Trust and reputation models comparison
Félix Gómez Mármol
NEC Europe Ltd, Heidelberg, Germany, and
Gregorio Martínez Pérez
Departamento de Ingeniería de la Información y las Comunicaciones,
University of Murcia, Murcia, Spain

Abstract
Purpose – The purpose of this paper is to analyse and describe several trust and reputation models for distributed and heterogeneous networks and compare some of them in order to provide an evaluation amongst some of the most relevant works in this field.

Design/methodology/approach – The authors have developed a trust and reputation models simulator for wireless sensor networks, called TRMSim-WSN, and implemented several trust models for distributed networks in order to test their accuracy as well as their resilience against a set of specific security threats that can be applied in these particular systems, as the paper will show.

Findings – The analysis of the outcomes obtained from the experiments revealed that while some models have a reasonably good performance against certain security threats, none of them behaves as would be desired under any circumstances.

Research limitations/implications – Ongoing work is focused on the implementation of several trust and reputation models in the simulator TRMSim-WSN, in order to have a wider range of possibilities for comparison. Furthermore, the authors are planning to include additional security threats that allow the testing of those models under new undesirable situations.

Practical implications – The experiments show that when deciding which trust and/or reputation model is more suitable or adequate to be applied, it is crucial to study and analyse the specific features of the distributed network where such model is to be deployed, as well as the possible security threats that can spoil its accuracy.

Originality/value – As far as is known, this is one of the few works in the field of trust and reputation in distributed systems where no new model is presented, but a comparison and analysis of some of the current most representative ones is carried out.

Keywords Trust, Modelling, Distributed databases, Computer networks

Paper type Research paper

Introduction and motivation
Trust and reputation management has arisen in the last few years as an innovative strategy in order to deal with the lack of centralised entities in purely and highly distributed environments providing reliable information regarding the actual behaviour of every member in the system (Josang et al., 2007; Sabater and Sierra, 2005).

In fact, several trust and/or reputation management proposals have been developed, applying a wide variety of different mechanisms, techniques or approaches,

This work has been supported by a Séneca Foundation grant within the Human Resources Research Training Program 2007 (code 15779/PD/10). Thanks also to the Funding Program for Research Groups of Excellence granted as well by the Séneca Foundation with code 04552/GERM/06.
constituting thus a research field which has captured the attention of a number of scientific groups in both industry and academia.

Nevertheless, although there are a great number of different proposals, as far as we know there are only a few works where no new approach is presented, but a comparison among the most representative ones is carried out, describing their main characteristics and analysing their strengths and weaknesses.

Therefore, this paper presents some of the most relevant trust and reputation models found in the literature. An implementation of those models has been done and included in a generic trust and reputation models simulator, called TRMSim-WSN (Gómez and Martínez, 2009a). This framework has allowed us to test those models and provide some results regarding their accuracy as well as their resilience against certain security threats specifically applicable to those systems.

The rest of the paper is organized as follows: the next section presents a short survey of some of the most relevant and innovative current trust and/or reputation models for distributed systems. In the following section we will describe in detail the selected approaches to be compared. Such comparison will be actually held in the penultimate section, describing the experiment settings and analysing the outcomes achieved. The final section concludes this work and presents some future research lines.

Related work
Nowadays, trust and reputation management is widely considered and accepted as an efficient solution for several environments where there is a lack of information about the entities composing a system and interacting among them (Marsh, 1994; Gómez and Martínez, 2009b; Mui, 2002). By using these mechanisms, an entity is able to decide which other entity to have an interaction with, according to the global reputation given to the latter, the direct trust given by the former, or a combination of both.

In this sense, on the one hand some works related to the analysis of trust and reputation systems have been carried out (Sun and Yang, 2007; Lam and Riedl, 2004; Gómez and Martínez, 2009c; Marti and Garcia-Molina, 2006). And even some others related to simulation tools for those systems (Moloney, 2005; Gómez and Martínez, 2009a).

On the other hand, many researchers have focused their efforts in providing new trust and/or reputation models in the last decades. We have surveyed the related literature and have realised that most of those authors just concentrate on describing their approaches. Some of them present a number of experiments in order to prove the accuracy of their proposals under certain conditions or circumstances (Chen et al., 2008; Wang et al., 2007; Kamvar et al., 2003; Gómez and Martínez, 2010; Gui et al., 2007; Gui et al., 2008; Xiong and Liu, 2004). And only a few ones additionally present a comparison of their models with some other works (Chen et al., 2007; Zhou and Hwang, 2007). There are even a number of them that do not carry out any set of experiments (Azzedin et al., 2007; Abdul-Rahman and Hailes, 2000).

Nevertheless, we have not found any paper where an implementation of several trust and/or reputation models as well as an exhaustive comparison amongst them is provided, without presenting a new proposed model (see Table I). Therefore, to our best knowledge, this is one of the first posing such a work.
<table>
<thead>
<tr>
<th>Category</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust and reputation management theory</td>
<td>Marsh (1994)</td>
<td>One of the first works in this field. A PhD thesis establishing some the bases of trust and reputation management</td>
</tr>
<tr>
<td></td>
<td>Gómez and Martinez (2009b)</td>
<td>This work surveys the state of the art regarding trust and reputation management in P2P networks.</td>
</tr>
<tr>
<td></td>
<td>Mui (2002)</td>
<td>Another early work in this field working for multi-agents systems</td>
</tr>
<tr>
<td></td>
<td>Sun and Yang (2007)</td>
<td>An interesting work formalising some aspects of trust and reputation management</td>
</tr>
<tr>
<td></td>
<td>Lam and Riedl (2004)</td>
<td>This work describes some vulnerabilities intrinsically related to recommenders systems</td>
</tr>
<tr>
<td></td>
<td>Gómez and Martinez (2009a)</td>
<td>This paper presents those security threats specifically applicable in trust and reputation systems, as well as some recommendations to tackle them</td>
</tr>
<tr>
<td></td>
<td>Martí and Garcia-Molina (2006)</td>
<td>This work describes the generic steps to be followed by a reputation mechanism in P2P networks</td>
</tr>
<tr>
<td></td>
<td>Moloney (2005)</td>
<td>This paper presents some simulations of a simple distributed recommendation system for pervasive networks</td>
</tr>
<tr>
<td></td>
<td>Gómez and Martinez (2009a)</td>
<td>This work presents a reputation model for P2P networks</td>
</tr>
<tr>
<td></td>
<td>Chen et al. (2008)</td>
<td>This work proposes a trust model for P2P networks taking into account direct experiences as well as context</td>
</tr>
<tr>
<td></td>
<td>Wang et al. (2007)</td>
<td>This work shows TRMSim-WSN, an open-source trust and reputation models simulation framework</td>
</tr>
<tr>
<td></td>
<td>Kamvar et al. (2003)</td>
<td>This work presents EigenTrust, probably the most cited model (around 1,870 cites) in this field. It is aimed to work in P2P networks</td>
</tr>
<tr>
<td></td>
<td>Gómez and Martinez (2010)</td>
<td>BTRM-WSN is shown in this paper, one of the few novel works applying an ant colony system in order to manage trust and reputation in wireless sensor networks</td>
</tr>
<tr>
<td></td>
<td>Gui et al. (2007)</td>
<td>This work presents a reputation aggregation mechanism by means of fuzzy sets and fuzzy logic</td>
</tr>
<tr>
<td></td>
<td>Gui et al. (2008)</td>
<td>This work describes a method for filtering dishonest feedbacks based on Dempster-Shafer theory</td>
</tr>
<tr>
<td></td>
<td>Xiong and Liu (2004)</td>
<td>PeerTrust is presented here. It is also one of the most cited works (around 814 cites) in the field of trust and reputation management in P2P networks</td>
</tr>
<tr>
<td></td>
<td>Chen et al. (2007)</td>
<td>This work shows CuboidTrust, a reputation-based trust model for P2P networks</td>
</tr>
<tr>
<td></td>
<td>Zhou and Hwang (2007)</td>
<td>PowerTrust, presented in this paper, might be considered as an enhancement of EigenTrust. It has also received much attention from the international research community (around 174 cites)</td>
</tr>
<tr>
<td></td>
<td>Azzedin et al. (2007)</td>
<td>This work makes use of fuzzy reasoning to deal with uncertain information regarding others’ trustworthiness in P2P networks</td>
</tr>
<tr>
<td></td>
<td>Abdul-Rahman and Hailes (2000)</td>
<td>This is also one of the first works in this field, describing some generic properties of trust and reputation systems</td>
</tr>
</tbody>
</table>

Table I. Review of existing trust and reputation model literature
Compared trust and reputation models

In this section we will describe every trust and reputation model we have selected to be implemented and compared. Some of them have been mainly chosen because of their novelty or goodness, and other ones because of their relevance and importance in this research field. As shown in Table I, apart from our previous novel work, BTRM-WSN, the other three models constitute an important reference for other researchers within this specific field.

BTRM-WSN (Gómez and Martínez, 2010) is a bio-inspired trust and reputation model for wireless sensor networks, whose novelty is the accurate application of an ant colony system (Dorigo et al., 2006) in order to help a node to find the most trustworthy sensor providing a certain service, and to reach such sensor through the most reputable path. To do so, ants are sent and spread throughout the network and, while they are exploring it, they also leave some pheromone traces that in turn will help forthcoming ants to follow the appropriate route.

Therefore, in this model every sensor stores a pheromone trace value for each one of its neighbours. These pheromone traces \( \tau \in [0, 1] \), will determine the probability of ants choosing a certain path, and can be seen as the amount of trust given by a sensor to another one. Additionally, the heuristic values \( \eta \in [0, 1] \) are defined as the inverse of the distance between every pair of sensors. Both elements together, pheromone traces and heuristic values, take part in the transition rule which decides the next sensor an ant will move forward.

Once the launched ants (or a subset of them) return to the source (i.e. the sensor applying BTRM-WSN to find the most trustworthy service provider), it is necessary to evaluate the quality of every found path. The metric used to assess such adequacy, \( Q(S_k) \), takes into account the average pheromone of the path found by ant \( k \), the length of such path as well as the percentage of ants that have selected the same solution as ant \( k \). Hence, the path with highest quality (which means the shortest one with highest pheromone traces) is selected.

Furthermore, every time an ant crosses a sensor, it modifies the pheromone trace associated to the link connecting the sensors it is travelling through so that links with lower pheromone traces and higher heuristic values receive a greater reinforcement, while those with high pheromone traces and low heuristic values have a lower increase. \( \varphi \in [0, 1] \) is a parameter used to adjust and control this pheromone local updating. Additionally, each edge of the best path found receives an extra pheromone reinforcement which is directly proportional to the quality of the best path, the heuristic value, the previous pheromone trace, and the value of a controlling parameter \( \rho \in [0, 1] \).

Finally, once the most trustworthy service provider has been selected and the transaction has been performed, according to the satisfaction of the client with the received service, a punishment or reward step takes place. This punishment or reward is done again in terms of pheromone evaporation or contribution, respectively, along the path leading to the selected service provider.

Ants’ ability to find alternative paths when the current one is no longer promising makes BTRM-WSN especially resilient against topology changes in the network as well as against nodes behavioural oscillations.

**EigenTrust**

EigenTrust (Kamvar et al., 2003) is a trust and reputation model for P2P networks where each peer is assigned a global trust value, based on that peer’s history of...
transactions. It has been proved to achieve a significant reduction of fraudulent transactions with a moderate overhead in the system.

A very important key of this proposal is the assumption of the presence of some special peers that are pre-trusted and, moreover, help the model to have a rapid convergence and to break up malicious collectives. Hence, an accurate and wise selection of those pre-trusted peers is crucial for the correct behaviour of EigenTrust. Authors propose to select the peers designer of the network (the very first who formed it) as pre-trusted ones.

Each peer $i$ in EigenTrust stores a global trust vector at time $k$, $\tilde{t}_i^{(k)}$, which in turn contains a global trust value for every other node in the network. This global trust vector is computed as follows:

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

where $a \in [0, 1]$ is a constant used to weight the influence of pre-trusted peers in the global trust computation, $\tilde{p}$ is a distribution over pre-trusted peers (being $P$ the set of pre-trusted peers, $p_i = 1/P$ if peer $i \in P$, and $p_i = 0$ otherwise) and $C^T$ is the transpose of matrix $[c_{ij}]$.

The normalised local trust value $c_{ij} \in [0, 1]$ represents the percentage of trust deposited on peer $j$ by peer $i$, based on the difference between satisfactory and unsatisfactory transactions of peer $i$ with peer $j$. However, if peer $i$ does not know anybody, or does not trust anybody, it will choose to trust pre-trusted peers.

The computation iteration of $t_i^{(k+1)}$ is repeated until it converges to the left principal eigenvector of matrix $C$. That is, every node in the network will reach the same values for their respective global trust vectors by applying the notion of transitive trust.

Once a node reaches this global trust vector, EigenTrust provides two mechanisms to select the peer to interact with. The first one, called deterministic, simply chooses the peer with highest trust value $t_{\text{max}}$. On the other hand, an improved method called probabilistic chooses peer $i$ as service provider with probability $\frac{t_i}{\sum t'_j}$ and with probability of 10 per cent selects a peer $j$ that has a trust value $t_j = 0$.

Finally, EigenTrust is resilient against a set of security threats (Gómez and Martínez, 2009c) such as different types of malicious collusions or Sybil attack (Douceur, 2002), for instance.

PeerTrust

PeerTrust (Xiong and Liu, 2004) is a reputation-based trust model aimed to work in P2P networks, where several factors influence in the computation of a trust value for each peer. Authors describe two strategies for implementing each of the two proposed basic trust metrics.

The general trust metric for peer $u$, $T(u)$, is defined as:

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

where $\alpha$ and $\beta$ denote the normalised weight factors for the feedback-based evaluation and the community context factor according to different situations, respectively; $I(u)$ is 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 $i$th transaction; $S(u, i)$ represents the normalised amount of satisfaction peer $u$ receives from peer $p(u, i)$ in its $i$th transaction; $Cr(v)$ denotes the credibility of the feedback submitted by peer $v$; $TF(u, i)$ is the adaptive transaction context factor for peer $u$’s $i$th transaction; and $CF(u)$ represents the adaptive community context factor for peer $u$.

The last element ($CF(u)$) might be used to encourage peers to provide ratings by defining it as $F(u)/I(u)$, where $F(u)$ is the total number of feedbacks peer $u$ gives to others. The transaction factor $TF(u, i)$ may reflect the size, importance or cost, for instance, of a transaction, so some specific transactions can have a higher weight in the global trust computation.

Regarding the credibility $Cr(v)$, authors of PeerTrust propose two different metrics. The first one, called Trust Value Metric (TVM), is directly proportional to the general trust metric $T(u)$, so the more trustworthy a peer is provisioning a service, the more credible it is as well for providing recommendations.

On the other hand, the second metric, called Peer Similarity Metric (PSM), uses the similarity between the feedbacks provided by two different nodes in order to effectively filter out dishonest recommendations, since benevolent peers will usually provide similar feedbacks to similar services or service providers.

Both metrics can be implemented under two different strategies. One is called dynamic trust computation (DTC), which uses fresh trust data collected at runtime to compute the trust value. The other is called approximate trust computation (ATC), which uses a cache in order to speed up the trust computation process.

The computed trust values in PeerTrust can be used either to select the most trustworthy node (i.e., the one with a highest trust value), or to help a node $u$ to determine whether to interact with another peer $w$ or not. In the latter case, a simple rule for peer $u$ to have a transaction with peer $w$ can be $T(u) > T_{\text{threshold}}(w)$, where $T_{\text{threshold}}(w)$ is the threshold trust value for peer $w$ to trust another peer.

Finally, in order to address the potential dynamic behaviour of peers, authors propose a simple adaptive time windows-based algorithm which reduces the time window to reflect the peer’s most recent behaviour when the peer is dropping its performance over a threshold.

**PowerTrust**

The main innovation of PowerTrust (Zhou and Hwang, 2007), a novel reputation model for P2P networks, relies on considering the distribution of peers feedbacks in such environments. Thus, this reputation mechanism is the first one which effectively and accurately takes advantage of that fact. Authors studied the eBay transaction trace over 10,000 users and discovered that feedbacks in those systems followed a power-law distribution, i.e., the node with a few feedbacks is common, whereas the node with a large number of feedbacks is extremely rare.

Therefore, PowerTrust leverages on those nodes with a higher amount of feedbacks, called power nodes, to aggregate users feedbacks and compute the global reputation scores $v_i \in [0, 1]$ owned by every peer $i$. Furthermore, the set of power nodes is updated dynamically after each round of aggregation as the set of the current $m$ most reputable nodes (the ones with highest reputation scores).

And in order to compute the reputation score $v_i$ of a node $i$, it first collects all the reputation scores $v_j$ and the normalised local trust scores $r_{ji} \in [0, 1]$ from those nodes $j$ who have interacted with $i$ in the past. On the one hand, $r_{ji}$ is defined as follows:
where $s_{ij}$ represents the satisfaction of node $i$ about the last interaction with node $j$; in other words, the most recent feedback that node $i$ rates node $j$. On the other hand, the aggregation needed to calculate $v_i$, in case $i$ is considered as a power node, follows the next expression:

$$v_i = (1 - \alpha) \cdot \sum_j (v_j \times r_{ji}) + \alpha / m$$

being $\alpha$ the greedy factor which determines the weight of power nodes. Otherwise, i.e., if $i$ is not a power node, then its global reputation score is computed as:

$$v_i = (1 - \alpha) \cdot \sum_j (v_j \times r_{ji}).$$

In summary, each node is rated with a global reputation score aggregated from local trust scores weighted by the global reputations of all the other nodes who have had an interaction with the former. Additionally, power nodes receive an extra reinforcement.

PowerTrust achieves high aggregation speed and accuracy, robustness to resist malicious peers, and high scalability to support large-scale P2P applications. It is specifically resilient against dynamic behaviour of peers due to the replaceable set of power nodes.

**Comparison**

**Framework for comparison: TRMSim-WSN**

The framework we selected in order to perform our experiments was TRMSim-WSN (Gómez and Martínez, 2009a), an open-source Java-based Trust and Reputation models for distributed networks simulator. The easiness that such a tool provides when developing and incorporating a new trust and reputation model allowed us to carry out the comparison between the selected models without too much effort. Its specific orientation to trust and reputation models simulation, as well as the implementation of several security threats were also two important reasons for choosing it as the selected comparison framework.

Moreover, TRMSim-WSN provides a generic API composed of five steps where almost every trust and reputation model might fit appropriately. Those steps are depicted in Figure 1.

As it can be observed, there is a first stage where behavioural information about the members of the community is collected. Second, that information is used to provide a score that will determine the reputation and/or trustworthiness of every user (or a subset of them) in the system. Then, the most trustworthy and/or reputable entity is generally selected and a transaction is performed with it, evaluating next the satisfaction of the requester with the received service. According to that satisfaction, a last step of punish or reward takes place, modifying the previous given score to the selected party.
Experiments description
All the experiments carried out consisted of 100 wireless sensor networks whose nodes were randomly deployed over a plane of 100 units$^2$. Of the nodes, 15 per cent acted as clients, requesting 100 times a certain service and applying a specific trust and/or reputation model in order to find the best server to interact with. From the 85 per cent left, a 5 per cent of nodes did not offer any service, acting therefore just as relay nodes.

These percentages were selected on the basis that the accuracy of a model is fully demonstrated when the number of clients providing recommendations (and hence, the information available in the system) is much lower than the number of servers (either cheating or behaving properly).

Since we expected every sensor to have more or less the same number of neighbours regardless the size of the network, their radius range was computed as:

$$l = \frac{n}{100^2} \cdot \pi \cdot R^2 \Rightarrow R = 100 \sqrt{\frac{l}{n \cdot \pi}}$$

where $l$ is the average number of links (equivalently, the number of neighbours) a sensor has, $n$ is the total number of sensors composing the network and $R$ is the radius of every sensor. We chose a value of $l = 5$ for our experiments, i.e. every sensor had, on average term, five direct neighbours.

Those are the common settings shared by all the experiments. Specifically, we designed and carried out four different experiments called “static”, “dynamic”, “oscillating” and “collusion”.

In the first experiment, static, neither the topology of the networks, nor the goodness of the sensors varied along the time, so they both remained immutable. In this case, we tested each one of the selected trust and/or reputation models with networks composed by 100, 200, 300 and 400 sensors, respectively, following the default configuration described above. The percentage of servers (those sensors offering the requested service) who were actually malicious was of the 70 per cent (in global terms, a 56.525 per cent = 85 per cent of servers * 95 per cent of servers providing the requested service * 70 per cent of servers providing the requested service and behaving maliciously). We considered that a lower value would not describe appropriately the accuracy of each model.
The rest of the experiments took wireless sensor networks composed by 100 sensors and increased the percentage of malicious servers from 20 per cent to a maximum of 80 per cent, maintaining all the other settings as mentioned before.

The second experiment, the dynamic one, consisted of a network where a sensor could swap into an idle state (if it received too many requests) and wake up later, modifying dynamically the topology of the network. In a WSN, the energy consumption is a critical issue that should not be undervalued. Thus, by switching off for a while, an important energy saving might be achieved.

In the third experiment, called “oscillating”, the dynamism came from the behavioural changes of the servers along the time. From time to time a redistribution of malicious sensors occurs, that is, one sensor can remain with its current benevolence or, on the contrary, it can swap its benevolence and become the opposite it formerly was (if it was benevolent, become malicious and vice versa). In any case, the percentage of malicious nodes remains constant after this behavioural oscillation. It is important to test the resilience of trust and/or reputation models against this type of threats, since it is not realistic to suppose that a sensor’s behaviour will never change in its whole lifetime.

The last experiment carried out, known as “collusion”, consisted of networks where the malicious nodes colluded in order to unfairly praise themselves and, additionally, drive down the reputation of benevolent sensors. This is also a quite common scenario which can be found in this kind of systems, where the more reputable or trustworthy you are, the more probabilities you have to be selected as a service provider.

For each experiment we measured the accuracy of the model (i.e. the percentage of success when selecting an actually benevolent or trustworthy server), as well as the average length of the paths leading to the selected service providers, as an indicative measure of the energy consumption of each model. For highly distributed networks, such as wireless sensor networks, this is a crucial issue, since its nodes are resource-constrained devices with limited capabilities in terms of battery, processing or bandwidth, for instance.

Regarding the parameter values for each one of the tested trust and/or reputation models, Table II shows the ones which were chosen for our experiments (note that we did not adjust those parameters according to the performed experiments, but we consider it might be an interesting future work).

It is important to mention that we have tried to make the parameters values as similar as possible to the ones presented by the authors of their corresponding models. For more information about the meaning of every parameter, please refer to the source article of each model.

Outcomes and results
In this section we will describe and analyse the outcomes got from each performed experiment, extracting as well some interesting ideas from the comparisons carried out.

Experiment 1: static. The outcomes provided by this first experiment can be observed in Figure 2.

It can be checked that the four models have a quite good performance regardless the size of the network. And, although EigenTrust increases its accuracy as the network size also grows (due to a greater number of pre-trusted peers in absolute terms), it is not
appropriate for large wireless sensor networks since it needs every sensor to store the matrix $C$, whose management, when the network size is big enough, becomes completely unfeasible.

This is the easiest environment for a trust and reputation model and all the tested ones have proved to be fully scalable in terms of accuracy, but not in the average length of the path leading to the most reputable server. In this case, BTRM-WSN has a constant path length, whereas in the rest of models it increases with the number of sensors composing the network, which makes BTRM-WSN a more suitable model for this kind of networks since it gets similar performances, with fewer resources necessities.
Experiment 2: dynamic. Figure 3 depicts the outcomes from the second experiment, where the topology of the networks changed dynamically.

As it could be expected, the greater the percentage of malicious nodes is, the worse the results got by every model are. PowerTrust is the one which behaves the best against this kind of situations, since its power nodes are selected dynamically as well. Additionally, in PowerTrust power nodes are the ones with a higher number of feedbacks, whereas in EigenTrust, pre-trusted nodes are the ones which initially formed the network. BTRM-WSN and PeerTrust obtain similar outcomes for each percentage of malicious nodes. For its part, EigenTrust is the more vulnerable to the topology changes, becoming useless (an accuracy less than 50 per cent) when the percentage of malicious nodes approximately exceeds the threshold of 50 per cent. The rest of models, except for PowerTrust, lose their utility when such percentage is greater than a 70 per cent, more or less.

Regarding the path length, the value obtained by EigenTrust, PeerTrust and PowerTrust is nearly independent from the percentage of malicious nodes, while in BTRM-WSN this value is directly proportional to such ratio, besides that it achieves the lowest value in all the cases.

Experiment 3: oscillating. Results of the third experiment, consisting of a network whose servers vary their goodness over time, are exposed in Figure 4.

All the models obtain reasonably good outcomes against this kind of threat, PowerTrust being the one which behaves the best on average terms. Nevertheless, the performance of all those models gets worse as the percentage of malicious nodes increases, especially in the case of BTRM-WSN. This is a very common attack in trust and reputation systems and hence authors have not neglected the response that their models should have against this kind of situations.

In PowerTrust and EigenTrust, the presence of power nodes and pre-trusted nodes, respectively, avoids this threat to success, since those nodes never oscillate their behaviour and therefore they are always available to provide trustworthy services.

Since the good performance of PeerTrust is based on the similarity between the ratings given by the truster and the trustee, and the windows size in our experiments was fixed (instead of varying dynamically), PeerTrust becomes more vulnerable to these behavioural fluctuations.
Regarding the path length, BTRM-WSN again obtains values that increase with the increment of malicious nodes in the network, whereas the rest of models reach similar values than in previous experiments.

Experiment 4: collusion. The outcomes achieved with our last experiment, where a set of malicious nodes collude to unfairly praise themselves and drive down the reputation of actually benevolent nodes, can be observed in Figure 5.

In this case it can be checked that EigenTrust and PowerTrust are the ones whose performance against this kind of threat is better. This is due, as with the previous experiment, to the existence of pre-trusted peers and power nodes, respectively, which are always good candidates to interact with, regardless the collusion formed by other nodes in the network.

BTRM-WSN and PeerTrust, however, achieve a poor accuracy, being PeerTrust ineffective whenever the percentage of malicious servers is greater than or equal to 20 per cent. Some reasons for such a bad performance are the lack of a punishment scheme (which allows, therefore, a malicious server to be selected repeatedly), the deterministic selection of the service provider with a highest score (which allows malicious nodes to deceive benevolent ones, making the latter think the former are trustworthy) and, again, a fixed windows size (which does not allow to adjust the “memory” of a node in order not to quickly forget bad experiences).
In BTRM-WSN, although there is a punishment mechanism, if the pheromone traces are deceptive, ants get the wrong way and travel towards the malicious servers.

Regarding the average length of the paths leading to the selected service providers, all the tested models get nearly constant values regardless the percentage of malicious nodes in the system. However, BTRM-WSN has clearly a better performance than the rest of the models, providing shorter paths under all the situations.

**Practical implications**
As a consequence of the performed experiments, we can also find the following analysis, which describes how each one of the selected trust and/or reputation models manages each one of the steps described previously and shown in Figure 1.

Thus, Table III exposes such analysis, where it can be observed, for instance, that the only model applying a probabilistic entity selection is EigenTrust.

Moreover, BTRM-WSN is the only model from the studied ones that actually has a differentiated step of punishing or rewarding an entity performing a transaction.

The fourth step is essentially the same for every model. The difference lies in how the satisfaction assessment is carried out. That satisfaction value could come as a fuzzy set, a value from a continuous interval, a linguistic label, just a binary value, etc.

Usually the gathering information step is the one who will mostly determine the overhead introduced by each model, so more efforts need to be concentrated on this phase if a lightweight trust and/or reputation model is desired.

Therefore, the decision about which trust and reputation model to apply for a specific scenario should be taken on an exhaustive analysis of the intrinsic properties of such scenario, as well as its design requirements or its feasible threats.

Thus for instance, if we are facing a system close to the definition made in experiment 1 (static networks), any of the proposed models would be convenient for dealing with malicious users, being EigenTrust, however, the less recommendable.

On the other hand, if the topology of the network is highly dynamic (with many nodes leaving and entering the community quite often), then the most suitable trust and reputation mechanism would be PowerTrust.

Nevertheless, if it is the behavioural patterns of the members within the network which are very dynamic (with many nodes swapping from benevolent to malicious and vice versa), then our suggestion would be to deploy BTRM-WSN, provided that the percentage of malicious users is less than or equal to 60 per cent.

Finally, if the malicious entities present in the system (or a subset of them), collude amongst them in order to achieve a higher profit in terms of (fake) recommendations, then using either EigenTrust or PowerTrust would be a wise decision.

**Conclusions and future work**
In this paper we have shown the comparison between some of the most relevant and innovative trust and/or reputation models for distributed networks.

All the studied models (BTRM-WSM, EigenTrust, PeerTrust and PowerTrust) show a reasonably good performance under almost any circumstances (dynamic networks, oscillating behaviour, collusion ...). Nevertheless, some of them proved to be more suitable for certain situations and more resilient to some specific threats. For instance, all the tested models, except for EigenTrust, are immune to the networks size
### Selected trust and/or reputation models

<table>
<thead>
<tr>
<th>Steps</th>
<th>BTRM-WSN</th>
<th>EigenTrust</th>
<th>PeerTrust</th>
<th>PowerTrust</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gather information</strong></td>
<td>Ants explore the network, leaving pheromone traces</td>
<td>Each node builds the matrix ( C = {c_{ij}} )</td>
<td>Each client collects other clients' satisfactions to compute their credibility ( Cr(n) )</td>
<td>Each server ( i ) collects ( r_i ) and ( v_j ) from each interacted client ( j )</td>
</tr>
<tr>
<td><strong>Score and rank</strong></td>
<td>Every path is given a score ( Q(S_k) )</td>
<td>Vector ( \tau_i^{(k)} ) is computed for each node ( i )</td>
<td>Each client computes ( T(u) ) for each reachable server ( u )</td>
<td>Each server ( i ) computes ( v_i )</td>
</tr>
<tr>
<td><strong>Entity selection</strong></td>
<td>The path with highest quality is selected</td>
<td>Probabilistic selection with probability ( \frac{\tau_i^{(k)}}{\sum_j \tau_j^{(k)}} )</td>
<td>Server ( u ) with ( \max_u {T(u)} ) is selected</td>
<td>Server ( k ) with ( \max_k {v_k} ) is selected</td>
</tr>
<tr>
<td><strong>Transaction</strong></td>
<td>The client assesses her satisfaction with the received service</td>
<td>The client assesses her satisfaction with the received service</td>
<td>The client assesses her satisfaction with the received service</td>
<td>The client assesses her satisfaction with the received service</td>
</tr>
<tr>
<td><strong>Punish and reward</strong></td>
<td>Pheromone evaporation</td>
<td>Not applied</td>
<td>Not applied</td>
<td>Not applied</td>
</tr>
</tbody>
</table>

**Table III.** Trust and reputation models parameters values
and they always get quite satisfactory results when those networks are static, i.e. their topology remains invariable over time.

Nodes oscillating behaviours are those which have been dealt more efficiently by all the studied models, obtaining quite good outcomes under almost any circumstances. EigenTrust and PowerTrust are the approaches that better manage those networks where a collusion is formed, although if the percentage of malicious nodes is too high, the models performance worsens.

Regarding the path length, in BTRM-WSN it depends on the percentage of malicious nodes (except for the case of collusion, where it remains nearly constant) and it is independent from the network size. In the rest of the models this parameter depends exclusively on the network size and it is totally independent from the percentage of malicious nodes in the system.

In summary, the importance has been demonstrated of analysing the specific and intrinsic features of the network where a trust and/or reputation model is to be applied, as well as the feasible security threats that could be found on it. This study and analysis will provide us with the information required to choose the more suitable model for a given network.

As a future work, we are planning to develop a controller for each implemented model in order to dynamically and automatically adjust their parameters according to the situation, circumstances and environment. Furthermore, we would like to include additional trust and reputation models in our simulator TRMSim-WSN, and to implement more security threats.

References


Further reading

Corresponding author
Gregorio Martinez Pérez can be contacted at: gregorio@um.es

To purchase reprints of this article please e-mail: reprints@emeraldinsight.com
Or visit our web site for further details: www.emeraldinsight.com/reprints