Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 1119 – 1128

Procedia Computer Science

International Conference on Computational Science, ICCS 2012

Agent-based trust management and prediction using D3-FRT

Olufunmilola Onolaja[a,*], Rami Bahsoon[a], Georgios Theodoropoulos[b,1]

[a] *University of Birmingham, Birmingham, United Kingdom*
[b] *IBM Research, Dublin, Ireland*

Abstract

Reputation and trust management systems have been useful in domains that rely on the cooperation of members to function correctly and to fulfil their purposes. Despite the advent of these systems, having trusted communications remains a challenge. This is as a result of relying on the domain members for reputation information. These systems lack well analysed approaches for determining the bias of the members. A semi-distributed framework D3-FRT, which is inspired by the Dynamic Data-Driven Simulation paradigm, is presented in this paper. The framework adopts an agent-based modelling approach to make predictions about domain members. The D3-FRT framework is novel as it uses past, online and predicted data to identify misbehaving members. In this paper, the accuracy of the prediction is tested and a report on the framework's performance in different network scenarios is also presented. The results of experiments and simulations using the D3-FRT approach are discussed.

Keywords: Agent-based simulation, DDDAS, Reputation, Trust

1. Introduction

The spread of Internet usage, proliferation of mobile devices, computing, and online market places and the rapid growth of wireless networks has changed the landscape of security in terms of trust. This change is because of users collaborating anonymously with others in these domains, resulting in users being exposed to risks. The domains primarily rely on cooperative member behaviour for their reliable operation; else, they are unable to fulfil their functions.

To reduce risks and improve reliability, applications must manage trust relationships between users by motivating cooperation and honest participation [1]. For example, Peer-to-Peer (P2P) networks are undergoing rapid progress and have had numerous developments in recent years. For such networks to be effective in fulfilling their purpose of anonymous sharing, they should be relatively reliable, efficient and secure [2]. However, due to their anonymous and open nature, malicious users can abuse the system by disseminating inauthentic files or acting together to commit as much damage as possible (*collusion attack*).

Furthermore, Internet applications have evolved from centralised and private computing platforms to distributed and collaborative computing systems: the wide spread of social computing, ecommerce and the advent of cloud

*Corresponding author
Email addresses: o.o.onolaja@cs.bham.ac.uk (Olufunmilola Onolaja), r.bahsoon@cs.bham.ac.uk (Rami Bahsoon), georgheo@ie.ibm.com (Georgios Theodoropoulos)

[1] This research was initiated while Georgios Theodoropoulos was with the School of Computer Science, University of Birmingham, UK.

doi:10.1016/j.procs.2012.04.121

federation models are some timely examples of how corroboration is a fundamental Internet computing requirement. As a result, there is an urgent need for an approach of dynamically managing trust and predicting reputation effectively.

Managing trust and its dynamic nature in such large-scale distributed applications and domains is a difficult challenge, but one well suited for reputation and trust management. Reputation and trust management research is highly interdisciplinary [3], involving researchers from networking and communications: Mobile Ad-hoc Networks (MANETs); Wireless Sensor Network (WSNs) and P2P; data management and information systems; e-commerce and online communities: YouTube, Amazon and eBay; Artificial Intelligence; also in the Social Sciences and Evolution Biology.

Reputation and trust management models (RTM) have gained popularity because they have shown to be promising in the area of trust management and have provided solutions to the issue of trusted communications. These models aim to collect, aggregate, and disseminate feedback about users' behaviour (reputation), based on some premise. Reputation and trust management is useful for establishing a healthy and efficient collaboration among a network of participants and users that might not have sufficient prior knowledge of each other [3]. Consider eBay for example, that has several millions of auctions open at a time and serving as a listing service, where buyers and sellers assume all associated risks with transactions [4]. Though there are fraudulent transactions on the eBay system, there is still a high rate of successful transactions as well, which is attributed to the reputation management system on eBay called the *Feedback Forum*.

RTMs aim to provide mechanisms to produce a metric encapsulating reputation (referred to as *Trust Value* (TV)) in a given domain, for each identity in the domain [5]. Despite the proliferation of RTMs, building reliable systems remains a challenge. Several unaddressed threats still limit the effectiveness of the models, as in the process of solving the issues, other problems are introduced [1]. Existing RTMs focus on historical and online information in determining the reputation of domain members. However, the dynamic nature of reputation and trust requires an equally dynamic approach to computing and resolving trust related issues in the application domains. We propose that possible future TVs of entities in the systems should be anticipated to aid predictions and thereby resulting in more reliable and agile reputation systems.

In [6], we introduced a novel agent-based and data-driven predictive framework called D3-FRT, and we provided an initial evaluation. This framework takes advantage of past behaviour, anticipates future and online events for reputation and trust management. In this paper, we discuss in more details, the attributes of the framework and provide a qualitative validation of the proposed approach, confirming that data-driven agent based simulation is a method that can be effectively utilised in this problem domain. We also present a more extensive quantitative evaluation of D3-FRT which demonstrates its efficiency under different conditions.

The rest of this paper is organised as follows: Section 2 describes the reputation management architectures in detail. The use of agent-based modelling for trust management is discussed in section 3, while section 4 summarises the D3-FRT framework proposed. Section 5 presents a set of experimental results and analysis of the framework and section 6 describes significant reputation systems in literature. Finally, the conclusions are stated in section 7.

2. Reputation system architecture

There are two types of RTM architecture: the *centralised* and *distributed* architectures. In centralised architectures, trust is managed by a trusted central server(s) (*Trusted Third Party* (TTP)) that is connected to all or some of the identities in the system. The centralised architecture has been successfully deployed in real life applications such as eBay and Amazon. On eBay's feedback forum for example, buyers and sellers can rate each other after each transaction. Members receive: $+1$ point for each positive rating, 0 points for each average rating, -1 point for each negative rating. The overall reputation of each entity is the sum of these ratings for some months. This simple approach to trust management has drawbacks because the approach used is linear. This implies that a rater gives either positive or negative scores per transaction, therefore, failing to capture the dynamic nature of reputation effectively. For example, an eBay seller with 50 positive feedback scores is rated the same as a seller with 100 positive feedback scores and 50 negative scores. Also generally, the downsides of centralised architecture are the performance bottleneck of the central entity and the resulting lack of scalability.

Contrarily, a purely distributed approach requires each entity or domain member (subsequently referred to as *agent* in this paper) to maintain trust-related information about other agents in the system. This implies that reputation

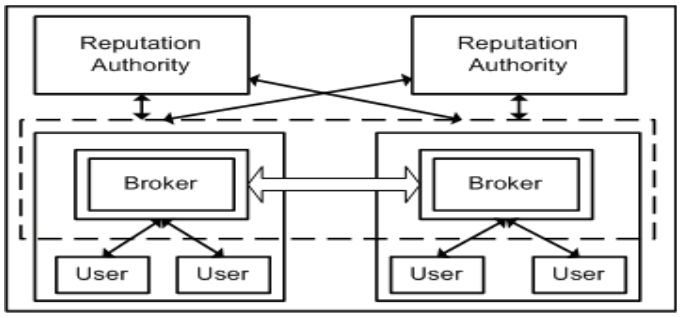

Figure 1: System architecture

management is determined, spread and shared among the agents. This approach introduces the problems of obtaining and propagating information across the system. In such an agent-centric model, each agent monitors, analyses and calculates the trustworthiness of other agents of interest based on some defined metrics. A purely distributed approach results in some real-life problems such as the corruption of trust decisions through recommendations made by agents. This approach exposes the system to *false praise/accusation* and *collusion attack* [7, 8]. This vulnerability is attributed to existing RTMs lacking well analysed approaches to determining the bias of each agent [9]. The use of a distributed architecture is an issue because the correct functioning of the system relies on the cooperation of domain members in terms of good behaviour, which cannot always be guaranteed. Following these downsides of purely distributed and centralised systems, a semi-distributed approach is more desirable, as it combines the upsides of both types of architecture.

Some previous work [10, 11] in the area of RTM research claim to be distributed, but still employ the use of TTP(s) for trust management. These RTMs are indeed not completely distributed. For example, in the work of Lin et *al* [10] for web applications, the RTM comprises of 3 components: *users*, *brokers* and *reputation authorities* as depicted in Fig 1. Users do not rely on a database managed by the same users but on the brokers to collect reputation information. The reputation authority is a last resort for information, in case there is insufficient information about any other user. The reliance on a TTP such as the broker and the reputation authority therefore, implies that the RTM is not completely distributed but may be regarded as a semi-distributed approach.

The semi-distributed approaches are similar to the concept of D3-FRT framework as they recognise the need for a distributed architecture but with a form of control to aid reputation management. In the D3-FRT framework, the reputation of other agents is not entirely determined and managed by individual agents but by super-agents (this is similar to the hierarchical intrusion detection systems in [12]) and through simulation of the domain. Therefore, a semi-distributed approach is more desirable, as it combines the upsides of both types of architecture. The semi-distributed architecture does not rely on only individual agents alone for making trust decision but employs the use of a TTP.

The ultimate aim of any RTM is to achieve trusted communications among a network of agents by meeting certain requirements. Firstly, there is a requirement for monitoring the behaviour of agents at runtime and providing feedback to the reputation system and the domain. The monitoring requirement is an important one especially in critical domains such as traffic management systems and military applications etc. Secondly, the prediction of agent TVs is essential and a more proactive approach to the detection of misbehaving agents.

The Dynamic Data Driven Application Systems (DDDAS) [13, 14] paradigm makes provisions to meet these requirements. The concepts of the paradigm, namely: *measurement*, *simulation*, *feedback* and *control* have the potential of providing dynamism in the detection of malicious agents and prediction of future ratings of each agent. The runtime measurements (qualitative behaviour of agents which is converted to quantitative TVs) are simulated to gain a better understanding and provide a more accurate prediction of the level of trust for each agent. The simulation dynamically measures trust levels, and continually incorporates new measurements at runtime. This will enable the reputation system to determine and give a feedback of the reputation of each agent. The output of the simulation controls the system relative to the decisions to be made in order to maintain trusted communications. Therefore, the presence of a TTP in a semi-distributed architecture to monitor, simulate, feedback measurements in the system which

allows agents to collaborate and at the same time to provide unbiased monitoring and feedback in the system is ideal, therefore resulting in a better trust management.

3. Using agent-based modelling

Due to the complexity of the systems in use these days, their interactions and interdependencies, a dynamic approach to predicting future events is required to capture all requirements, in order to provide a close representation of reality. Agent-Based Modelling and Simulation (ABMS) is an approach to modelling complex systems composed of interacting, autonomous agents. ABMS offers ways to model social systems that are composed of interacting agents that influence each other, learn from their experiences, and adapt their behaviours [15].

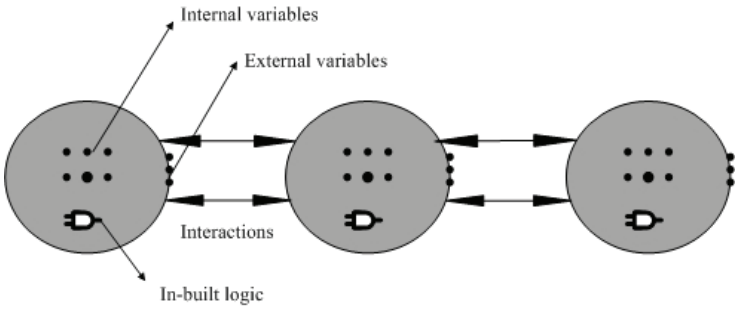

Figure 2: Properties of agents in simulation

ABMS is very applicable as behavioural emergence is a major consideration of this research. Provision of useful information to control the systems is another justification for the use of ABMS. When agents optimise their collective behaviour through simple exchanges of information as is done in an ant colony optimisation [16] for instance, the purpose is to achieve a desired end-state. This is an optimised system rather than the simulation of a dynamic process for its own sake. ABMS is suitable for reputation and trust management and its use is supported by its potential to provide insightful views of the past, present and future states of a system. The use of ABMS in the framework therefore enables D3-FRT to be proactive in terms of making predictions about future states of agents in the domain it is applied.

4. D3-FRT

For an RTM to be reliable and effective in trust management, trust has to be predictable. Generally RTMs make use of past events as pointers for the future. This is because it is generally assumed that the predictive power of an RTM depends on the supposition that past behaviour is an indication of the future. In previous papers [6, 7], we have shown that this supposition might not hold in the case of misbehaviours such as *intoxication*. Intoxication occurs when an agent behaves as expected for a sustained period of time to obtain a good reputation in the domain and only starts to misbehave afterwards. By way of illustration of such an attack on the eBay system, an agent builds a high feedback rating with low-valued transactions and then misbehaves with a high-valued one.

D3-FRT extends the supposition further by considering the past, present and possible future behaviours. D3-FRT uses a semi-distributed approach and is data-driven as data obtained from agent interactions is incorporated in the simulation and subsequently used for making informed decisions in the domain.

For easy referencing, the subsequent frequently used notations are listed below:

tv_h^R, tv_h^S	Historical TVs in the system and simulation respectively
tv_o^R	Current online TV
tv_f^S	Predicted TV
N	Number of agents

In D3-FRT, agent behaviours are described by simple rules. Interactions with other agents in turn influence the behaviours of agents. The agents have internal and external properties referred to as state variables that are depicted

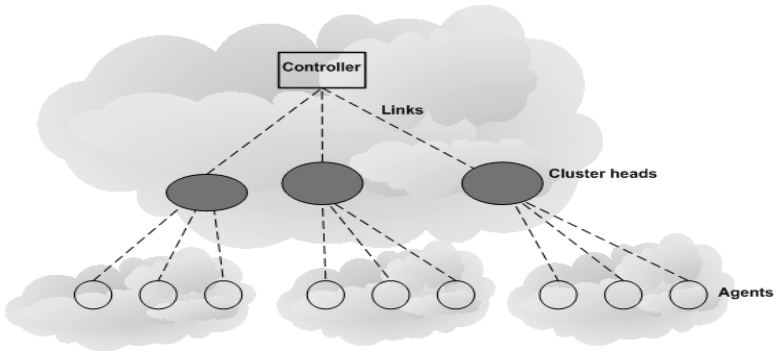

Figure 3: Hierarchical topology

in Fig 2. Some variables such as trust value (tv_h^R, tv_h^S, tv_o^R and tv_f^S), agent identifier, trust region, agent type are static while others may change with time. Changes in the *state variables* may either be as a consequence of the influence of the internal in-built logic of the agent or as a result of cooperating with another agent or domain member. D3-FRT comprises agents, *Cluster Heads* (CHs) and a *Controller*, as shown in Fig 3.

Fig 4 shows a normal collaboration scenario of an agent A, for example, that registers its presence with a CH and obtains a unique AgentID with neutral tv_o^R of 2.5 (agents' TVs range between 0 and 5). The network of agents is partitioned into clusters, where each cluster has a head; a CH that is a super-agent. Each CH has a direct connection with every other member of the cluster.

As a substitute to each agent monitoring the actions of every other agent, the CH is responsible for monitoring and obtaining information about agent interactions. Agents obtain reputation information from the CH and TVs of other agents of interest. The CH in turn shares information about the agents. Agents are encouraged to be cooperative and to collaborate with reputable agents through the incentive of increase in TVs and decrease in TVs for misbehaviour which could eventually result in exclusion from the system. In the scenario of Fig 4, if agent A wishes to collaborate with another agent B for example, agent B's reputation will impact on A's TV. Agent A requests the TV of B from the nearest CH. The CH that B is connected to and which stores the TV provides the current TV of agent B from its table.

The output from the interactions is reported by both interacting agents to their corresponding CH(s). This information is used in computing the tv_o^R of the agents and the values are stored for future reference. The controller obtains data about events in the system from the CHs to compute an updated TV of each agent. This implies that at specific time intervals, D3-FRT selects useful data from a stream of data from the system, which is dynamically injected into a controller to compute the current TVs of all agents. The data is obtained from agent interactions that occurred within a specific time frame, which includes the feedback from the interacting agent and observations captured by CH. The simulation component of D3-FRT then determines the future TVs of the agents, using historical data, online events in the system and anticipated future activities.

Considering the computation of TVs, similar to the approach of [17, 18], the current behaviour of an agent carries more weight in D3-FRT. This is to prevent agents from obtaining a good reputation with high TVs and subsequently misbehaving (intoxication).

To make predictions, the simulation considers different '*what-if*' scenarios in which an agent may be in the future. This component utilises the processed data to predict the tv_f^S of agents. Historical data (tv_h^R and tv_h^S) about agent behaviour is also considered in making the predictions. Depending on their TVs, agents logically belong to *risk-regions* in the domain. This enables the RTM to focus on agents that are of high-risk to the system. A detailed description of the D3-FRT components is given in previous papers [6, 7].

5. Simulations and analysis

In this section, we evaluate the performance of the D3-FRT framework in terms of its predictive capability in the presence of collusion and intoxication attacks. The performance of the framework in different network sizes with different numbers of misbehaving agents is also evaluated.

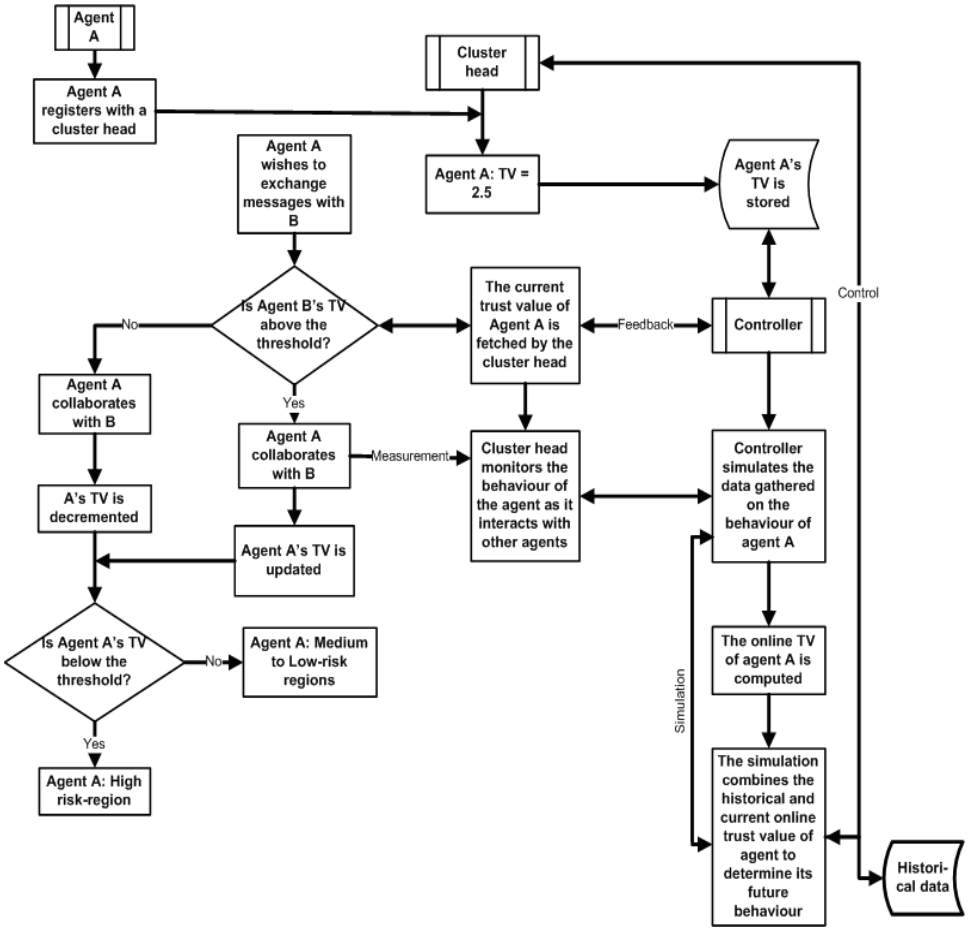

Figure 4: D3-FRT flowchart

Fig 5 provides an indication of the predictive capability of D3-FRT. The following parameters were used throughout the simulation: $N = 50$ in a *super-peer* P2P file-sharing network with 2 CHs and all agents with an initial tv_o^R of 2.5. A super-peer network is simply a P2P network consisting of super-peers and their clients [19]. An agent with a TV of 1 and below is regarded as misbehaving and is automatically excluded from the network. The total time for this experiment was 1000 *ticks*, which is a compression of time. Throughout the experiment, 30% of the agents misbehaved. The predicted TV is computed using the formula [7, 6]: $\mu_h tv_h^R + \mu_o tv_o^R + \mu_f tv_f^S$ and the following scaling factors were used: $\mu_o = 0.3, \mu_h = 0.5$ and $\mu_f = 0.17$. Higher weighting was given to recent interactions (that is, $\mu_o > \mu_h$) to reduce the possibility of intoxication.

In the experiment, agents interact randomly as in a P2P network and these interactions are captured at every 1 tick. The predictive accuracy of D3-FRT is measured by comparing the agents' TV in the network with the predicted TVs by D3-FRT. We then obtain the Magnitude Relative Error (MRE): $\left|\frac{TV_{actual} - TV_{predicted}}{TV_{actual}}\right|$ for all agents for the duration of the simulation averaged over a set of randomly selected agents. Where TV_{actual} is the value computed for nodes in the network while $TV_{predicted}$ is obtained from the simulation. The MRE remained below 1.0 throughout the duration of the simulation.

Fig 6 shows the results of the Mean Magnitude Relative Error (MMRE): $\frac{1}{N}\sum_{i=1}^{N}\left|\frac{TV_{actual} - TV_{predicted}}{TV_{actual}}\right|$ where $N = 10$ from the data obtained from the previous experiment. The result ranged between 0.39 and 0.5 and the overall MMRE averaged at ≈ 0.46. This value is attributed to the simulation not having any prior knowledge of the network initially. However as the system evolves, the simulation converges as shown in the following experiments. From these results,

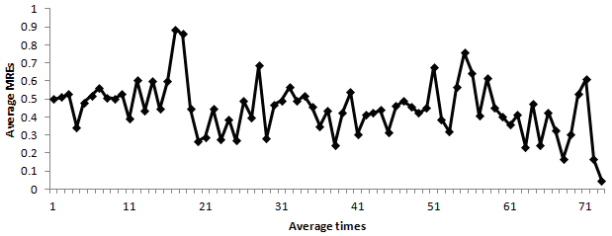

Figure 5: Average trust value prediction error

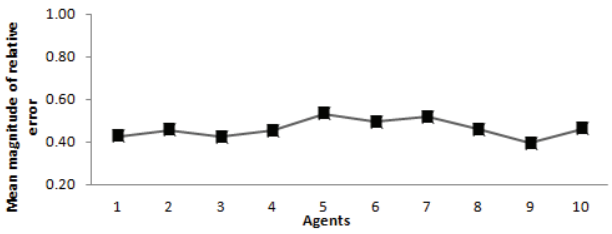

Figure 6: Mean magnitude of relative error per agent

the degree of variance between the actual TVs and the predicted values might account for possible false-positives and false-negatives in the predictions. Note that the aim of D3-FRT is not to accurately make predictions about the agents but near-accurate predictions.

In order to test the reliability of the framework, the error rate in predictions made is tested with different parameters. The acceptable error rate of D3-FRT will depend on the criticality of the application domain and the risk appetite in the domain. With penalty of 0.4 and 0.5 for intoxication and collusive behaviour respectively, and a reward of 0.5 for normal behaviour, the following parameters are used: Scaling factors $\mu_h = 0.3, \mu_o = 0.5$. For every 100-tick cycle of the simulation, $N = 50$ collaborating agents with 12% and 10% of the agents exhibiting collusive and intoxicating behaviours and for the duration of 9000 ticks. As shown in Figures 7 and 8, there was 5.1% prediction error rate at an allowable error threshold of ±0.6 in the domain.

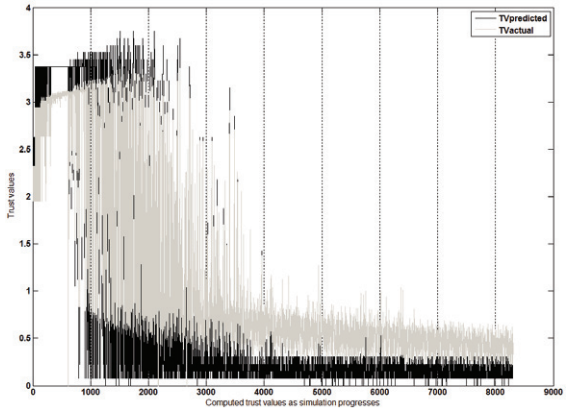

Figure 7: TV_r and TV_f when $\mu_o = 0.5$

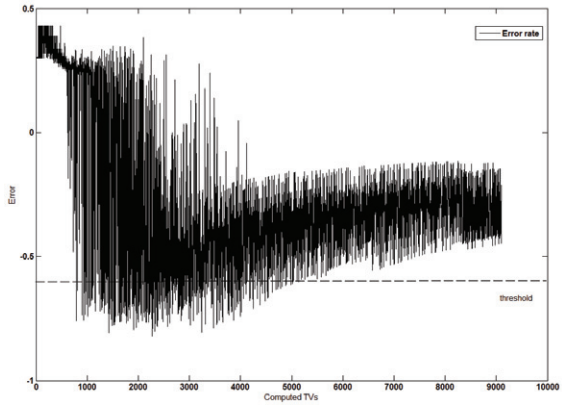

Figure 8: Error in prediction when $\mu_o = 0.5$

However, with μ_o set to 0.6 and error threshold of ±0.6, D3-FRT made some inaccurate predictions about agent trust values at the rate of 15%; this is much higher than error rate of when $\mu_o = 0.5$. The graph in Fig 9 shows the overall TV values of the agents compared with the predicted values. In Fig 10, from approximately 5500 tick in the simulation, the error rate reduced considerably and remained above the lower bound −0.6 of the threshold.

To test the behaviour of D3-FRT when the scaling factors are equal, μ_o was set to 0.3. That is, $\mu_o = \mu_h = 0.3$. The results of this experiment are depicted in figures 11 and 12 and the error rate is 21%. Fig 12 shows that the errors in prediction initially fluctuated above and within the threshold value but later remained within the threshold value from around 3800 and for the rest of the simulation. Therefore, the best values for the scaling factors are $\mu_h = 0.3, \mu_o = 0.5$ as the best results were obtained with these values. This clearly shows that making D3-FRT adaptable allows for obtaining the best results in terms of predictions.

The performance of D3-FRT is evaluated by considering the effect of network size in terms of the total number of agents and population of misbehaving agents. The observation is that as the network size increases with an increase in the number of misbehaving agents, so does the Time-To-Detect (TTD) all the agents with TVs less than the threshold of 1. This is illustrated in Fig 13. With $N = 400$, out of which 15% are misbehaving agents, the TTD is $\approx$ 90 ticks.

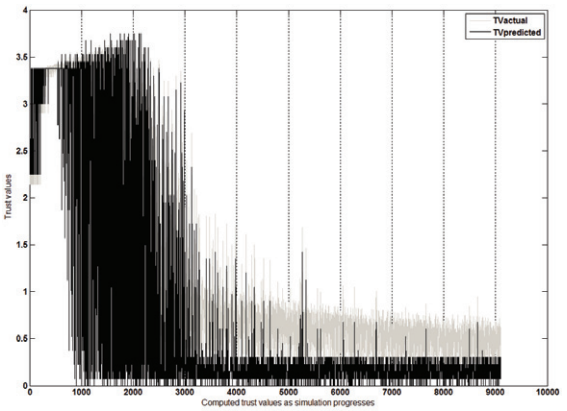

Figure 9: TV_r and TV_f when $\mu_o = 0.6$

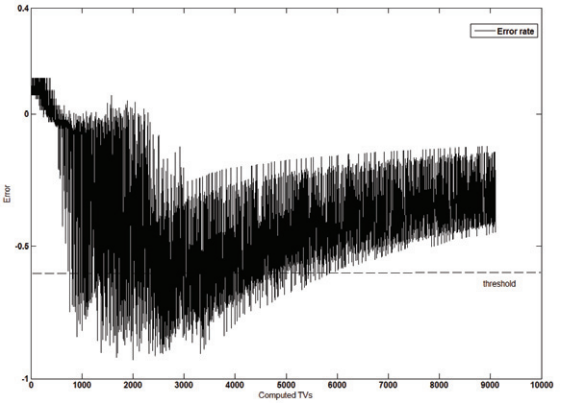

Figure 10: Error in prediction when $\mu_o = 0.6$

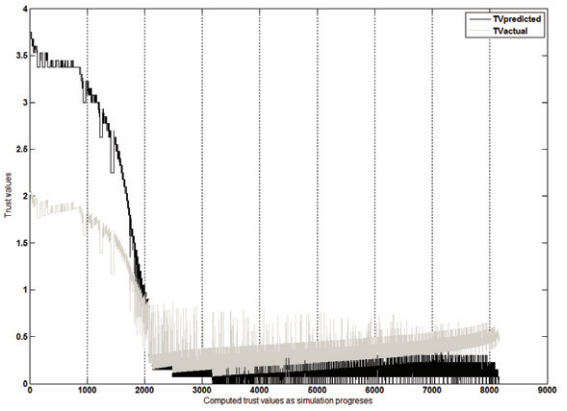

Figure 11: TV_r and TV_f when $\mu_h = \mu_o = 0.3$

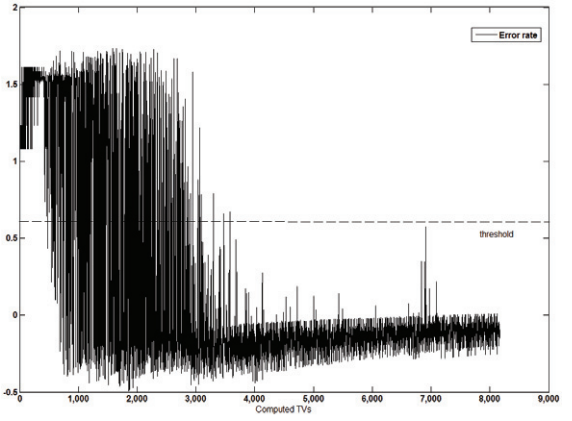

Figure 12: Error in prediction when $\mu_h = \mu_o = 0.3$

However with 1600 agents and with 6.25% of the population misbehaving, TTD is $\approx$ 120 ticks. This result shows that the number of interacting agents has an impact on performance of the D3-FRT.

6. Related work

RTMs that have shown positive results and contributed significantly to trust management in literature are discussed in this section. Michiardi and Molva [20] proposed a model where reputation is formed and updated over time by direct observations and information provided by other members of the network. In their model, nodes have to contribute continuously to remain trusted or their reputation will be degraded until they are excluded from the network. The model gives a higher weight to past behaviour. The authors argue that a more recent sporadic misbehaviour should have minimal influence on a node's reputation that has been built over a long period of time.

A file-sharing P2P reputation system's algorithm: EigenTrust [21], similar to the popular PageRank aims to identify sources of inauthentic file and to prevent peers downloading from them. The algorithm assigns each peer a unique global TV, based on the peer's history of uploads. EigenTrust's susceptibility to collusion has been demonstrated in [22], where certain colluding peers are able to obtain high TVs.

Buchegger *et al.* [17] proposed a protocol that aims to detect and isolate misbehaving nodes, making it unattractive for any node to deny cooperation with others. In the protocol, each node maintains a reputation and a trust rating about every other node of interest. Only fresh reputation is propagated in the network, with more weight given to the current behaviour of a node over its past behaviour. Nodes monitor and detect misbehaviour in their neighbourhood by means

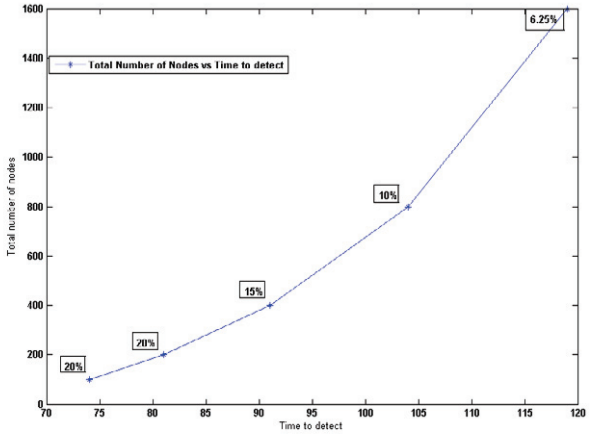

Figure 13: The number of misbehaving agents versus time taken to detect

of an enhanced *packet acknowledgment* mechanism; where the confirmation of acknowledgment comes indirectly by overhearing the next node forward the packet [18, 23].

Similarly to the scaling factors used in D3-FRT, TrustGuard [24] system which includes a TTP, can be implemented with different degrees of centralisation. TrustGuard allows for flexibility by giving different trust components varying weights. The system uses a strategic oscillation guard based on a controller to combat malicious sudden or oscillatory behaviour among nodes.

In the work of Ganeriwal *et al.* [25]; which is applicable to WSNs, each sensor node maintains reputation metrics. These metrics represent the past behaviour of other nodes and are used as an inherent aspect in predicting their future behaviour. The model relies on network members to maintain the reputation of others based on their experiences and uses this to evaluate their trustworthiness.

More recent studies on RTMs are discussed in [9, 26, 27]. A common problem seen in the models is the vulnerability to collusion attacks [28]. Models applicable in the mobile networks domain, make use of a component resident on each node called *watchdog* mechanism. This component monitors its neighbourhood and gathers data by *promiscuous observation*. Promiscuous observation refers to the situation where each node overhears the transmission of neighbours to detect misbehaviour. Watchdog requires that every node report to the originator about the next node and once misbehaviour is detected, a negative TV is stored. However, nodes may decide to cover up for one another, thereby deceiving the reputation system.

7. Conclusion

Leveraging on our previous work, in this paper we have presented a more extensive qualitative and quantitative evaluation of D3-FRT, an agent-based predictive framework for reputation and trust management. This semi-distributed framework uses agent-based modelling simulation approach, and exploits the DDDAS paradigm in managing reputation. These have shown to be useful in aiding the prediction of domain events and agents' trust values.

D3-FRT's simulation component anticipates future events and considers available information to make predictions about participants in a P2P network scenario. D3-FRT has the potential to be adaptable as it allows for varying variables, to then select the best input values that will provide the closest set of predictions. This therefore allows for informed decisions to be made to prevent misbehaviour in the system. This paper shows how the D3-FRT framework fulfils its purpose in identifying misbehaving agents and in making predictions. It can be concluded that the use of monitoring, simulation and feedback, can potentially improve the reliability of reputation and trust management systems.

References

[1] G. Swamynathan, K. Almeroth, B. Zhao, The design of a reliable reputation system, Electronic Commerce Research 10 (3-4) (2010) 239 – 70.

[2] J. Chen, H. Lu, S. Bruda, A reputation-based approach for countering vulnerabilities in p2p networks, in: 2nd International Conference on E-Business and Information System Security, EBISS2010, 2010, pp. 263 – 266.

[3] L. Lui, W. Shi, Trust and Reputation Management, Internet Computing, IEEE 14 (5) (2010) 10 –13.

[4] P. Resnick, R. Zeckhauser, E. Friedman, K. Kuwabara, Reputation Systems, Communications of the ACM 43 (12) (2000) 45 – 8.

[5] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Computing Surveys 42 (1) (2009) 1–31.

[6] O. Onolaja, G. Theodoropoulos, R. Bahsoon, A Data-Driven Framework for Dynamic Trust Management, Procedia Computer Science 4 (2011) 1751 – 1760, proceedings of the International Conference on Computational Science, ICCS 2011.

[7] O. Onolaja, R. Bahsoon, G. Theodoropoulos, Conceptual framework for dynamic trust monitoring and prediction, Procedia Computer Science 1 (1) (2010) 1235 – 1244, ICCS 2010.

[8] O. Onolaja, R. Bahsoon, G. Theodoropoulos, Trust Dynamics: A Data-Driven Simulation Approach, in: IFIP International Federation for Information Processing, 2011, pp. 323 – 334.

[9] V. Balakrishnan, V. Varadharajan, P. Lucs, U. Tupakula, Trust enhanced secure mobile ad-hoc network routing, in: Advanced Information Networking and Applications Workshops, AINAW'07, Vol. 1, 2007, pp. 27–33.

[10] K. Lin, H. Lu, T. Yu, C. Tai, A reputation and trust management broker framework for web applications, in: Cheung, W and Hsu, J (Ed.), 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Proceedings, 2005, pp. 262–269.

[11] O. Bamasak, N. Zhang, A distributed reputation management scheme for mobile agent based e-commerce applications, in: 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Proceedings, 2005, pp. 270 – 275.

[12] M. Rafsanjani, A. Moveghar, F. Koroupi, Investigating Intrusion Detection Systems in MANET and Comparing IDSs for Detecting Misbehaving Nodes, in: Proceedings of world academy of Science, Engineering and Technology, Vol. 34, 2008, pp. 2070 – 3740.

[13] F. Darema, Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements, in: International Conference on Computational Science, 2004, pp. 662–669.

[14] C. Douglas, Dynamic Data Driven Applications Systems, in: International Conference on Computational Science ICCS (3), Vol. 5103 LNCS, 2008, pp. 3–4.

[15] C. Macal, M. North, Tutorial on agent-based modelling and simulation, Journal of Simulation 4 (2010) 151–162.

[16] C. Macal, M. North, Agent-based modeling and simulation, in: Proceedings of the 2009 Winter Simulation Conference (WSC 2009), 2009, pp. 86 – 98.

[17] S. Buchegger, J. Le Boudec, Performance analysis of the CONFIDANT protocol (Cooperation of nodes: Fairness In Dynamic Ad-hoc Networks), in: Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, 2002, pp. 226–236.

[18] S. Buchegger, J. Le Boudec, Self-policing mobile ad hoc networks by reputation systems, IEEE Communications Magazine 43 (7) (2005) 101–107.

[19] B. Yang, H. Garcia-Molina, Designing a super-peer network, in: International Conference on Data Engineering, 2003, pp. 49 – 60.

[20] P. Michiardi, R. 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, Vol. 100, Kluwer, B.V., 2002, pp. 107–121.

[21] S. Kamvar, M. Schlosser, H. Garcia-Molina, The Eigentrust algorithm for reputation management in P2P networks, in: Proceedings of the 12th international conference on World Wide Web, WWW '03, ACM, 2003, pp. 640–651.

[22] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, X. Li, An Empirical Study of Collusion Behavior in the Maze P2P File-Sharing System, Proceedings of the 27th IEEE International Conference on Distributed Computing Systems 0 (2007) 56.

[23] A. Srinivasan, J. Teitelbaum, W. Jie, DRBTS: Distributed reputation-based beacon trust system, in: 2nd IEEE International Symposium on Depenable, Autonomic and Secure Computing, 2006, pp. 277 – 283.

[24] M. Srivatsa, L. Liu, Securing decentralized reputation management using TrustGuard, Journal of Parallel and Distributed Computing 66 (9) (2006) 1217 – 1232.

[25] S. Ganeriwal, L. K. Balzano, M. B. Srivastava, Reputation-based framework for high integrity sensor networks, ACM Transactions on Sensor Networks 4 (3) (2008) 15:1–37.

[26] H. Chen, H. Wu, J. Hu, C. Gao, Event-based trust framework model in wireless sensor networks, in: NAS '08: Proceedings of the 2008 International Conference on Networking, Architecture, and Storage, IEEE Computer Society, 2008, pp. 359–364.

[27] Q. He, D. Wu, P. Khosla, SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks, in: Proceedings of WCNC Wireless Communications and Networking Conference, Vol. 2 of IEEE Wireless Communications and Networking Conference, 2004, pp. 825–830.

[28] J. Hu, M. Burmester, LARS - A locally aware reputation system for mobile ad hoc networks, in: Proceedings of the ACM SE Regional Conference, Vol. 2006, 2006, pp. 119 – 123.