IJoFCS (2007) 75-86

e-forensic PRESS

The International Journal of FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Group trust yields improved scalability and anomaly detection for P2P systems

Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Júnior, Lorena de Souza Bezerra and Giselle Rosa de Lima

Abstract - This paper implements an existing computational model of trust and reputation applied to a P2P environment, and extends the approach using a novel group trust calculation that demonstrates improved scalability and anomaly detection for P2P systems. Our analysis is based on results obtained by simulating a P2P environment using the JXTA open source platform. A trust and reputation model was implemented in the same platform, allowing to constructing a baseline for the behavior of the nodes using combined trust and reputation coefficients in a scenario without malicious nodes. Then simulations were conducted with malicious nodes and the effect of trust and reputation factors were analyzed regarding their influence on the anomaly detection capacity and scalability in P2P communications. Several simulation scenarios were configured and explored, considering the presence of different number of malicious nodes in the P2P environment, with both constant and variable behavior. Other scenarios included calculations of combined trust and reputation for node groups. The results show that group trust ensure more interactions among nodes, even in the presence of a large number of malicious nodes (60% of the total), besides providing focused identification of malicious nodes inside groups.

Key words: botnets, DNS, denial of service

1 - Introduction

The problem of node authentication is one of the main security issues in peer-to-peer (P2P) systems. With advances in distributed systems technology, thousands of machines may potentially comprise a P2P overlay network, allowing malicious nodes to act among and possibly harm normal nodes. Current research aims to provide secure interactions among nodes through protocols that enable authentication and establish a distributed trust and reputation database [1] [2] [3]. In this paper, we discuss how security issues of P2P systems can be addressed by constructing interaction rules based on trust and reputation. We also show that extending the focus to group trust brings better scalability and anomaly detection to P2P systems.

The test environment used here was configured for simulations of a P2P system

using the JXTA platform [4]. This P2P system is associated with the TRAVOS trust and reputation model, as proposed by [1], and implemented in the same platform. This allows us to construct a baseline for the behavior of the nodes using trust and reputation models that we can then use to explore factors that affect P2P communications. Previous simulations considered factors such as appropriate utilization of trust and reputation concepts, calculation of trust coefficients, network convergence regarding reputation values, system performance, and existence of malicious peers and the initial configuration of node behavior in peer groups.

This paper is organized as follows. In section 2 we present a review of trust and reputation concepts and the model used in our work. In section 3 we discuss the main characteristics of the JXTA platform [4] and consider decisions regarding distributed environments based on this platform as well. In section 4 we propose a group trust approach, and in section 5 we present our implementation and analysis of the results. Finally, we conclude our work in section 6.

2 - Trust and Reputation Concepts

The concept of trust described here refers to an agent making decisions and interacting successfully with other agents, even in a dynamic and uncertain environment [1] [5]. According to Patel [1], the concept of trust in computational relations deals with operations such as the optimized selection of communication partners, the delegation of functions to individuals, and the establishment of agreements between two or more members of the network prior to an interaction or communication.

Gambetta [6] defines trust as a particular level of subjective probability that an agent can evaluate in order to judge if another agent (or group of agents) will perform a particular action. This is computed before the evaluating agent can monitor such an action, such that the results will affect the evaluating agent's decision about performing the action. According to Marsh [5], this definition permits a quantization of trust between zero and one, representing the degree of trust that an agent (truster) can evaluate by directly interacting with another agent (trustee).

The concept of reputation represents a form of indirect trust and involves asking for the opinion of other parties who have previously interacted with the trustee in the past. Patel [1] defines reputation as the common opinion of others regarding an entity, which may be used in the absence of trust formed from direct opinions.

In the trust and reputation approach, the model that will execute the trust management in a virtual system must use both trust and reputation properties and the entire historical context of the environment to be able to perform trust and reputation calculations.

2.1 - Trust in P2P systems

Rowstron and Druschel [3] define peer-to-peer as distributed systems in which all nodes have identical capabilities and responsibilities, and in which all communication is symmetric. The peers in a P2P system represent individual agents with their own interests and motivations, and they depend on beneficial and just relations to fulfill their functions in a virtual organization. Therefore, according to Patel [1], agents constitute and maintain social agreements and structure in a virtual community which mimics a community in the real world. We extend this view to our P2P approach.

Trust in a P2P context is also seen as the effect of a relation in which a peer may or may not satisfy the expectations of another peer according to a predetermined and expected behavior. This naturally assumes that certain specific functions are expected from that peer, such as a non-corrupted file transfer or a query delivery matching a predetermined level of quality. When the outcome of an interaction satisfies the expectation, then the peer is determined to be trustworthy.

According to Aberer and Despotovic [7], trust management comprises any mechanism that allows peers to establish mutual trust. Therefore, the concept of reputation is inserted into trust management since whenever a peer requests the reputation of another, this behavior represents a derivative measure of the direct and indirect trust acquired from past interactions.

2.2 - The TRAVOS Trust and Reputation Model

The trust and reputation model used in this paper is based on the Patel approach to trust, which is called the Trust and Reputation model for Agent-based Virtual Organisations (TRAVOS) [1]. This approach was built to establish direct and indirect trust values among the relations between machines in a grid-like virtual organization. We chose this algorithm for our study because it provides a framework for P2P environments to calculate trust in direct interactions and to calculate reputation in the absence of direct trust experience. The extension of this approach for interactions among groups of peers is described in Albuquerque [9].

Based on the definitions stated by Gambetta [6], TRAVOS is based on a probabilistic view of trust. It contains mechanisms to obtain a trust value based on past interactions and from reputation information obtained from agents in a multi-agent system.

Among the concepts central to the TRAVOS model, three main aspects concerning the proper management of trust were used as a foundation for tests in this paper:

1. The history of previous interactions with an agent (direct trust).
2. The opinions provided by other agents in the network (reputation).
3. The combination of direct interactions with reputation (combined trust).

In this model, the value of direct trust is calculated through a Beta distribution based on the observations by an agent a_1 of results of interactions with an agent a_2, and is represented by the equation:

$$\tau^{d}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}, \quad (1)$$

where $\hat{\alpha} = m^{1:t}_{a_1,a_2} + 1$ and $\hat{\beta} = n^{1:t}_{a_1,a_2} + 1$ are the successful and unsuccessful observed results of interactions, respectively.

The parameters α and β may be calculated by adding the number of successful outcomes to the initial value of α and the number of unsuccessful outcomes to the initial value of β. This formulation actually denotes α and β by $\hat{\alpha}$ and $\hat{\beta}$, because these are assumed to be estimated values. The basic idea is that the most reliable evidence for predicting the behavior of an agent is through the history of past interactions with this agent. Unfortunately, in some cases, the assessing agent has limited or no experience with the potential interaction partner. In these cases, reputation is used as a means of evaluating the agent.

An agent a_1 that desires to calculate the reputation $\tau^{r}_{a_1,a_2}$ of agent a_2 must collect the opinions provided by the other agents. Therefore, the successful and unsuccessful interactions are enumerated, resulting in the parameters M_{a_1,a_2} and N_{a_1,a_2}. These values can then be used to calculate shape parameters of the Beta distribution. The parameters are calculated by:

$$N_{a_1,a_2} = \sum_{k=0}^{p} \hat{n}_{a_k,a_2}, \quad (2)$$

$$M_{a_1,a_2} = \sum_{k=0}^{p} \hat{m}_{a_k,a_2}, \quad (3)$$

where p is the number of reports, and

$$\hat{\alpha} = M_{a_1,a_2} + 1, \hat{\beta} = N_{a_1,a_2} + 1, \quad (4)$$

and

$$\tau^{r}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}. \quad (5)$$

However, this value of reputation will only be reliable if two aspects are considered. First, the behavior of the trustee must be independent

of the identity of the truster; and second, the reputation provider must report its observations accurately and truthfully.

It is also possible to combine agent-personal observations with the opinions provided by other agents. This situation represents the approach of combined trust. To combine the personal experience (direct trust) with the opinions (reputations), we must first enumerate all of the opinions collected (M_{a_1,a_2}, N_{a_1,a_2}) and the direct trust values ($m^t_{a_1,a_2}, n^t_{a_1,a_2}$) in order to get the Beta shape parameters used to calculate the combined trust $\tau^{c}_{a_1,a_2}$, namely

$$\hat{\alpha} = M_{a_1,a_2} + \overline{m}^t_{a_1,a_2} + 1, \quad (6)$$

$$\hat{\beta} = N_{a_1,a_2} + \overline{n}^t_{a_1,a_2} + 1, \quad (7)$$

and

$$\tau^{c}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}. \quad (8)$$

The limitation of combined trust is that even using the combination of methods, there may still exist some agents capable of influencing the trust calculation through faked or inconsistent opinions. In response to this limitation, Patel [1] created a mechanism that enables the agent to determine the level of confidence it has in its own observations and then only seek the opinions of others if this level of confidence is insufficient.

The confidence metric γ_{a_1,a_2} measures the probability that the actual value of B_{a_1,a_2} (which is the probability that a_2 fulfils its obligations) lies within an acceptable level of error ε about τ_{a_1,a_2}. This level ε influences how confident an agent is, given the same number of observations. If the number of observations remains constant, a higher value of ε causes an agent to be more confident in its calculations than a lower value of ε. The parameter confidence is calculated as follows:

$$\gamma_{a_1,a_3} = \frac{\int_{\tau_{a1,a2}+\varepsilon}^{\tau_{a_1,a_2}-\varepsilon} (\mathrm{B}_{a_1,a_2})^{\alpha-1}\left(1-\mathrm{B}_{a_1,a_2}\right)^{\beta-1} d\mathrm{B}_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1} dU}. \quad (9)$$

Therefore, an agent is capable of deciding if it will use only direct trust or reputation, or a combination of the two for calculating the overall trust.

3. The JXTA Platform

The JXTA Project consists of open protocols responsible for the execution of all necessary functions in a P2P network, such as node research, resource discovery, query publishing, etc. All of these functions are defined in the JXTA framework [8] and executed through publishing and exchanging notifications (advertisements) and XML messages among peers. Project JXTA is based on a hybrid P2P

architecture that uses a DHT (distributed hash table) to store all information related to peers.

The JXTA [4] platform keeps an ensemble of Java open codes and enables specific codes to be easily added. The JXTA shell [4] is an application based in JXTA that simulates a command-structure environment similar to a UNIX shell. It allows Java classes to be created and added to the basic structure of the code, facilitating the P2P platform to be used for distinct studies in distributed environments. This therefore provides a P2P platform with a framework for determining trust and reputation in accordance with the objectives of our study.

In our work, we have created new commands in the JXTA shell in order to develop a P2P simulated environment system and test the TRAVOS model for establishing connections based on trust and reputation between machines.

4. Group trust approach

The concept of trust represents a unidirectional (1:1) relation. However, as with human relationships, there is no way to establish a communication based only on direct trust values in large societies. Instead, P2P networks may hold several, variable amount of nodes. Therefore, bigger networks make it more difficult to establish direct relations based only on direct trust and reputation requests. Using a group configuration

among the nodes then allows relations to expand using 1:N and N:M relations.

According to Albuquerque [9], in order for the group trust to reflect the experience of individual peers, each group must contain a leader according to a predefined model. As an example, the leader may be the most trustworthy peer in the network using the number of interactions and historical trustfulness as a baseline.

To be able to calculate the group trust, the leader asks for the reputation value from all other members of the group and calculates the average reputation of each node according to

$$\omega_g^n = \frac{\sum_{i=1}^{j} R_i}{j}, \qquad (10)$$

where j > 0. Thus, the trust value that one group places in another is calculated by adding the combined trust of every peer in the group and then dividing by the total numbers of peers of the destination group, that is,

$$\bar{\tau}_g = \frac{\sum_{i=1}^{x} \omega_g^n}{x}, \qquad (11)$$

where x is the number of peers in the group and τ_g is the final trust value of group g.

There are not many algorithms that can be used in a group trust calculation. In the leader election process, most of the models are based on voting. In the trust and reputation approach, a consensus for trust and reputation is desirable in order to identify the most trustworthy member of the group. This paper focuses on an efficient manner of trust and reputation calculation in a P2P system. The reader is referenced to Albuquerque [9] for more detailed information about trust in group contexts.

5. Implementation and Analysis

In this section, we present the implementation, the test environment of the P2P communication system used, and the analysis of our results. The main configuration settings applied during the tests are also defined and the procedures for quantifying trust and reputation are given.

The tests were accomplished without considering the real possibility that peers may lie about trust values in the network. In other words, all peers propagated their real trust and reputation values, even for those with malicious behavior in the system. In this simulation, all participants made the same number of interactions, with the intention of trying to find standards and verify certain behaviors about trust and reputation values in the system.

Trust on the computational level also includes the fact that a peer may or may not correspond to expectations of a predetermined behavior. When interactions follow such rules and protocols, it can be determined if the peer is trustful or not.

In our simulation, some behavior patterns of the peers were considered as desirable in a P2P system aimed at allowing file transfers among peers. First, there should be no errors in the transmission of a file, thus the file must arrive non-corrupted at the destination peer (the hash of the file is checked); Second, the time for transmitting the file is compared to predetermined levels that will determine the overall quality level of the transmission (the time in seconds for the transmission to complete). These parameters were chosen in order to simplify the P2P environment, thus allowing us to focus of the analysis of trust and reputation values during peer interactions.

5.1 - Developed JXTA Commands

We developed several shell commands in the course of our analysis and incorporated them into the JXTA shell structure. Such commands are recognized and executed normally by the shell prompt, allowing the analysis of TRAVOS in P2P systems to be evaluated and used for studying the group trust approach. These commands are given below.

JXTA> trust –init peerName - Automatically initiates each peer in the P2P network through the argument *peerName*. It is responsible for seeking the peers who are members of the network, erasing the cache that kept the name of the peers from former interactions, registering the peer and authorizing the file transfer through SFTP. Finally, the command creates a directory named *Tables* which will store the trust values of the peers.

JXTA> trust –test srcName destName filename – Transfers a file from one peer to another using SFTP. It is also responsible for preparing the destination peer to receive the file, and for calculating its hash using the MD5 algorithm. Each time a file is transmitted, the duration and speed of the transfer is calculated and the trust and reputation values are updated.

JXTA> trust –interaction srcName fileName [destName1 destName2 etc.] – Automates the trust and reputation tests in the network. Each time the command is set, a sequential interaction among all peers is issued by "trust –test".

JXTA> trust –directtrust srcName destName – Calculates the direct trust value among two peers in the network using the formulas given in section 2.2.

JXTA> trust –reputation srcName destName [otherName1 otherName2 etc.] - Calculates the reputation of a specific peer in the network. Each time this command is set, several requests are sent from the source peer to all the other peers (except for the destination peer) in the network.

JXTA> trust –combinedtrust srcName destName [otherName1 otherName2 etc.] - Calculates the combined trust value among two peers in the network. This command is used in the simulation to calculate the trust rate that one peer has for another. The parameter confidence is also computed in order to measure the level of confidence of the direct trust value.

JXTA> trust –grouptrust srcGroup srcLeader [srcOtherName1 srcOtherName2 etc.] destGroup [destOtherName1 destOtherName2 etc.] - Calculates the trust value one group has for another using the combined trust values of each peer in the destination group.

5.2 - Procedures, Simulations and Initial Settings

The simulations were executed on machines with the JXTA shell installed and configured. Five peers were set up on each machine using standard JXTA shell commands. Communication between machines was established with a Layer2 switch using 10/100Mbps ports. The purpose of this setup was to represent a P2P network connected directly to a LAN. The peers were configured in the same network segment with no additional hops. Each peer replied to a different TCP port between 9700 and 9709. This is to permit the P2P network to establish connections on different ports.

The transfer delay and the integrity of the file were parameters used to decide whether a peer was malicious or not. The interval of time values for the transfer was defined after several file transfer tests were executed. Several interactions were completed for a file with fixed size (100 Kbytes), and a standard time was then defined for a successful interaction. Table 1 gives the results of this test. Based on this information, it was determined that the expected transfer time of a file would be under 1s, a slightly-delayed transfer time would be between 1 and 2s, and completely delayed transfer time would be above 2s. The integrity of the received file was also defined using a hash calculation to verify the file's condition.

Table 1 – Test parameters.

Source Peer	Destination Peer	Time (s)	Speed (KB/s)
peer1	peer2	0.121	3363
peer3	peer4	0.280	723
peer1	peer6	0.441	459
peer2	peer7	0.510	797
peer8	peer3	0.480	1221
peer9	peer4	0.701	1161
peer10	peer5	0.881	923
peer6	peer7	0.160	635
peer7	peer8	0.210	1937

The transmission time and file integrity measure all combinations of possible file transfer

situations in the system, and are described by variables a and b, respectively. The parameter P represents the weight (importance) that the network administrator allocates to the integrity of the file. In our case, a value of 0.75 was chosen for P because we considered the integrity of the file to be more important than the time for transmission. Quantifying the combined file transmission time and integrity is then done by defining

$$C = P\,a + (1 - P)\,b. \qquad (12)$$

Considering these initial settings, Table 2 summarizes the possible situations that may occur.

	a	b	C
1. File corrupted and on time	0.00	1.00	0.250
2. File corrupted and a little delayed	0.00	0.50	0.125
3. File corrupted and completely delayed	0.00	0.00	0.000
4. File not corrupted and on time	1.00	1.00	1.000
5. File not corrupted and a little delayed	1.00	0.50	0.875
6. File not corrupted and completely delayed	1.00	0.00	0.750

In this simulation, values of *C* higher than or equal to 0.8 resulted in a defined successful interaction, causing the value of the expectation of the contract O_{a_1,a_2} (from TRAVOS model) to be equal to 1. This indicates that the peer who sent the file fulfilled its obligations by delivering a trustworthy and high-quality file. Any other value of *C* forces a null value to O_{a_1,a_2}, meaning that the interaction is associated with a malicious peer.

As related to the behavior of the peers during the tests, peers only accomplished interactions with appropriate parameters in order to verify the convergence of trust and reputation values. The tests started with a fixed number of malicious peers which was gradually increased. In this stage of the process, when a peer was initiated with a certain behavior (malicious or not), it maintained the behavior until the end of the tests. These characteristics were also implemented in the group approach.

In the first test, all peers of the network were trustworthy. Then 20% of the peers were configured as malicious. In both situations, the values of direct trust, reputation and combined trust were calculated. The final stage considered the scenario of changing a peer's behavior after it was already interacting on the network. The main objective was to verify if TRAVOS allowed the peers to realize that some members of the network changed their behavior and thus reacted to the change accordingly. In this test, all peers started as trustworthy but then changed their behavior at predetermined moment. At first 20%, then 50%, and lastly 70% of the peers turned malicious. For each situation, the values of direct trust, reputation and combined trust were collected for every communication channel.

5.3 - Results and Analysis

Our simulations attempted to reproduce situations inside an ideal environment in which all peers initiate from within the network and interact with each other the same number of times, therefore avoiding external factors that could influence the tests results. To simplify the demonstration of our results, we denote peer1 as our reference peer.

The parameter ε used for calculating the TRAVOS confidence was set to $\varepsilon = 0.2$, indicating our acceptable error level for an observation. The confidence parameter was defined as $\gamma = 0.95$. This means that if the confidence level of a peer is below this value, it will search for the opinions of other peers in the network. The values of the parameters γ and ε were defined through simulations from Patel [1], thus representing good values to simulate all features of TRAVOS.

A minimum trust value of 0.7 was set for a peer to consider another peer trustworthy and decide to continue interacting with it. For values below 0.7, the peer is considered malicious and the exchange of data among them is terminated. It is important to remember that this value is only a representative component, and may be adjusted according to the security or sought-after behavior between peers. We decided on this value because we desired a rigorous trust value.

5.3.1 - Trust calculation without behavior change

The simulation was set up assuming that each peer maintains its initial behavior. This means that malicious peers are initiated as malicious (at *t=0*), and keep the same behavior until the end of the interactions. In this test, peers do not keep historic data of trust values and the test is therefore called "trust calculation without historic". Two tests were simulated for this scenario, using 0% and 20% total malicious peers in the network.

The trust coefficient is equal among peers with the same behavior due to the fact that all the peers are initiated at the same time. In a real network, this would not occur because peers would have distinct behaviors and therefore different trust values. Ten interactions were performed in each test. This number is appropriate to describe the behavior of the trust values in the network (every interaction is performed using a trust approach).

The first simulation was treated as a baseline to compare all other tests. In this case, all peers are trustworthy and the network behaves ideally. The trust value initiates to null and increases during the elapsed time. The trust coefficient grows according to the Beta distribution, and the trust value tends to increase until stabilizing at a value near 1 (or a very high trust). It is important to remember that the coefficient will never reach the maximum value of 1 due to the properties of the Beta distribution. Figure 1 shows this initial configuration.

In this case, all peers answer the reputation request with positive values, inducing the combined trust values to increase faster than the direct trust values. During a certain moment of the test, an abrupt decrease of the trust coefficient occurs. This happens when the confidence value γ is larger than or equal to the established threshold of 0.95. When this occurs, the reputation values are no longer considered and the combined trust value is equal to the direct trust value. Therefore, all peers are considered trustworthy and interact accordingly.

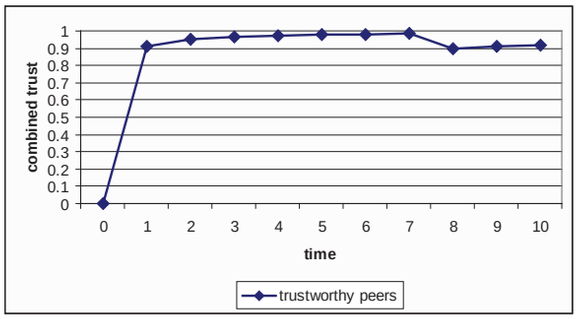

Figure 1 – Combined trust without malicious peers.

In the second test, 20% of the peers in the network are malicious. As seen in figure 2, the trust values remain low for malicious peers and increase for trustworthy peers (similar to the previous test). Again, the trust value relative to malicious peers tends to remain at a value near zero, but the coefficient never reaches the minimum value of zero due to the properties of the Beta distribution.

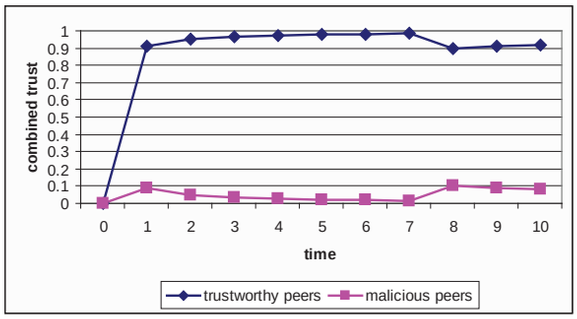

Figure 2 – Combined trust with 20% of peers malicious.

During a particular moment of the test (interactions 7 and 8), an abrupt decrease of the trust coefficient of the trustworthy peers occurs and coincides with an increase of the trust value of the malicious peers. Again, this happens because this point corresponds to a confidence value γ larger than or equal to our 0.95 confidence threshold. Therefore, in this case, trustworthy peers will never trust the malicious peers.

5.3.2 - Trust calculation with behavior change

This simulation was tested in a network in which peers change their behavior, similar to

current P2P networks. All peers were initiated as trustworthy peers (at *t=0*) and some peers changed their behavior as interactions unfolded. In this test, peers keep historic data about trust so this data is taken into consideration at the moment they became malicious and the test is therefore called "trust calculation with historic". During predetermined moments, some peers became malicious: first 20% of the peers (peer9 and peer10), then 50% of the peers (peer6 through peer10, inclusive) and finally 70% of the peers (peer4 through peer10, inclusive). A larger number of interactions were simulated in this test in order to better analyze the behavior of the trust coefficients. Figure 3 summarizes the behavior of the peers in this simulation through their combined trust values.

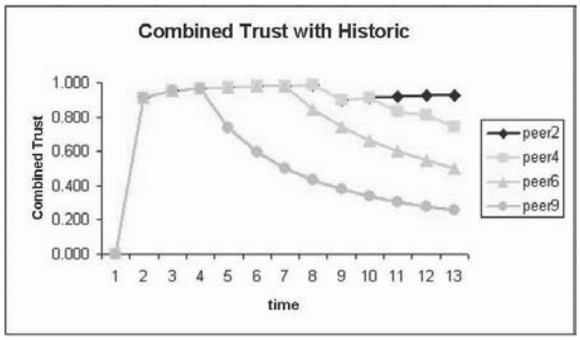

Figure 3 – Combined trust with historic.

It can be seen that peer2 was the only peer that remained trustworthy in the network until the end of the simulation, thus its trust value was not corrupted and remained close to 1. The trust values of the other peers decreased at specific moments once their intentions became malicious.

5.3.3 - Trust calculation in groups

The simulation in a group context was also performed with two different scenarios: with and without behavior change. In this test, the odd peers constituted the source group (the group that requested the trust value) and the even peers constituted the destination group (the group from which the trust value was calculated). All the collected results are presented having peer1 as the reference peer. A total of twelve interactions were performed in each test. This number was considered satisfactory to describe the trust behavior of a determined group in the network because it represents 12 possible interactions in a real scenario. The final group trust value is calculated in each test.

In the first simulation, without behavior change, there were no malicious peers and only good interactions were fulfilled. The test was then restarted with 20%, 40% and 60% of the members of the group acting as malicious peers. The results are relative to the combined trust values of the peers belonging to the destination group, according to peer1. In the event of no malicious peers in the network, the group trust value is equal to the individual trust value of each peer in the group, because all members have the same behavior. In this case, the group is considered extremely trustworthy and its trust coefficient tends to stabilize to a value near 1. Figure 4 summarizes this scenario.

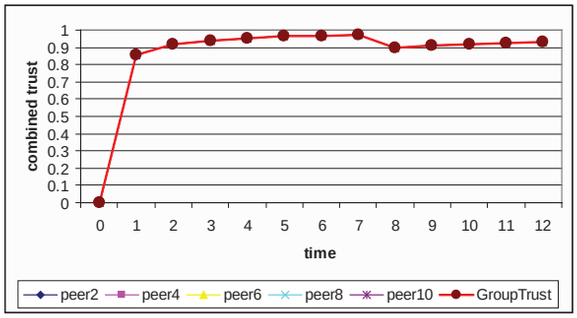

Figure 4 – Group trust without malicious peers.

In the second test, 20% of the peers behaved in a malicious manner. The group trust value increased and stabilized near 0.8. This result has one malicious peer, meaning 80% of the peers in the group are considered trustworthy. The analysis shows that the increase of the trust coefficient of the good peers overcomes the decrease of the coefficient of the malicious peers. In this case, the group is considered trustworthy as well ($\tau < 0.7$), despite having one malicious member.

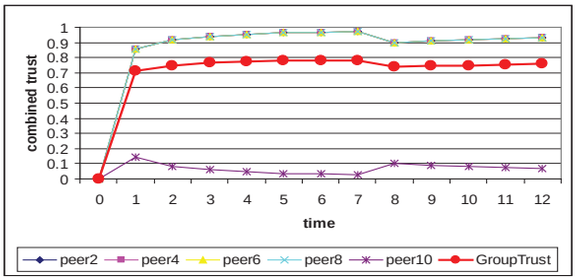

Figure 5 - Group trust with 20% of peers malicious.

Figure 5 shows the results for the group trust approach. It is important to note that this approach does discover the malicious peer in the group. Depending on the configuration, that particular peer can then be isolated from communicating on the P2P network.

When 40% of the group members are malicious, the trust coefficient tends to stabilize at a value near 0.6, representing 60% of the peers in the group are trustworthy. In this case, the group is not considered trustworthy because it is below our threshold of $\tau = 0.7$. Lastly, when 60% of the group members are malicious, the trust coefficient tends to stabilize at a value near 0.4, also making the group not trustworthy. Figures 6 and 7 show the results for 40% and 60% of the peers in the group being malicious, respectively.

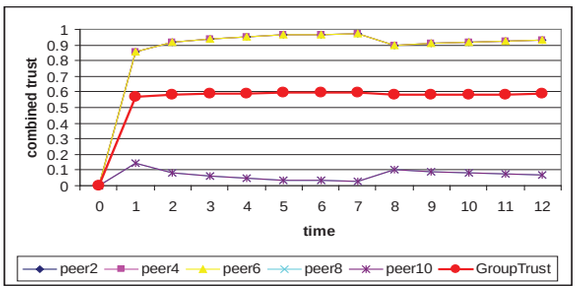

Figure 6 – Group trust with 40% of peers malicious.

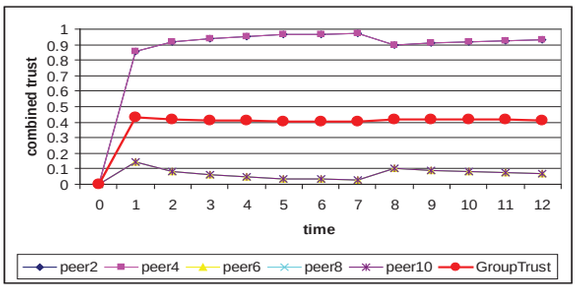

Figure 7 – Group trust with 60% of peers malicious.

During the second simulation, former interactions were considered and peers changed their behavior at predetermined moments. Initially, 20% of the peers (peer10), then 40% of the peers (peer8 and peer10) and finally 60% of the peers (peer6, peer8 and peer10) became malicious.

Figure 8 summarizes the combined trust values of the destination group and of the group members. The trust value of the group was initially high (the moment in which all peers have good behavior). The value then started to decrease at the moment the first peer in the group became malicious. When the other peers changed their behavior, the group trust coefficient decreased faster, thus making a group with constantly changing behavior not trustworthy. With a higher number of interactions, this value tends to stabilize near 0.4.

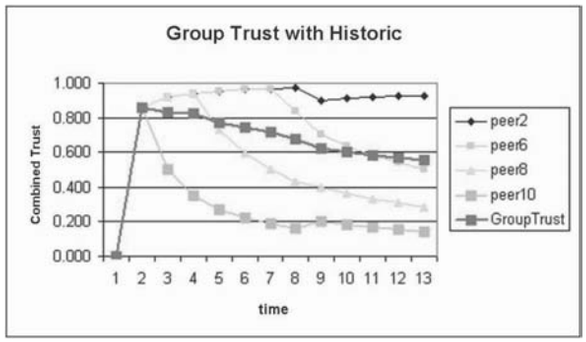

Figure 8 – Group trust with historic and changing behavior.

In both simulations, the individual behavior of each group member influenced the trust value of the group as a whole. This result was acceptable since all peers initiated in the network at the same moment and interacted with each other the same number of times. During the times of behavior change, the group's trust value was influenced. For instance, if a group leader realizes that their trust value is below expectations, it may decide to isolate malicious peers from the network. After that, the group trust value may again rise above expectations and result in more interactions. In the case that successful interactions represent incentives for continued collaboration, groups tend to maintain good behavior most of the

time and isolate malicious peers because they represent a threat to the group as a whole.

6. Conclusion

In our simulations, the JXTA shell appeared to be flexible and sufficiently structured in order to correctly emulate a decentralized P2P network, and the TRAVOS implementation was an effective trust and reputation model that gave peers the chance to react quickly (and with a small number of interactions) to realize changes in the network. However, since it has no means of calculating group trust and trust consensus, which is an open question for trust in distributed systems, we extended the model with our approach to group trust. This basic implementation and our extension were then evaluated.

The main objectives of this paper were fulfilled. In particular, our approach was able to identify malicious peers and exclude them from communication. It was observed that, after some number of interactions, peers with good behavior established a general agreement about the identification of the malicious peers in the network. Therefore, malicious peers were isolated form further communication after a certain number of interactions.

In the simulation where peers changed their behavior during the interactions, the network as a whole learned the identification of the malicious peers more slowly. However, within a reasonable number of interactions, the network converged to a consensus of malicious peer identification, again permitting their isolation from further communication.

Using the concept of trust in a group approach, the model explored in this paper could be used in larger and more complex P2P architectures. It is important to consider a leader election approach for the group and it must be elected by other peers considering the trust approach adopted here. It is also correct to think that the trust value of an older peer in the network should have more significant influence for the group trust coefficient than the trust value of more recent peers. Using a balanced average that considers the standard deviation of the individual trust values would result in a more precise group trust coefficient, thus improving group trust results.

References

[1] Patel, J. "A Trust and Reputation Model for Agent-Based Virtual Organizations". Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. University of Southampton. January, 2007.

[2] Sabater, J. and Sierra, C. "Review on Computational Trust and Reputation Models". Artificial Intelligence Review (2005) 24:33–60, Springer 2005

[3] Rowstron, A., Druschel, P. "Scalable, decentralized object location and routing for large-scale peer-to-peer systems". Middleware 2001: 329-350. 2001.

[4] Wilson, B. "JXTA". New Riders Publishing, June 2002.

[5] Marsh, S. "Formalising Trust as a Computational Concept". PhD thesis, University of Stirling, UK, 1994.

[6] Gambetta, D. Can we trust trust? in Gambetta, D., Trust: Making and Breaking Cooperative Relations, Sociology Department, Oxford University, Chapter 13. (2000)

[7] Aberer, K., Despotovic, Z. "Managing Trust in a Peer-2-Peer Information System". CIKM 2001: 310-317. 2001.

[8] JXTA. "JXTA – Get Connected". Available at http://www.jxta.org

[9] Albuquerque, R. "Propuesta de un Modelo de Confianza y Consenso Computacional en Grupos en Ambientes Distribuidos". Universidad Complutense de Madrid, 2007.

R. O. Albuquerque was born in Taguatinga – DF, Brazil, on January 30, 1976. He graduated in Computer Science, Catholic University of Brasília, Brasília – DF, Brazil, 1999, and got his Master Degree in Electrical Engineering, University of Brasília, Brasília – DF, 2003. He is pursuing his Doctorate Degree at University of Brasília, Brasília – DF, Brazil, and at Universidad Complutense de Madrid, Madrid, Spain. His field of study is Network and Information Security. He is a network security expert and his professional experience includes IT consulting for private organizations and the Brazilian Federal Government. His fields of interest and research include Network Management, Network Systems, Software Agents, Wireless Networks and Open Source Software.

R. T. de Sousa, Jr., was born in Campina Grande – PB, Brazil, on June 24, 1961. He graduated in Electrical Engineering, Federal University of Paraíba – UFPB, Campina Grande – PB, Brazil, 1984, and got his Doctorate Degree in Telecommunications, University of Rennes 1, Rennes, France, 1988. His field of study is Network Engineering, Management and Security. His professional experience includes technological consulting for private organizations and the Brazilian Federal Government. He received the 3rd Telecommunications Telexpo Golden Medal for his work on the management of network services quality. He is a Network-Engineering Professor at the Electrical Engineering Department, University of Brasília, Brasília – DF 70910-900 Brazil, and his current research interest is trust management for spontaneous self-organized networks.

L. S. Bezerra was born in Belém - PA, Brazil, on June 04, 1982. She graduated in Network Engineering, University of Brasília - DF, Brazil, 2007. She is taking her Master Degree in Electrical Engineering at University of Brasília - DF, Brazil. She works in the Brazilian Federal Government with Network and Information Security. Her professional experience includes information security policies, detection and intrusion prevention, IT security and network managemant. Her fileds of interest are information and network security, comunications network, and distributed systems.

G.R. de Lima was born in Luziânia – GO, Brazil , on October 14, 1986. She graduated in Network Engineering, University of Brasília - DF, Brazil, 2008. She works in a private company dealing with several network technologies. Her professional experience includes designing network solutions along with CCNP certification. Her fields of interest are information and network security, network systems management and routing and switching design..