor Networks.

We can see that the standard deviation remains quite low and nearly undistinguishable when the size of the network is greater than or equal to 200 sensors and the percentage of malicious servers is less than or equal to 80%, which means that, in those cases BTRM WSN is very accurate and almost always finds the same percentage of trustworthy servers.

If we are dealing with a smaller Wireless Sensor Network or the proportion of malicious servers is greater than or equal to 90%, however, this standard deviation increases remarkably, being its maximum value a 19.35%, when the tested WSN is composed of 100 sensors.

5.2.3 Average path length leading to trustworthy servers

As it can be checked in Fig. 6, the average path length obtained in a dynamic environment is greater than in a static one (see Fig. 3).

However, although the size of the network can reach high values, the average path length never exceeds 8.66 hops in any case, which is still a good outcome for Wireless Sensor Networks.

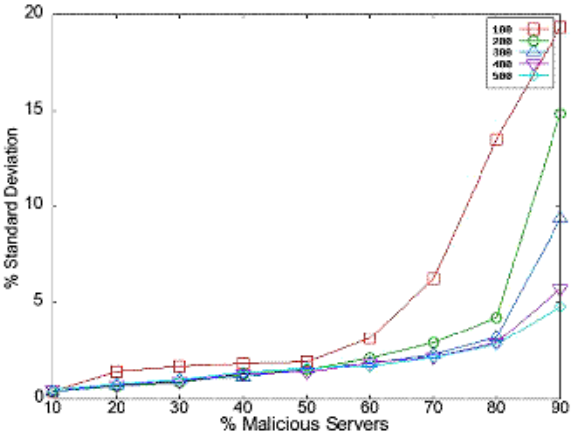


Fig. 5 Dynamic WSNs. Standard deviation of the selection percentage of trustworthy servers

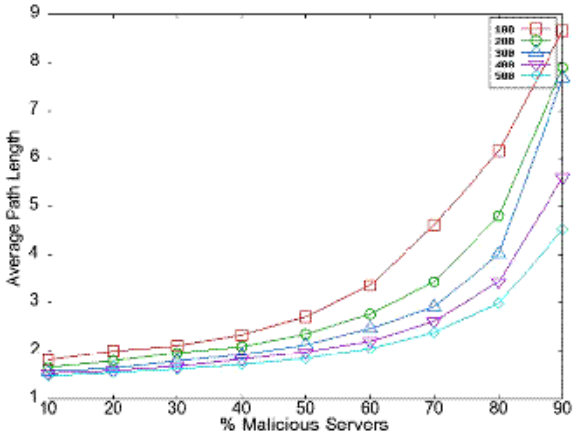


Fig. 6 Dynamic WSNs. Average path length leading to trustworthy servers

Equally to the static scenario, when the size of the network is lower and the percentage of malicious servers composing the network is greater, then the average path length also increases.

This experiment together with the selection percentage of trustworthy servers and the standard deviation of that selection percentage constitute the proof that BTRM WSN obtains quite good and accurate outcomes over dynamic Wireless Sensor Networks, with a low influence from the size of the networks and the percentage of malicious servers.

We can state, therefore, that BTRM WSN presents a technique to identify trustworthy servers that is suitable for dynamic Wireless Sensor Networks.

5.3 Experiments and results over oscillating WSNs

Another tested scenario developed consisted of Wireless Sensor Networks where the goodness of the servers belonging to them could change along the time.

How a sensor decides to be benevolent or malicious at each time is out of scope of this paper. We designed, therefore, our particular proposal as follows: after every 20 transactions are carried out (i.e., after every client has had 20 transactions) all the benevolent servers composing the Wireless Sensor Network become malicious.

Now, in order to preserve the same percentage of malicious servers, the number of previous benevolent servers, let say n_b, is kept. Then n_b random servers are selected (note that all of them will be malicious) and their goodnesses are swapped so they become benevolent and the percentage of malicious servers remains equal to the stage previous the oscillation.

With an oscillation scheme like this a benevolent server could maintain its positive goodness since it could be randomly selected to become benevolent when it indeed previously was benevolent.

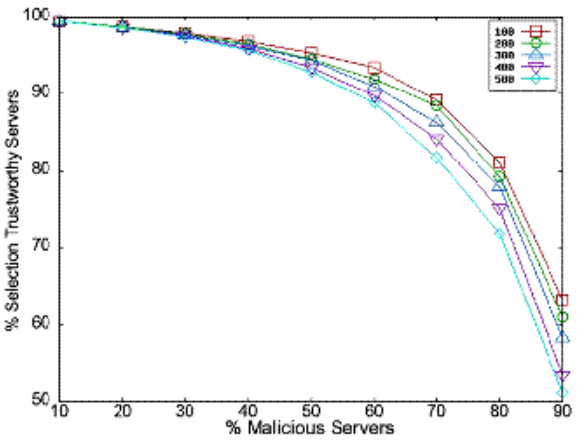


Fig. 7 Oscillating WSNs. Selection percentage of trustworthy servers

5.3.1 Selection percentage of trustworthy servers

As we can see in Fig. 7, outcomes got here are quite similar to those obtained in the previous experiment with dynamic WSNs.

It can be checked that the selection percentage of trustworthy servers is greater than 90% if the percentage of malicious servers is approximately less than or equal to 60%, regardless the size of the Wireless Sensor Network.

Moreover, reasonably good outcomes (those with a selection percentage above the 70%) are obtained always the proportion of fraudulent servers is less than or equal to 80%.

5.3.2 Standard deviation of the selection percentage of trustworthy servers

Again, similar outcomes to the ones shown for static WSNs about the standard deviation of the selection percentage

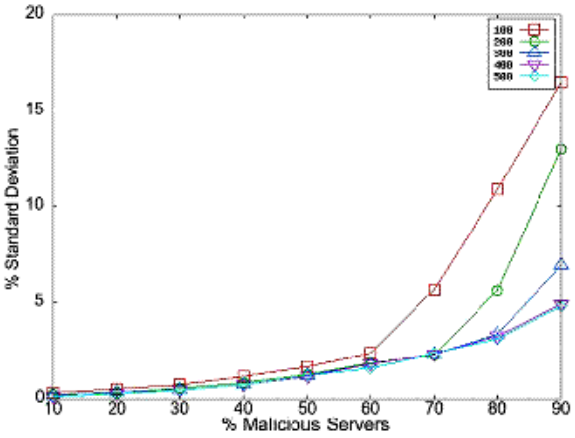


Fig. 8 Oscillating WSNs. Standard deviation of the selection percentage of trustworthy servers

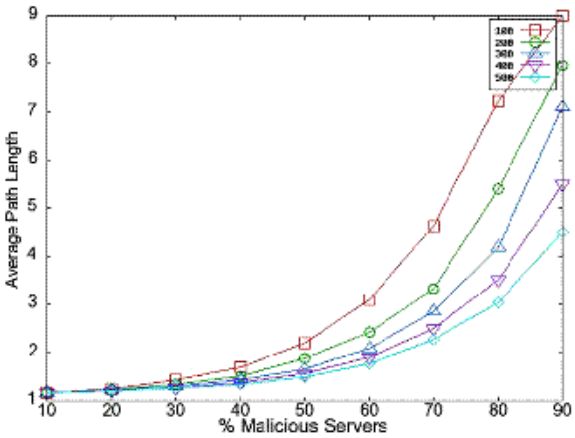


Fig. 9 Oscillating WSNs. Average path length leading to trustworthy servers

of trustworthy servers are achieved here, with oscillating WSNs. Figure 8 depicts these results.

One more time the more variable behavior of the model happens when the size of the tested WSNs is less than 200 sensors. In such situation a maximum standard deviation value of 16.5% is reached when the proportion of fraudulent servers is 90%.

Nonetheless, if the tested WSNs are composed of 200 sensors or more, and the percentage of malicious servers is less than or equal to 70%, then the standard deviation is undistinguishable and less than 2.4%.

5.3.3 Average path length leading to trustworthy servers

Figure 9 shows the outcomes about the average path length of those routes found by BTRM WSN leading to a trustworthy server over oscillating WSNs.

It can be checked that these results are very similar to the ones shown in Fig. 6, so the same conclusions can be obtained.

The three last experiments demonstrate that BTRM WSN is also a feasible technique in order to find the most trustworthy server over oscillating WSNs.

5.4 Experiments and results over static WSNs with collusion

The last tested scenario consisted of static Wireless Sensor Networks where a collusion among all malicious servers composing the network was built. As explained in Sect. 4, since in BTRM WSN every sensor is the only one who can manage the pheromone traces associated with its neighbors, malicious servers could collude an falsely praise themselves or slander benevolent servers.

We chose the worst situation, where both things occurred, that is, every malicious server always had the maximum

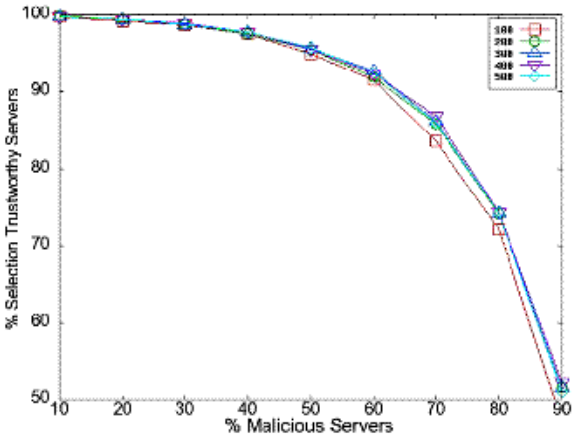


Fig. 10 Static WSNs with collusion. Selection percentage of trustworthy servers

pheromone value for those of its neighbors who were also malicious, and the minimum pheromone value for those neighbors who were benevolent.

How every sensor knows if its neighbors are malicious or benevolent is out of the scope of this paper.

5.4.1 Selection percentage of trustworthy servers

The worst outcomes about the selection percentage of trustworthy servers among all the tested experiments are reached here and can be seen in Fig. 10.

Again, the first appreciation that can be done is the high similarity of the five graphics corresponding to the five WSN sizes tested. This means that BTRM WSN is highly scalable in a collusion scenario.

As it can be checked the selection percentage of trustworthy servers remains greater than a 90% when the percent

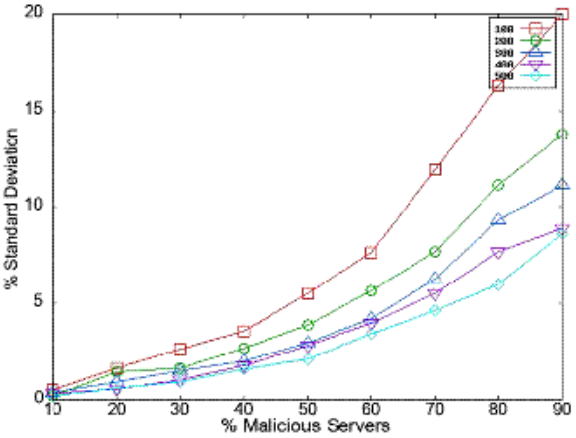


Fig. 11 Static WSNs with collusion. Standard deviation of the selection percentage of trustworthy servers

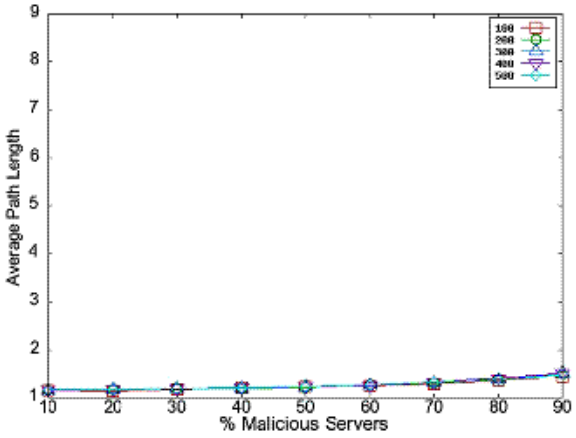


Fig. 12 Static WSNs with collusion. Average path length leading to trustworthy servers

age of malicious servers ir less than or equal to 60%. And it produces qualified outcomes (above the 70% of selection percentage) when the proportion of malicious servers is less than or equal to 80%.

But if this percentage increases, however, then our model is quite near to the limit of being useful in any way. Notice that if the selection percentage is under the 50%, then the model is completely useless.

5.4.2 *Standard deviation of the selection percentage of trustworthy servers*

In Fig. 11 the standard deviations of the selection percentage of trustworthy servers obtained by BTRM WSN over static WSNs with a collusion, are shown.

An interesting behavior happens here. In the previous experiments the standard deviation remained very similar when the size of the network was high and the proportion of malicious servers was low.

However, here the standard deviation grows when the percentage of fraudulent servers also increases and the size of the tested Wireless Sensor Networks decreases.

This means that BTRM WSN is less independent of the number of sensors composing the network when a collusion is formed.

5.4.3 *Average path length leading to trustworthy servers*

On the other hand, the best outcomes regarding the average length of the routes leading to trustworthy servers found by BTRM WSN over all the tested experiments are obtained when a collusion takes place, as it can be observed in Fig. 12.

This average path length never exceeds 1.55 hops, which is a very low value. This means that most of the trustworthy servers found are very near to the client. It can also mean that in such an adverse situation like this one (static WSNs with collusion), BTRM WSN is unable to find benevolent servers which are too far from the clients.

And it makes sense getting these values. If the proportion of malicious servers is low, it will be probable that some benevolent servers stay near the clients. And if that percentage is high, then malicious colluding servers will avoid clients' ants to travel quite far in order to find benevolent servers.

5.5 Energy consumption

Energy consumption is a critical issue when dealing with wireless sensor networks, since these ones are commonly composed by resource constrained devices with limited features in terms of processing, memory and communicating capabilities.

Therefore, we could not ignore this topic in our trust and reputation model, so we developed a last experiment aimed to measure the average energy consumption needed by our approach.

As pointed out by [19, 20] the power required by a sensor in a WSN can be seen as a function of the distance. Different energy models can be used to estimate the energy required by a sensor s to send a message far enough to reach another sensor placed at distance d. In the most commonly used model, the energy consumption for transmitting a message at distance d is:

$$E(d) = d^{\alpha} + C$$

where $\alpha \in [2, 6]$ represents the media attenuation factor and C is a constant denoting the power used to process the radio signal (note it is a dimensionless measurement).

Table 2 Static WSNs. Average energy consumption per sensor

	Number of sensors				
	100	200	300	400	500
10%	1.2×10^{14}	1.1×10^{16}	1.0×10^{16}	6.8×10^{17}	3.6×10^{18}
20%	1.3×10^{14}	1.0×10^{16}	1.1×10^{16}	7.0×10^{17}	3.7×10^{18}
30%	1.6×10^{14}	1.2×10^{16}	1.1×10^{16}	7.5×10^{17}	3.8×10^{18}
40%	2.0×10^{14}	1.3×10^{16}	1.2×10^{16}	7.8×10^{17}	4.0×10^{18}
50%	1.9×10^{14}	1.4×10^{16}	1.4×10^{16}	8.5×10^{17}	4.4×10^{18}
60%	2.8×10^{14}	1.4×10^{16}	1.5×10^{16}	1.0×10^{18}	5.0×10^{18}
70%	2.7×10^{14}	2.2×10^{16}	1.9×10^{16}	1.2×10^{18}	6.2×10^{18}
80%	5.0×10^{14}	3.0×10^{16}	3.1×10^{16}	1.8×10^{18}	8.6×10^{18}
90%	9.3×10^{14}	0.9×10^{17}	0.7×10^{18}	6.8×10^{18}	2.6×10^{19}

Following these authors' direction, we have chosen a value of $\alpha = 4$ and $C = 10^8$. Additionally, the sensors belonging to our generated networks are spread along a 10000 m^2 area, and each of them has a radio range of 10 meters.

We have collected the energy measurements from the experiments developed over static wireless sensor networks, which constitutes the worst scenario, since every sensor is permanently awake. The outcomes can be observed in Table 2.

Two direct consequences can be deducted from the table. On the one hand, the bigger the number of sensors is, the higher is the energy consumption. And on the other hand, the greater the percentage of malicious sensors is, the higher is the power consumption as well.

The increase of energy needed as the size of the network grows is explained because it implies an increase in the density of the network too. So the average number of links increases rapidly and, therefore, a higher number of messages are sent.

Regarding the percentage of malicious sensors, in a network where this kind of sensors are in majority it is more difficult to find a benevolent one and, consequently, more messages need to be sent as well.

5.6 TRMSim WSN. Trust & reputation models simulator for WSNs

In order to carry out all the explained experiments we have developed a Java based Trust & Reputation Models Simulator for WSNs, called TRMSim WSN [21].

It allows a user to test BTRM WSN over all the scenarios described in this paper (static, dynamic, oscillating and collusion), and even combinations of them, deciding the number and size of WSNs and the number of transactions or executions of the model carried out by every client. It also allows to set the percentage of clients, relay servers (those not providing the service requested by the clients), and malicious servers.

The wireless range of every sensor can be set as well, determining thus the topology of the network by means of determining every sensor's neighbors.

Currently it only implements BTRM WSN and PeerTrust [22] models and allows to tune their parameters, but we are planning to implement additional trust and reputation models for WSNs in order to make a comparison among them.

Figure 13 shows a snapshot of TRMSim WSN, which can be downloaded from [21], where a more complete documentation of the simulator can be found.

6 Conclusions and future work

Managing trust and reputation in Wireless Sensor Networks (WSNs) in an efficient, accurate and robust way has not been completely solved yet. Providing this management would notably increase the security in such a sentient environment, supporting thus its development and deployment.

In this paper we have proposed a Bio inspired Trust and Reputation Model for WSNs, called BTRM WSN. It is based on the Ant Colony System (ACS) and a complete description of its main features has been shown. We have seen how the pheromone traces deposited by ants help next ants to find the most trustworthy server through the most reputable path all over the network.

Specifically we have explained how the pheromone updating is carried out, as well as how to measure the quality of a path or how to punish or reward a server depending on its behavior. We have described the ants transition and stop condition scheme followed in our model, too.

A set of experiments over static WSNs (not changing its topology along the time) have been carried out. The outcomes achieved in the three developed experiments demonstrate that BTRM WSN fulfills reasonably well the initially stated expectations about security, scalability and lightness in WSNs.

We have also tested our model against dynamic Wireless Sensor Networks, where some nodes switched off for awhile if certain conditions occurred (changing thus the topology of the network), against oscillating WSNs, where the goodness of the servers changed along the time, and against static networks with a collusion scenario among malicious servers. It has been demonstrated that BTRM WSN obtains accurate, robust and scalable outcomes in most of the situations.

Specifically, if the percentage of malicious servers is below the 60%, the selection percentage of trustworthy servers remains above 90% in most of the cases with a standard deviation never greater than 7.62%, regardless the size of the

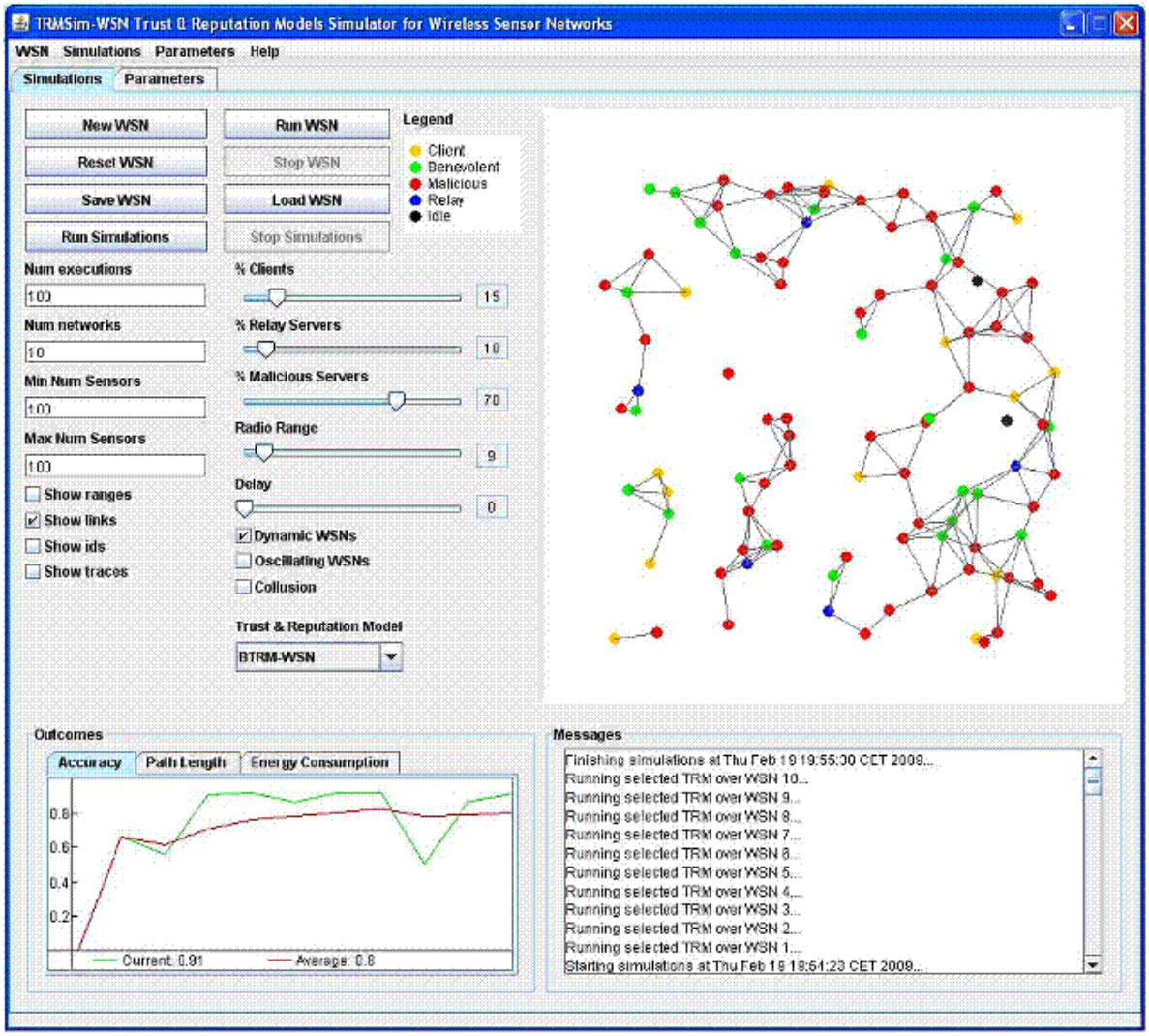


Fig. 13 Snapshot of TRMSim-WSN, a trust & reputation models simulator for WSNs

network. And if the proportion of fraudulent servers is under a 80%, we can reach a selection percentage of trustworthy servers up to 70%.

It has therefore been proved that BTRM WSN is highly scalable, accurate, light and robust. Its main deficiencies come when the percentage of malicious servers is greater than or equal to 90%. So the key factor that makes our model failing when searching the most trustworthy server through the most reputable path is that proportion of fraud ulent servers.

We have proposed two versions of our model, depending on the capabilities of the WSN we are dealing with and on the security restrictions we would like to apply. Thus, the first version is lighter and more scalable while the second is more resilient and accurate.

Regarding security, we have identified and described some security threats that could be applied in our model and other similar trust and reputation models for WSNs.

This paper opens, however, several future ways. For in stance, we have used the same parameters of our model in every case. We think it would be better if each client could decide the values of its parameters and, even more, auto adjust them along the time, in order to get a better perfor mance.

We also need to improve the outcomes got when the pro portion of malicious servers is equal to 90%, and specifi cally, when a collusion is formed.

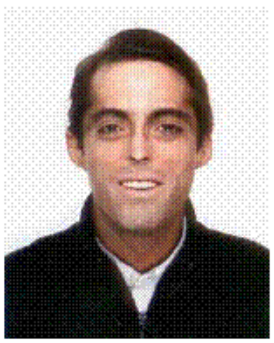
Further theoretical explanations about the performance variation between the scenarios in Sect. 5 will be considered as an extension of our work.

Finally, we are planning to add more functionality to our visual simulator TRMSim-WSN. Our intention is to make it as much generic as possible, so it can be easily used in order to test any other trust and reputation model over WSNs.

Acknowledgements This research work has been partially funded by SWIFT (Secure Widespread Identities for Federated Telecommunications, FP7-ICT-2007-1, Grant No.: 215832) EU IST project, and by a Séneca Foundation grant within the Human Resources Researching Training Program 2007 (06826/FPI/07). Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

References

1. Römer, K., & Mattern, F. (2004). The design space of wireless sensor networks. *IEEE Wireless Communications*, *11*(6), 54–61.
2. Li, F., & Wang, Y. (2007). Routing in vehicular ad hoc networks: a survey. *Vehicular Technology Magazine IEEE*, *2*(2), 12–22.
3. Marsh, S. P. (1994). *Formalising trust as a computational concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling.
4. Marti, S., & Garcia-Molina, H. (2006). Taxonomy of trust: categorizing P2P reputation systems. *Computer Networks*, *50*(4), 472–484.
5. Dorigo, M., & Gambardella, L. (1997). Ant colony system: a cooperative learning approach in the traveling salesman problem. *IEEE Transaction on Evolutionary Computing*, *1*(1), 53–66.
6. Dorigo, M., Gambardella, L., Birattari, M., Martinoli, A., Poli, R., & Stützle, T. (2006). Ant colony optimization and swarm intelligence. In *LNCS: Vol. 4150. 5th international workshop, ANTS 2006*. Brussels: Springer.
7. Cordón, O., Herrera, F., & Stützle, T. (2002). A review on the ant colony optimization metaheuristic: basis, models and new trends. *Mathware and Soft Computing*, *9*(2–3), 141–175.
8. Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Bradford Book
9. Boukerche, A., Xu, L., & El-Khatib, K. (2007). Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, *30*(11–12), 2413–2427.
10. Dhurandher, S. K., Misra, S., Obaidat, M. S., & Gupta, N. (2009). An ant colony optimization approach for reputation and quality-of-service-based security in wireless sensor networks. *Security and Communication Networks*, *2*(2), 215–224.
11. Chen, H., Wu, H., Zhou, X., & Gao, C. (2007). Agent-based trust model in wireless sensor networks. In *Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing, SNPD'03* (pp. 119–124).
12. Ganeriwal, S., & Srivastava, M. B. (2004). Reputation-based framework for high integrity sensor networks. In *SASN'04: Proceedings of the 2nd ACM workshop on security of Ad hoc and sensor networks* (pp. 66–77). New York: ACM.
13. Michiardi, P., & Molva, R. (2002). CORE: a collaborative reputation mechanism to enforce node cooperation in mobile Ad hoc networks. In *Proceedings of the IFIP TC6/TC11 sixth joint working conference on communications and multimedia security* (pp. 107–121). Deventer: Kluwer, B.V.
14. Srinivasan, A., Teitelbaum, J., & Wu, J. (2006). DRBTS: distributed reputation-based beacon trust system. In *DASC'06: Proceedings of the 2nd IEEE international symposium on dependable, autonomic and secure computing* (pp. 277–283). Washington: IEEE Computer Society.
15. Buchegger, S., & Le Boudec, J. Y. (2004). A robust reputation system for P2P and mobile Ad-hoc networks. In *Proceedings of the second workshop on the economics of peer-to-peer systems, Cambridge MA, USA*.
16. Buchegger, S., & Boudec, J.-Y. L. (2002). Performance analysis of the CONFIDANT protocol: cooperation of nodes. In *Proceedings of IEEE/ACM symposium on mobile Ad hoc networking and computing (MobiHOC)*. Lausanne: IEEE.
17. Almenárez, F., Marín, A., Campo, C., & García, C. (2004). PTM: a pervasive trust management model for dynamic open environments. In *Privacy and trust, first workshop on pervasive security and trust, Boston, USA*.
18. Glover, F. W., & Kochenberger, G. A. (2003). *Handbook of metaheuristics (International series in operations research & management science)*. Berlin: Springer.
19. Li, L., & Halpern, J. Y. (2001). Minimum-energy, mobile wireless networks revisited. In *IEEE International Conference on Communications, ICC 2001* (Vol. 1, pp. 278–283).
20. Sánchez, J.A., & Ruiz, P.M. (2006). Improving delivery ratio and power efficiency in unicast geographic routing with a realistic physical layer for wireless sensor networks. In *Proc. 9th Euromicro conference on digital system design (DSD'06)* (pp. 591–597).
21. Gómez, M.F. TRMSim-WSN, a trust & reputation models simulator for wireless sensor networks, http://ants.dif.um.es/~felixgm/research/trmsim-wsn.
22. Xiong, L., & Liu, L. (2004). PeerTrust: supporting reputation-based trust in peer-to-peer communities. *IEEE Transactions on Knowledge and Data Engineering*, *16*(7), 843–857.



Félix Gómez Mármol is a PhD student in the Department of Information and Communications Engineering of the University of Murcia. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received an MSc in computer engineering from the University of Murcia. Contact him at felixgm@um.es

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security and management of IPv4/IPv6 communication networks. He received an MSc and PhD in computer engineering from the University of Murcia. Contact him at gregorio@um.es

5

Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems

Title:	Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Computer Standards & Interfaces, Special Issue on Information and Communications Security, Privacy and Trust: Standards and Regulations
Impact factor (2008):	1.074 (position 42/86)
Publisher:	Elsevier
Volume:	32
Number:	4
Pages:	185-196
Year:	2010
Month:	June
DOI:	http://dx.doi.org/10.1016/j.csi.2010.01.003
State:	Published

Table 5: Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems

Computer Standards & Interfaces 32 (2010) 185–196

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems

Félix Gómez Mármol *, Gregorio Martínez Pérez

Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30.071 Murcia, Spain

ARTICLE INFO

Article history:
Received 30 January 2008
Received in revised form 15 January 2010
Accepted 23 January 2010
Available online 2 February 2010

Keywords:
Trust and reputation management
Trust and reputation models taxonomy
Trust and reputation standardization

ABSTRACT

Different trust and/or reputation models have arisen in the last few years. All of them have certain key processes in common such as scoring, ranking, rewarding, punishing or gathering behavioral information. However, there is not a standardization effort for these kinds of models. Such effort would be beneficial for distributed systems such as P2P, ad-hoc networks, multi-agent systems or Wireless Sensor Networks. In this paper we present a pre-standardization approach for trust and/or reputation models in distributed systems. A wide review of them has been carried out, extracting common properties and providing some pre-standardization recommendations. A global comparison has been done for the most relevant models against these conditions, and an interface proposal for trust and/or reputation models has been proposed.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Designing an accurate and efficient trust and/or reputation model for distributed and heterogeneous environments is nowadays a research challenge. Different issues have to be taken into consideration when developing such models.

The problem to be solved here consists of deciding in a distributed environment which entity is the most reliable to interact with, in terms of confidence and reputation. That is, having a system where different entities offer some services or goods and other ones are requesting those services, the former will always look for the best self profit, while the latter will demand the best services with respect to some quality characteristics, properties or attributes.

Nevertheless, most of the times it is not feasible or realistic to assume the existence of service level agreements (SLA) or the presence of a centralized entity or architecture (such as a PKI), supplying reliable information regarding the actual and current behavior of every service provider in the system.

Hence, requesters have to determine on their own which service providers are the best ones according to certain criteria. Under these conditions, trust and/or reputation models are aimed to select the most trustworthy entity all over the system offering a certain service.

And although several authors have proposed different models in this way, there is a lack of standardization orientations when designing a trust and/or reputation model for distributed systems.

Many scenarios would benefit from the existence of some trust and/or reputation model standardization recommendations, such as P2P networks, multi-agent systems, ad-hoc networks, Wireless Sensor Networks, file-sharing systems, etc, since these scenarios already have standards in many other issues, but not in managing trust and/or reputation between different entities.

The rest of the paper is organized as follows: we present a review of the main existing trust and/or reputation models in Section 2. A classification of the studied models has been done in Section 3. Section 4 describes the components of a general trust and/or reputation model and gives some designing recommendations that might be considered as part of a pre-standardization approach. Finally, Section 5 exposes some conclusions and future work to be done towards pre-standardization of trust and/or reputation models.

2. Analysis of main trust and reputation models

Different trust and reputation model proposals have been suggested recently. In this section we will describe the most representative ones, exposing their main characteristics. Our intention will consist of extracting certain common features from them and providing a set of recommendations for a pre-standardization process.

2.1. Multi-agent system models

Multi-agent systems are supposed to reflect the collective behavior of human societies, since intelligent agents aim to represent human reasoning and behavior in electronic communities.

In such systems, individuals may collaboratively decide who to interact with, forming thus a social network which improves the quality of the decisions to be made.

Several trust and/or reputation models have been developed in this field in order to achieve those goals.

* Corresponding author. Tel.: +34 868 88 78 66; fax: +34 868 88 41 51.
E-mail addresses: felixgm@um.es (F. Gómez Mármol), gregorio@um.es (G. Martínez Pérez).

0920-5489/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2010.01.003

2.1.1. Sporas

In [44], for instance, authors suggest *Sporas* as a reputation mechanism in agent systems where the reputation is computed recursively and where the more recent a rating is, the more weight it has.

Thus, the reputation rating at time i, R_i, is obtained recursively from the previous reputation R_{i-1} and the purchase rating W_i as

$$R_i = R_{i-1} + \frac{1}{\theta} \cdot \Phi(R_{i-1}) \cdot (W_i - R_{i-1}).$$

Where θ is the effective number of ratings taken into account in an evaluation ($\theta > 1$), W_i represents a rating given by user i, and function Φ is defined in order to slow down the incremental changes for very reputable users as follows:

$$\Phi(R_{i-1}) = 1 - \frac{1}{1 + e^{-\frac{R_{i-1} - D}{\sigma}}}.$$

Where D is the maximum possible reputation value and σ is the acceleration factor of the damping function Φ. Hence, the smaller the value of σ, the steeper the damping factor $\Phi(R)$.

2.1.2. Regret

The model proposed in [30], called *Regret* (one of the most representative trust and reputation models in multi-agent systems), manages the reputation from three different dimensions: the individual one, given from direct interactions with the agent; the social one, from previous experiences of group members with the agent and its acquaintances; and the ontological one, given by the combination of multiple aspects in order to build a reputation about complex concepts.

Authors define $\iota = (a, b, o, \varphi, \tau, W)$ as the impression of agent a about outcome o of agent b for a certain subject φ at time τ, being $W \in [-1, 1]$ the subjective opinion of a.

Thus, an individual reputation at time τ from agent a's point of view and satisfying pattern p, $R^i(IDB_p^a)$, is computed as

$$R^i\left(IDB_p^a\right) = \sum_{\iota_i \in IDB_p^a} \rho(\tau, \tau_i) \cdot W_i$$

where IDB_p^a is agent a's impressions database satisfying the pattern p, $\rho(\tau, \tau_i) = \frac{f(\tau_i, \tau)}{\sum_{\iota_j \in IDB_p^a} f(\tau_j, \tau)}$, and $f(\tau_i, \tau)$ is a time dependent function that gives higher values to values closer to τ.

The social reputation of agent b's group, g, from the point of view of agent a, about subject φ at current time, $R_{a-g}(\varphi)$, is

$$R_{a-g}(\varphi) = \sum_{b_i \in g} \omega^{ab_i} \cdot R_{a-b_i}(\varphi)$$

where $\sum_{b_i \in g} \omega^{ab_i} = 1$. Social reputations R_{g-b} and R_{g-g} are also computed in a similar way. And finally, a reputation measure that combines both individual reputation and social one is defined as follows:

$$SR_{a-b}(\varphi) = \xi_{ab} \cdot R_{a-b}(\varphi) + \xi_{ag} \cdot R_{a-g}(\varphi) + \xi_{gb} \cdot R_{g-b}(\varphi) + \xi_{gg} \cdot R_{g-g}(\varphi)$$

where $\xi_{ab} + \xi_{ag} + \xi_{gb} + \xi_{gg} = 1$.

At last, being $C_\varphi = children(\varphi)$, the reputation of node φ in an ontological graph is defined as

$$OR_{a-b}(\varphi) = \begin{cases} \sum_{\phi \in C_\varphi} w_\phi \cdot OR_{a-b}(\phi) & \text{if } C_\varphi \neq \varnothing \\ SR_{a-b}(\varphi) & \text{otherwise.} \end{cases}$$

2.1.3. AFRAS

Authors of *AFRAS* [7] propose a reputation mechanism in multi-agent systems whose main characteristic is the modelling of an agent reputation and the interaction rating as fuzzy sets.

Thus, the reputation fuzzy set of an agent at time i, R_i, is computed from the i-th satisfaction fuzzy set, S_i, as follows:

$$R_i = R_{i-1} \cdot W_2 + S_i \cdot W_1$$

where W_1 and W_2 weights determine how much the last reputation value, R_{i-1}, and the satisfaction, S_i, respectively contribute; and both are defined from a sole weight $W \in [0, 1]$ as follows:

$$W_1 = 1 - \frac{W}{2} \quad W_2 = \frac{W}{2}$$

So, the reputation R_i can be expressed as

$$R_i = R_{i-1} + \frac{(S_i - R_{i-1}) \cdot (1 - W)}{2} = R_{i-1} + \frac{(S_i - R_{i-1}) \cdot (1 - \rho_i)}{2}$$

where ρ_i may be understood as a function based on historic transactions indicating the weight of past reputation values at time i, and can be recursively computed as follows

$$\rho_i = \frac{\rho_{i-1} + \Delta(R_{i-1}, S_i)}{2}$$

where $\Delta(R_{i-1}, S_i) \in [0, 1]$ measures the similarity between the two fuzzy sets R_{i-1} and S_i. Having this, we can deduce that, if the prediction was right (i.e. $\Delta \approx 1$), then memory at time i, ρ_i, is increased by $1/2 + \rho_{i-1}/2$. Otherwise ($\Delta \approx 0$), the memory at time i is halved.

2.1.4. MTrust

In *MTrust* [33], authors use a Bayesian network in order to compute the trust value among entities in the network. It is focused on a mobile agent system, where the cooperative interactions among these agents and their respective visited hosts is ensured.

Therefore, $T^M_{O_i^T - vh_i} \in [0, 1]$ is defined as a trust value between a truster owner of agents O_i^T (that is, a node who generates a set of mobile agents $A^O = \{ma_1^O, \ldots, ma_j^O\}$) and a visited host vh_i, computed using a method from set M, where $M = \{$predefined trust value, general trust value, Feedback Aggregation method for inexperienced truster (FA_{IN}), Feedback Aggregation method for experienced truster (FA_{EX}), Bayesian Network (BN), a combination of FA_{EX} and $BN\}$.

A predefined trust value is a trust value deduced from a truster's behavior. A general trust value is computed from an averaging of all trustees' trust values. If a BN is used, every owner develops a simple BN model which is updated with every received feedback.

FA_{IN}: A truster computes each trustee's reputation perceived by each rater O_i^R as an expectation of beta distribution as follows

$$Rep_{O_i^R - vh_i} = \frac{\alpha}{\alpha + \beta}$$

where $\alpha = N_p + 1$, $\beta = N_N + 1$ and N_p and N_N are the number of positive and negative consistent feedbacks, respectively. A general form of a trustee's trust value is presented next:

$$T^{FA_{IN}}_{O_i^T - vh_i} = \sum_{k=1}^{Num} w_k \cdot \frac{\sum_{i=1}^{N_k} Rep_{O_i^R - vh_i}}{N_k}$$

where Num is the number of ranges, w_k is the weight of range k and N_k is the number of raters in range k.

To use FA_{EX} there are three situations a truster must consider. The first, if all raters are unknown, then the truster applies FA_{IN} with a

weight factor of 1 for each rater. The second, if all raters are known, FA_W is utilized with weighted reputation value. The third, if the raters consist of a combination of known and unknown raters, then a truster separates raters into an unknown and a known group, and applies the appropriate methods as described previously.

MTrust is also composed of a reputation system relying on a truthful feedback submission algorithm and on a set of distributed feedback information storages. It supposes the existence of a PKI and lacks in capability to prevent a strategically malicious trustee who dynamically changes its pattern of service.

[2], [26] and [28] expose trust and reputation models for multi-agent systems. [2] also emphasizes its use in P2P networks and [28], for example, makes use of fuzzy sets in order build its model.

Finally, [1] is one of the first works done in the field of trust and reputation management in virtual communities and presents a trust and reputation model for a multi-agent system.

2.2. P2P network models

P2P on-line communities are currently receiving much attention as means of sharing and distributing information. However, they are commonly perceived as an environment offering both opportunities and threats. Thus for instance, their open nature offers an almost ideal scenario for the spread of wrong data, noise information, or even malicious services.

One way to minimize such threats in such networks is to use community-based reputations in order to help estimating the trustworthiness of peers [19]. Many trust and/or reputation mechanisms have been proposed so far for P2P networks.

2.2.1. DWTrust

The novelty and strong point of the model proposed in [18], DWTrust, is that all the factors that have influence on the trust are represented as dynamic weights that adapt themselves depending on the trust policy of each node, due to some feedbacks. This kind of representation simplifies the modelling as well as the computing of the trust itself.

In DWTrust the trust of node x about node y at time t is modelled as a tuple:

$$w^x_{y,t} = \left(b^x_{y,t}, d^x_{y,t}, u^x_{y,t}, a^x_{y,t}\right)$$

where $b^x_{y,t}$, $d^x_{y,t}$, $u^x_{y,t}$ and $a^x_{y,t}$ represent the credibility, uncredibility, uncertainty, and relative atomicity functions of x about y at time t satisfying $b^x_{y,t} + d^x_{y,t} + u^x_{y,t} = 1$.

Let $\alpha_i, \beta_i, \gamma_i \in [0, 1]$ and $\lambda_i \in [0, \infty)$ be the success weight, the failure weight, the time weight and the importance weight of the i_{th} transaction, respectively. Let $R_i \in \{0, 1\}$ be the binary outcome of the i_{th} transaction. Then we have:

$$b_i = \frac{\sum_{k=0}^{i} \alpha_k \lambda_k R_k \gamma_k}{\sum_{k=0}^{i} \alpha_k \lambda_k R_k + \sum_{k=0}^{i} \beta_k \lambda_k (1 - R_k) + 1}$$

$$d_i = \frac{\sum_{k=0}^{i} \beta_k \lambda_k (1 - R_k) \gamma_k}{\sum_{k=0}^{i} \alpha_k \lambda_k R_k + \sum_{k=0}^{i} \beta_k \lambda_k (1 - R_k) + 1}$$

$$u_i = \frac{1 + \sum_{k=0}^{i} (1 - \gamma_k) \alpha_k \lambda_k R_k + \sum_{k=0}^{i} (1 - \gamma_k) \beta_k \lambda_k (1 - R_k)}{\sum_{k=0}^{i} \alpha_k \lambda_k R_k + \sum_{k=0}^{i} \beta_k \lambda_k (1 - R_k) + 1}$$

$$\alpha_i = \alpha_{i-1} + \Delta\alpha$$

$$\begin{cases} \alpha_i = \alpha_{i-1} + \Delta\alpha R_i \\ \beta_i = \beta_{i-1} + \Delta\beta (1 - R_i) \\ \gamma_i = \gamma_{i-1} + \Delta\gamma. \end{cases}$$

For optimistic nodes, α is assigned a bigger value than β in order to increase the effect of a successful transaction and to incentive cooperation. On the other hand, a pessimistic node would adopt $\alpha < \beta$ in order to slow down trust behavior.

Feedback rules are defined in feedback control unit in order to modify α and β dynamically, with $\Delta\alpha$ and $\Delta\beta$.

Finally, the propagation of trust can be computed with both the combination of discounting operation $w^A_{x,r} \equiv w^B_{x,r} \otimes w^B_{x,s}$ (recommendation) and consensus operation $w^{A,B}_{x,r} \equiv w^A_{x,r} \oplus w^B_{x,s}$ (combination).

2.2.2. AntRep

AntRep [41] is a novel model where reputation evidences are distributed over a P2P network, based on the swarm intelligence paradigm [21]. Specifically, authors propose the use of an ant system [9] for building trust relationships in P2P networks efficiently.

In AntRep each peer has a reputation table (RT) which is very similar with the distance-vector routing table [36], but differs from: (i) each peer in the RT corresponds to one reputation content; (ii) the metric is the probability of choosing each neighbor as the next hop instead of the hop count to destinations.

There are two kinds of forward ants sent out for a particular reputation:

1. Unicast ants are sent out to the neighbor with the highest probability in the reputation table.
2. Broadcast ants are sent out when there is no preference to neighbors. This happens either when no path to the reputation has been explored or the information the node has is outdated.

Once forward ants find the required evidence (reputation information), a backward ant is generated. When the backward ant visits each node i, it updates the reputation table at the same time. This updating is carried out due to the next reinforcement rule:

$$P_i(r) = \frac{[\tau_i(r)]^{\alpha} [\eta_i(r)]^{\beta}}{\sum_{j \in N} [\tau_j(r)]^{\alpha} [\eta_j(r)]^{\beta}}$$

where η_i is the goodness value of the link between current node and its neighbor node j and τ_i is the pheromone deposit, which is defined as follows: if at time $t + \Delta t$, current node receives a backward ant from node i, then

$$\tau_i(t + \Delta t) = f(\tau_i(t), \Delta t) + \Delta p$$

$$\tau_j(t + \Delta t) = f\left(\tau_j(t), \Delta t\right), j \in N, j \neq i$$

$\Delta p = \frac{k}{f[c]}$, being $k > 0$ a constant, $f[c]$ a nondecreasing function of cost c and c could be any parameter revealing the information of evidence or the scenario of current network. $f[\tau_i(t), \Delta t)$ is the pheromone evaporation function:

$$f(\tau_i(t), \Delta t) = \frac{\tau_i(t)}{e^{\Delta t / k}}.$$

Finally, α and β are constants varied in different network environments.

Another function of the pheromone is to decide when to send out broadcast forward ants. When node k receives a request at time t, it first searches if there is an entry for the desired evidence. If no such entry exists, it simply sends out broadcast ants. Otherwise, it finds the one with the highest probability.

2.2.3. EigenTrust

EigenTrust [20] has become one of the most cited and compared trust models for P2P networks. It achieves decreasing the number of downloads of inauthentic files in a P2P file-sharing network by assigning

each peer a unique global trust value, based on the peer's history of uploads.

Let $sat(i,j)$ be the number of satisfactory transactions peer i has had with peer j (equally, $unsat(i,j)$). Then, the local trust value s_{ij} is defined as follows:

$$s_{ij} = sat(i,j) \quad unsat(i,j).$$

Some distribution $\vec{p}$ (with $p_i \in [0,1]$) over pre-trusted peers is defined. For instance, if some set of peers P are known to be trusted, then $p_i = 1/|P|$ if $i \in P$, and $p_i = 0$ otherwise. Having this, a normalized local trust value $c_{ij} \in [0,1]$ can be defined as:

$$c_{ij} = \begin{cases} \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)} & \text{if } \sum_j \max(s_{ij},0) \neq 0 \\ p_j & \text{otherwise.} \end{cases}$$

Thus, if peer i does not know anybody, or does not trust anybody, it will choose to trust the pre-trusted peers.

In EigenTrust, the global reputation of each peer i is given by the local trust values assigned to peer i by other peers, weighted by the global reputation of the assigning peers. Therefore, the aggregation of normalized local trust values is computed as follows:

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

where t_{ik} represents the trust that peer i places in peer k based on asking his friends. If C is defined to be the matrix $[c_{ij}]$ and $\vec{t}_i$ to be the vector containing the values t_{ik}, then $\vec{t}_i = C^T \vec{c}_i$.

In order to get a wider view, peer i may wish to ask his friends' friends ($\vec{t}_i^{\,[2]} = (C^T)^2 \vec{c}_i$). If he continues in this manner ($\vec{t}_i^{\,[n]} = (C^T)^n \vec{c}_i$), he will have a complete view of the network after n iterations.

If n is large enough, the trust vector $\vec{t}_i$ will converge to the same vector for every peer i. Namely, it will converge to the left principal eigenvector of C. In other words, t is a global trust vector in this model and its elements, t_i, quantify how much trust the system as a whole places in peer j.

Finally, in order to avoid malicious collectives in P2P networks, the global trust value is re-defined as:

$$\vec{t}^{\,(k+1)} = (1-a) C^T \vec{t}^{\,(k)} + a\vec{p}$$

where a is some constant less than 1 and $\vec{t}^{\,(0)} = \vec{p}$.

2.3. Ad-hoc network models

Mobile ad-hoc networks (MANETs) comprise computer nodes which communicate over wireless links without any central control. Hence, countermeasures for node misbehavior and selfishness are mandatory requirements in this kind of systems.

Estimation of trust and reputation in ad-hoc networks constitutes an inevitable basis for hybrid networks to inter-operate as well as for users to establish a communication among them.

2.3.1. RRS for P2P and MANETs

In [6] authors present an enhancement of CONFIDANT [5], which is a robust reputation system for P2P and mobile ad-hoc networks where everyone maintains a reputation rating R_{ij} and a trust rating T_{ij} about everyone else they care about.

From time to time first-hand reputation information F_{ij} is exchanged with others and only second-hand reputation information that is not incompatible with the current reputation rating is accepted, using a modified Bayesian approach.

Every node i keeps a summary record of first-hand information about node j in a data structure called F_{ij}, having the form (α, β).

Let's assume node i makes one individual observation about j and let be $s = 1$ if this observation is qualified as misbehavior, and $s = 0$ otherwise.

Then the updating of first-hand information is carried out as follows

$$\alpha = u\alpha + s$$

$$\beta = u\beta + (1 - s)$$

where u is a discount factor for past experiences, which serves as the fading mechanism and is defined as $u = 1 - \frac{1}{m}$, being m the order of magnitude of the number of observations over which it is believed it makes sense to assume stationary behavior.

The reputation rating R_{ij} is also defined by two numbers, say (α', β'). Initially it is set to $(1, 1)$ and it is updated on two types of events: (1) when first-hand observation is updated and (2) when a first-hand information F_{kj} published by some other node k is accepted and copied.

In the former case, the update is the same as for the first-hand information. In the latter case, if node i considers k trustworthy, then F_{kj} is considered by node i, who modifies R_{ij} according to

$$R_{ij} = R_{ij} + wF_{kj}$$

Here, w is a small positive constant. Otherwise i considers k untrustworthy and uses the results of the deviation test as follows.

Let $\mathbb{E}(Beta(\alpha, \beta))$ be the expectation of the distribution $Beta(\alpha, \beta)$ and let $F_{kj} = (\alpha_f, \beta_f)$ and $R_{ij} = (\alpha', \beta')$. Thus, the deviation test is

$$\mathbb{E}(Beta(\alpha_f, \beta_f)) \quad \mathbb{E}\left(Beta\left(\alpha', \beta'\right)\right) \geq d$$

where d is a positive constant (deviation threshold). If the deviation test is positive, the first-hand information F_{kj} is considered incompatible and is not used. Else F_{kj} is incorporated as shown before.

The trust rating T_{ij} is equal to (γ, δ). Initially $(\gamma, \delta) = (1, 1)$ and an update is performed whenever node i receives a report by some node k on first-hand information about node j.

Let $s = 1$ if the deviation test succeeds, and $s = 0$ otherwise. The trust rating $T_{ij} = (\gamma, \delta)$ is updated by

$$\gamma = v\gamma + s$$

$$\delta = v\delta + (1 - s)$$

where v is a discount factor for trust, similar to u.

Finally, how node i considers other nodes as trustworthy or misbehaving is carried out as follows:

$$\begin{cases} normal & \text{if } \mathbb{E}\left(Beta\left(\alpha', \beta'\right)\right) < r \\ misbehaving & \text{if } \mathbb{E}\left(Beta\left(\alpha', \beta'\right)\right) \geq r \end{cases}$$

$$\begin{cases} trustworthy & \text{if } \mathbb{E}(Beta(\gamma, \delta)) < t \\ untrustworthy & \text{if } \mathbb{E}(Beta(\gamma, \delta)) \geq t \end{cases}$$

2.3.2. PTM

PTM [3] is a decentralized trust model for pervasive dynamic open environments where human intervention has been minimized and a recommendation protocol has been defined. Authors also implemented the model using the J2ME Personal Profile [25].

Each node has its own key pair, a list of trustworthy and untrustworthy users, behavioral information and available certificates. Trust relationships are expressed through fuzzy logic, fulfilling certain properties, such as: reflexive, non-symmetrical, conditionally transitive and dynamic. These relationships can be established as direct or indirect. Let A and B be two unknown users.

- In the first case, A will trust B without intervention of third parties. For that A takes into account some available previous knowledge about B, otherwise A will use an inference engine to interpret the established rules. Such rules are based on the user's security context.
- The indirect trust relationships are given by recommendations from TTPs. A TTP is a peer who has a trust value higher than a certain threshold. Such recommendations are distributed using a pervasive recommendation protocol (PRP) among close entities or using public key certificates.

Once the trust relationship is established, A calculates B's degree of trust, T_i:

$$T_i = \begin{cases} T_{i-1} + \omega \cdot V_{a_i}(1 - T_{i-1}) & \text{if } V_{a_i} > 0 \\ T_{i-1}\left(1 - \omega + \omega \cdot V_{a_i}\right) & \text{otherwise} \end{cases}$$

where the strictness factor ω is related to the user's disposition regarding the present and the past, and V_a is computed according to the related weight to each kind of action, W_a, which is rewarded or punished according to the past behavior, both positive, a^+, and negative, a^-, as follows:

$$V_{a_i} = W_{a_i}^{|m|} \cdot \frac{(a^+ - a^-)\left((a^+ - a^-) \cdot \sigma\right)^{2m}}{(a^+ - a^-)\left((a^+ - a^-) \cdot \sigma\right)^{2m} + 1}$$

$\sigma \in (0, 0.05]$ determines the increment and can be calculated from the security level $m \geq 1$.

The calculation of action value (either positive, a^+, or negative, a^-) includes classic a priori probabilities about the user behavior. From a priori probabilities we can calculate posteriori ones applying the Bayes' theorem.

$$P\left(a^i \mid H_{a\sigma}\right) = \frac{P\left(H_{a\sigma} \mid a^i\right) P\left(a^i\right)}{P(H_{a\sigma})}$$

where $H_{a\sigma}$ is the historical behavior. The density function represents the distribution of these probabilities for binary events in the interval $[0, 1]$, so:

$$f_k(X \mid H_{a\sigma}) = \frac{(a^+ + a^- + 1)!}{(a^+)!(a^-)!} X^{a^+}(1 - X)^{a^-}.$$

This probabilistic model could be of use for evaluating the risk implicitly considered within the trust definition.

2.4. Wireless Sensor Network models

Wireless Sensor Networks have recently gained a high attention because of their multiple and innovative applications as well as their singular characteristics.

They are, however, susceptible to a large number of security threats, some of which might be effectively mitigated with an accurate trust and reputation management [12,29].

The main goals of reputation and trust-based systems in wireless communication networks [34], consist of providing information that allows nodes to distinguish between trustworthy and non-trustworthy nodes, encouraging nodes to be trustworthy and discouraging participation of nodes that are untrustworthy.

Additionally, two more goals of a reputation and trust-based system from a wireless communication network perspective can be identified. The first goal is to cope with any kind of observable misbehavior. And the second goal is to minimize the damage caused by insider attacks.

2.4.1. QDV

Authors of [10] present an Ant Colony Optimization approach for reputation and quality-of-service-based security in WSNs. They specifically propose a quality-based distance vector protocol known as QDV, where the more reputation a node has, the more reliable it is for communication purposes.

QDV is able to protect the network against packet injection by those malicious nodes which have been detected. This protection is made by identifying those nodes who drop the packets forwarded to them.

In this model reputation is based on pheromone content of a path for communication. Thus, a path having more deposits of pheromone, τ_{ij}, is considered more secure. On the other hand, QoS considers the distance between two communicating nodes, η_{ij}. Therefore:

$$\phi_{ij}(t) = \frac{\sum_{k=1}^{n_i} \tau_{kj}}{n_i}$$

where τ_{kj} is the pheromone trace between nodes k and j, n_i is the number of i's neighbors, and if $\phi_{ij}(t) < \tau_{min}$, misbehavior or security violation is detected, which means node i has less forwarding capabilities.

In the same direction, QoS is defined as the percentage of exposed traffic according to:

$$\theta_{ij}(t) = \frac{\sum M_s(t) + \sum M_r(t) - \sum M_d(t)}{\sum M_s(t) + \sum M_r(t)} \eta_{ij}$$

being $\sum M_s(t)$, $\sum M_r(t)$ and $\sum M_d(t)$ the total number of generated, received and dropped packets, respectively.

Finally, the quality-of-security, QSec, depends on the two previous parameters and defines the communication and transfer between two nodes. It is the deciding factor as to which node needs to be selected as the next node in the path and is computed as the weighted sum of reputation and QoS:

$$W_{ij}(t) = w_1 \phi_{ij}(t) + w_2 \theta_{ij}(t)$$

where w_1 and w_2 are the weight parameters for the respective factors Reputation and QoS, and $w_1 + w_2 = 1$.

2.4.2. ATRM

ATRM [4] is an agent-based trust and reputation management scheme for Wireless Sensor Networks (WSN), where trust and reputation management is carried out locally with minimal overhead in terms of extra messages and time delay.

It is based on a clustered WSN with backbone, and its core is a mobile agent system. It requires a node's trust and reputation information to be stored respectively in the forms of t-instrument and r-certificate by the node itself. In addition, ATRM requires every node to locally hold a mobile agent that is in charge of administrating the trust and reputation of its hosting node.

Considering any two nodes n_i and n_j, the t-instrument issued by n_i to n_j under context $C_\star$ is defined as:

$$TI\left(n_i, n_j, C_\star\right) = E_{sk}(D, H(D))$$

where $E_{sk}\{M\}$ is an encryption function using n_i's symmetric key, $H\{M\}$ is a hash digest function, $D = \{ID(n_i), ID(n_j), C_\star, T, \tau_{ij}\}$, T is a time-stamp

implying the time when the t–instrument is issued and t_{ij} is the trust evaluation made by n_i on n_j.

If there are k concerned contexts, for any node n_i, its r–certificate is defined as:

$$RC(n_i) = E_{AK}(R, H(R))$$

where $R = (ID(n_i), T, ((r_1, C_1), (r_2, C_2), \ldots, (r_k, C_k)))$, which means that n_i's reputation is r_1 under context C_1, r_2 under context C_2,...,r_k under context C_k at time point T.

Before starting any transaction between n_i and n_j, the former asks its local mobile agent to obtain the r–certificate of the latter by directly querying n_j's local mobile agent. Based on n_j's r–certificate, n_i decides whether or not to start the transaction.

After the transaction is finished, n_i makes a trust evaluation on n_j based on the quality of the service it gets, and then submits this evaluation to its local mobile agent which then accordingly generates a t–instrument for n_j and sends it to n_j's local mobile agent.

Based on the collected t–instruments, a mobile agent periodically issues its hosting node updated r–certificates. But since mobile agents are designed to travel over the entire network and run on remote nodes, they must be launched by trusted entities. Therefore, in ATRM it is assumed that (1) there is a trusted authority that is responsible for generating and launching mobile agents, and (2) mobile agents are resilient against the unauthorized analysis and modification of their computation logic.

Several additional proposals have been developed in the last few years in the field of trust and reputation management for Wireless Sensor Networks. For instance, ATSN [8] based on agent systems, BTRM-WSN [17], based on ant colony systems, RFSN [14], making use of Bayesian theory or DRBTS [35], suggesting an analytic formulation.

3. Trust and reputation models classification

Once we have shown and described several representative trust and reputation models, we are ready to give some recommendations to classify them. First of all we need to extract the main characteristics that most of them share. We do not expect to give an exhaustive and complete classification, but just an approach to be considered as part of a pre-standardization process of trust and reputation models in distributed systems.

The first classification could be done according to what kind of model they are dealing with. As we have seen, it might be either a trust or a reputation model, or even a hybrid one if it manages both issues. These three groups include all the models we have analyzed.

The next level of classification is the scope, environment or target of the model. For instance, we have analyzed trust and/or reputation models for Peer-to-Peer, Ad-Hoc or Wireless Sensor Networks. We have seen models for multi–agent systems as well. And again, some models could be included in more than one of these categories.

Another categorization could depend on what technic or methodology has been used in order to develop the model. We are talking, for example, about using Fuzzy Logic, Bayesian Networks, Social Networks or bio–inspired algorithms. Other authors just give some analytic expressions in order to model trust and/or reputation concepts.

In this way, Table 1 shows a classification of all trust and reputation models exposed in Section 2 and some others, such as TPOD [39] and GroupRep [40], for instance. It includes 30 mechanisms or proposals for trust and/or reputation management, 15 of them applicable in multi–agent systems, 12 in P2P networks, 5 in ad–hoc ones and 7 in WSNs.

Note how some models fit well in more than one category. It is also clear that most of the studied models use analytic expressions to model trust and reputation, and are aimed to work in a P2P network or a multi–agent system.

In fact, 6 out of 30 of the approaches exposed in Table 1 make use of Fuzzy logic, 5 use Bayesian networks or Bayesian theory, 4 apply a bio–inspired mechanism, 3 benefit from social networks research and 12 describe their suggested trust and/or reputation management through an analytic expression, or a set of them.

Therefore, a very appropriate field of investigation (since there is less work done) could be the development of trust and reputation models whose scope are Ad–hoc and Wireless Sensor Network, and whose technic or methodology are a social network or bio–inspired algorithms.

4. Recommendation for pre-standardization

4.1. Trust and reputation definition

Being our purpose the development of a set of recommendations aimed to reach a standardization approach for trust and/or reputation

Table 1
Trust and reputation models comparison.

	Technic					
	Fuzzy	Bayesian	Bio–inspired	Social network	Analytic	
Agent	PATROL-F [38] [1,28]	MTrust [33] BNBTM [42]	AntRep [41]	[26,45]	ATRM [4] [2] ATSN [8]	Trust
	AFRAS [7] PATROL-F [38] [1,28]	MTrust [33] BNBTM [42]	AntRep [41]	Social Regret [31] [26]	Sporas [44] Regret [30] ATRM [4] [2]	Reputation
P2P	PATROL-F [38] PTM [3]	BNBTM [42] PTM [3]	AntRep [41] TDTM [47]		[43] DWTrust [18] TPOD [39] GroupRep [40] EigenTrust [20] [2]	Trust
	PATROL-F [38] PTM [3]	BNBTM [42] PTM [3] RRS [6]	AntRep [41]		[43] GroupRep [40] EigenTrust [20] [2]	Reputation
Ad–hoc	[1]	PTM [3]			ATRM [4] [27]	Trust
	[1]	PTM [3] RRS [6]			ATRM [4]	Reputation
WSN	[22]	RFSN [14]	BTRM-WSN [17] QLOV [10] BTRM-WSN [17]		ATRM [4] DRBTS [35] ATSN [8] ATRM [4] DRBTS [35]	Trust Reputation

models in distributed and heterogeneous systems, the first thing we need is a commonly accepted definition of trust and reputation.

Different definitions of trust and reputation have been given along several research works [1,13,23]. However, we have chosen just one definition for each one. In that way, the definition of trust by [13] is:

> *Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.*

Equally, the definition of reputation by [1] is:

> *A reputation is an expectation about an agent's behavior based on information about it or observations of its past behavior.*

Some authors do not explicitly distinguish between these two concepts whereas others make a clear separation between trust and reputation. We will consider them as different but closely related issues.

4.2. Trust and/or reputation model components

From the analysis of the state of the art presented in Section 2, we can deduce that a trust and/or reputation model is generally composed of five components [24]: gathering behavioral information, scoring and ranking entities (peers, nodes, agents, and sensors), entity selection, transaction, and rewarding and punishing entities. Fig. 1 depicts these elements. Any standard model should be designed having these components in mind.

The first component of a standard trust and/or reputation model, gathering information, should be responsible for collecting behavioral information about the entities in the system. This information would be used to determine how trustworthy an entity is (either on an absolute scale or relative to other entities).

This information might come from several sources such as direct experiences with the targeting entity, neighbors, acquaintances, belonging group or organization, and even witnesses. Pre-trusted entities would be also acceptable as it already happens in some existing models such as EigenTrust [20] or PowerTrust [46].

If this information does not come from direct experiences, another point should be taken into account: the information integrity. That is, the confidence one has on other entities' recommendations or opinions. This is the reason why some models not only evaluate entities as service providers, but also as raters, as well. This is the case of Regret [30], MTrust [33] or [43], for instance.

At this point, several specifically applicable security threats [15,37] may be taken into consideration. Thus, the possibility of malicious entities of forming a collusion or performing a *Sybil* attack [11], for instance, should be avoided or mitigated.

Fig. 1. Trust and/or reputation model elements: general view.

Once an entity's transaction history has been collected and properly weighted, a trust and/or reputation score should be computed for that entity. This computation could be done in different ways. Examples of them could be fuzzy logic, Bayesian networks, analytic expressions or bio-inspired algorithms.

At the end, either a local or a global trust and/or reputation rating is done. This rating, used in order to help an entity to decide which entity to interact with, can be modelled as a binary value (e.g. trusted, untrusted), a scaled integer (e.g. 1, 2, ..., 9, 10), an element from a set of linguistic labels (e.g. {"very trustworthy", "trustworthy", "untrustworthy", "very untrustworthy"}), a value within a continuous interval (e.g. [0, 1]), etc.

It is important to mention the granularity of such score. Some trust and/or reputation mechanisms consider the behavior of an entity as a whole, while others distinguish among the different services offered, by assigning a different rating related to each one of them. As we will see later, an incorrect handling of this issue might lead to important risks when selecting the final service provider due to incoherent aggregation of trust ratings.

Thus, the next component of a standard model would consist of deciding which entity to interact with according to its trust and/or reputation values. At this point not many models explain explicitly how they select that entity, since most of the times it is understood that the most trustworthy entity will be selected, but this has not to be done compulsorily in this way.

In fact, a utility function could be used in order to determine the quality of a service, and therefore help in the final service provider selection.

Having selected an entity to interact with, the transaction itself would be carried out between both entities, giving a certain service or good as a result.

Finally, after receiving the requested service, the client entity should assess that transaction so it can reward or punish the entity who provided that service. In our opinion, this is a very important component in a trust and/or reputation model, since the accuracy of trust and/or reputation ratings highly depend on how fair an entity is evaluated from a given service.

However, not many authors talk about this issue in their works, and they mainly focus on how the behavioral information is gathered and how the trust and/or reputation rating is computed.

Fig. 2 shows an extended view of the elements recommended for a standard trust and/or reputation model in a distributed environment, as we have just described.

4.3. Trust and/or reputation model interfaces

From Fig. 2 representing our approach, an interface can be deduced. For example, a gathering information function should receive a collection of experiences and a collection of pre-trusted entities as parameters. The experiences could be represented as reputation values, or as a collection of positive and negative previous transactions. That function should make use of an inside function in order to check the integrity of the gathered information by measuring the reliability of the sources of information.

A possible scoring and ranking function would receive the outcome of the gathering function as its primary input. That is, it would receive either a collection of recommendations about which entity is more trustworthy, or a collection of weighted transactions (or maybe both).

This function would make use of any mechanism (such as fuzzy logic, Bayesian networks, bio-inspired algorithms, etc.) in order to provide a local or global (or maybe both) value of reputation and/or trust of some or all the entities belonging to the network.

An entity selection function should determine which entity to interact with, according to the received values of trust and/or reputation from the scoring and ranking function. Most of the times just the most trustworthy entity is selected, but this is not mandatory.

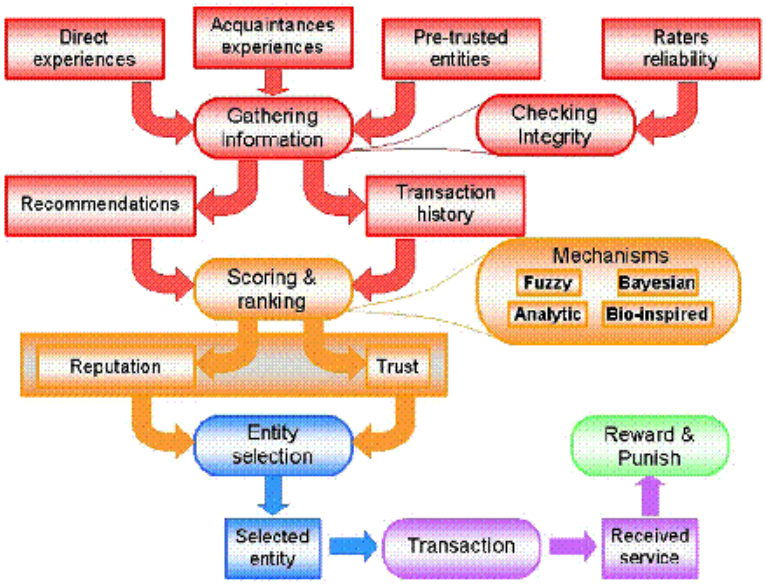


Fig. 2. Trust and/or reputation model elements: extended view.

Once the transaction is carried out and the requested service is received, a reward and punish function should determine the subjective satisfaction with that certain service or good. Thus, according to that satisfaction and to the trust and/or reputation values of the involved entity it will reward or punish that entity in terms of considering it more or less trustworthy or reputable.

As a proof supporting this generic specification, a trust and reputation models simulator, called TRMSim-WSN [16], has been developed implementing these steps within several models included in the tool.

4.4. Additional recommendations towards pre-standardization

Classical trust and/or reputation schemes quite often relied on a centralized Public Key Infrastructure with its Certification Authorities and so on. However, the trust and/or reputation model standardization approach we develop here does not suppose the existence of such PKI. Instead, it expects to define a trust and/or reputation model for distributed and heterogeneous environments, such as P2P, ad-hoc networks, multi-agent system or Wireless Sensor Networks.

In the previous section we described the components a standard trust and/or reputation model should have. Here we present a number of recommendations that, in our opinion, trust and/or reputation models should follow. Nevertheless, the final application of each of those recommendations will depend on the specific restrictions and characteristics of the system and/or application scenario where to utilize this kind of models. We consider some recommendations as more critic and unavoidable in any trust and reputation mechanism, whereas others are more susceptible to each specific environment.

1. Within the gathering information component, a "good" trust and/or reputation model should not rely on the existence of identities. In other words, it may allow entities' anonymity since, as a specific application of privacy, users may only accept participating if a certain amount of anonymity is guaranteed.
   This may vary from no anonymity requirements, to hiding real-world identities behind pseudonyms [32], for instance. Another solution would consist of cryptographically generated unique identifiers and secure hardware modules, as it is the case of RRS (Section 2.3.1).
   However, a purely anonymous system might not be feasible, since the sources of two different transactions would be indistinguishable and unlinkable.
2. When weighting the collected behavioral information and computing the trust and/or reputation values, the more recent a transaction is, the more weight it should has. Fig. 3 shows an example of a function weighting transactions along time.
   By giving a higher consideration to more recent interactions, it is easier to accurately estimate the current behavior of an entity and prevent behavioral oscillations along time. This is the case of Regret (Section 2.1.2), with the use of function $f(t_i, t)$.
   It is important to note that the decision of which discounting function to use has a strong implication in the performance and accuracy of the model.
3. It should be allowed the subjectivity in the assessment of a transaction, i.e., each entity may have its own criteria when evaluating a received service. An entity may evaluate a performed transaction alone or with the opinion and even consensus of other

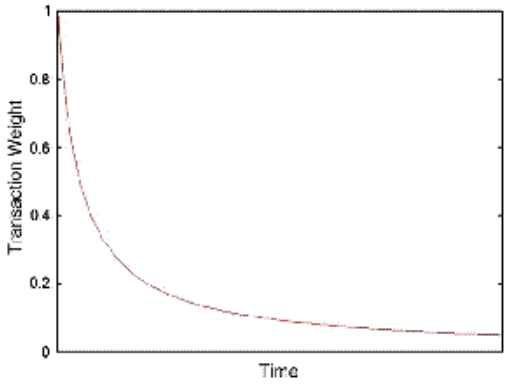


Fig. 3. Weight of a transaction along time.

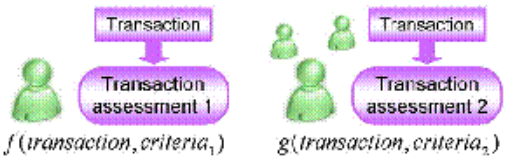


Fig. 4. Subjective transaction assessment.

parties, as it can be checked in Fig. 4.
Allowing subjective evaluation of transactions makes trust and/or reputation scores to fit more and better with the specific goals and necessities of each entity in the system.
In the case of MTrust (Section 2.1.4), an agent can decide which method from set M to use in order to compute the trust between a truster owner of agents and a visited host.

4. Redemption of past malicious entities who has become benevolent should be possible. Otherwise an entity who has had a misbehavior once in the past could not be selected anymore to have a transaction with, even if that transaction could provide a better service than the current most trustworthy entity's transactions.
As mentioned before, the introduction of a utility function might be in some cases very useful since it can help to distinguish which is the best service, not only in terms of the trust and/or reputation of the entity which is providing such service.
This behavior is illustrated in Fig. 5, where a malicious entity (with an initial low selection probability) turns into benevolent (reaching thus a high selection probability).
With this kind of redemption an actually benevolent entity is not eternally punished for a misbehavior maybe due to an involuntary action such as a hardware failure, for example. Furthermore, a malicious entity who has swapped her goodness may have some opportunities to be selected as well.
In AntRep (Section 2.2.2), for instance, the pheromone traces left by the ants, allow former malicious nodes to recover their reputation.
5. Benevolent newcomers should have the opportunity to participate even if there are already trustworthy entities in the system, as it is exemplified in Fig. 6. Once again, the application of utility functions could help to let a newcomer be selected, even if it has a low trust/reputation in the system, but offering a certainly profitable service. This measure prevents a benevolent entity (or a set of them) to achieve a high trust and/or reputation score and, consequently, monopolize all the transactions performed in the system.
Moreover, the fact of assigning an appropriate initial trust and/or reputation value to each newcomer is a problem that should not be neglected, since it will directly determine its selection probability.
In this case, QDV (Section 2.4.1) has a good performance, since its pheromone traces initialization, together with the evaporation of such traces in case of malicious behavior, allows newcomers to have opportunities to be selected as service providers.
Furthermore, the assignment of a new identity in the system could have an associated cost (not necessarily economic), in order to avoid a misbehaving entity to easily leave and re-enter the system as a newcomer without paying a penalty period.

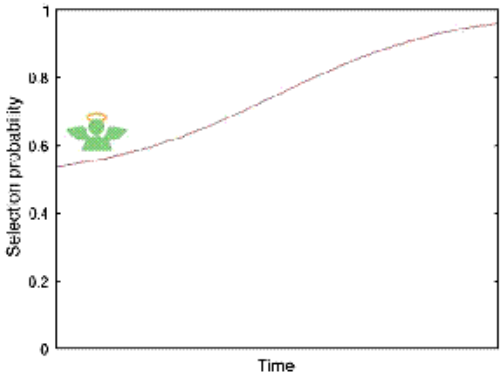


Fig. 6. Chance for benevolent newcomers.

6. An abuse of a good achieved reputation should be avoided. Otherwise an entity could reach a high level of reputation and then misbehave for a long time. This prevention is shown in Fig. 7, where an entity repeatedly oscillates its behavior.
A quick, accurate and effective detection of a repeated misbehavior by a malicious entity who reached a high reputation should be carried out by a system making use of a trust and/or reputation model. To do so, an implementation of a transaction history or "memory" might be effectively developed.
In fact, PTM (Section 2.3.2) maintains a historical behavior H_{acc} used in order to calculate the posterior probabilities needed to estimate the trustworthiness of a user in the system. It is done by the application of the Bayes' theorem.
7. Newcomers should not have more opportunities than non malicious remaining nodes in the network. Otherwise an entity could achieve enough reputation to interact with other ones, then keep cheating until its level of reputation did not allow him to interact again (at least

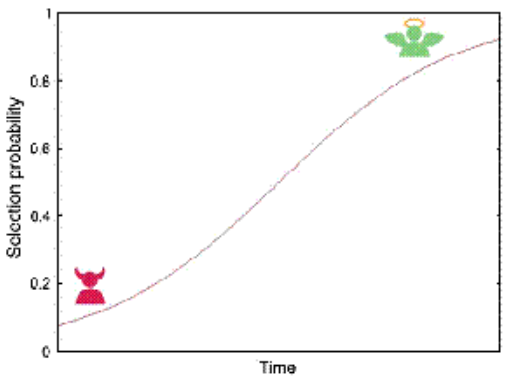


Fig. 5. Redemption of malicious entities.

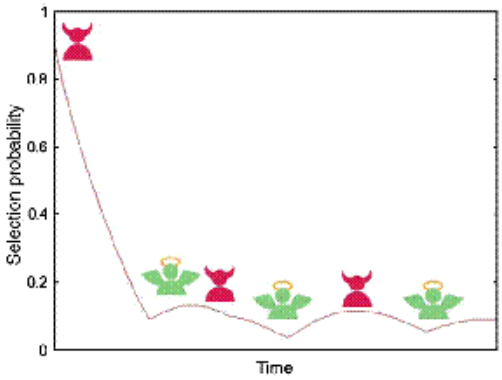


Fig. 7. Exploitation of good reputation prevention.

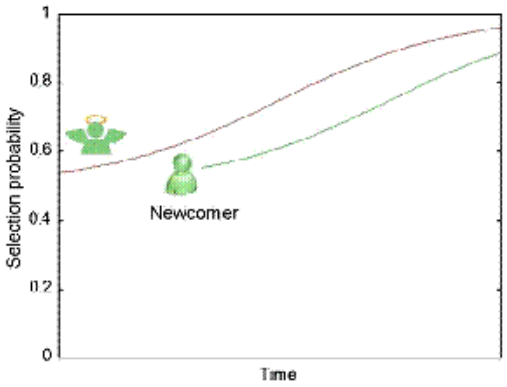


Fig. 8. Newcomers vs previously belonging entities.

for awhile), and then leave and re-enter the network as a newcomer and start again. Fig. 8 depicts this issue.
Therefore a newcomer should not be able to be selected as service provider immediately if a proved trustworthy entity already belongs to the network.
That is the case, for instance, of Sporas (Section 2.1.1), where newcomers are given the minimum reputation value, so they have to build up such reputation bit by bit.

8. Every entity should receive a different trust and/or reputation rating depending on the type of service it is providing. Otherwise an entity could gain a high level of trust and/or reputation because of its benevolent behavior providing a certain service but, at the same time, use that position to misbehave when serving another different goods or tasks. Fig. 9 depicts this scenario.
The only model from the analyzed ones in Section 2 which allows this differentiation is Regret, since in that approach each impression ι of an agent is related to one specific subject φ.
9. Bandwidth (as well as energy consumption) is a very critical issue in certain networks, such as Wireless Sensor Networks. Therefore, trust and/or reputation models aimed to work in this kind of environments ought to take care about the overhead they introduce in the system. ATRM (Section 2.4.2) is highly aware of this problem and obtains a good accuracy while preserving a low rate of massages besides a small delay, incurring thus in a minimal overhead in the network.
10. The importance of a transaction or its associated risk should influence in the subsequent punishment or reward accordingly. The more relevant a transaction is, the higher it should be punished if it is not properly carried out, and vice versa, as it happens with DWTrust (Section 2.2.1) when using λ_4.

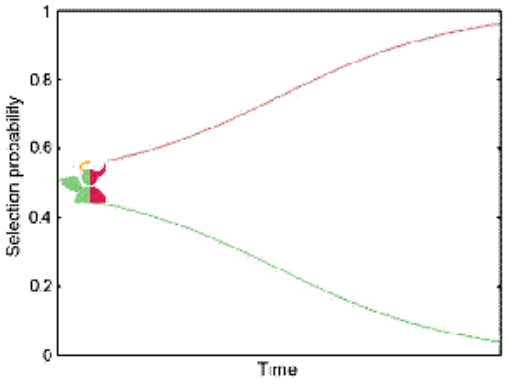


Fig. 9. Different ratings for different services.

4.5. Models fulfillment of pre-standardization recommendations

Every model described in Section 2 fulfills some of the recommendations given in the previous section, but none of them fulfills all together. Next we will outline some of those recommendations which are indeed applied in the aforementioned trust and/or reputation mechanisms.

Thus, for instance, Sporas [44] assigns new users a minimum reputation value, and they build up reputation during their activity on the system. Furthermore, the reputation value of a user never falls below the reputation of a new user, no matter how unreliable it is.

Regret [30] model gives a higher weight to those transactions which are more recent (using a time dependent function $\rho(r, r_i)$) and allows subjectivity when evaluating a transaction, but it does not permit anonymity.

Agents applying AFRAS [7] might be more or less suspicious depending on the cost or relevance of their decisions. Newcomers are given the opportunity to interact with present agents in the system even if they know each other and they are actually benevolent.

MTrust [33] has been proved to be resilient against strategic raters and oscillating malicious agents. The former consists of a single or a collusive group of raters who strategically provide a set of unfair feedbacks aiming to destroy an agent's reputation or to boost its partners' one. The latter is an agent fluctuating its performance by cooperating or defecting its partners unevenly in an acceptable range, so that it can still engage itself in future interactions.

DWTrust [18], with its dynamic weights can manage many issues like redemption of past malicious nodes or prevention of a good built exploitation. However, it does not take into account the anonymity or the introduced overhead.

Both AntRep [41] and QDV [10] use an ant colony optimization in order to manage reputation and trust, so both benefit from its advantages. For instance, both allow the redemption of past malicious entities, since pheromone traces increase as ants travel trough the network. On the other hand, the pheromone evaporation implies that i) the benevolent newcomers have the chance to be selected as service providers and ii) an exploitation of good built reputation is avoided.

Anonymity is one of the main design considerations of EigenTrust [20], and it also manages the overhead introduced in the network, nevertheless every transaction has the same weight.

RRS [6] considers the question of identity as central in reputation systems. This model enforces redemption of nodes over time by the combination of two mechanisms: periodic re-evaluation and reputation fading.

Authors of PTM [3], counting with a transactions history which, together with a posteriori probabilities computation through Bayes' theorem, make an exploitation of good reputation unfeasible.

Finally, in ATRM [4] the reduced energy consumption is one of its main features since it is focused on WSN where the devices have high constraints in memory and computation; however, only the very last transactions are taken into account.

Table 2 summarizes the fulfillment of the previous recommendations by some of the most representative trust and/or reputation models in distributed systems. As it can be observed, managing anonymous identities, dealing with the overhead introduced by each model, punishing and rewarding according to the importance of a transaction and, especially, assigning different scores to an entity depending on the service it is providing, are the recommendations which less models followed.

5. Conclusions and future work

In this paper we have described several trust and/or reputation models found in the literature. We have also given some classification

Table 2
Fulfillment of pre-standardization recommendations.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Sporas [44]		✓		✓		✓	✓			
Regret [30]		✓	✓			✓	✓	✓		
AFRAS [7]		✓	✓	✓	✓	✓				✓
MTrust [33]			✓			✓	✓			
DWTrust [18]		✓	✓	✓	✓	✓	✓			✓
AntRep [41]		✓		✓	✓	✓	✓			
EigenTrust [20]	✓			✓	✓	✓	✓		✓	
RRS [6]	✓	✓		✓	✓	✓	✓			
PTM [3]				✓	✓	✓				
QDV [10]		✓		✓	✓	✓	✓			
ATRM [4]		✓	✓	✓	✓	✓	✓		✓	

recommendations according to the type of model (trust, reputation or hybrid), its scope (multi-agents systems, P2P networks, ad-hoc networks, Wireless Sensor Networks or a combination of these) and the technic used to compute trust and/or reputation values (fuzzy logic, Bayesian networks, bio-inspired algorithms, social networks or analytic expressions).

We have also described the components we consider a future trust and/or reputation standard model should have: gathering behavioral information, scoring and ranking entities, selecting the entity to interact with, performing a transaction with the selected entity and rewarding or punishing that entity.

We have also proposed an interface aimed to supply a common layer when designing a trust and/or reputation model for distributed environments, describing its main parameters and outcomes. This interface was implemented in TRMSim-WSN simulator [16].

A pre-standardization approach has been also proposed, giving some designing advices that we think trust and/or reputation models should include. Thus, someone interested in developing such a model, will have a starting point to begin his/her design, with possible functions or steps provided with different input and output parameters and parameter values.

As we have seen, neither all the scopes nor all the technics defined in this paper for trust and/or reputation models have received the same attention. For instance, Wireless Sensor Network and Ad-hoc Networks have less trust model works dealing with them. It equally happens with bio-inspired algorithms and social networks. Therefore, this can be a good field of research and future work.

As future work we will integrate additional trust and reputation models in TRMSim-WSN, accomplishing the generic steps described in this paper. Moreover, several security threats [15] will be included as well.

Acknowledgements

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2007. Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

References

[1] A. Abdul-Rahman, S. Hailes, Supporting Trust in Virtual Communities, Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, USA, 2000.
[2] K. Aberer, Z. Despotovic, Managing trust in a Peer-2-Peer information system, in: H. Paques, L. Liu, D. Grossman (Eds.), Proceedings of the Tenth International Conference on Information and Knowledge Management, ACM Press, 2001, pp. 310–317.
[3] F. Almenárez, A. Marín, C. Campo, C. García, PTM: a Pervasive Trust Management Model for Dynamic Open Environments, Privacy and Trust, First Workshop on Pervasive Security and Trust, Boston, USA, Aug. 2004.
[4] A. Boukerche, L. Xu, K. El-Khatib, Trust-based security for wireless ad hoc and sensor networks, Computer Communications 30 (11–12) (2007) 2413–2427.
[5] S. Buchegger, J.-Y.L. Boudec, Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), IEEE, Lausanne, CH, Jun 2002.
[6] S. Buchegger, J.-Y. Le Boudec, A Robust Reputation System for P2P and Mobile Ad-hoc Networks, Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems, Cambridge MA, USA, Jun 2004.
[7] J. Carbó, J. Molina, J. Dávila, Trust management through fuzzy reputation, International Journal of Cooperative Information Systems 12 (Mar 2003) 135–155.
[8] H. Chen, H. Wu, X. Zhou, C. Gao, Agent-based trust model in Wireless Sensor Networks, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 03, 2007, pp. 119–124.
[9] O. Cordón, F. Herrera, T. Stützle, A review on the ant colony optimization metaheuristic: basis, models and new trends, Mathware and Soft Computing 9 (2–3) (2002) 141–175.
[10] S.K. Dhurandher, S. Misra, M.S. Obaidat, N. Gupta, An ant colony optimization approach for reputation and quality-of-service-based security in wireless sensor networks, Security and Communication Networks 2 (2) (2009) 215–224.
[11] J.R. Douceur, J.S. Donath, The sybil attack, Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), 2002, pp. 251–260.
[12] M.C. Fernandez-Gago, R. Roman, J. Lopez, A survey on the applicability of trust management systems for wireless sensor networks, International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007, pp. 25–30.
[13] D. Gambetta, Can we trust trust? in: D. Gambetta (Ed.), Trust: Making and Breaking Cooperative Relations, 2000, pp. 213–237, Published Online, Ch. 13.
[14] S. Ganeriwal, M.B. Srivastava, Reputation-based framework for high integrity sensor networks, SASN '04: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, New York, NY, USA, 2004, pp. 66–77.
[15] F. Gómez Mármol, G. Martínez Pérez, Security threats scenarios in trust and reputation models for distributed systems, Elsevier Computers & Security 28 (7) (2009) 545–556.
[16] F. Gómez Mármol, G. Martínez Pérez, TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks, Proceedings of the IEEE International Conference on Communications, Communication and Information Systems Security Symposium, Dresden, Germany, Jun 2009.
[17] F. Gómez Mármol, G. Martínez Pérez, Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique, Telecommunication Systems Journal 46 (2) (2010).
[18] C. Huang, H. Hu, Z. Wang, A dynamic trust model based on feedback control mechanism for P2P applications, Autonomic and Trusted Computing, No. 4158 in LNCS, Springer, Wuhan, China, Sep. 2006, pp. 312–321.
[19] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems 43 (2) (2007) 618–644.
[20] S. Kamvar, M. Schlosser, H. Garcia-Molina, The EigenTrust Algorithm for Reputation Management in P2P Networks, Proc. of the International World Wide Web Conference (WWW), Budapest, Hungary, May 2003.
[21] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
[22] T.K. Kim, H.S. Seo, A trust model using fuzzy logic in wireless sensor network, Proceedings of World Academy of Science, Engineering and Technology, vol. 32, Aug. 2008, pp. 69–72.
[23] Marsh, S.P., Apr. 1994. Formalising trust as a computational concept. Ph.D. thesis, Department of Computing Science and Mathematics, University of Stirling.
[24] S. Marti, H. Garcia-Molina, Taxonomy of trust: categorizing P2P reputation systems, Computer Networks 50 (4) (Mar. 2006) 472–484.
[25] Muchow, J.W., 2002. Core J2ME Technology and MIDP. Sun Microsystems Press Java series. Prentice Hall PTR.
[26] L. Mui, M. Mohtashemi, A. Halberstadt, A Computational Model of Trust and Reputation, Proceedings of the 35th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Washington, DC, USA, 2002, pp. 188–196.
[27] M. Omar, Y. Challal, A. Bouabdallah, Reliable and fully distributed trust model for mobile ad hoc networks, Computers and Security 28 (3–4) (2009) 199–214.
[28] S. Ramchurn, C. Sierra, L. Godo, N. Jennings, A computational trust model for multi-agent interactions based on confidence and reputation, 6th International Workshop of Deception, Fraud and Trust in Agent Societies, 2003, pp. 69–75.
[29] R. Roman, M.C. Fernandez-Gago, J. Lopez, Featuring trust and reputation management systems for constrained hardware devices, Autonomics '07: Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems, Rome, Italy, 2007, pp. 1–6.
[30] J. Sabater, C. Sierra, REGRET: reputation in gregarious societies, in: J.P. Müller, E. Andre, S. Sen, C. Frasson (Eds.), Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press, Montreal, Canada, 2001, pp. 194–195.
[31] J. Sabater-Mir, C. Sierra, Social ReGreT, a reputation model based on social relations, SIGecom Exch 3 (1) (2002) 44–56.
[32] A. Singh, L. Liu, TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems, IEEE International Conference on Peer-to-Peer Computing, 2003, pp. 142–149.
[33] S. Songsiri, MTrust: a reputation-based trust model for a mobile agent system, Autonomic and Trusted Computing, No. 4158 in LNCS, Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 374–385.
[34] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, M. Cardei, On Trust Establishment in Mobile Ad-Hoc Networks, Reputation and Trust-based Systems for Ad Hoc and Sensor Networks, Wiley & Sons, 2007.
[35] A. Srinivasan, J. Teitelbaum, J. Wu, DRBTS: Distributed Reputation-based Beacon Trust System, DASC '06: Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, IEEE Computer Society, Washington, DC, USA, 2006, pp. 277–283.
[36] W. Stallings, Data and Computer Communications, 7th ed. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[37] Y. Sun, Z. Han, K. Liu, Defense of trust management vulnerabilities in distributed networks, IEEE Communications Magazine 46 (2) (Feb. 2008) 112–119.
[38] A. Tajeddine, A. Kayssi, A. Chehab, H. Artail, PATROL-F — a comprehensive reputation-based trust model with fuzzy subsystems, Autonomic and Trusted Computing.No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 205–217.
[39] Y. Tang, L. Sun, J. Luo, S. Yang, Y. Zhong, TFCD: a trust-based incentive mechanism for peer-to-peer live broadcasting, Autonomic and Trusted Computing. No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 332–341.
[40] H. Tian, S. Zou, W. Wang, S. Cheng, A group based reputation system for P2P networks, Autonomic and Trusted Computing.No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 342–351.
[41] W. Wang, G. Zeng, L. Yuan, Ant-based reputation evidence distribution in P2P networks, CCC. Fifth International Conference on Grid and Cooperative Computing, IEEE Computer Society, Changsha, Hunan, China, Oct. 2006, pp. 129–132.
[42] Y. Wang, V. Cahill, E. Gray, C. Harris, L. Liao, Bayesian network based trust management, Autonomic and Trusted Computing.No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 246–257.
[43] F. Yu, H. Zhang, P. Yan, S. Cao, An improved global trust value computing method in P2P system, Autonomic and Trusted Computing. No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 238–267.
[44] G. Zacharia, P. Maes, Trust management through reputation mechanisms, Applied Artificial Intelligence 14 (2000) 881–907.
[45] Y. Zhang, H. Chen, Z. Wu, A social network-based trust model for the semantic web, Autonomic and Trusted Computing. No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 183–192.
[46] R. Zhou, K. Hwang, PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing, Transactions on Parallel and Distributed Systems, 2007.
[47] T. Zhuo, L. Zhengding, L. Kai, Time-based dynamic trust model using ant colony algorithm, Wuhan University Journal of Natural Sciences 11 (6) (2006) 1462–1466.

Félix Gómez Mármol is a PhD student in the Department of Information and Communications Engineering of the University of Murcia. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received an MSc in computer engineering from the University of Murcia. Contact him at felixgm@um.es (http://ants.dif.um.es/felixgm).

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security and management of distributed communication networks. He received an MSc and PhD in computer engineering from the University of Murcia. Contact him at gregorio@um.es (http://webs.um.es/gregorio).

6

Security Threats Scenarios in Trust and Reputation Models for Distributed Systems

Title:	Security Threats Scenarios in Trust and Reputation Models for Distributed Systems
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	Journal
Journal:	Computers & Security
Impact factor (2008):	1.028 (position 56/99)
Publisher:	Elsevier
Volume:	28
Number:	7
Pages:	545-556
Year:	2009
Month:	October
DOI:	http://dx.doi.org/10.1016/j.cose.2009.05.005
State:	Published

Table 6: Security Threats Scenarios in Trust and Reputation Models for Distributed Systems

COMPUTERS & SECURITY 28 (2009) 545–556

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Security threats scenarios in trust and reputation models for distributed systems

*Félix Gómez Mármol**, *Gregorio Martínez Pérez*

Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30.071 Murcia, Spain

ARTICLE INFO

Article history:
Received 17 December 2008
Received in revised form
3 April 2009
Accepted 1 May 2009

Keywords:
Trust
Reputation
Security threats
Distributed environments
Threats taxonomy

ABSTRACT

Trust and reputation management over distributed systems has been proposed in the last few years as a novel and accurate way of dealing with some security deficiencies which are inherent to those environments. Thus, many models and theories have been developed in order to effective and accurately manage trust and reputation in those communities. Nevertheless, very few of them take into consideration all the possible security threats that can compromise the system. In this paper, we present some of the most important and critical security threats that could be applied in a trust and reputation scheme. We will describe and analyze each of those threats and propose some recommendations to face them when developing a new trust and reputation mechanism. We will also study how some trust and reputation models solve them. This work expects to be a reference guide when designing secure trust and reputation models.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Trust and reputation models have been recently proposed by many researches as an innovative solution for guaranteeing a minimum level of security between two entities belonging to a distributed system that want to have a transaction or interaction.

Thus, many studies, works and models have been designed, carried out and developed in this direction, leading to a current solid research field on which both academia and industry are focusing their attention.

Many methods, technologies and mechanisms like fuzzy logic (Tajeddine et al., 2006), bayesian networks (Wang et al., 2006b) or even bio inspired algorithms (Gómez Mármol, 2008) have been proposed in order to manage and model trust and reputation in systems such as P2P networks (Almenárez et al., 2004), ad hoc ones (Moloney and Weber, 2005), wireless sensor networks (Boukerche et al., 2007) (WSN) or even multi agent systems (Sabater and Sierra, 2001).

Analyzing and studying some of these models (Josang et al., 2007; Sabater and Sierra, 2005) we realized that there are some security threats directly related to this specific kind of models, which are common and applicable to most of these approaches.

Nevertheless, we also noticed that each author proposed his/her own threats when testing their developed models, revealing the lack of a commonly agreed process of checking the robustness of a trust and reputation model against the mentioned risks.

This paper presents the most important security threat scenarios that can be found in the area of trust and reputation in a distributed system where some entities request some services and other ones provide those services. As far as we know, this is one of the first research works making such a thorough analysis.

We will describe each threat and propose a possible solu tion for tackling it. We will additionally study how some of the most representative models deal with those threats and analyze their proposed solutions.

* Corresponding author. Tel.: +34 868 887866.
E-mail addresses: felixgm@um.es (F.G. Mármol), gregorio@um.es (G.M. Pérez).
0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.05.005

Every accurate and robust trust and reputation model should have some mechanisms to effectively overcome all the threats that could be applied to it. Therefore, this work is intended to serve as a reference guide for developing secure trust and reputation models.

The rest of the paper is organized as follows: Section 2 makes a brief description of trust and reputation management and discusses the importance of dealing with the security threats exposed in Section 3, where some solutions are also proposed. A taxonomy of these threats is described in Section 4 and how some trust and reputation models face them is shown in Section 5. Finally, some conclusions and future work are depicted in Section 6.

2. Trust and reputation management

Trust and reputation management has recently become a very useful and powerful tool in some specific environments where a lack of previous knowledge about the system can lead participants to undesired situations, specifically in virtual communities where users do not know each other at all or, at least, do not know everyone.

It is in those cases where the application of trust and reputation mechanisms is more effective, helping a peer to find out which is the most trustworthy or reputable participant to have an interaction with, preventing thus the selection of a fraudulent or malicious one.

We have noticed that most of the current trust and reputation models in the literature follow these four general steps (Marti and Garcia Molina, 2006) (as shown in Fig. 1):

1. Collecting information about a certain participant in the community by asking other users their opinions or recommendations about that peer.
2. Aggregating all the received information properly and somehow computing a score for every peer in the network.
3. Selecting the most trustworthy or reputable entity in the community providing a certain service and effectively having an interaction with it, assessing a posteriori the satisfaction of the user with the received service.
4. According to the satisfaction obtained, a last step of punishing or rewarding is carried out, adjusting consequently the global trust (or reputation) deposited in the selected service provider.

Additionally, each model manages concepts such as trust or reputation in many different ways. For instance, some models like PTM (Almenárez et al., 2004; Almenárez et al., 2006) or AFRAS (Carbó et al., 2003) make use of fuzzy logic in order to deal with those topics.

On the other hand, bayesian networks are used by authors of MTrust (Songsiri, 2006) and BNBTM (Wang et al., 2006b). And even bio inspired algorithms are used in AntRep (Wang et al., 2006a) or TACS (Gómez Mármol et al., 2008). Other models like Eigentrust (Kamvar et al., 2003) or PeerTrust (Xiong and Liu, 2004) just give some analytic expressions.

However, we also realized that not all the models address all the possible threats that could be found and applied in those scenarios. In fact, some of them do not even deal with these risks at all.

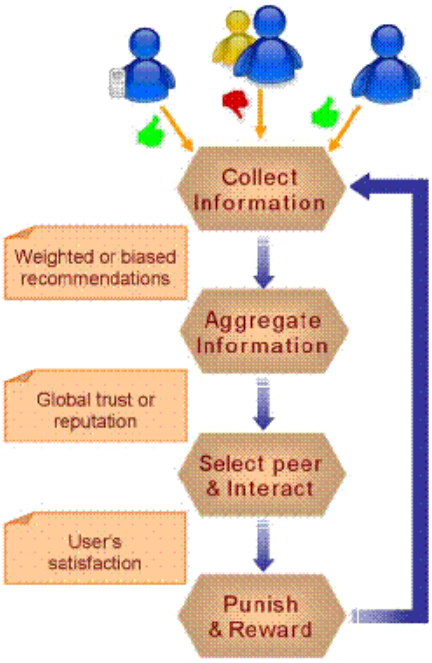


Fig. 1 – Trust and reputation models' steps.

In our opinion, this is an issue that should not be underestimated when designing and developing a new trust and reputation model over distributed and heterogeneous systems, since an inaccurate management of these threats could result in important security deficiencies and weaknesses.

It is also worth mentioning that the distinction between a trust and a reputation model is not always clear. However, in our opinion, those models making an explicit use of other participants' recommendations could be categorized as reputation models while the rest could be considered just as trust models.

Finally, some scenarios where a trust and reputation model may prove useful could be, among many others, a P2P file sharing system, an ad hoc routing protocol or a streaming service in case of accident over a WSN.

3. Security threats

In this section we will present and describe the most common security threats applicable in the field of trust and reputation management over distributed environments. Moreover, an approach aimed to tackle and solve each of those threats will be also proposed.

It is important to note that, although all of these threats can be applied to some trust and reputation models, not all of them can be applied to any model, since some threats are specific of one or another type of trust and reputation model.

Without loss of generality we will consider a scenario where several participants (entities, nodes, peers, agents, users, ...) belong to a virtual community (P2P network, WSN, ad hoc network, multi agent system, ...) where a certain set of services is offered.

When a specific participant is requested to provide one of the services it offers it can effectively provide the offered service and act, therefore, in a benevolent way or, on the other hand, it can provide a worse service, acting thus fraudulent or maliciously.

3.1. *Individual malicious peers*

Description. Malicious peers always provide bad services when selected as service providers (Fig. 2).

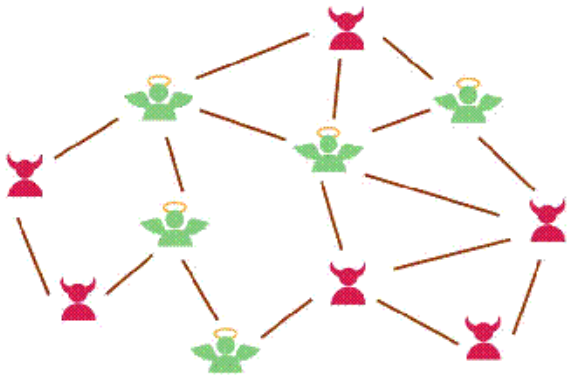


Fig. 2 – Individual malicious peers.

Discussion. This is the simplest threat that can be found in a trust and reputation system. Every trust and reputation model deals with this kind of attack.

Solution. The way of preventing such a misbehavior is by decreasing the level of trust or reputation of those participants who always provide bad services, categorizing them, therefore, as malicious peers.

3.2. *Malicious collectives*

Description. Malicious peers always provide bad services when selected as service providers. Malicious peers form a malicious collective by assigning the maximum trust value to other malicious peers in the network (Fig. 3).

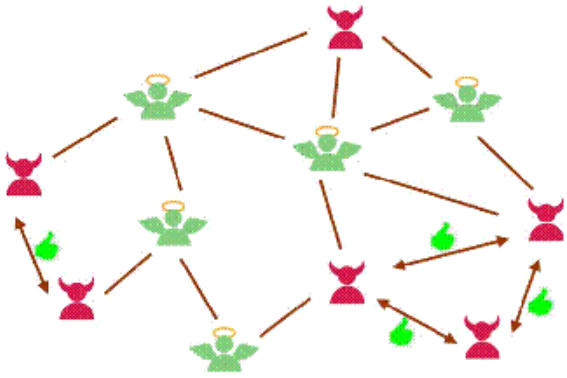


Fig. 3 – Malicious collectives.

Discussion. Not many trust and reputation models treat the problem arisen from the constitution of a collusion among malicious peers, having thus an important security deficiency.

Solution. The first thing needed to be able to overcome this threat is to somehow manage, not only the goodness of every user when supplying services, but also their reliability when giving recommendations about other peers. Thus, a user who provides unfair ratings will be also discarded as a service provider.

3.3. *Malicious collectives with camouflage*

Description. Malicious peers provide bad services in p% of all cases when selected as service providers. Malicious peers form a malicious collective by assigning the maximum trust value to other malicious peers in the network (Fig. 4).

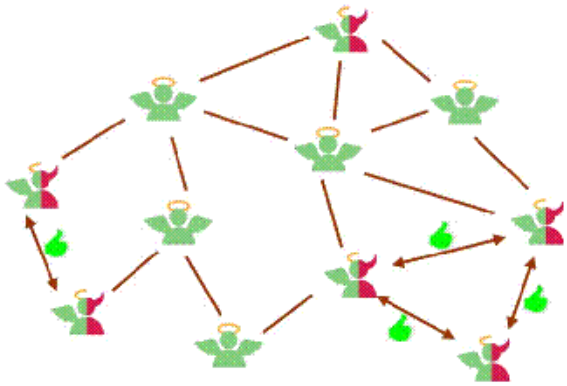


Fig. 4 – Malicious collectives with camouflage.

Discussion. This is, in many cases, a threat which is not always easy to tackle, since its resilience will mostly depend on the behavioral pattern followed by malicious peers. That is, it is not equal to battle against an oscillating pattern (being fully benevolent for a period of time, and fully fraudulent for the next period, and so on, as shown in Fig. 5(a)), for instance, than against an increasing and decreasing one (Fig. 5(b)), or even a random pattern (Fig. 5(c)).

Furthermore, the variable behavior is not even considered as a threat in many models in the sense that they do not punish that kind of behavior, but they just try to adjust the trust and reputation given to a peer to its real and current goodness. Other models (Kamvar et al., 2003), however, demonstrate the uselessness for malicious peers to behave in this way.

Solution. The first topic to address is to somehow distinguish the confidence deposited in a peer as a recommender and the trust deposited in the same peer as a service provider. This mechanism can be very helpful when trying to avoid unfair ratings from malicious entities. Additionally, the variable behavior of a peer, when detected, could be punished and avoided.

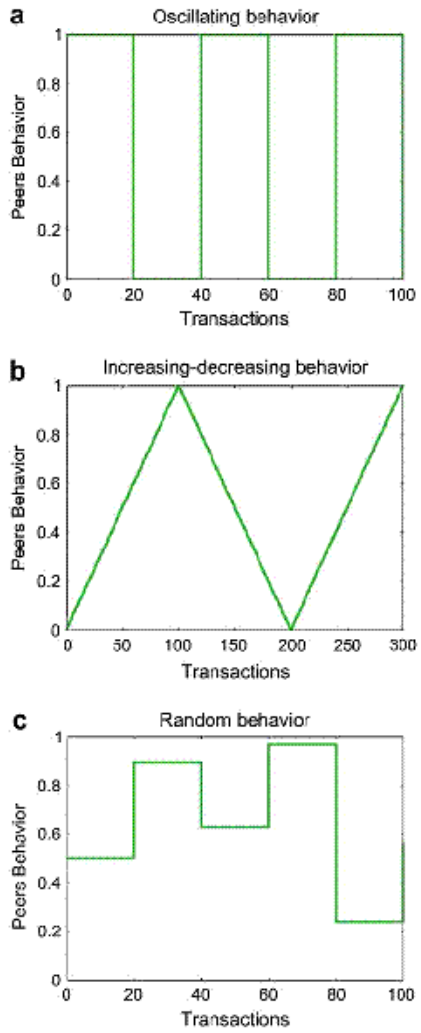


Fig. 5 – Variable behavior.

3.4. *Malicious spies*

Description. Some malicious peers always provide bad services when selected as service providers. Those malicious peers form a malicious collective by assigning the maximum trust value to other malicious peers in the network. Other distinct malicious peers, known as malicious spies, always provide good services when selected as service providers, but they also give the maximum rating values to those malicious peers who always provide bad services (Fig. 6).

Discussion. In this threat, the malicious spies may gain a high level of trust and reputation, since they always provide good services, being able then to easily subvert the trust and reputation mechanism applied in the system. Most of the times, this kind of attack has not a trivial or easy way of being effectively tackled.

Solution. Like in previous threats, an accurate management of the reliability of the peers, not only as service providers, but also as recommendation providers may effectively help to prevent this kind of abuse, although it will probably take longer (more effort and more resources needed, therefore) in order to be able to identify both the malicious peers and the malicious spies.

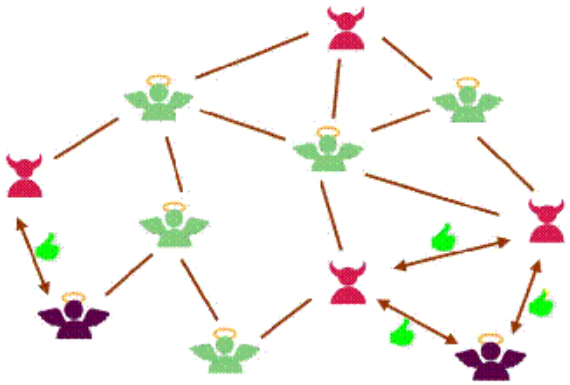


Fig. 6 – Malicious spies.

3.5. *Sybil attack*

Description. An adversary initiates a disproportionate number of malicious peers in the network. Each time one of the peers is selected as a service provider, it provides a bad service, after which it is disconnected and replaced with a new peer identity (Fig. 7) (Douceur and Donath, 2002).

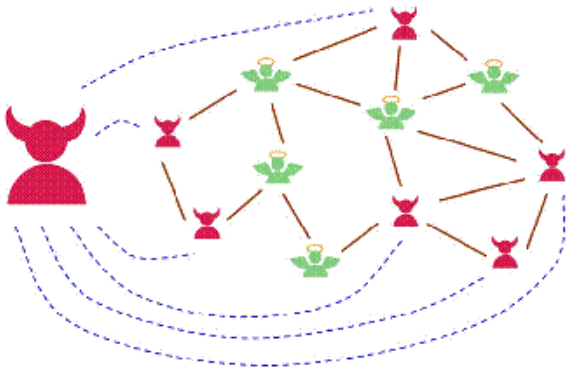


Fig. 7 – Sybil attack.

Discussion. This kind of attack might prove quite problematic since it could prevent good peers from being able to gain a good reputation, since they might not be selected most of the times.

Again, not many trust and reputation models deal with such an important and potentially dangerous threat like the Sybil attack leading thus to an underestimated but great risk.

Solution. One of the most common solutions proposed in the literature for this kind of threat consists of associating a cost to the generation of new identities in the community.

This cost is not necessarily economic, but it can also be a cost in terms of time or resources, for instance.

Another suggested way of dealing with this problem (Girao et al., 2006) makes use of a central entity managing (virtual) identities in the system, or even a set of identity providers ensuring that every participant in the community has a unique and immutable identity.

3.6. *Man in the middle attack*

Description. A malicious peer can intercept the messages from a benevolent service provider peer to the requestor and rewrite them with bad services, making therefore the reputation of the benevolent peer to decrease. That participant could even maliciously modify the recommendations given by an honest peer, in order to benefit his/her own interests (Fig. 8).

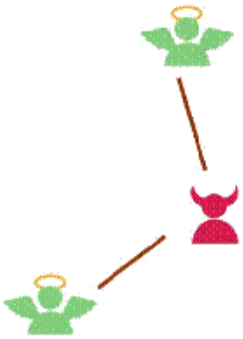


Fig. 8 – Man in the middle attack.

Discussion. One more time, this is a threat which has not been associated with trust and reputation systems traditionally. Most of the authors consider or assume the authenticity of the peer providing either a service or a recommendation. Nevertheless, as explained before, this attack can cause a great damage and effect in the system if its application is possible.

Solution. A simple way of avoiding this risk could be by the use of cryptography schemes in order to authenticate each user in the system (maybe with a digital signature or any similar mechanism). However, and unfortunately, it is not always feasible to apply such a solution, above all in highly distributed environments like wireless sensor networks.

3.7. *Driving down the reputation of a reliable peer*

Description. Malicious peers always provide bad services when selected as service providers. Malicious peers form a malicious collective by assigning the maximum trust value to other malicious peers in the network. Additionally, they give the worst rating to those benevolent peers, who indeed provide good services (Fig. 9).

Discussion. This kind of attack can be even worse than the ones named malicious collectives and malicious spies, since in this case benevolent peers also receive unfair critics from

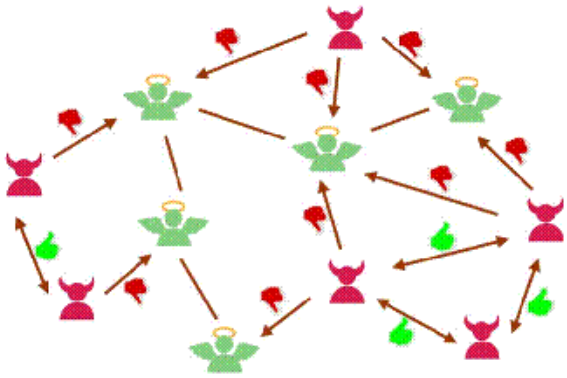


Fig. 9 – Driving down the reputation of a reliable peer.

malicious pees. In such a situation if an interaction with a malicious peer is carried out it can be identified as malicious, but if an interaction has never been performed with a peer which is actually benevolent but whose reputation has been driven down by malicious participants, then that peer will not probably be chosen as the peer to have an interaction with.

Solution. The differentiated management of the trust given to a participant when supplying services and the reliability of his/her recommendations can be very useful in this scenario as well. However, there are some trust and reputation models (Gómez Mármol, 2008) where this distinction is not explicitly done but, due to their dependency on the topology of the network, are able to find the most trustworthy path leading to the most reputable peer offering a certain service.

3.8. *Partially malicious collectives*

Description. Malicious peers always provide bad services when selected as certain service providers. However, they always provide good services when selected as other different service providers (Fig. 10).

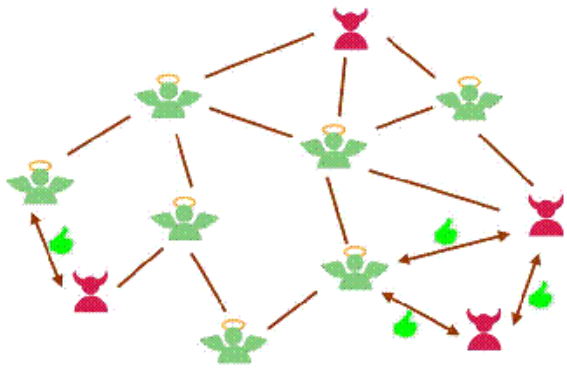


Fig. 10 – Partially malicious collectives.

That is, for certain services they behave properly, while for other specific services, they act maliciously. Malicious peers

form a malicious collective by assigning the maximum trust value to other malicious peers in the network.

Discussion. There are some trust and reputation models which are not resilient to this kind of attack since they just perform a global computation of the trust and/or reputation of a peer, regardless the service they are providing. In such a situation some distortion can emerge, considering a peer as fully or quite benevolent (malicious) although it can also provide some fraudulent (good) services.

Solution. By just considering a different score for every service offered by a peer, this threat is mitigated most of the times. However, it is not always possible to make this distinction since in some environments (for instance, those with a great amount of services offered) it could lead to some scalability problems.

3.9. *Malicious pre-trusted peers*

Description. Some or all the pre trusted benevolent peers become malicious ones, maybe by always providing bad services when selected as service providers or by rating with maximum trust value other malicious peers who always provide bad services when selected as service providers.

Discussion. First it is worth mentioning that it is not always feasible to find a set of peers that can be trusted before any transaction is carried out in the system (Fig. 11).

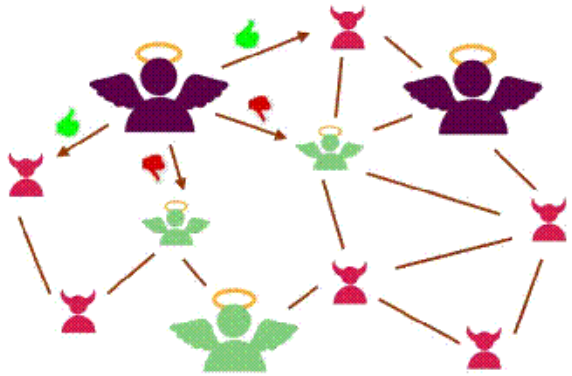


Fig. 11 – Malicious pre-trusted peers.

Some models (like Eigentrust (Kamvar et al., 2003), for instance) base their strategy on this kind of participants. However, and maybe in a paranoic way of thinking, every user in a virtual community can behave inappropriately at some point. If such a thing occurred with a pre trusted peer, those models mentioned before would be in a risk.

Solution. Our suggestion for such situations would be to be able to decide at any time which peers belong to the set of pre trusted ones, depending on their behavior.

4. Security threats taxonomy

This section will describe several properties or dimensions related to a generic security threat for trust and reputation systems (Lam and Riedl, 2004). These dimensions will help us to create a taxonomy of the previously exposed threats, analyzing and categorizing each one of them according to these properties. In fact, a summary table (Table 1) has been designed and included showing that classification.

- *Attack intent*

An adversary may have several different goals when trying to subvert a trust and reputation system. Two straightforward intents are to fraudulently praise an entity in order to increase her reputation in the system and, conversely, to drive down the reputation of a reliable entity.

A third possible goal could be just to damage the reputation system as a whole, so users may decrease their trust in it and, eventually, stop using it.

Thus for instance, malicious collectives, collectives with camouflage and malicious spies attacks will try to unfairly praise and increase the reputation of some entities which actually do not deserve it. The rest of threats will just try to subvert the whole system in one or another way.

- *Targets*

Some security threats focus their efforts on a subset of users or entities belonging to the system, whereas other threats center on specific individual users. There are even some threats which do not distinguish and are applicable to the whole community.

In this sense, individual malicious peers and man in the middle attacks can be classified as individual attacks, while driving down the reputation of a reliable peer affects all the members of the community. Other threats' targets are composed by a subset of the entities belonging to the system.

- *Required knowledge*

The amount of information needed to be gathered or collected from the system in order to effectively perform an attack is another important issue in these scenarios. Thus, some threats will require a comprehensive knowledge about the whole system or about some particular entities, while some other threats will work properly with a small knowledge about the trust and reputation system (its users, the trust and reputation model applied, ratings distribution, etc.).

Regarding this point, creating a collusion, for instance, will need more information about the system (each member of the collusion needs to know the rest of them) than an individual attack such as individual malicious peers or Sybil attack. If they also need to know, for example, the goodness of each member for every given provided service, then the amount of required knowledge in order to perform the attack is higher.

- *Cost*

The less expensive an attack is, the more beneficial is its application. Once again, the cost of running an attack is not necessarily economic, but it can be also measured in terms of resources or time requirements, for instance.

Table 1 – Security threats taxonomy.

Security threats	Attacks dimensions					
	Attack intent	Target	Required knowledge	Cost	Algorithm dependence	Detectability
Individual malicious peers	Whole	Individual	Low	Low	Generic	High
Malicious collectives	Praise	Subset	Medium	Medium	Generic	Medium
Malicious collectives with camouflage	Praise	Subset	Medium	Medium	Generic	Low
Malicious spies	Praise	Subset	High	High	Generic	Low
Sybil attack	Whole	Subset	Low	Medium	Generic	Low
Man in the middle attack	Whole	Individual	Medium	Medium	Generic	Medium
Driving down the reputation of a reliable peer	Whole	All	High	High	Generic	Low
Partially malicious collectives	Whole	Subset	High	High	Generic	Low
Malicious pre-trusted peers	Whole	Subset	High	High	Specific	Low

Thus, some threats will have a higher associated cost and will be therefore more difficult to be performed, while others will be easily applicable, since their corresponding cost will make them worthy.

As it can be observed in Table 1 the cost of applying an attack is directly related to its associated amount of required knowledge. The only case where both dimensions do not match is for the Sybil attack, because although it needs (nearly) no knowledge about the system, it is not usually so easy to create a disproportionate number of entities enough to cause a really important damage to the community.

- *Algorithm dependence*

Some security threats take advantage of a specific trust and reputation algorithm or model vulnerability and exploits it in order to create a great damage to the system. On the other hand, other attacks are more generic and, consequently, applicable in a wider set of scenarios or environments.

Most of the described security threats for trust and reputation system could be applied in almost any scenario or environment. Malicious pre trusted peers, however, is an specific attack related and, therefore, only applicable to those trust and reputation algorithms or models which actually make use of pre trusted peers, as we will see later in the case of EigenTrust (Kamvar et al., 2003).

- *Detectability*

Finally, an attack over trust and reputation systems is desired to be as less detectable as possible. Later an attack is detected, the higher might be the damage caused. That is the reason why most of the threats act trying not to induce suspicion as much as possible, i.e., they do not cause drastic changes in the system, but they rather make slight ones.

In some way, the detectability of an attack or threat is a measurement of its resilience and effectiveness. Thus, the easiest threat of the previously presented ones to be detected would be the individual malicious peers. As the collaboration between attackers and their gathered knowledge about the system increases, those attacks become more and more undetectable. That is the reason why all the threats based on a collusion are, generally, more difficult to tackle.

5. Dealing with main security threats in major trust and reputation models

This section will present some of the most representative trust and reputation models for distributed systems and will show how each of them face the threats exposed in Section 3.

Some experimental results taken from the reference papers highlight how each model is reacting against certain attacks they are covering.

5.1. *EigenTrust*

5.1.1. *Brief introduction*

The first trust model we will describe is called EigenTrust (Kamvar et al., 2003), and it is one of the most known and cited ones in this field. It is characterized by the assignment of a unique global trust value to each peer in a P2P file sharing system, based on the peer's history of contributions.

Thus, authors define s_{ij} as the local trust of peer i about peer j, in the following way:

$$s_{ij} = \mathrm{sat}(i,j) - \mathrm{unsat}(i,j)$$

i.e., the difference between the satisfactory and unsatisfactory interactions of peers i and j. Moreover, they also define a normalized local trust value $c_{ij} \in [0, 1]$ as:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

Peer i's global reputation is given by the local trust values given to it by other peers, weighted by the global reputation of the assigning peers. Let C be the matrix $[c_{ij}]$ and $\vec{c}_i$ a vector defined as follows:

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1j} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2j} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ c_{i1} & c_{i2} & \cdots & c_{ij} & \cdots & c_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nj} & \cdots & c_{nn} \end{pmatrix}, \quad \vec{c}_i = \begin{pmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{ij} \\ \vdots \\ c_{in} \end{pmatrix}$$

Having this, t_{ik} represents the trust that peer i places in peer k based on asking his friends, and defined as:

$$\vec{t}_i = C^T \vec{c}_i = \left(\sum_{j=1}^{n} c_{ij} c_{j1}, \ldots, \sum_{j=1}^{n} c_{ij} c_{jk}, \ldots, \sum_{j=1}^{n} c_{ij} c_{jn} \right)$$

By querying his friends' friends, peer i gets a wider view of peer's k reputation, that is:

$$\vec{t}_i = (C^T)^2 \vec{c}_i$$

Going on in this way, after a large enough number m of queries, peer i will get the same eigenvector $\vec{t}_i = (C^T)^m \vec{c}_i$, as every other peer in the system.

Additionally, authors propose more sophisticated ways of computing this eigenvector based on pre trusted peers. They also consider that a peer who is honest providing a service (in their case sharing a file) is also likely to be honest in reporting its local trust values, which, as we have seen before, has not to be necessarily always like this.

5.1.2. Security analysis

Regarding the security threats this model covers, when a set of individual malicious peers is present in the system, those peers receive high local trust values only from other malicious peers, since they are the only ones who value the supply of malicious services (i.e., they compute $s_{ij} = \text{unsat}(i, j\}) - sat(i, j\})$. And even that only occasionally, since malicious peers have to meet each other through an interaction. Because of their low trust values, malicious peers are rarely chosen as service providers (around 10% of the times).

Forming a malicious collective does not increase the global trust values of malicious peers enough in order for them to have impact on the network due to the presence of pre trusted peers. A user will always have the opportunity to perform a transaction with one of those pre trusted peers and if an interaction is performed with a malicious peer (which occurs again around 10% of the times), it will be identified as malicious by the whole system.

However, it is worthy to mention that authors do not consider a collusion exactly in the same way we defined it previously, since in their scenario, every peer belonging to the collusion gives the maximum rate to the "next" peer in the collusion (and the minimum to everybody else), forming thus a ring or chain.

Moreover, the optimum scenario for a malicious collective with camouflage in this model consists of providing 50% of the times a fraudulent service (in that case 28% of the transactions correspond to a malicious service). Kamvar et al. (2003) demonstrate the unworthiness of such behavior for malicious peers relying on the cost those peers have in order to sometimes provide a service properly.

Finally, authors also deal with the problem of Sybil attack by imposing some kind of cost to the generation of new identities, but they also show the vulnerability of their model against malicious spies, since their opinions and recommendations will be taken into account (even when rating malicious peers) due to their proper behavior when supplying services.

5.2. PeerTrust

5.2.1. Brief introduction

PeerTrust (Xiong and Liu, 2004) is a trust and reputation model that combines several important aspects related to the management of trust and reputation in distributed systems, such as: the feedback a peer receives from other peers, the total number of transactions of a peer, the credibility of the recommendations given by a peer, the transaction context factor and the community context factor.

This accurate aggregation is performed through the following expression, representing the trust value of peer u, $T(u)$:

$$T(u) = \alpha \sum_{i=1}^{I(u)} S(u,i) CR(p(u,i)) TF(u,i) + \beta \times CF(u)$$

where $I(u)$ denotes the total number of transactions performed by peer u with all other peers, $p(u, i)$ denotes the other participating peer in peer u's ith transaction, $S(u, i)$ denotes the normalized amount of satisfaction peer u receives from $p(u, i)$ in its ith transaction, $CR(v)$ denotes the credibility of the feedback submitted by v, $TF(u, i)$ denotes the adaptive transaction context factor for peer u's ith transaction, and $CF(u)$ denotes the adaptive community context factor for peer u.

On the other hand, the credibility of v from w's point of view, is computed as:

$$Cr(p(u,i)) = \frac{Sim(p(u,i), w)}{\sum_{j=1}^{I(u)} Sim(p(u,j), w)}$$

where

$$Sim(v,w) = 1 - \sqrt{\sum_{x \in [S(v,w)} \left(\frac{\sum_{i=1}^{I(x,v)} S(x,i)}{I(x,v)} - \frac{\sum_{i=1}^{I(x,w)} S(x,i)}{I(x,w)} \right)^2 \Big/ |[S(v,w)|}$$

being $I(u, v)$ the total number of transactions performed by peer u with peer v, $[S(v)$ the set of peers that have interacted with peer v and $[S(v, w)$ the common set of peers that have interacted with both peer v and w, computed as $[S(v) \cap [S(w)$.

Additionally this model introduces a trust based peer selection scheme, according to the third step described in Section 2 and depicted in Fig. 1. A simple rule for peer w to decide whether to have an interaction with peer u or not could be $T(u) > T_{threshold}(w)$, where the value of $T_{threshold}(w)$ depends on several factors such as the importance of the transaction, or the disposition of w to trust unknown peers or not, among many others.

5.2.2. Security analysis

The accurate management of the credibility of a peer as a recommender, as well as the context factor or the community one allows PeerTrust model to effectively overcome many of the security threats described previously.

Thus, malicious individual peers, malicious collectives, malicious collectives with camouflage and driving down the reputation of a reliable peer are some of the threats that are solved by PeerTrust.

This ability to deal with those threats is due to, among other factors, the definition of credibility in terms of the

similarity between two peers, which allows the model to accurately detect and identify in the community malicious service providers as well as malicious recommenders.

Additionally it stimulates the community to supply recommendations by building incentives or rewards to those peers who provide feedbacks to others. And this is done through the context factor, with the following definition:

$$CF(u) = \frac{F(u)}{I(u)}$$

where $F(u)$ represents the total number of feedback peer u gives to others. This stimulation also helps and is very useful to avoid (almost any kind of) malicious peers to gain a high reputation in the system and therefore, to be selected many times as service providers.

When a threat of the type malicious individual peers, malicious collectives or driving down the reputation of a reliable peer occurs, PeerTrust achieves the selection of fraudulent peers to remain less than a 10% of the times, being the worst case that one where 50% of the peers are malicious.

Regarding malicious collectives with camouflage, authors test the oscillating scenario described before (Fig. 5(a)) obtaining reasonably good outcomes due to the use of a time windows based metric that discounts the old feedbacks of peers.

Finally, PeerTrust can also overcome the threats of partially malicious collectives (since it introduces a context factor to measure the importance of each transaction) and the man in the middle attack.

The latter is tackled making use of cryptographic mechanisms. Specifically, authors propose that every identity is established by a public key corresponding to a unique private key, avoiding thus the spoofing of an identity without the knowledge of such private key. Additionally, any content properly signed will not have its integrity or origin compromised.

5.3. BTRM-WSN

5.3.1. Brief introduction

BTRM WSN (Gómez Mármol, 2008) is a novel trust model for wireless sensor networks (WSN) based on the bio inspired algorithm of ant colony system (ACS, Dorigo et al., 2006). It allows to find the most trustworthy path leading to the most reputable service provider in a network. Its intrinsic nature makes it to be easily adaptable to sudden changes in the topology of the network as well as in the behavior of its participants.

In this model, a set of ants (artificial agents) is launched through the WSN. While they are searching for the most reputable service provider, they leave some pheromone traces in every link connecting two nodes. That pheromone between sensors a and b, denoted as τ_{ab}, is identified with the confidence sensor a has on finding the most trustworthy path through sensor b.

At each node, every ant has to decide which next sensor to move towards. In order to carry out this decision, a probability is given to each arc not visited yet by that ant as follows:

$$p_k(r,s) = \begin{cases} \dfrac{[\tau_{rs}]^{\alpha}[\eta_{rs}]^{\beta}}{\sum_{u \in J_k(r)}[\tau_{ru}]^{\alpha}[\eta_{ru}]^{\beta}}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases}$$

being $p_k(r,s)$ the probability of ant k to move from sensor r to s, η_{rs} the heuristic associated with the link joining r and s, identified with the distance that separate both sensors, $J_k(r)$ the set of neighbors of node r not visited yet by ant k, and α and β, two parameters to balance the pheromone and the heuristic.

Every time an ant crosses a link, it modifies its pheromone trace in the following way:

$$\tau_{s_1 s_2} = (1-\varphi)\tau_{s_1 s_2} + \varphi\Omega$$

where $\Omega = (1+(1-\varphi)(1-\tau_{s_1 s_2}\eta_{s_1 s_2}))\tau_{s_1 s_2}$ is the convergence value of $\tau_{s_1 s_2}$ and ϕ is a parameter controlling the amount of pheromone left by an ant.

In the same way, the best path found by all ants receives an additional updating, as follows:

$$\tau_{rs} = (1-\rho)\tau_{rs} + \rho(1+\tau_{rs}\eta_{rs}\,Q(S_{Global\_Best}))\tau_{rs}$$

being $Q(S_{Global\_Best})$ the quality of such path. The quality of a path S_k is measured in terms of the average pheromone of the edges belonging to that path, $\overline{\tau}_k$, the percentage of ants that have selected that precise path as the most trustworthy, $\%A_k$, and its length, as it can be observed next:

$$Q(S_k) = \frac{\overline{\tau}_k}{\sqrt{\text{Length}(S_k)}}\,\%A_k$$

Furthermore, when ant k finds a peer offering the desired service, it has to decide whether to stop and return that found service provider, or to travel ahead trying to find a better (more reputable) one. In order to make that decision, the average pheromone trace of the edges composing the current path is computed, $\overline{\tau}_k$.

If $\overline{\tau}_k$ is greater than a given threshold, then ant k stops and returns current solution with a probability equal to $\overline{\tau}_k$ (which means that better paths have more probabilities to be chosen). Otherwise, if $\overline{\tau}_k$ is less than or equal to that certain threshold, ant k considers current service provider not enough trustworthy and keeps trying a better one.

As we indicated in Section 2, the last general step of every trust and reputation model consists of punishing or rewarding the selected service provider, according to the user's satisfaction. In BTRM WSN this step is explicitly performed in terms of pheromone evaporation (punishment) or reinforcement (reward) of the path leading to the selected peer, as shown next:

$$\tau_{rs} = (\tau_{rs} - \varphi \times df_{rs})\frac{Sat}{df_{rs}}$$

where ϕ is the same parameter used in the local pheromone updating, Sat is the user's satisfaction and df_{rs} represents a distance factor of the link joining sensors r and s, which is defined as follows:

$$df_{rs} = \sqrt{\frac{d_{rs}}{L(S_k)(L(S_k) - d_{rs} + 1)}}$$

being d_{rs} the distance of link joining sensors r and s from the client and $L(S_k)$ the length of the solution found by ant k, S_k.

5.3.2. Security analysis

Regarding the performance of BTRM WSN against certain threats, it has been demonstrated its accuracy in situations of malicious individual peers, malicious collectives, malicious

collectives with camouflage and driving down the reputation of a reliable peer.

When a peer is selected as a service provider and it supplies a worse service than the one it initially offered, not only the path leading to that server is punished (by means of phero mone evaporation), but also all the links or edges falling into that node, hindering this way other ants to choose that peer as the next hop in their route.

Malicious individual peers are, in this way, accurately identified in the community. Less than 10% of the times they are wrongly selected when the 90% of the nodes are individual malicious peers, in a WSN composed by of 100 sensors.

Due to the definition of the algorithm, where every peer only stores the pheromone traces of its neighbors, if a mali cious peer forms a collusion and gives unfair ratings (in terms of pheromone traces) to its neighbors, ants are able to over come this situation and find alternative paths (if they exist) leading to the most reputable nodes. This definition allows the resilience against a man in the middle attack, as well.

BTRM WSN is therefore resilient in the presence of mali cious collectives. In this case the selection percentage of malicious service providers (also called the error of the model) remains under the 10% regardless the size of the wireless sensor network, when the percentage of malicious peers forming a collusion is below the 60%.

Actually, the collusion threat model implemented in BTRM WSN corresponds to the threat we called here driving down the reputation of a reliable peer, which is in fact a particular case of a collusion. So this model has been demonstrated to be able to overcome both threats.

Once again the oscillating scenario of Fig. 5(a) has been chosen in order to test the model against malicious collectives with camouflage. In this case, since there are some benevolent peers not belonging to this collusion, they gain a high trust level in the system and are, therefore, selected most of the times as service providers, obtaining similar outcomes than in the case of malicious collectives (less than 10% of error when the percentage of malicious peers is under 60%).

Partially malicious collectives are also avoided since BTRM WSN uses different and independent pheromone traces for each service offered by the WSN.

5.4. *PowerTrust*

5.4.1. *Brief introduction*

PowerTrust (Zhou and Hwang, 2007) is a robust and scalable P2P reputation system which leverages the power law feed back characteristics found applicable in dynamically growing P2P networks, either structured or unstructured.

Authors made several comprehensive experiments over a data set extracted from eBay transactions and concluded that the feedback numbers in eBay follow a power law distribution. Even more, they demonstrate that power law feedback distri bution is applicable to every P2P reputation system in general.

The power law distribution implies that the node with a few feedbacks is common, whereas the node with a large number of feedbacks is extremely rare. Therefore, only a few nodes have much higher degree than others, and specifically those nodes are dynamically selected as power nodes and considered as most reputable in the system.

Nevertheless, power nodes can be dynamically replaced if they become less active or demonstrate unacceptable behavior. Actually, the m most reputable nodes are selected using a distributed ranking mechanism which in turn applies a locality preserving hashing in order to sort all nodes with respect to their global reputation scores.

To do so, PowerTrust builds a trust overlay network (TON) on top of all nodes in a P2P system where every peer evaluates each other whenever a transaction takes place between a pair of them. Therefore, all nodes have local trust scores and the system aggregates those scores in order to calculate the global reputation score of each participating peer. All global scores form a reputation vector $V = \{v_1, v_2, \ldots, v_n\}$ fulfilling that $\sum v_i = 1$.

In order to compute vector V, consider the trust matrix $R = (r_{ij})$ defined over an n node TON, where $r_{ij} \in [0, 1]$ is the normalized local trust score defined by $r_{ij} = s_{ij} / \sum_i s_{ij}$ (with $\sum_i r_{ij} = 1$), and s_{ij} is the most recent feedback score that node i rates node j. Next an initial reputation vector $V_{(0)}$ is set assuming, for instance, $v_i = 1/n$. And while $|V_{(k)} - V_{(k-1)}| > \varepsilon$ the successive reputation vectors are recursively computed as:

$$V_{(k+1)} = R^T \times V_{(k)}$$

After a sufficient number of k iterations, the global reputation vector will converge to the eigenvector of the trust matrix R. Finally, this global reputation scores updating is carried out by power nodes.

5.4.2. *Security analysis*

The use of reliable power peers as global reputation scores updaters makes PowerTrust a resilient model against a wide variety of security threats. Specifically, authors demonstrate the robustness and accuracy of their approach through a set of developed experiments.

Thus, PowerTrust has been proved to be resistant against an individual malicious peers attack, achieving good outcomes in presence of this type of adversaries (less than a 35% of error).

Even more, since authors consider that a node providing corrupted services is highly likely to issue dishonest scores, PowerTrust is also resilient (with experiments supporting this fact) against malicious collectives, malicious collectives with camouflage and driving down the reputation of a reliable peer.

Nevertheless, it is vulnerable to a malicious pre trusted peers threat, because in this model, power nodes are consid ered as fully reliable peers (as pre trusted peers are in Eigen Trust). So if those power peers become malicious, they can cause a great damage in the system.

5.5. *Tackling summary*

In this section, we present a summary table (Table 2) indi cating for each one of the described trust and reputation models which threats can be overcome, which not and which are just not applicable. In order to make a more complete table, we have also included some models (ATSN (Chen et al., 2007) and DWTrust (Huang et al., 2006)) not described in this paper.

As it can be observed, individual malicious peers, mali cious collectives and malicious collectives with camouflage

Table 2 – Tackling summary.

Security Threats	Trust and reputation models					
	EigenTrust	PeerTrust	BTRM-WSN	PowerTrust	ATSN	DWTrust
Individual malicious peers	✓	✓	✓	✓	✓	✓
Malicious collectives	✓	✓	✓	✓	✓	✓
Malicious collectives with camouflage	✓	✓	✓	✓	✓	✓
Malicious spies	|	|	|	|	|	|
Sybil attack	✓	|	|	|	|	|
Man in the middle attack	|	✓	✓	|	|	|
Driving down the reputation of a reliable peer	|	✓	✓	✓	|	|
Partially malicious collectives	|	✓	✓	|	|	✓
Malicious pre-trusted peers	|	Φ	Φ	|	Φ	Φ

✓, Resilient; |, vulnerable; Φ, not applicable.

are the most common tackled threats, while malicious spies and Sybil attack are not overcome by any or nearly any trust and reputation model.

Additionally, we have only found two models where the last threat (malicious pre-trusted peers) can be applied, which are EigenTrust and PowerTrust.

It is important to note that none of the presented models can absolutely prevent all the threats and that the proposed solutions given in Section 3 are just some helpful guides that aim to decrease the impact of each one of the associated threats, but they cannot (and the do not pretend to) completely overcome them.

6. Conclusions and future work

Trust and reputation management over distributed and heterogeneous systems has emerged in the last few years as a novel and accurate way of dealing with some security risks related to these environments.

Nevertheless, the application of such mechanisms involves the arising of new specific and related threats that should not be underestimated. As far as we know, this is one of the first works mainly focused on describing such threats and proposing solutions to overcome them.

In this paper, we have analyzed the main security threats that can be applied in most of trust and reputation schemes. Moreover, we have discussed them and suggested a possible way of tackling each one of those risks in the design phase.

A complete taxonomy of those threats or attacks has been developed as well, describing several possible dimensions of an attack over trust and reputation systems, and categorizing the exposed threats according to these dimensions or properties.

Additionally, we have presented some representative trust and reputation models and shown how they deal with those threats that can be applied to them, revealing that not all the threats are paid the same attention and none of them is categorically solved.

As for future work, we consider that an implementation and comparison of several of the most representative trust and reputation models, in terms of their response against some of the threats presented in this paper could be an interesting research line. In that way, we will focus on the development of a validation tool allowing researchers to perform such tests.

Finally, we hope this work helps to the development of this research field by constituting a guide for new trust and reputation model designers.

Acknowledgements

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2007. Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

REFERENCES

Almenárez F, Marín A, Campo C, García C. PTM: a pervasive trust management model for dynamic open environments. In: Privacy and trust. First workshop on pervasive security and trust, Boston, USA; Aug 2004.

Almenárez F, Marín A, Díaz D, Sánchez J. Developing a model for trust management in pervasive devices. In: PERCOMW '06: proceedings of the 4th annual IEEE international conference on pervasive computing and communications workshops. Washington, DC, USA: IEEE Computer Society; 2006. p. 267.

Boukerche A, Xu L, El-Khatib K. Trust-based security for wireless ad hoc and sensor networks. Computer Communications 2007;30(11–12):2413–27.

Carbó J, Molina J, Dávila J. Trust management through fuzzy reputation. International Journal of Cooperative Information Systems March 2003;12:135–55.

Chen H, Wu H, Zhou X, Gao C. Agent-based trust model in wireless sensor networks. In: Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing, SNPD 03; 2007. p. 119–24.

Dorigo M, Gambardella L, Birattari M, Martinoli A, Poli R, Stützle T. Ant colony optimization and swarm intelligence. In: Fifth international workshop, ANTS 2006. LNCS, vol. 4150. Brussels, Belgium: Springer; 2006.

Douceur JR, Donath JS. The Sybil attack. In: Proceedings for the 1st international workshop on peer-to-peer systems (IPTPS '02); 2002. p. 251–60.
Girao J, Sarma A, Aguiar R. Virtual identities – a cross layer approach to identity and identity management. In: Proceedings for the 17th wireless world reseach forum, Heidelberg, Germany; Nov 2006.
Gómez Mármol F, Martínez Pérez, G. Providing trust in wireless sensor networks using a bio-inspired technique. In: Proceedings of the networking and electronic commerce research conference, NAEC'08. Lake Garda, Italy; Sep 2008.
Gómez Mármol F, Martínez Pérez G, Gómez Skarmeta AF, TACS, a trust model for P2P networks. Wireless personal communications, special issue on "information security and data protection in future generation communication and networking"; 2008.
Huang C, Hu H, Wang Z. A dynamic trust model based on feedback control mechanism for P2P applications. In: Autonomic and trusted computing, Third international conference, ATC. LNCS, vol. 4158. Wuhan, China: Springer; 2006. p. 312–21.
Josang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. Decision Support Systems 2007;43(2):618–44.
Kamvar S, Schlosser M, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks, Budapest, Hungary; May 2003.
Lam SK, Riedl J, Shilling recommender systems for fun and profit. In: WWW '04: proceedings of the 13th international conference on World Wide Web; 2004.
Marti S, Garcia-Molina H. Taxonomy of trust: categorizing P2P reputation systems. Computer Networks 2006;50(4):472–84.
Moloney M, Weber S, A context-aware trust-based security system for ad hoc networks. In: Workshop of the 1st international conference on security and privacy for emerging areas in communication networks, Athens, Greece; Sep 2005. p. 153–60.
Sabater J, Sierra C. REGRET: reputation in gregarious societies. In: Müller JP, Andre E, Sen S, Frasson C, editors. Proceedings of the fifth international conference on autonomous agents. Montreal, Canada: ACM Press; 2001. p. 194–5.
Sabater J, Sierra C. Review on computational trust and reputation models. Artificial Intelligence Review 2005;24(1): 33–60.
Songsiri S. MTrust: a reputation-based trust model for a mobile agent system. In: Autonomic and trusted computing, Third international conference, ATC. LNCS, vol. 4158. Wuhan, China: Springer; 2006. p. 374–85.
Tajeddine A, Kayssi A, Chehab A, Artail H. PATROL-F – a comprehensive reputation-based trust model with fuzzy subsystems. In: Autonomic and trusted computing, Third international conference, ATC. LNCS, vol. 4158. Wuhan, China: Springer; 2006. p. 205–17.
Wang W, Zeng G, Yuan L. Ant-based reputation evidence distribution in P2P networks. In: GCC: fifth international conference on grid and cooperative computing. Changsha, Hunan, China: IEEE Computer Society; 2006a. p. 129–32.
Wang Y, Cahill V, Gray E, Harris C, Liao L. Bayesian network based trust management. In: Autonomic and trusted computing, Third international conference, ATC. LNCS, vol. 4158. Wuhan, China: Springer; 2006. p. 246–57.
Xiong L, Liu L. PeerTrust: supporting reputation-based trust in peer-to-peer communities. IEEE Transactions on Knowledge and Data Engineering 2004;16(7):843–57.
Zhou R, Hwang K. PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing. Transactions on Parallel and Distributed Systems 2007.

Félix Gómez Mármol is a PhD student in the Department of Information and Communications Engineering of the University of Murcia. His research interests include authorization, authentication and trust management in distributed and heterogeneous systems, security management in mobile devices and design and implementation of security solutions for mobile and heterogeneous environments. He received an MSc in computer engineering from the University of Murcia. Contact him at felixgm@um.es.

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering of the University of Murcia. His research interests include security and management of IPv4/IPv6 communication networks. He received an MSc and PhD in computer engineering from the University of Murcia. Contact him at gregorio@um.es.

7

TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks

Title:	TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks
Authors:	Félix Gómez Mármol, Gregorio Martínez Pérez
Type:	International Conference
Conference:	IEEE International Conference on Communications (IEEE ICC 2009), Communication and Information Systems Security Symposium
Location:	Dresden, Germany
Publisher:	IEEE Communications Society
Year:	2009
Month:	June
DOI:	http://dx.doi.org/10.1109/ICC.2009.5199545
State:	Published

Table 7: TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks

Félix Gómez Mármol and Gregorio Martínez Pérez
Departamento de Ingeniería de la Información y las Comunicaciones
University of Murcia
Facultad de Informática, Campus de Espinardo, s/n
30.071, Murcia, Spain
{felixgm, gregorio}@um.es

Abstract—**Trust and reputation models research and development for distributed systems such as P2P networks, Wireless Sensor Networks (WSNs) or Multi-agent systems has arisen and taken importance in the last recent years among the international research community. However it is not always easy to check the correctness and accuracy of a model and even more, to compare it against other trust and reputation models. This paper presents TRMSim-WSN, a Java-based trust and reputation models simulator aimed to provide an easy way to test a trust and/or reputation model over WSNs and to compare it against other models. It allows the user to adjust several parameters such as the percentage of malicious nodes or the possibility of forming a collusion, among many others.**

I. Introduction

Relevance and utility of distributed networks is improving everyday due to the numerous applications they have and all the research efforts that are focused on them. Specifically Wireless Sensor Networks [1] (WSNs) are widely spread and employed in multiple scenarios such as fire detection, weather measurements and even traffic management in Vehicular-to-Vehicular [2] (V2V) networks.

Nevertheless, this kind of networks has its own drawbacks. Their wireless way of communication, their battery and bandwidth constraints or their location in open environments, for example, lead them to some security threats. Recently, trust and reputation management has become a novel way of dealing with some of these important issues. Thus, several trust and/or reputation models over WSNs [3], [4], [5], [6] have been developed and studied.

But most of them provide their own and particular test set in order to demonstrate their accuracy and goodness, leading to a more difficult way of objectively comparing them with other models. Furthermore, it is usually hard to design a new trust and/or reputation model without any guideline. Therefore, in this paper we present TRMSim-WSN [7], a trust and reputation models simulator for WSNs aimed to provide a generic tool in order to test and compare trust and reputation models. We have included some of the most common experiments found in the literature for this kind of models and also developed an API which constitutes a template for easily including new trust and reputation models into our simulator.

A number of network simulators [8], [9] has been proposed and developed in the last few years. But most of them focus their attention on complex communication protocols, which many times require an expert knowledge. TRMSim-WSN is one layer above. It abstracts developers from low level communication issues and centers specifically on trust and reputation models.

The rest of the paper is organized as follows. Section II describes some simulation environments for trust and/or reputation models. In section III we explain the generic API designed to implement new trust and reputation models and how these are added to the simulator. TRMSim-WSN is presented in section IV and section V shows some conclusions and future work.

II. Related work

Lots of network simulators have been developed in order to test new communication protocols and check their correctness, robustness or accuracy. Authors of [8], for instance, present a survey of P2P network simulators, describing their main features and also their limitations.

However, while the number of those network simulators is considerable there is a lack of trust and reputation models simulators for distributed networks. One of the few ones that have been designed and published is TOSim [10]. It has been created to be highly modular and configurable, without incurring in excessive overload both in terms of memory and time. In order to simulate behavior related to trust, authors consider four threat models of malicious peers to cause insecure files to be uploaded to the system.

Another important platform for simulating reputation models is ART testbed [11] which has become in recent years a reference environment in the field of reputation models for multi-agent systems. This testbed serves, on the one hand, as a competition forum in which researchers can compare their models against objective metrics and, on the other hand, as a set of tools with flexible parameters, allowing researchers to perform customizable and easily-repeatable experiments.

Nevertheless, we have not found any simulator, competition or environment targeting trust and reputation systems for WSNs. Therefore, as far as we know, TRMSim-WSN is the first trust and reputation model tool covering this objective.

978-1-4244-3435-0/09/$25.00 ©2009 IEEE
Authorized licensed use limited to: IEEE Transactions on SMC Associate Editors. Downloaded on March 06,2010 at 15:49:13 EST from IEEE Xplore. Restrictions apply.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

III. Generic Trust and Reputation Model

Each trust and reputation model has its own specific characteristics and particularities. However, most of them share the same abstract schema or pattern about what steps have to be given in order to complete a whole transaction in a distributed system making use of a trust and/or reputation model.

Therefore, one of the main targets followed by our work was to design and provide a trust and reputation models interface as generic as possible. So first of all, we identified the four main steps to be done in most of this kind of models [12], [13]. Figure 1 shows these steps.

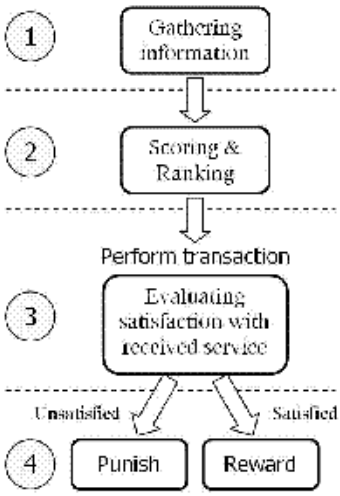


Fig. 1. Generic Trust and Reputation Model Scheme

We have developed an abstract Java class called `TRModel_WSN` containing one attribute: a set of generic parameters for trust and reputation models (abstract class `TRMParameters` containing the name of the parameters file and the parameters themselves in the form <parameter=value>).

In order to add a new trust and/or reputation model to the simulator both subclasses of `TRModel_WSN` and `TRMParameters` have to be implemented. A subclass of `Service` class could be also defined in order to specify more details or characteristics (such as associated costs or quality parameters, for instance) of a certain service.

Additionally, class `TRModel_WSN` defines the five public abstract methods shown in table I in order to accomplish the steps illustrated in figure 1.

The first method, `gatherInformation`, is responsible for collecting or gathering the necessary information from other nodes in the network (indirect experiences, recommendations, reputation values, etc.) if we are dealing with a pure reputation model, direct experiences or pre-trusted nodes, if what we have is a pure trust model, or a combination of both, which is the most common case.

Returned Value	Method Name	Arguments
`GatheredInfo`	`gatherInformation`	`Client Service`
`Vector<Sensor>`	`scoreAndRanking`	`Client GatheredInfo`
`Outcome`	`performTransaction`	`Vector<Sensor> Service`
`Outcome`	`reward`	`Vector<Sensor> Outcome`
`Outcome`	`punish`	`Vector<Sensor> Outcome`

TABLE I
`TRModel_WSN` ABSTRACT METHODS

Its first parameter is the `Client` who is requesting the desired service and, therefore, needs the application of the trust and reputation model in order to find the most trustworthy or reputable server offering the `Service` given as a second parameter.

It returns a `GatheredInformation` object. Currently this class only contains the paths leading to those servers which are candidates to be selected as service providers. Each model can create a subclass of this one including the specific information needed to work.

The second method, `scoreAndRanking`, receives the gathered information from the previous one and scores each path leading to a server, returning either a sorted collection of these servers (according to the score received) or the path leading directly to the most trustworthy server found.

The third abstract method belonging to class `TRModel_WSN`, called `performTransaction`, receives as a parameter the path found in the previous step, so it can actually apply for the required service to the server selected as most trustworthy or most reputable by the implemented model.

Then the server, according to its goodness, will provide exactly the same service it has been asked for, a worse one or even a better one, in some cases. Once the client receives the service, it assesses its satisfaction and returns its value in an `Outcome` object (necessary to perform some statistics in order to evaluate the accuracy of the model). Some models would store in this step that transaction satisfaction as a direct experience.

Finally, the last two methods, `reward` and `punish`, carry out the fourth step pointed out in the scheme shown in figure 1. That is, they perform the reward and punishment, respectively, to the server who has been selected to have the transaction with. Depending on the satisfaction of the client with the supplied service, one or the other will be applied.

They both receive two parameters: the path leading to the most trustworthy or reputable server found in the second step and the outcome got in the third one, containing, among other things the satisfaction or dissatisfaction of the client with the received service.

It is worth mentioning that there are, however, some trust and reputation models which do not apply any additional punishment and/or reward to those nodes the interaction has

Authorized licensed use limited to: IEEE Transactions on SMC Associate Editors. Downloaded on March 06,2010 at 15:49:13 EST from IEEE Xplore. Restrictions apply.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

been carried out with. Thus, these two methods may not have any particular code associated depending on the particular trust and reputation model being implemented and addapted to the TRMSim-WSN proposed architecture.

Regarding the parameters needed for the trust and reputation model, abstract class `TRMParameters` defines several protected methods, used to store and retrieve generic parameters of any of the primitive types.

Finally, figure 2 shows a brief class diagram including the main classes and their relationships involved in the design of our generic trust and reputation models interface.

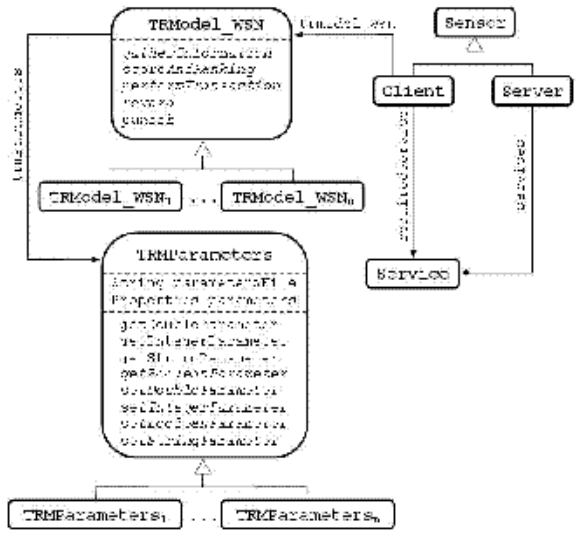


Fig. 2. Class diagram of main classes of the generic trust and reputation models interface

As it can be observed, each `Client` has its required service as an attribute, while each `Server` has a collection of offered services. Both are subclasses of `Sensor` class. Therefore, every client uses its trust and reputation model in order to: i) search the most trustworthy and/or reputable server offering the desired service, ii) apply for that certain service to the selected server and assess its satisfaction with the actually received service, and iii) punish or reward the service provider according to that satisfaction.

A further and more detailed explanation on how to add successfully a new trust and/or reputation model to the TRMSim-WSN simulator can be found in [7].

IV. TRMSim-WSN

In this section we will formally present and describe our proposal of Trust and Reputation Models Simulator for Wireless Sensor Networks, called TRMSim-WSN [7]. A screenshot of the main window of TRMSim-WSN can be observed in figure 3.

A. Network settings

The very first step to be carried out when using our simulator is to create a new WSN. To do that, there are two fields where we can establish the maximum and the minimum number of sensors we want our networks to have, as well as a slide bar to set the wireless range of every sensor. Those three parameters will determine the links density of the network (i.e., the neighborhood of every node).

Additionally, we can select which percentage of the nodes we want to act as clients requiring a default service. The rest of them will act, therefore, as servers. We can also say which percentage of those servers will not offer the required service and will then only act as relay nodes. Finally, regarding the servers who actually offer the desired service it is possible to determine the percentage of them who will be malicious ones, that is, they will not provide the service the are actually offering, but a worse one or even any service.

Once we have set all those parameters according to our needs, a new random WSN can be created just by pushing the bottom labelled "New WSN". It is also possible to load a WSN from a XML file by pushing "Load WSN" button, and to save the current one into a XML file through the "Save WSN" button.

If we want to evaluate the WSN we currently have, but with different links density, we can change the wireless range parameter and push "Reset WSN" button.

B. Simulation settings

The next thing to configure are the simulation settings. First we can determine the number of executions we want for our simulations, that is, the number of times every client in the network will ask for its default service, making use of the selected trust and reputation model. We can set the number of different random WSNs we want as well, according to the settings described in the previous subsection.

We can take some decisions regarding the visual or graphic presentation of the networks to be tested in our simulations. For instance, we can decide whether we want the wireless ranges to be shown or not, as well as the links connecting sensors or the identifier of each one of them.

TRMSim-WSN is initially released with two trust and reputation models: BTRM-WSN [3] and PeerTrust [14]. Furthermore, the parameters panel allows us to set the input parameters file, or to manually specify the value of each parameter needed by the current selected trust and reputation model.

Since one of the main characteristics of WSNs are their constraints about battery and energy consumption, a dynamic WSN can also be simulated, where some sensors swap into an idle state for awhile if they do not receive any request within a certain period of time. A sensor in an idle state does not receive nor transmit any message or packet. After a certain timeout they wake up again.

Once we have established all the previous settings, we are ready to start our simulations. If we want to run a simulation only over current network, we should press "Run WSN"

Authorized licensed use limited to: IEEE Transactions on SMC Associate Editors. Downloaded on March 06,2010 at 15:49:13 EST from IEEE Xplore. Restrictions apply.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

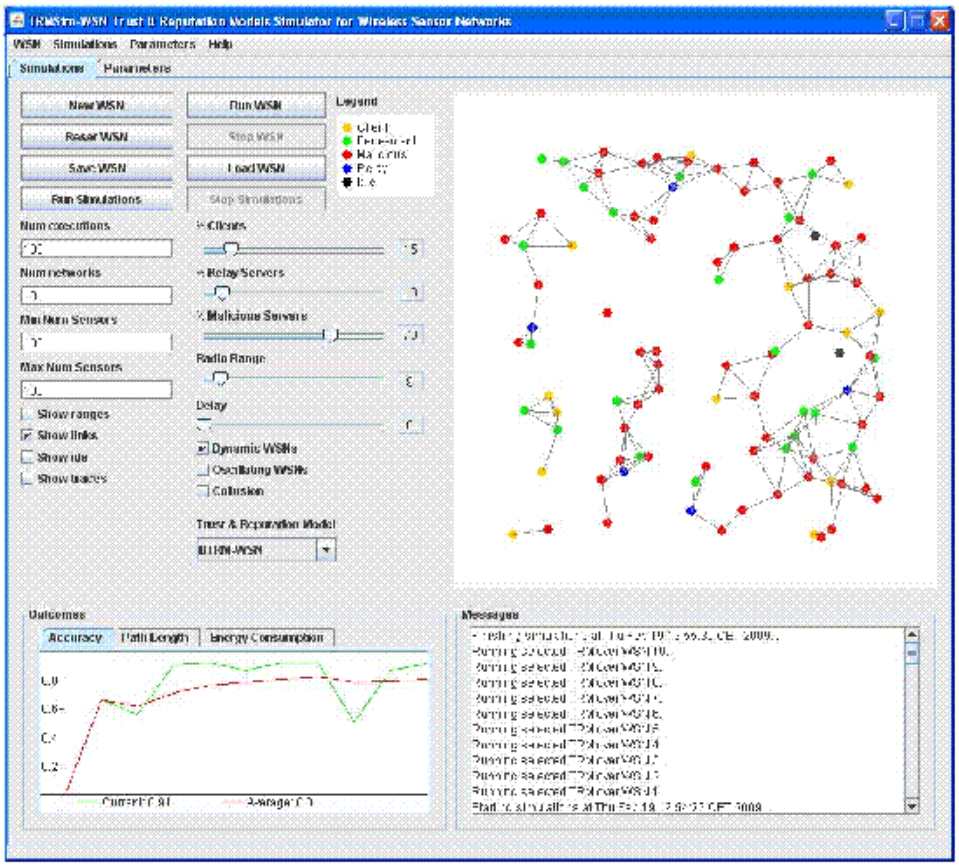


Fig. 3. TRMSim-WSN. Trust and Reputation Models Simulator for Wireless Sensor Networks

button. "Stop WSN" button allows us to force the finishing of that simulation.

Otherwise, if what we want is to run a simulation over a given number of random WSNs, then we have to push button labelled "Run Simulations". We can stop that simulation whenever we want by pressing "Stop Simulations" button, and current outcomes will be shown.

Finally we can also add some delay between each simulated network, if we need to check the topology of every tested WSN. The maximum value corresponds to one second.

C. Oscillating behavior and collusion

In order to test the accuracy of every simulated trust and reputation model we have included two security threats [15] to our simulator. First one has to do with the oscillating behavior of the servers offering the requested service.

Therefore, if that option is selected, after every 20 executions (i.e. transactions or interactions), each malicious server becomes benevolent. Then the same percentage of previous malicious servers are randomly chosen to be now malicious (note that with a scheme like this a malicious server could remain malicious after 20 executions).

The second security threat introduced consists of the possibility for the malicious servers to form a collusion among themselves. That implies that every malicious sensor will give the maximum rating for every other malicious sensor, and the minimum rating for every benevolent one.

A good trust and reputation model should quickly react against these behavioral changes and collusions and readapt itself in order to prevent selecting a malicious node as the most trustworthy or reputable one.

D. Outcomes and messages

Finally, two panels help us to know what has happened or is currently happening in the simulator, and which are the results of the last simulation.

In the messages panel, for instance, several messages are shown containing useful information like the instant when the last simulation started or finished, or which is the current WSN being tested. Moreover, every action such as creating a

Authorized licensed use limited to: IEEE Transactions on SMC Associate Editors. Downloaded on March 06,2010 at 15:49:13 EST from IEEE Xplore. Restrictions apply.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

new WSN, loading or saving current one or showing ranges, identifiers and links, among others, are also recorded and shown there.

On the other hand, the outcomes panel lets us know the results of the current simulated network, or the average outcomes for a whole simulation. Three important values can be observed here: the accuracy of the model, the average length of all the paths found by every client of every simulated network, and the energy consumed by the model (for future work). Additional panels can be easily added if required in order to show more details about the experiments.

The average satisfaction is computed collecting the satisfaction of every client belonging to each one of the tested WSNs. However, clients who can not reach any benevolent server are not taken into account for computing these outcomes (since any trust and reputation model is useful in that situation).

In figure 3 we can observe that a simulation over 10 random dynamic WSNs (with 100 sensors each one) has been carried out using **BTRM-WSN** model. There were a 15% of clients, an 8.5% (85% · 10%) of relay sensors, a 53.55% (85% · 90% · 70%) of malicious servers and a 22.95% (85% · 90% · 30%) of benevolent ones. The average number of hops needed to reach the most trustworthy server was 6.04 and the average percentage of times that the model selected a benevolent server as the most trustworthy one was 80%.

V. Conclusions and Future Work

A number of network simulators can be found nowadays, allowing us to test low level communication protocols. Nevertheless, there is a lack of simulators aimed to check the correctness and accuracy of trust and reputation models for distributed systems and, specifically, for WSNs.

In this paper we have presented **TRMSim-WSN**, a novel trust and reputation models simulator for wireless sensor networks. As far as we know, this is the first simulator of these characteristics for WSNs. We have shown the generic trust and reputation models interface we have designed and developed and explained how a new trust and reputation model can be easily added to the simulator.

We have also described the main features and possibilities that **TRMSim-WSN** offers, and how to configure it in order to carry out customized simulations.

Nonetheless, several improvements and enhancements could be applied to **TRMSim-WSN**. For instance, we are planning to add an energy consumption module to determine the overhead introduced by each simulated model. Another interesting option would be the possibility of selecting a specific sensor and changing its properties (services offered, goodness, etc.). Mobile wireless sensor networks, where nodes can move and change their position along the time is also a new feature we are planning to incorporate to the **TRMSim-WSN** simulator.

Acknowledgment

This work has been partially funded by project *Middleware de gestión de Identidades de Seguridad en TRansacciones electrónicAs basado en código Libre* (**MISTRAL**) with code **TIC-INF** 07/01-0003), and by project *Infraestructura de Servicios Ubicuos y de Comunicaciones en Redes Vehiculares* with code **TIN2008-06441-C02-02**. It has been also funded by a Séneca Foundation grant within the Human Resources Research Training Program 2007. Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.

References

[1] K. Römer and F. Mattern, "The Design Space of Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, dec 2004.

[2] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE*, vol. 2, no. 2, pp. 12–22, jun 2007.

[3] F. Gómez Mármol and G. Martínez Pérez, "Providing Trust in Wireless Sensor Networks using a Bio-inspired Technique," in *Proceedings of the Networking and Electronic Commerce Research Conference, NAEC'08*, 2008.

[4] A. Boukerche, L. Xu, and K. El-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[5] S. Buchegger and J. Y. Le Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks," in *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge MA, USA, jun 2004.

[6] F. Almenárez, A. Marín, C. Campo, and C. García, "PTM: A pervasive trust management model for dynamic open environments," in *Privacy and Trust*. Boston, USA: First Workshop on Pervasive Security and Trust, aug 2004.

[7] Félix Gómez Mármol, "TRMSim-WSN, a Trust & Reputation Models Simulator for Wireless Sensor Networks," http://anis.dil.um.es/~felixgm/research/trmsim-wsn.

[8] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai, "A survey of peer-to-peer network simulators," *Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06)*, 2006.

[9] B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," 2005, pp. 477–482.

[10] Y. Zhang, W. Wang, and S. Lü, "Simulating trust overlay in p2p networks," in *International Conference on Computational Science (1)*, 2007, pp. 632–639.

[11] K. K. Fullam, T. Klos, G. Muller, J. Sabater-Mir, K. Barber, and L. Vercouter, "The Agent Reputation and Trust (ART) Testbed," in *Trust Management*, ser. LNCS, no. 3986, Fourth International Conference, iTrust 2006. Pisa, Italy: Springer, may 2006, pp. 439–442.

[12] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, mar 2006.

[13] Y. Sun and Y. Yang, "Trust Establishment in Distributed Networks: Analysis and Modeling," in *Proceedings of the IEEE International Conference on Communications (IEEE ICC 2007), Communication and Information Systems Security Symposium*, Glasgow, Scotland, jun 2007.

[14] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[15] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proc. of the International World Wide Web Conference (WWW)*, Budapest, Hungary, may 2003.

[16] E. Aivaloglou, S. Gritzalis, and C. Skianis, "Trust establishment in sensor networks: behaviour-based, certificate-based and a combinational approach," *International Journal of System of Systems Engineering*, vol. 1, no. 1-2, pp. 128–148, 2008.

[17] A. Mitseva, E. Aivaloglou, M. Marchitti, N. R. Prasad, C. Skianis, S. Gritzalis, A. Waller, T. Baug, and S. Pennington, "Towards adaptive security for convergent wireless sensor networks in beyond 3g environments," *Wireless Communications and Mobile Computing*, 2008.

[18] E. Aivaloglou, S. Gritzalis, and C. Skianis, "Towards a flexible trust establishment framework for sensor networks," *Telecommunication Systems. Modeling, Analysis, Design and Management*, vol. 35, no. 3-4, pp. 207–213, 2007.

Authorized licensed use limited to: IEEE Transactions on SMC Associate Editors. Downloaded on March 06,2010 at 15:49:13 EST from IEEE Xplore. Restrictions apply.

Appendices

A

Acceptance letters

Title	State	Date
State of the Art in Trust and Reputation Models in P2P networks	Published	March 2010
TACS, a Trust Model for P2P Networks	Published	October 2009
META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm	Accepted	May 2009
Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique	In press	February 2010
Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems	Published	June 2010
Security Threats Scenarios in Trust and Reputation Models for Distributed Systems	Published	October 2009
TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks	Published	June 2009

Table 8: Publications summary

META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm

29-May-2009

Coordinating author, Gregorio Martinez Perez (gregorio@um.es).

Paper: IASC 1537R - META-TACS: A trust model demonstration of robustness through a genetic algorithm

Dear Dr. Martinez-Perez:

I am pleased to inform you of the acceptance of your paper after the latest suggestions have been taken into account. In preparation of your manuscript for final processing, we are asking you to do the following:
1) Prepare the revised paper in MS Word format with additional files for the figures and email them directly to Ms. Debbie Rivard (debrarivard@yahoo.com) and riv686@aol.com. A template can be found from the link http://wacong.org/autosoft/auto/forms.html
2) Provide a recent photo (on say .jpg, etc.) of all authors and a 50-word biographical sketch (in MS word) with the same email. Please put all of the files and email them also to Ms. Debbie Rivard
3) Please go to AutoSoft web site (see above) under Authors' kit and download a TSI Press copyright form and fill it in and send it by airmail to the address at the top of this letter.
4) Please send all these items as soon as possible so that our publishing schedule would not be delayed.

Today, AutoSoft Journal (1st journal on soft computing in the world) papers are being abstracted in over ten scientific and technological indices (ISI, Elsevier, EBSCD, etc.), and WAC papers from WAC meetings books and CD ROM proceedings go into IEEE Xplore. Thanks to fine contribution like yours, we will keep up the quality and standard of this publication. We certainly hope that you can realize our situation, that without this modest support it would be very difficult for the journal to survive. Thank you for your contribution to the AutoSoft Journal.

We look forward to seeing you in Japan in 2010 (see wacong.org for details and call for papers).
Please promote this congress among all your peers.

Sincerely yours,

Jila Salari
Publisher
WAC 2010 Co-General Chairperson

PS: Address of Production Manager:

Cc. Ms. Debbie Rivard, Production Manager Mailing Address : 176 High Ridge Trail SE Rio Rancho, NM 87124-3398 USA Riv686@aol.com, debrarivard@yahoo.com

Publications relevance

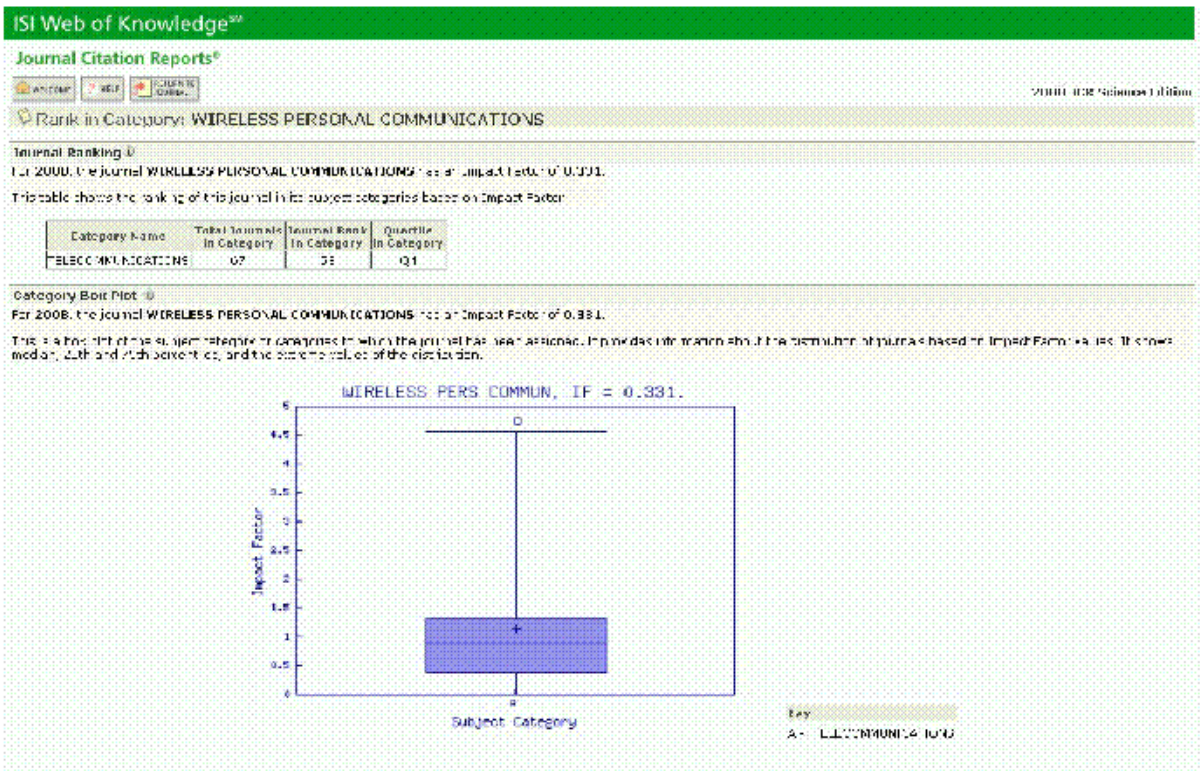
State of the Art in Trust and Reputation Models in P2P networks

The article entitled 'State of the Art in Trust and Reputation Models in P2P networks', has been published as a book chapter in the Springer Handbook of Peer-to-Peer Networking.

Springer is a global publishing company which publishes books, e-books and peer-reviewed journals in science, technical and medical (STM) publishing. Within STM, Springer is the largest book publisher, and second-largest journal publisher worldwide (the largest being Elsevier).

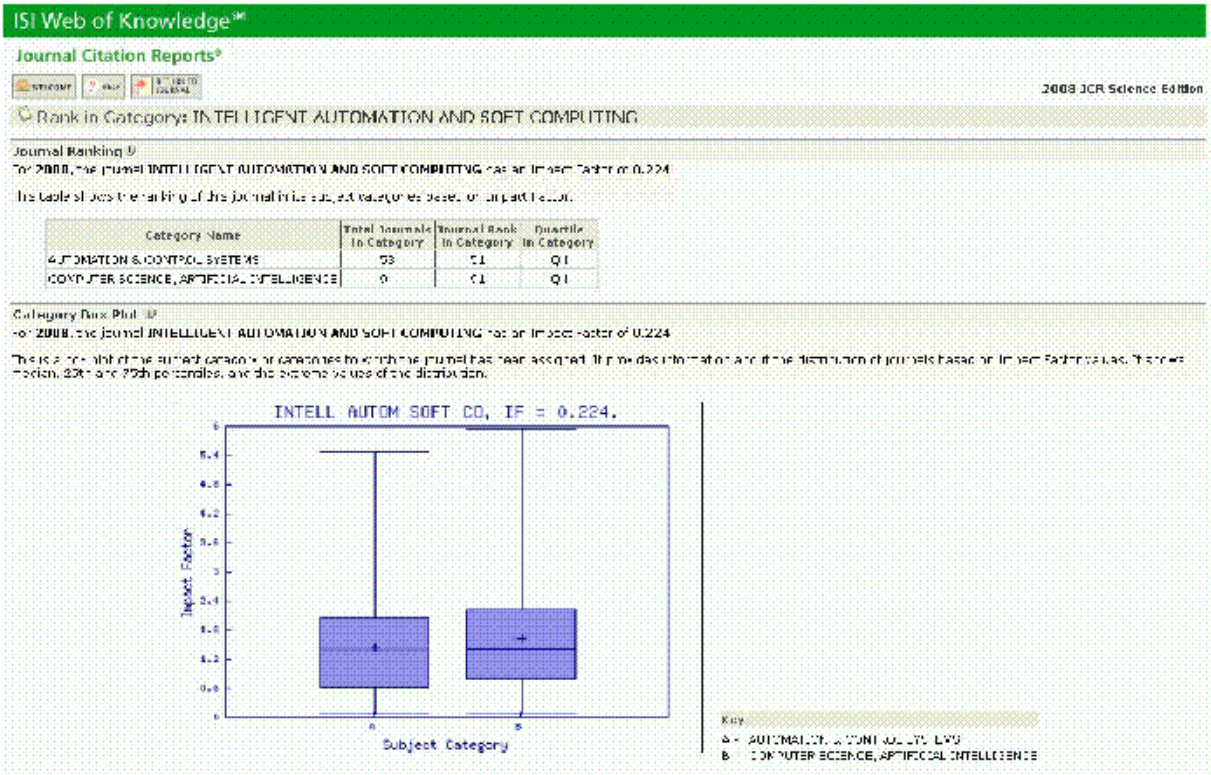
TACS, a Trust Model for P2P Networks

The article entitled 'TACS, a Trust Model for P2P Networks', has been published in Wireless Personal Communications Journal, whose relevance and impact factor can be observed next:



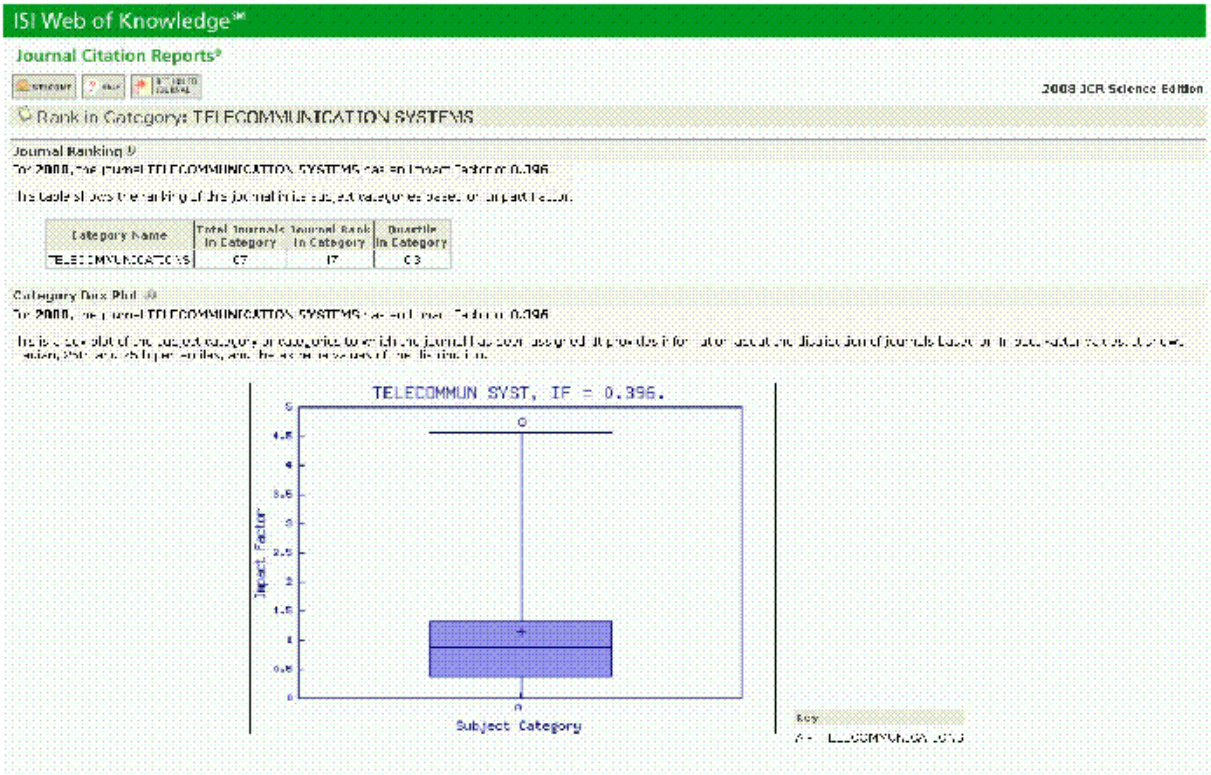
META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm

The article entitled 'META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm', has been accepted for publication in Intelligent Automation and Soft Computing Journal, whose relevance and impact factor can be observed next:



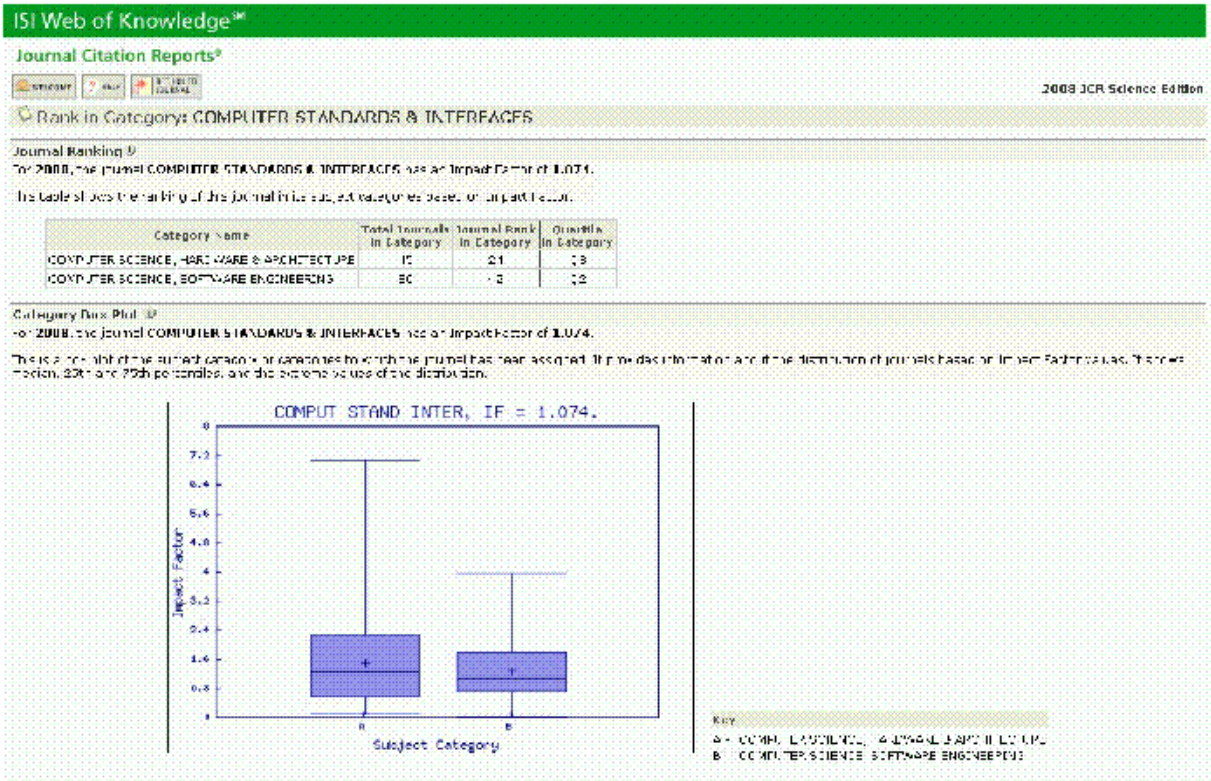
Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique

The article entitled 'Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique', has been accepted for publication in Telecommunication Systems Journal, whose relevance and impact factor can be observed next:



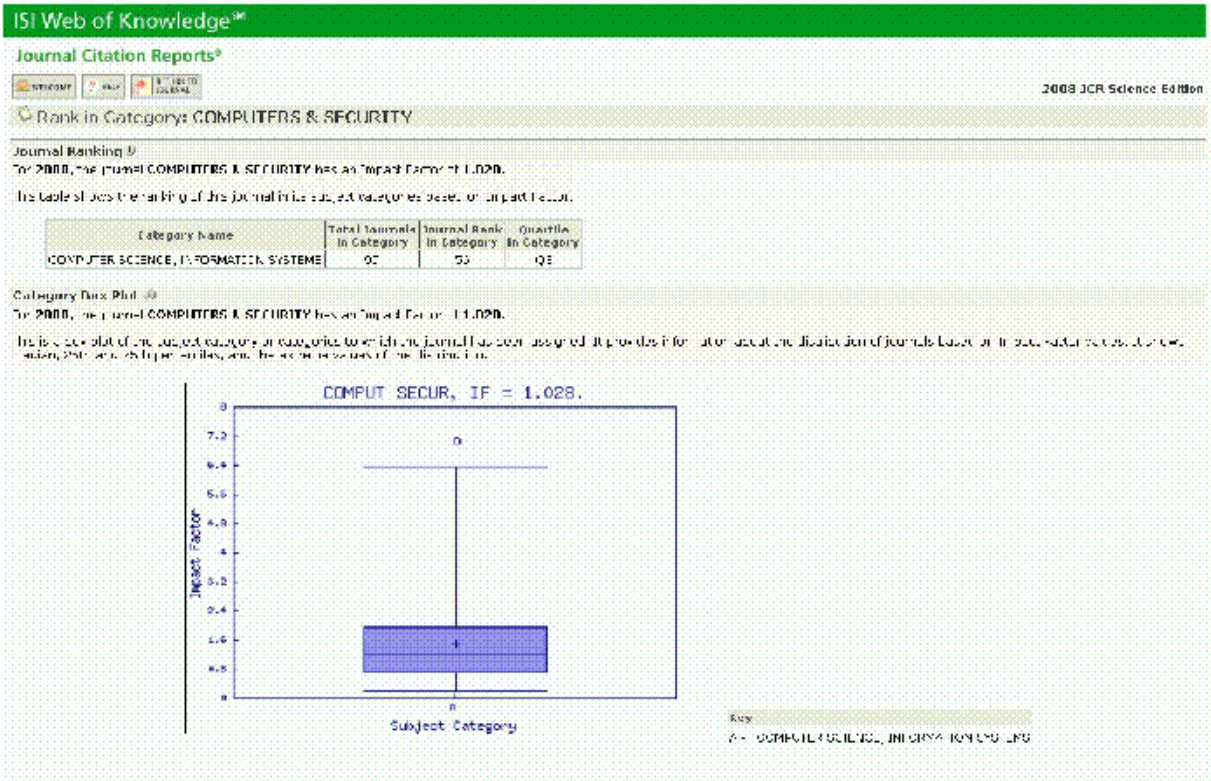
Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems

The article entitled 'Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems', has been accepted for publication in Computer Standards & Interfaces Journal, whose relevance and impact factor can be observed next:



Security Threats Scenarios in Trust and Reputation Models for Distributed Systems

The article entitled 'Security Threats Scenarios in Trust and Reputation Models for Distributed Systems', has been published in Computers & Security Journal, whose relevance and impact factor can be observed next:



Additionally, this paper was in the top 10 most downloaded articles of this journal in the last quarter of 2009.

TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks

The article entitled 'TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks', was presented and subsequently published in the proceedings of the IEEE International Conference on Communications (IEEE ICC 2009).

The International Conference on Communications (ICC) is an annual international academic conference organized by the Institute of Electrical and Electronics Engineers' Communications Society. Some major telecommunications discoveries have been announced at ICC, such as the invention of turbo codes. Recent ICCs have been attended by 1200 - 1400 people.

Acronyms

AAA, Authentication, Authorization and Accounting

ACO, Ant Colony Optimization

ACS, Ant Colony System

AFRAS, A Fuzzy Reputation Agent System

API, Application Programming Interface

AS, Ant System

ATRM, Agent-based Trust and Reputation Management

ATSN, Agent-based Trust model in wireless Sensor Networks

BNBTM, Bayesian Network Based Trust Management

BTRM-WSN, Bio-inspired Trust and Reputation Model for Wireless Sensor Networks

C2C, Car-to-Car

CA, Certification Authority

CHC, Cross generational elitist selection, Heterogeneous recombination, Cataclysmic mutation

CORE, COllaborative REputation mechanism

DRBTS, Distributed Reputation-based Beacon Trust System

EAS, Elitist Ant System

EC, Evolutionary Computation

ECDLP, Elliptic Curve Discrete Log Problem

FP, Framework Provider

GA, Genetic Algorithm

IdM, Identity Management Systems

IdP, Identity Provider

ITS, Intelligent Transportation Systems

J2ME, Java 2 Micro Edition

MANET, Mobile Ad-hoc NETwork

MMAS, Max-Min Ant System

P2P, Peer-to-Peer

PGP, Pretty Good Privacy

PKI, Public Key Infrastructure

PTM, Pervasive Trust Management

QDV,Quality-based Distance Vector

QoS, Quality of Service

REGRET, REputation in GREgarious societies

RFSN, Reputation-based Framework for high integrity Sensor Networks

RP, Recommendation Provider

RRS, Robust Reputation System

SLA, Service Level Agreement

TACS, Trust Ant Colony System

TDTM, Time-based Dynamic Trust Model

TRIMS, Trust and Reputation model for Identity Management Systems

TRMSim-WSN, Trust and Reputation Simulator for Wireless Sensor Networks

TTL, Time-To-Live

V2V, Vehicular-to-Vehicular

WSAN, Wireless Sensor and Actuator Network

WSC, Web Service Consumer

WSN, Wireless Sensor Network

WSP, Web Service Provider

WWW, World Wide Web

Bibliography

I Trust and reputation models

[1] Aameek Singh and Ling Liu. TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems. *IEEE International Conference on Peer-to-Peer Computing*, pages 142–149, 2003.

[2] Avinash Srinivasan, Joshua Teitelbaum, and Jie Wu. DRBTS: Distributed Reputation-based Beacon Trust System. In *DASC '06: Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pages 277–283, Washington, DC, USA, 2006. IEEE Computer Society.

[3] Ayman Tajeddine, Ayman Kayssi, Ali Chehab, and Hassan Artail. PATROL-F - a comprehensive reputation-based trust model with fuzzy subsystems. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 205–217, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[4] Azzedine Boukerche, Li Xu, and Khalil El-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30(11-12):2413–2427, 2007.

[5] Chenlin Huang, Huaping Hu, and Zhiying Wang. A dynamic trust model based on feedback control mechanism for P2P applications. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 312–321, Wuhan, China, sep 2006. Springer.

[6] Fajiang Yu, Huanguo Zhang, Fei Yan, and Song Gao. An improved global trust value computing method in P2P system. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 258–267, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[7] Félix Gómez Mármol and Gregorio Martínez Pérez. TACS - Trust Ant Colony System. http://sourceforge.net/projects/tacs.

[8] Félix Gómez Mármol and Gregorio Martínez Pérez. Providing Trust in Wireless Sensor Networks using a Bio-inspired Technique. In *Proceedings of the Networking and Electronic Commerce Research Conference, NAEC'08*, Lake Garda, Italy, sep 2008.

[9] Félix Gómez Mármol and Gregorio Martínez Pérez. Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique. *Telecommunication Systems Journal*, 46(2), 2010.

[10] Félix Gómez Mármol, Gregorio Martínez Pérez, and Javier Gómez Marín-Blázquez. META-TACS: a Trust Model Demonstration of Robustness through a Genetic Algorithm. *Intelligent Automation and Soft Computing (Autosoft) Journal*, 2010 (in press).

[11] Félix Gómez Mármol, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta. TACS, a Trust Model for P2P Networks. *Wireless Personal Communications, Special Issue on "Information Security and data protection in Future Generation Communication and Networking"*, 51(1):153–164, 2009.

[12] Florina Almenárez, Andrés Marín, Celeste Campo, and Carlos García. PTM: A pervasive trust management model for dynamic open environments. In *Privacy and Trust*, Boston, USA, aug 2004. First Workshop on Pervasive Security and Trust.

[13] Florina Almenárez, Andrés Marín, Daniel Díaz, and Juan Sánchez. Developing a model for trust management in pervasive devices. In *PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops*, page 267, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

[15] Haiguang Chen, Huafeng Wu, Xi Zhou, and Chuanshan Gao. Agent-based Trust Model in Wireless Sensor Networks. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD*, 03:119–124, 2007.

[16] Huirong Tian, Shihong Zou, Wendong Wang, and Shiduan Cheng. A group based reputation system for P2P networks. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 342–351, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[17] Javier Carbó, Jose M. Molina, and Jorge Dávila. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems*, 12:135–155, mar 2003.

[18] Jordi Sabater and Carles Sierra. Social ReGreT, a reputation model based on social relations. *SIGecom Exch*, 3(1):44–56, 2002.

[19] Jordi Sabater and Carles Sierra. REGRET: reputation in gregarious societies. In Jörg P. Müller, Elizabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 194–195, Montreal, Canada, 2001. ACM Press.

[20] Li Xiong and Ling Liu. PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[21] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. A Computational Model of Trust and Reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, page 188, Washington, DC, USA, 2002. IEEE Computer Society.

[22] Maria Moloney and Stefan Weber. A context-aware trust-based security system for ad hoc networks. In *Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 153–160, Athens, Greece, sep 2005.

[23] Mawloud Omar, Yacine Challal, and Abdelmadjid Bouabdallah. Reliable and fully distributed trust model for mobile ad hoc networks. *Computers and Security*, 28(3-4):199–214, 2009.

[24] Mudhakar Srivatsa, Li Xiong, and Ling Liu. Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 422–431, New York, NY, USA, 2005. ACM.

[25] Pietro Michiardi and Refik Molva. CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, 2002. Kluwer, B.V.

[26] Ruichuan Chen, Xuan Chao, Liyong Tang, Jianbin Hu, and Zhong Chen. CuboidTrust: A Global Reputation-Based Trust Model in Peer-to-Peer Networks. In *Autonomic and Trusted Computing*, number 4610 in LNCS, pages 203–215, Hong Kong, China, jul 2007. 4th International Conference, ATC 2007, Springer.

[27] Runfang Zhou and Kai Hwang. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *Transactions on Parallel and Distributed Systems*, 18(4):460–473, 2007.

[28] Sarvapali D. Ramchurn, Carles Sierra, Lluis Godo, and Nicholas R. Jennings. A Computational Trust Model for Multi-Agent Interactions based on Confidence and Reputation. In *6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, 2003.

[29] Sanjay Kumar Dhurandher, Sudip Misra, Mohammad S. Obaidat, and Nidhi Gupta. An ant colony optimization approach for reputation and quality-of-service-based security in wireless sensor networks. *Security and Communication Networks*, 2(2):215–224, 2009.

[30] Saurabh Ganeriwal and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 66–77, New York, NY, USA, 2004. ACM.

[31] Sepandar D. Kamvar, Mario T. Schlosser, and Héctor Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the International World Wide Web Conference (WWW)*, Budapest, Hungary, may 2003.

[32] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, jun 2002. IEEE.

[33] Sonja Buchegger and Jean Y. Le Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge MA, USA, jun 2004.

[34] Suphithat Songsiri. MTrust: A reputation-based trust model for a mobile agent system. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 374–385, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[35] Tae Kyung Kim and Hee Suk Seo. A trust model using fuzzy logic in wireless sensor network. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 32, pages 69–72, aug 2008.

[36] Tang Zhuo, Lu Zhengding, and Li Kai. Time-based dynamic trust model using ant colony algorithm. *Wuhan University Journal of Natural Sciences*, 11(6):1462–1466, 2006.

[37] Wei Wang, Guosun Zeng, and Lulai Yuan. Ant-based reputation evidence distribution in P2P networks. In *GCC*, pages 129–132, Changsha, Hunan, China, oct 2006. Fifth International Conference on Grid and Cooperative Computing, IEEE Computer Society.

[38] Yong Wang, Vinny Cahill, Elizabeth Gray, Colin Harris, and Lejian Liao. Bayesian network based trust management. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 246–257, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[39] Yu Zhang, Huajun Chen, and Zhaohui Wu. A social network-based trust model for the semantic web. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 183–192, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

[40] Yuan Wang, Ye Tao, Ping Yu, Feng Xu, and Jian Lu. A Trust Evolution Model for P2P Networks. In *Autonomic and Trusted Computing*, number 4610 in LNCS, pages 216–225, Hong Kong, China, jul 2007. 4th International Conference, ATC 2007, Springer.

[41] Yun Tang, Lifeng Sun, Jianguang Luo, Shiqiang Yang, and Yuzhuo Zhong. TPOD: A trust-based incentive mechanism for peer-to-peer live broadcasting. In *Autonomic and Trusted Computing*, number 4158 in LNCS, pages 332–341, Wuhan, China, sep 2006. Third International Conference, ATC 2006, Springer.

II Networks simulators

[42] Anelia Mitseva, Efthimia Aivaloglou, Maria Marchitti, Neeli Rashmi Prasad, Charalabos Skianis, Stefanos Gritzalis, Adrian Waller, Tim Baugé, and Sarah Pennington. Towards adaptive security for convergent wireless sensor networks in beyond 3g environments. *Wireless Communications and Mobile Computing*, 2008.

[43] Ben L. Titzer, Daniel K. Lee, and Jens Palsberg. Avrora: scalable sensor network simulation with precise timing. In *Fourth International Symposium on Information Processing in Sensor Networks, IPSN*, pages 477–482, 2005.

[44] Jonathan Polley, Dionysys Blazakis, Jonathan Mcgee, Dan Rusk, and John S. Baras. Atemu: A fine-grained sensor network simulator. In *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004.

[45] Jordi Pujol, Rubén Mondéjar, Helio Tejedor, Marc Sánchez, Pedro García, Carles Pairot. PlanetSim: An Overlay Network Simulation Framework. http://planet.urv.es/planetsim.

[46] Stephen Naicken, Anirban Basu, Barnaby Livingston, and Sethalat Rodhetbhai. A survey of peer-to-peer network simulators. *Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06)*, 2006.

[47] Thomer M. Gil, Frans Kaashoek, Jinyang Li, Robert Morris, Jeremy Stribling. P2PSim: A Simulator for Peer-to-Peer (P2P) Protocols. http://pdos.csail.mit.edu/p2psim.

II.1 Trust and reputation models simulators

[48] Efthimia Aivaloglou, Stefanos Gritzalis, and Charalabos Skianis. Towards a flexible trust establishment framework for sensor networks. *Telecommunication Systems: Modeling, Analysis, Design and Management*, 35(3-4):207–213, 2007.

[49] Efthimia Aivaloglou, Stefanos Gritzalis, and Charalabos Skianis. Trust establishment in sensor networks: behaviour-based, certificate-based and a combinational approach. *International Journal of System of Systems Engineering*, 1(1-2):128–148, 2008.

[50] Félix Gómez Mármol and Gregorio Martínez Pérez. SourceForge.net: TRMSim-WSN. http://sourceforge.net/projects/trmsim wsn.

[51] Félix Gómez Mármol and Gregorio Martínez Pérez. TRMSim-WSN, a Trust & Reputation Models Simulator for Wireless Sensor Networks. http://ants.dif.um.es/~felixgm/research/trmsim wsn.

[52] Félix Gómez Mármol and Gregorio Martínez Pérez. TRMSim-WSN, Trust and Reputation Models Simulator for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications (IEEE ICC 2009), Communication and Information Systems Security Symposium*, Dresden, Germany, jun 2009.

[53] Karen K. Fullam, Tomas Klos, Guillaume Muller, Jordi Sabater-Mir, K.Suzanne Barber, and Laurent Vercouter. The Agent Reputation and Trust (ART) Testbed. In *Trust Management*, number 3986 in LNCS, pages 439–442, Pisa, Italy, may 2006. Fourth International Conference, iTrust 2006, Springer.

[54] Seamus Moloney. Simulation of a distributed recommendation system for pervasive networks. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1577–1581, 2005.

[55] Yan Zhang, Wei Wang, and Shunying Lü. Simulating trust overlay in p2p networks. In *International Conference on Computational Science (1)*, pages 632–639, 2007.

III Trust and reputation management

[56] Avinash Srinivasan, Joshua Teitelbaum, Huigang Liang, Jie Wu, and Mihaela Cardei. *On Trust Establishment in Mobile Ad-Hoc Networks*, chapter Reputation and Trust-based Systems for Ad Hoc and Sensor Networks. John Wiley & Sons Ltd., 2007.

[57] Adolfo Morrone, Noemi Tontoranelli, and Giulia Ranuzzi. How good is trust?: Measuring trust and its role for the progress of societies. OECD Statistics Working Papers 2009/3, OECD, Statistics Directorate, 2009.

[58] Alfarez Abdul-Rahman and Stephen Hailes. Supporting Trust in Virtual Communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Hawaii, USA, 2000.

[59] Audun Josang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[60] Diego Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Published Online, 2000.

[61] Félix Gómez Mármol and Gregorio Martínez Pérez. *Handbook of Peer-to-Peer Networking*, chapter State of the art in trust and reputation models in P2P networks. Springer, pages 761–784, 2010.

[62] Félix Gómez Mármol and Gregorio Martínez Pérez. Towards Pre-Standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems. *Computer Standards & Interfaces*, 2010.

[63] Félix Gómez Mármol and Gregorio Martínez Pérez. Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. *Elsevier Computers & Security*, 28(7):545–556, 2009.

[64] George Metakides et al. Trust in the Information Society. A report of the advisory board RISEPTIS, Advisory Board of Research and Innovation for Security, Privacy and Trustworthiness in the Information Society, 2010.

[65] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[66] Karl Aberer and Zoran Despotovic. Managing Trust in a Peer-2-Peer Information System. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 310–317. ACM Press, 2001.

[67] M. Carmen Fernandez-Gago, Rodrigo Roman, and Javier Lopez. A survey on the applicability of trust management systems for wireless sensor networks. In *International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pages 25–30, 2007.

[68] Rodrigo Roman, M. Carmen Fernandez-Gago, and Javier Lopez. Featuring trust and reputation management systems for constrained hardware devices. In *Autonomics '07: Proceedings of the 1st international conference on Autonomic computing and communication systems*, pages 1–6, Rome, Italy, 2007.

[69] Sergio Marti and Hector Garcia-Molina. Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484, mar 2006.

[70] Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 393–402, New York, NY, USA, 2004.

[71] Stephen Paul Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling, apr 1994.

[72] Yan Sun and Yafei Yang. Trust Establishment in Distributed Networks: Analysis and Modeling. In *Proceedings of the IEEE International Conference on Communications (IEEE ICC 2007), Communication and Information Systems Security Symposium*, Glasgow, Scotland, jun 2007.

[73] Yan Sun, Zhu Han, and K.J.R. Liu. Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine*, 46(2):112–119, feb 2008.

III.1 Trust and reputation EU Projects

[74] Fraunhofer Institute for Intelligent Analysis & Information Systems and Symantec and Tiscali and Nortel Networks and Katholieke Universiteit Leuven. AntiPhish (Anticipatory Learning for Reliable Phishing Prevention). http://www.antiphishresearch.org.

[75] CETIC and Science & Technology Facilities Council and Italian National Research Council and Vrije Universiteit Amsterdam and Interplay Software and Hewlett-Packard and Istituto Geografico DeAgostini and Moviquity. GridTrust. http://www.gridtrust.eu.

[76] Centre for Research & Technology Hellas and University of Basel and FORENAP FRP and Fraunhofer Institute for Industrial Engineering IAO and Institute of Communication & Computer Systems and Multitel ASBL and Siemens VDO Automotive and Starlab Barcelona and University of Stuttgart and Volvo Technology AB and University of Pisa and NETSMART and Telefónica Investigación y Desarrollo. HUMABIO (Human Monitoring and Authentication using Biodynamic Indicators and Behavioral Analysis). http://www.humabio eu.org.

[77] Technikcon and AMD and Comneon and Infineon and IBM and University of Cambrigde and Politecnico di Torino and Technische Universitat Munchen and HP and Technische Universitat Dresden and University of London et all. OpenTC (Open Trusted Computing). http://www.opentc.net.

[78] IBM Belgium and Karlstads Universitet and Universiteit van Tilburg and IBM Zürich Research Laboratory and Hewlett-Packard and Technische Universitat Dresden and Institut EURECOM and T-Mobile and Johann Wolfgang Goethe-Universitat Frankfurt am Main and Deutsche Lufthansa and Unabhangiges Landeszentrum fur Datenschutz and Katholieke Universiteit Leuven et all. PRIME (Privacy and Identity Management for Europe). https://www.prime project.eu.

[79] Vitali Isabella, Consiglio Nazionale Delle Ricerche. REPUTATION (Using trust and reputation to Improve security in virtual societies).

[80] University of Trento and France Telecom Spain and Create-Netand and DoCoMo Euro-Labs and Fast and Kungliga Tekniska Hogskolan and Katholieke Universiteit Leuven and Moviquity and Omnys and Sintef and Trusted Logic and Vrije Universiteit Amsterdam. S3MS (Security of Software and Services for Mobile Systems). http://www.s3ms.org.

[81] Commissariat a l'energie atomique LETI, The University of Surrey, ALMA Consulting Group, Ove Arup & Partners International, Ericsson AB, NEC Europe Ltd., SAP AG, and et all Telefónica Investigación y Desarrollo. SENSEI (Integrating the Physical with the Digital World of the Network of the Future). http://www.ict sensei.org.

[82] Carles Sierrra García, Consejo Superior de Investigaciones Científicas. SOCIALREP (Toward the next generation of computational trust and reputation models).

[83] Fraunhofer SIT and Alcatel-Lucent and Deutsche Telekom and Dracotic and IT Aveiro and NEC and Portugal Telecom and University of Murcia and University of Stuttgart. SWIFT (Secure Widespread Identities for Federated Telecommunications). http://www.ist swift.org.

[84] Cahill Vinny, Trinity College Dublin. TRUSTREP (Creation and use of trust in virtual communities through reputation Management).

IV Bio-inspired algorithms

[85] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[86] Fred W. Glover and Gary A. Kochenberger. *Handbook of Metaheuristics (International Series in Operations Research & Management Science)*. Springer, jan 2003.

[87] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[88] Larry Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 265–283, San Mateo, CA, 1991. Morgan Kaufman.

IV.1 Ant colony systems

[89] Luca Maria Gambardella and Marco Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. In *International Conference on Evolutionary Computation*, pages 622–627, 1996.

[90] Luca Maria Gambardella, Eric D. Taillard, and Marco Dorigo. Ant Colonies for the QAP. *Journal of the Operational Research Society*, 50:167–176, 1999.

[91] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach in the traveling salesman problem. *IEEE Transaction on Evolutionary Computing*, 1(1):53–66, 1997.

[92] Marco Dorigo, Luca Maria Gambardella, Mauro Birattari, Alcherio Martinoli, Riccardo Poli, and Thomas Stützle. *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006*, volume 4150 of *LNCS*. Springer, Brussels, Belgium, 2006.

[93] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.

[94] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Book, 2004.

[95] Oscar Cordón, Francisco Herrera, and Thomas Stützle. A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware and Soft Computing*, 9(2-3):141–175, 2002.

[96] Oscar Cordón, Iñaki Fernandez de Viana, and Francisco Herrera. Analysis of the best-worst Ant System and its variants on the TSP. *Mathware and Soft Computing*, 9(2-3):177–192, 2002.

[97] Thomas Stützle and Holger H. Hoos. MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8):889–914, 2000.

V Wireless sensor networks

[98] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22, jun 2007.

[99] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, Aug 2002.

[100] Jennifer Yicka, Biswanath Mukherjeea, and Dipak Ghosal. A survey on sensor networks. *Computer Networks*, 52(12):2292–2330, Aug 2008.

[101] Jens-Matthias Bohli, Christoph Sorge, and Dirk Westhoff. Initial observations on economics, pricing, and penetration of the internet of things market. *ACM SIGCOMM Computer Communication Review*, 39(2):50–55, 2009.

[102] Juan A. Sánchez and Pedro M. Ruiz. Improving Delivery Ratio and Power Efficiency in Unicast Geographic Routing with a Realistic Physical Layer for Wireless Sensor Networks. In *Proc. 9th Euromicro Conference on Digital System Design (DSD '06)*, pages 591–597, September 2006.

[103] Kay Römer and Friedemann Mattern. The Design Space of Wireless Sensor Networks. *IEEE Wireless Communications*, 11(6):54–61, dec 2004.

[104] Li Li and Joseph Y. Halpern. Minimum-energy mobile wireless networks revisited. In *IEEE International Conference on Communications, 2001. ICC 2001*, volume 1, pages 278–283, 2001.

[105] Mirko Presser, Payam M. Barnaghi, Markus Eurich, and Claudia Villalonga. The sensei project: integrating the physical world with the digital world of the network of the future. *Communications Magazine, IEEE*, 47(4):1–4, April 2009.

VI Identity Management Systems

[106] Gabriel Lopez, Oscar Canovas, Antonio F. Gomez-Skarmeta, and Joao Girao. A swift take on identity management. *Computer*, 42:58–65, 2009.

[107] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *CCS'02: Proceedings of the 9th ACM conference on Computer and communications security*, New York, NY, USA, pages 21–30, 2002.

[108] Joao Girao, Amardeo Sarma, and Rui Aguiar. Virtual Identities - A Cross Layer approach to Identity and Identity Management. In *Proceedings for the 17th Wireless World Reseach Forum, Heidelberg, Germany*, nov 2006.

[109] John R. Douceur and Judith S. Donath. The sybil attack. In *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pages 251–260, 2002.

[110] Sergio Marti and Hector Garcia-Molina. Identity crisis: anonymity vs reputation in P2P systems. In *Proceedings for the Third International Conference on Peer-to-Peer Computing (P2P 2003)*, pages 134–141, Linköping, Sweden, sep 2003.

VII Cryptography and cryptoschemes

[111] Einar Mykletun, Joao Girao, and Dirk Westhoff. Public key based cryptoschemes for data concealment in wireless sensor networks. In *IEEE International Conference on Communications*, Istanbul, Turkey, jun 2006. ICC2006.

[112] Josep Domingo-Ferrer. A Provably Secure Additive and Multiplicative Privacy Homomorphism. *Information Security Conference*, pages 471–483, 2002.

[113] Josep Domingo-Ferrer, Alexandre Viejo, Francesc Sebe, and Ursula Gonzalez Nicolas. Privacy homomorphisms for social networks with private relationships. *Computer Networks*, 52(15):3007–3016, October 2008.

[114] James M. Adler, Wei Dai, Richard L. Green, and C. Andrew Neff. Computational Details of the VoteHere Homomorphic Election System. *ASIACRYPT*, 2000.

[115] Jun Furukawa and Hideki Imai. An efficient group signature scheme from bilinear maps. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(5):1328–1338, 2006.

[116] Kun Peng, Riza Aditya, Colin Boyd, Ed Dawson, and Byoungcheon Lee. Multiplicative homomorphic e-voting. In *INDOCRYPT*, pages 61–72, 2004.

[117] Lidong Chen and Torben P. Pedersen. New group signature schemes. In *Advances in Cryptology, Eurocrypt'94*, pages 171–181. LNCS, 1995.

[118] Sergei Evdokimov, Matthias Fischmann, and Oliver Gunther. Provable security for outsourcing database operations. *Data Engineering, International Conference on*, 0:117, 2006.

[119] Taher El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *CRYPTO*, IT-31(4):469–472, jul 1985.

VIII Miscelaneous

[120] Nakhjiri Madjid and Nakhjiri Mahsa. *AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility*. John Wiley & Sons, Ltd, sep 2005.

[121] William Stallings. *Data and Computer Communications (7th ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[122] Yoav Shoham. Agent-Oriented Programming. *Artificial Intelligence*, 60(1):51–92, 1993.