DYNAMIC DATA-DRIVEN FRAMEWORK FOR REPUTATION MANAGEMENT

by

OLUFUNMILOLA OLADUNNI ONOLAJA

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
5th November 2012

Abstract

The landscape of security has been changed by the increase in online market places, and the rapid growth of mobile and wireless networks. Users are now exposed to greater risks as they interact anonymously in these domains. Despite the existing security paradigms, trust among users remains a problem. Reputation systems have now gained popularity because of their effectiveness in providing trusted interactions.

We argue that managing reputation by relying on history alone and/or biased opinions is inadequate for security, because such an approach exposes the domain to vulnerabilities. Alternatively, the use of historical, recent and anticipated events supports effective reputation management.

We investigate how the dynamic data-driven application systems paradigm can aid reputation management. We suggest the use of the paradigm's primitives, which includes the use of controller and simulation components for performing computations and predictions.

We demonstrate how a dynamic framework can provide effective reputation management that is not influenced by biased observations. This is an online decision support system that can enable stakeholders make informed judgments. To highlight the framework's usefulness, we report on its predictive performance through an evaluation stage. Our results indicate that a dynamic data-driven approach can lead to effective reputation management in trust-reliant domains.

for Dad and Mum

Acknowledgements

Firstly, I would like to thank my Supervisor, Dr. Rami Bahsoon for his support from the inception of this research. Throughout the study he has given invaluable suggestions and ideas. His contribution and guidance towards my PhD experience has been of great impact, which will always be remembered. This research has been funded by the School of Computer Science, University of Birmingham and I am grateful for the enabling environment that the School has provided. The members of my thesis group - Prof. Uday Reddy and Dr Volker Sorge, have provided valuable perspectives that helped in directing the course of this research. The biannual meetings were very beneficial and always gave me inspiration to continue; I am delighted to have worked with them.

This thesis will not be complete without my acknowledging some of the people that have been of immense help to me. Many thanks to some of my friends who studied drafts of this thesis - Tolu Fapojuwo, Adetayo Bankole, Freda Raymond-Obi, Ifedolapo Fajinmade, Debola Ogunoiki, Dayo Fashina and Kanayo Eruchalu. My thanks also goes to Rehana Yasmin and Vivek Nallur, who helped to critically review the initial drafts of the thesis. I am indeed indebted also to my senior friends - Tope and Debola Adeleye that have provided moral support and encouragement. In addition, my gratitude goes to Dr. Georgios Theodoropoulos, who was part of my PhD at the start, gave very valuable

insights and advice, and helped to shape the research in the right direction. Furthermore, I am grateful for all the good friends that have contributed to my success in different ways. I am constrained by space; else I would have listed all their names here. However, I feel highly honoured to have had great and supportive friends through this research.

I am thankful for my siblings - Adesola, Seun, Lanre, Taiwo and Kehinde, I am fortunate to have and to call them family.

Finally, I am extremely appreciative of my parents Prof. and Mrs Onolaja. Their support and prayers have gone a long way in helping me to achieve all that I have. They have taught me to always believe and not to give up but to push till something happens; they are both truly appreciated and this thesis is dedicated to them.

Contents

List of Tables

List of Figures

Part I

The purest treasure mortal times afford is

spotless reputation; that away,

men are but gilded loam or painted clay.

William Shakespeare

Chapter 1

Introduction

1.1 Background

The expansion of the global computing infrastructure such as the Internet, cloud etc, has raised new and difficult security challenges, such as anonymous collaboration among participants and changes in environmental conditions. Likewise, the global infrastructure is highly dynamic with continuously appearing and disappearing entities, services and changes in entity behaviour. In response to these challenges, the goals of reputation and trust management in the infrastructure are to allow collaborating entities reason about the trustworthiness of each other and to make security decisions on the basis of trust. These goals require the development of computational reputation and trust models that enable the entities to reason about trust and verify the security properties of any collaboration and interaction. For these reasons, it is essential that associated computational models are able to incorporate effective reputation management into their functionalities [IFI11].

Furthermore, with the increase in the use of the Internet, mobile devices, computers, online market places, and wireless networks, users are exposed to greater risk as they collaborate with one another. In order to reduce risk and improve performance (effectiveness and reliability), applications on the infrastructure must manage trust relationships among users, by motivating cooperation and honest participation. Introducing trust to such large-scale distributed applications and domains is a difficult challenge, but one well suited for reputation and trust management[SAB10].

During the past decade, reputation and trust management has provided cogent answers to emerging challenges in the global computing infrastructure relating to computer and network security, electronic commerce, virtual enterprises, social networks and cloud computing (in terms of trusted communications). It is vital in establishing a healthy and efficient collaboration among a group of participants and players that might have insufficient prior knowledge about each other [LS10]. This thesis aims to contribute to reputation management field by catering for the dynamism in the interactions and relationships among anonymous users.

1.2 Motivation

Computational reputation and/or trust is central to the use of the global computing infrastructure, especially in domains that are based on collaboration among members. Over the years, as these trust-reliant domains have evolved, risky collaboration with potentially unknown and misbehaving parties has resulted in new attacks. The need to minimise these risks resulted in the development of several Reputation and Trust-based Models (RTMs) that have been shown to be useful. Despite the proliferation of the models, there remain outstanding challenges of prediction, dynamism and susceptibility to attacks in the domains. These challenges include: the dynamic nature of reputation and trust, likely changes in entities' behaviour over time, dynamics of their interactions, emerging behaviour resulting from their interactions, new or emerging computing paradigms in global computing and their security concerns, new or unanticipated attacks, group dynamics and formulations, etc.

Additionally, behavioural expectation in any context can be motivated from a social perspective, where individuals within a society are expected to behave in certain ways. A disreputable person could redeem himself through consistent honest actions over time whilst a trusted person could become less reputable if they demonstrated deceit over time [AD05]. This implies that reputation can change randomly over time, and is therefore dynamic. We refer to this nature as *Trust Dynamics* in this thesis.

In the RTM research domain, it is usually assumed that the predictive power of a reputation and trust management model depends on the supposition that past behaviour is an indication of future behaviour. This premise is not always effective because it results in attacks on the reputation management system itself [OTB11, OBT12].

This research is motivated by these challenges. In order to enhance trusted communications among a domain of participants, reputation and trust management models should provide the essential primitives for reliable reputation and trust management, address trust dynamics and the making of predictions. Also in response to these challenges, this thesis proposes and investigates Dynamic Data-Driven Framework for Reputation Management (D3-FRT), which is a decision support system, that exploits the Dynamic Data-Driven Application Systems (DDDAS) [DDD06] approach for managing reputation.

1.3 This Thesis

Over the last decade, RTMs [BLB02, GBS08, MM02, QHC06] have gained popularity over the years borrowing ideas from both Game Theory and Bayesian networks. These models are described as systems that provide mechanisms to produce a metric encapsulating reputation for each identity in any given application domain [HZNR09]. The metric is subsequently referred to as Reputation Value (RV) in this thesis.

Basically, RTMs aim to provide information that allows members to distinguish between trustworthy and untrustworthy members. The models encourage the cooperation of domain members by the provision of incentives and discourage maliciousness by using punishment schemes such as isolation and service denial [OTB11]. RTMs have been adopted in applications that rely on members' cooperation in order for the application to function correctly. The models have been used extensively in various electronic commerce and online communities such as YouTube, Amazon and eBay. Some literatures also suggest their use in domains ranging from Peer-To-Peer (P2P) to mobile networks.

From our study of some of these models, we concluded that whilst they aim to solve trust related issues in their domains, the models invariably introduce other issues. There is therefore a need for a framework (presented in this thesis), that is capable of providing dynamic trust ratings at runtime and predicting the future ratings of entities within a domain. The framework does not only rely on collective opinion and ratings to determine the reputation of domain entities. Instead the framework uses its simulation, feedback and control mechanisms to make predictions about a potential compromise before it occurs. For these reasons, D3-FRT which is proposed in this thesis, is a proactive decision support system that provides useful information about domain events to enable the making of informed decisions. Therefore providing the necessary primitives for addressing the challenge of reputation and trust dynamics.

1.4 Research Perspective

Despite recent advances in the state-of-art and practice in areas related to computational models and frameworks for managing reputation and trust in networks, the problem of managing trust dynamics, which we explicate in this research, is still an open research challenge. In particular, this research by and large aims at understanding:

- The problem of trust dynamics. We consider how managing trust dynamics can be fundamentally different from other assumptions.

- The usefulness of anticipating futuristic events in the prediction of reputation. More precisely, we analyse the relationship between domain and anticipated events to identify any correlations.

- The underlying architectural choices, design decisions and primitives which are necessary for realising such requirements and for engineering frameworks for managing reputation and trust dynamics. In this context, we argue that the effectiveness of any dynamic reputation and trust management system is sensitive to the choices of the underlying architectural style - i.e. whether it is fully distributed, centralised or a combination of both architectures.

These are studied using relevant scenarios through our proposed and unique semi-distributed approach for predictive reputation management; a first in its kind. Our novel framework sought inspirations from the generic DDDAS paradigm and adopts much of its underlying primitives. The research aims to understand the extent to which DDDAS and its primitives can induce new methods for managing reputation and trust dynamics.

1.5 Introductory Example

This section introduces an application that is representative of the class of trust and reputation-reliant applications that this research targets. In particular, the focus is on the issues of trust dynamics, and the predictability of trust. The aim of this sub-section is to highlight these issues and introduce the value of this research to the reputation landscape.

1.5.1 Trust in Traffic Monitoring and Management

Consider a network of mobile sensor nodes that are deployed along the roadside to monitor vehicular movement in order to obtain real traffic flow data and conditions. The network runs a reputation and trust management system where the sensors are equipped with wireless interfaces with which they form the network. The nodes can sense, measure and gather information from the environment and, based on predetermined or some local decision process [YMG08], transmit the data to a central control server or network controller. This information is used by the server for decisions about the nodes and for traffic control in order to, for example, judge the cause of a road accident, traffic jam or redirect traffic to other routes.

Collaboration

The effectiveness of the traffic management system is dependent on the expected behaviour of the nodes. Nodes collaborate to collect and process the data that generate information about traffic conditions. When a sensor node receives information from another, it is combined and fused with local information. The information is sent to all nodes in the path for the same processing before being sent to the server.

Each node watches the transmission of neighbouring nodes, and then reports any deviation from expected behaviour and makes recommendations. In a normal situation, if a sensor node A needs to forward a message M to another node D, it relays the message to its next hop neigbour B and B forwards the message to C. Node C then relays M to node D.

Challenges: The nodes are vulnerable to various internal and external attacks due to their dynamic and volatile nature. An adversary may compromise a sensor node, which in turn compromises other nodes. It is possible that B colludes with C and does not report to A when C alters message M to $M\#$, before spreading the bogus message (*collusion attack*). Although a message was delivered to the recipient, the originator is unaware that a wrong message has been delivered. Therefore, trust decisions to be made by the controller have been corrupted through recommendations made by the colluding nodes.

Can nodes be trusted to provide accurate information about domain events? How can the system quickly and accurately distinguish misbehaving nodes from trusted nodes? How can misbehaviour be penalised and how can expected behaviour be effectively encouraged? How can the system identify colluding nodes that cover-up for one another?

The trust management system should be designed to be reliable and effective. Reputation and trust have to be predictable for each entity in the domain. The system should not rely only on the collective opinion and recommendations of the domain entities in making trust decisions.

Dynamism

The reputation management system provides a RV representing the reputation of each wireless sensor node in the network. The nodes are rated according to behaviours they exhibit, for example, sending more conflicting information when compared to other neighbouring nodes results in a negative rating whilst a consistent propagation of correlating information earns a positive value. After each event in the network, nodes' trust values are updated using the rating earned for the event. Past events that resulted in a node's current value help the reputation system identify nodes in the network that can be trusted.

Challenges: A node sends accurate information for an extended period of time in order to earn high ratings and trust value. The node subsequently suddenly begins to misbehave by propagating inaccurate information about vehicular movement across the network (*intoxication attack*).

Will the provision of recent ratings aid the identification of sudden behavioural changes? How can the anticipation of misbehaviour help in preventing this form of attack?

The assumption that past behaviour is an indication of futuristic behaviour is not valid with sudden behavioural changes and is inadequate in providing a trusted system. Historical events in the network should have less effect on more recent events when computing the trust of each member. Sudden changes in node behaviour should be captured and analysed by the reputation system. If misbehaviour is anticipated using historical, current and possible future events, such misbehaviours can most likely be prevented.

The example in this section has highlighted the consequences of ill-management of reputation trust, and therefore, highlighting the need for a new framework that focuses on the dynamic nature of reputation and trust.

1.6 Objectives and Contributions

The objective of this thesis is to investigate a novel approach for managing the dynamics of reputation. The goal is to be able to facilitate dynamic reputation computation, prediction and provision of useful information for domain stakeholders such as the network administrator. We aim to suggest solutions to the issue of corruption of trust decisions by domain entities, also to distinguish between members, penalise misbehaving ones and provide dynamic reputation management.

This thesis presents a piece of work that can be useful in critical domains such as in military networks, traffic management systems, and mobile ad-hoc or sensor networks and so on. Therefore, this thesis contributes a simulation framework for reputation management. The contribution is sub-divided in the following.

- In DDDAS an application can accept and respond dynamically to new data, and in reverse, the application can dynamically control itself. In essence, the DDDAS unifies complex computational models of a system with real-time data acquisition and aids the controlling of the system. Our research is the first of its use of the DDDAS paradigm for reputation management. This thesis shows that the use of monitoring, simulation, and feedback in terms of prediction and control mechanisms, can potentially improve on the reliability of systems that rely on reputation management to function.The proposed framework has the capability of providing a high level of dynamism by updating the ratings of domain members as the collaborate.

- The general assumption that past histories indicate future events has been found not to be effective and reliable. This is because the domain is then exposed to vulnerabilities such as *intoxication*.

We present in this thesis, an approach that does not rely on historical data alone for trust decisions. The approach also considers more recent data and anticipates possible futuristic events that could occur in the domain.

- In a distributed architecture for reputation and trust management, each participant maintains the trust information of other participants, whereas in a centralised system, a trusted entity manages the reputation of all participants. A semi-distributed reputation management system is more desirable as it combines the advantages of both distributed and centralised architectures.

  When compared to completely distributed reputation and trust-based models, a semi-distributed predictive framework may have a relatively higher performance overhead, but is timelier in terms of misbehaviour detection. The semi-distributed architecture for reputation management that is described here will be most applicable in niche environments where the security requirement is more critical compared to other non-functional requirements.

- As it was described earlier in Section 1.5.1, trust decisions that is generated by a reputation system can be corrupted by relying on domain members. In the approach suggested in this thesis, decision making is not dependent only on the collective opinion of domain members, but also based on observations captured by the controller of domain events within a specific time frame. This form of monitoring and feedback system prevents collusion attack among domain members. The *simulation* component of D3-FRT makes prediction about future events in the system. These predictions are not only useful at the network level but at a higher level, providing adequate and timely information that allows for countermeasures and for enabling the making of informed and security-aware decisions by stakeholders. The prediction gives the system enough time for preventive measures, making the framework proactive compared with some other models. The framework can be proactive in terms of providing controls such as downgrading the rating of suspect or misbehaving members before they perpetuate an attack.

1.7 List of Publications

The work presented in this thesis is based on and extends several papers that have been published in the last three and a half years. The papers include:

[OBT12]: O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Agent-based trust management and prediction using D3-FRT. In *ICCS'12: Proceedings of the International Conference on Computational Science in the Journal of Procedia Computer Science, volume 9*, pages 1119-1128, 2012.

[OBT11]: O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Trust Dynamics: A Data-Driven Simulation Approach. In *IFIPTM'11: 5th IFIP WG 11.11 International Conference on Trust Management of International Federation for Information Processing (IFIP), volume 358 of Advances in Information and Communication Technology (AICT)*, pages 323-334, Springer, 2011.

[OTB11]: O. Onolaja, G. Theodoropoulos, and R. Bahsoon. A Data-Driven Framework for Dynamic Trust Management. In *ICCS'11: Proceedings of the International Conference on Computational Science in the Journal of Procedia Computer Science, volume 4*, pages

1751-1760, 2011.

[OBT10]: O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Conceptual Framework for Dynamic Trust Monitoring and Prediction. In *ICCS'10: Proceedings of the International Conference on Computational Science in the Journal of Procedia Computer Science, volume 1*, pages 1235-1244, 2010.

[OBT09]: O. Onolaja, R. Bahsoon, and G. Theodoropoulos. An Architecture for Dynamic Trust Monitoring in Mobile Networks. In *OTM'09: 4th International Workshop on mobile and networking technologies for social applications, volume 5872 of LNCS*, pages 494-503, 2009.

This thesis should be regarded as the definitive account of the work.

1.8 Roadmap of Thesis

This thesis is made up of four parts that are sub-divided into eight chapters. Part I includes the first two chapters, introduces and gives the background of this thesis. Chapter 1 provides the motivating context and the focus of this thesis. In Chapter 2, the concepts of reputation, trust and reputation management are introduced. The chapter discusses the usefulness, objectives and some problems of reputation and trust management systems.

Part II details an overview of reputation and trust-based models and a background of the DDDAS paradigm. Chapter 3 considers the existing problems of reputation and trust management and critically reviews literature focused on reputation and trust management models with proposed solutions to the trust related problems that have shown useful results. The chapter describes how the existing models attempt to solve the problems; each with its merits and faults. Comparative and gap analysis of the models are discussed extensively in this chapter. Chapter 4 introduces the dynamic data-driven application systems paradigm and motivates the usefulness of the paradigm for reputation management. The chapter discusses the DDDAS computational model in the context of examples of novel capabilities enabled through its implementation in different application areas. In addition, notable literature that adopts the paradigm is comprehensively described. The chapter elaborates on the issues of trust dynamics. We also describe how the DDDAS coupled with an agent-based simulation approach can be useful for the problem of trust dynamics.

In part III, we present the approach adopted in this thesis and an evaluation. Chapter 5 builds on the gaps in the literature that are identified in Chapters 3 and 4. The chapter introduces the novel framework (D3-FRT) for the predictive reputation management system that adopts a rich agent-based simulation approach. The chapter describes how D3-FRT is capable of providing dynamic trust ratings of domain members at runtime and making predictions. Chapter 6 applies the proposed framework in a network situation to examine its effectiveness in providing trusted communications among participants. D3-FRT's performance in terms of its predictive capability, dynamism, and scalability and in varying scenarios is evaluated in this chapter.

Part IV summarises this thesis. Chapter 7 presents the overall conclusions based on the work done, reviews the contributions that this thesis has made and details some promising directions for future work.

Finally, Appendix A gives the attack modelling on the reputation system as they have been implemented in this research and in Appendix B, a sample of the data that is generated by the D3-FRT is presented.

Chapter 2

The Notion of Reputation and Trust

2.1 Background

The spread of Internet usage, proliferation of mobile devices, computers, and online market places, as well as the rapid growth of wireless networks and other related domains have changed the landscape of security. Users are exposed to greater risks as they collaborate anonymously with one another within diverse domains. The domains rely primarily on cooperative user behaviour for their effective operations because without this cooperation, they cannot fulfil their functions.

In order to reduce risks and improve performance, applications must manage trust relationships between users, by motivating cooperation and honest participation [SAB10]. P2P networks for example have undergone rapid progress and significant developments in recent years in this regard. However, due to their anonymous and open nature, malicious users can abuse the system by disseminating bogus files or acting together to commit as much damage as possible (collusion attack) [OBT12]. For such networks to be effective in fulfilling their purpose of anonymous sharing, they should be relatively reliable, efficient and secure [CLB10]. Moreover, Internet applications have evolved from centralised and private computing platforms to distributed and collaborative computing systems. Collaboration is in fact today a fundamental Internet computing requirement.

Reputation and Trust Management is well suited for the requirements and the research is highly interdisciplinary [LS10], involving researchers from networking and communications: Mobile Adhoc Networks (MANETs)[1], Wireless Sensor Network (WSNs)[2] and P2P, data management and information systems, e-commerce and online communities: YouTube, Amazon and eBay, Artificial Intelligence, and also the Social Sciences and Evolution Biology. The concepts of *trust* and *reputation* have been developed into Reputation and Trust-based Models (RTMs). These models have gained popularity because they have been shown to be promising in the area of reputation and trust management as they aim to collect, aggregate, and disseminate feedback about a user's behaviour, based on some predetermined premise.

Reputation and trust management is useful for establishing healthy and efficient collaborations among a network of participants and users that might not have sufficient prior knowledge about each other [LS10]. For example, eBay has several millions of auctions simultaneously open, serving as a listing service where buyers and sellers assume all associated risks with transactions [RZFK00]. There are occurrences of fraudulent transactions however there is a higher rate of successful transactions as well, which are primarily attributed to the reputation system on eBay called the *Feedback Forum.* This is

[1] Infrastructureless networks that have no fixed routers; all nodes are capable of movement and can be connected dynamically in an arbitrary manner [RCK99]

[2] These are a large number of densely deployed nodes that mainly use broadcast communication paradigm to monitor arge or complex physical environments [ASSC02]

a prime example of a RTM which is getting to be a commonplace in many other domains as well.

Despite the proliferation of RTMs in the trust-reliant domains, ensuring trusted collaborations among participants remains a challenge. Several unaddressed threats (discussed in the later part of this chapter) still limit the effectiveness of reputation systems.

Furthermore, some of the existing RTMs in literature focus on historical and recent 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 any domain.

2.2 Reputation and Trust

The reputation of a node in the survey of Djenouri *et al.* [DKB05] is the amount of trust that a trustor grants a trustee regarding the trustee's cooperation and participation in the domain. In this thesis, an entity is said to be trusted when it has obtained positive feedbacks from its interactions with others coupled with high ratings in a domain.

Reputation and trust have been used interchangeably and synonymously in literature. However, some studies [JIB07, BAS09] indicate that whilst there is some correlation between them, there is a clear difference between the concepts. An example is given by Josang *et al.* in [JIB07] detailing the differences between the concepts as:

1. Trust systems produce a score that reflects the relying party's subjective view of an entity's trustworthiness, whereas reputation systems produce an entity's reputation score as seen by all.

2. Transitivity is an explicit component in trust systems, whereas reputation systems usually only take transitivity implicitly into account.

3. Trust systems usually take subjective and general measures of trust as input, whereas information or ratings about specific events, such as transactions, are used as input in reputation systems.

Furthermore, Momani and Challa in [Mom10] defines trust as a derivation of the reputation of an entity. In the work, the authors state that based on the reputation, a level of trust is bestowed upon an entity. By motivating from a social perspective, reputation is built over time based on a person's history of behaviour, and determines the level of trust that is bestowed on the person in a society. Reputation is said to be the amount of trust inspired by a particular member of a community in a specific domain. These definitions show the interlink between both concepts (trust and reputation); which is also the position of this thesis. For simplicity, this thesis describes reputation as the overall status of an entity within a particular domain and context while an entity is said to be trusted if it obtains high ratings from its interactions (reputation). Also, we focus on reputation rather than the subjectivity of trust, the time-dependency of reputation and direct trust relationships are highlighted in this thesis.

2.2.1 Properties of Reputation and Trust

The general properties of reputation and trust in the work of Liu and Issarny [LI04], and Adam and Davies in [AD05] are discussed below:

- **Subjective**: The subjective property of trust implies that an entity in a domain may have different RVs depending on the individual perception of other entities.

That is, node A might have a very good reputation as perceived by B, but just an average reputation with C.

- **Context Dependent**: Reputation is context dependent because it is possible for an individual to exhibit contrasting behaviours in different situations. An example is a domain member that may be a good seller but not a good buyer on an online auctioning system.
- **Time Dependent and Dynamic**: From the social perspective, a person's behaviour in a society can fluctuate with time and the person can be classified as reputable at a certain time but become less reputable at another time. This implies that trust (and reputation) change over time and are therefore, dynamic and time dependent.
- **Transitive**: Transitivity can be described using the example: if a node A trusts B and B trusts C then A trusts C. There is a conflict about the transitivity of trust in literature. This is due to the fact that trust can either be treated as subjective or global. When trust is regarded as subjective, then it is not transitive whereas the reverse is the case when it is global [BAS09]. In this research, trust is treated as a global property and is therefore transitive.

2.2.2 Trust Relationships

Trust relationships, which are a major component of reputation management, shows the connections between participants. They help in capturing the behaviour of domain members through monitoring, whether cooperative or otherwise as evidence to compute their level of trustworthiness and of course, reputation.

According to Srinivasan *et al* [STW^{+}08], trust relationships can be broadly classified into:

1. **First-hand (direct trust)**: This is a direct trust relationship that represents a trust assertion of a member (*trustor*) about another (*trustee*). It is a one-to-one trust relation between both parties. The trust relation is maintained locally by trustors and represents the trustors' personal opinion about the trustee [SRIT11]. The system therefore uses direct observation or its own experience to update reputation. Typical examples of this are a trust relationship among friends in a social context such as on Facebook.

2. **Second-hand (Recommendation-based trust)**: This is a third-party's opinion about another entity. It can either be a one-to-many or many-to-one relationship. In this relationship, the system uses information provided by domain members about a certain participant.

Most systems proposed so far use both first-hand and second-hand information to update reputation. This allows the system to make use of the experience of its neighbours to form its opinions. Some systems choose not to use both types of information. In systems that use only first hand information, a node's reputation value of another node is not influenced by others.

2.3 Reputation and Trust-Based Models

RTMs have been studied, applied and found useful in diverse disciplines such as in Social Sciences, Economics and Computer Science. In Computer Science, RTMs aim to provide mechanisms to produce a metric encapsulating reputation (referred to as *Reputation Value*) for a given domain for each identity in the system [HZNR09].

These models are essential for establishing a healthy and efficient collaboration among a network of participants and players that may not have prior knowledge of one another [LS10]. An illustration can be given from some online auctioning systems that provide a means of obtaining quality ratings of participants of transactions by having the buyer and seller give each other feedback.

The main goals of an RTM as identified by Srinivasan *et al.* in [STW$^+$08] are to:

1. Provide information that allows nodes to distinguish between trustworthy and non-trustworthy members;
2. Encourage members to be trustworthy by incentivising good behaviour;
3. Discourage the participation of untrusted members;
4. Cope with any kind of observable misbehaviour;
5. Minimise the damage caused by insider attacks.

Furthermore, when designing reputation and trust management systems, the following properties ([ZH07]) should be considered:

- *High accuracy*: To help distinguish reputable members from non-reputable ones, the system should calculate the trust ratings as close to their real trustworthiness as possible.

- *Adaptability*: In domains where, members join and leave dynamically such as in an open P2P system, the RTM should adapt to these changes rather than being reactive.

- *Scalability*: The RTM should be able to scale to serve a large number of members in term of accuracy, convergence speed, and extra overhead per member.

- *Robustness to misbehaviour*: The system should be robust to various attacks by whether from independent domain members or collective.

- *Fast convergence*: Behaviour changes are expected in dynamic domains causing reputation to vary over time. Therefore, reputation aggregation should converge fast enough to reflect the true changes in behaviours.

- *Low overhead*: The system should only consume limited computation and bandwidth resources for reputation monitoring and evaluation.

Additionally, RTMs are a very robust solution to curtail insider attacks compared to the use of cryptography and other solutions. This is because the misbehaving or compromised nodes are a part of the domain and have access to the cryptographic keys of other members. However, RTMs also come along with added overhead, both in computation and communication and add another dimension of security consideration. That is, the fact that an adversary has another vector: the reputation system itself, to attack the system with.

2.3.1 Misbehaviour

Behavioural expectation in any domain can be considered from a social perspective, where individuals are expected to behave in certain ways within the society. The behaviour of an individual, whether good or bad, will determine how others will cooperate with the individual. Using the introductory example in Chapter 1, the expected behaviour of a sensor node set up for monitoring vehicular movement is expected to be cooperative with neighbouring sensor nodes in collecting and transmitting honest and observed data.

Misbehaviour is the deviation from the norm, i.e. the expected behaviour in the domain. Entities that misbehave are said to be *untrusted* or *misbehaving* in this thesis. An example of misbehaviour among nodes in a Mobile Adhoc Network (MANET) is *packet dropping* where a malicious node selectively drops packets that should be forwarded. Similarly, malicious peers in a file sharing P2P network may change or forge files, resulting in lack of integrity for the files in the networks; this is referred to as *modification*.

Misbehaviour can either be *selfish* or *malicious*. Nodes may for example selfishly save battery power or memory by not forwarding packets that are of interest in a network. However, malicious misbehaviour tends to maximise damage to the system even at the cost of the nodes benefit themselves. Hence, the only remedy for such members is exclusion from the domain [STW$^+$08].

Furthermore, misbehaving members can employ different strategies, like *brain washing*, *collusion*, *intoxication* and *identity spoofing* in influencing their reputation.

- *Brain washing* depicted in Figure 2.1, is described as when colluding lying nodes surround a node A for example, and the node is tricked into believing false information. When the node moves into a different neighbourhood with honest nodes, it will not

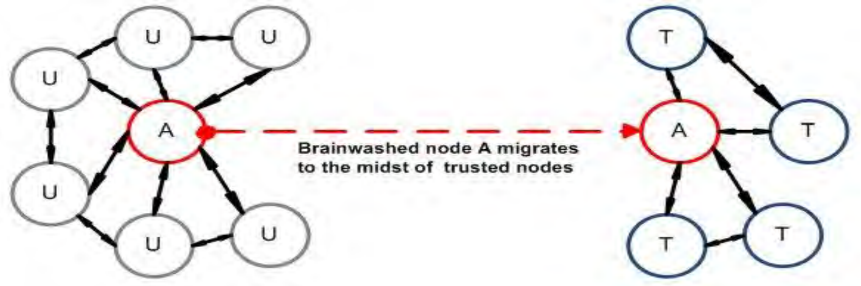


Figure 2.1: Brainwashing (U-untrusted, T-trusted)

believe them since their information deviates too much from its own. This is a form of collaborative attack because it takes two or more nodes to collude in order to brainwash another.

- A node may try to gain trust from others by behaving as expected over a sustained period of time and at a later time (when it has built a good reputation) it starts to misbehave as shown in Figure 2.2; this is referred to as *intoxication*. An example is with an online auctioning system with incidents such as buyers building a high rating with low-valued transactions and then suddenly misbehaving with a high-valued one.

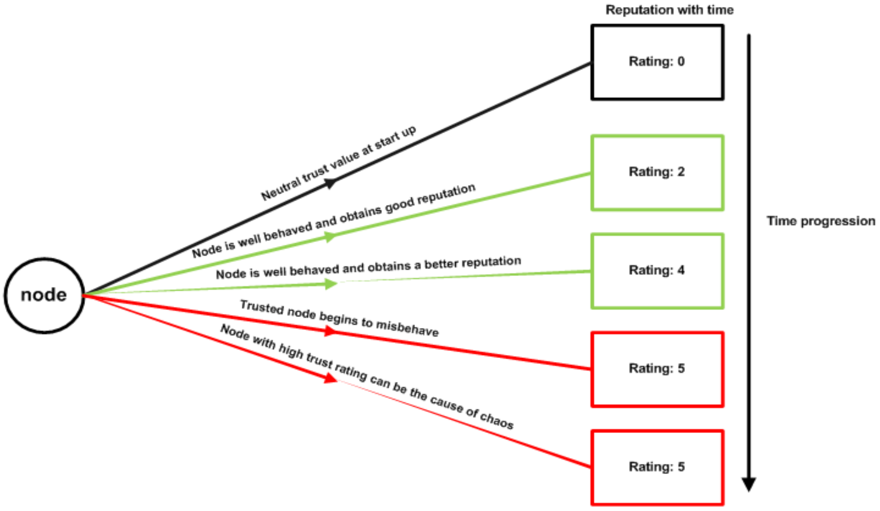

Reputation with time
Rating: 0
Rating: 2
Rating: 4
Rating: 5
Rating: 5
Time progression
node
Neutral trust value at start up
Node is well behaved and obtains good reputation
Node is well behaved and obtains a better reputation
Trusted node begins to misbehave
Node with high trust rating can be the cause of chaos

Figure 2.2: Node behaves as expected for some time to build a good reputation and suddenly begins to misbehave, such that the reputation fails to detect the node

- *Identity spoofing* in Figure 2.3 can occur in a network where there is no Identity

Persistence (IP) and allowing a badly rated node the chance to disappear and reappear with a different identity [AEKHES08, STW+08].

The existing RTMs fuse data from multiple sources in order to produce meaningful results for the end user. There is a reliance on domain members to provide accurate information in the models. An example of the problem that arises from the reliance on such recommendations is collusion, where two or more members team up to behave maliciously. Without countermeasures, the effects of this attack have been shown to dramatically affect the network performance as evidenced in poor reliability and quality of service, higher overhead and throughput degradation [BLB02, GBS08, LJT04]

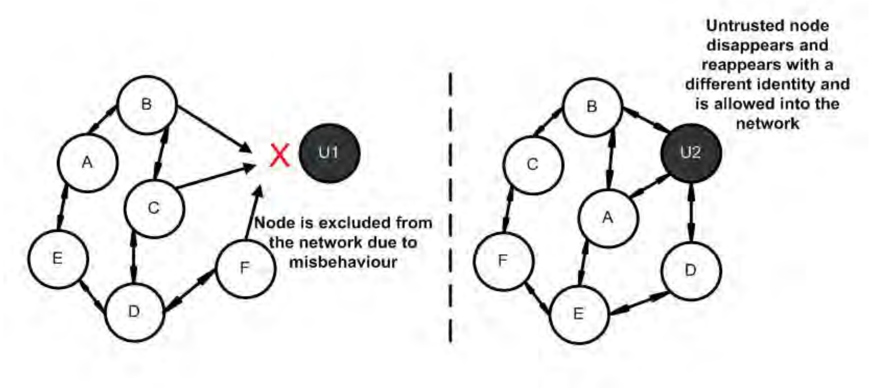

B
A
C
E
D
F
X
U1
Node is excluded from the network due to misbehaviour
Untrusted node disappears and reappears with a different identity and is allowed into the network
B
C
U2
A
F
D
E

Figure 2.3: The node returns with a different identity and is allowed to collaborate.

2.3.2 Reputation System Architectures

There are two main types of RTM architectures, the *centralised* and *distributed* architectures. In centralised architectures, trust is managed by a trusted central server(s) (*Trusted Third Party* (TTP) also referred to as *controller* in this thesis) 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 [eBa12] that is managed centrally 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 over a couple of months. The RTM uses a positive feedback percentage to rate members, which is:

$$\frac{Positives}{Positives + Negatives} * 100 \tag{2.1}$$

where *Positives* are the number of positive feedbacks within 12 calendar months and vice versa for *Negatives*.

This simple reputation 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. Also generally, the downsides of centralised architecture are the performance bottleneck of the central entity and the resulting lack of scalability. However, many e-commerce deployments of RTMs such as eBay have successfully utilised centralised systems which allow for long-term storage and better internal auditing of all reputation data [HZNR09].

Contrarily, a purely distributed approach requires each entity or domain member to maintain trust-related information about other nodes in the system. This indicates that the reputation management is determined, spread and shared among the nodes. This approach introduces the additional requirement of obtaining and propagating information across the system. In this member or node-centric architecture, each node monitors, analyses and computes the trustworthiness of other nodes of interest, based on some pre-defined metric(s). However, the low computational power of certain mobile device for example, does not allow for such analysis. Also, a purely distributed approach results in some real-life problems, such as the corruption of trust decisions through recommendations made by nodes. This approach exposes the system to *false praise/accusation* and *collusion attack* and can be considered as biased. These vulnerabilities are attributed to existing RTMs lacking well analysed approaches to determining the bias of each node [BVLT07]. 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.

There is also a semi-distributed approach (described in more details in Section 3.10) which is a combination of centralised and distributed architectures. This approach recognises the need for a distributed architecture but with a form of control to aid reputation management. The semi-distributed approach is adopted in this research where a domain member collaborates in a distributed manner with its reputation not being entirely determined and managed by other possibly biased members, but by a TTP or other entities (this is similar to the architecture of hierarchical intrusion detection systems in [RMK08]).

The ultimate aim of any RTM is to achieve trusted communications among a group of collaborating members by meeting the domain requirements. The downsides of purely distributed and centralised systems make a semi-distributed approach more desirable, as it combines the upsides of both types of architecture. The presence of a TTP in a semi-distributed architecture for trust management is therefore more appropriate allowing for collaboration and at the same time providing unbiased monitoring and feedback in the system, and resulting in a better reputation management.

2.4 Summary

In this chapter, the notions of trust and reputation have been introduced. We discussed how RTMs have been found useful in domains that rely on the cooperation of a network of participants. The models aid the collaboration required and the effectiveness of trust-reliant domains.

The use of reputation and trust management however, introduces an additional overhead, giving an adversary a vector to attack the system which is the reputation system itself. Intoxication, collusion, brainwashing, identity spoofing are some of the attacks that the models are exposed to. Without countermeasures, the effects of these attacks have been shown to dramatically affect the security and performance at runtime [BLB02] as evidenced in poor reliability and quality of service, higher overhead and throughput degradation. In order to address the problems of intoxication and collusion attack in reputation management, this thesis proposes a novel semi-distributed, predictive and dynamic data-driven framework that is described in Chapter 5.

Part II

That men do not learn very much from

the lessons of history is the most important

of all the lessons of history.

Aldous Huxley

Chapter 3

Computational Reputation and Trust Models

3.1 Introduction

Generally, RTMs [BVLT07, BLB02, CWHG08, GBS08, HWK04, MM02, QHC06] provide the mechanism to monitor, gather the behaviour of members in a network and compute reputations of nodes based on the information obtained by monitoring. They also coordinate approaches to storing and exchanging of reputation information among nodes within the domain. RTMs are described as systems that provide mechanisms to produce RVs or trust ratings for each identity in the domain. Generally, RTMs aim to provide information to distinguish between trustworthy and untrustworthy members. As stated in Chapter 1, the trust models encourage members to cooperate by providing incentives and discourage maliciousness by punishment schemes such as isolation and service denial.

Normally, RTMs rely on recommendations provided by entities in the domain to determine the reputation of others. Each model addresses some of the trust issues but not all of the problems or in the process of solving one issue, other issues are introduced. An example of the problems that arise from the reliance on these recommendations is *collusion*. Incentive policies that are used in P2P networks to ensure cooperation between peers are also generally susceptible to collusion attack [LZY$^+$07].

In this section, a comparative literature review is conducted on models that have contributed to reputation and trust management in literature and give insights to developing the framework proposed in this thesis. Researchers proposed trust models to solve trust related issues and they have shown positive results. Foundational distributed frameworks were already based on social trust considerations, in that they evolved trust based on first-hand experiences and recommendations, and they integrated some trust properties: context, subjectivity, and (only later) time. The general concept of trust in Computer Science was proposed by Marsh [Mar94]. Abdul-Rahman and Hailes [ARH00] later proposed the use of recommendations for managing context-dependent and subjective trust, based on Marsh's approach [Que09]. Their model is based on a word-of-mouth mechanism, which allows agents to decide which other agents' opinion they trust more. They use direct and indirect (recommendations) trust and they introduced the semantic distance of the ratings in their model [Mom10].This work was foundational but was architectural in style and for example lacked a process for trust evolution [QHC06]. A notable literature by Riegelsberger *et al* [RSM05], although targeted at the Human Computer Interaction (HCI) community suggests basic requirements in designing a trust system. We will not go into reviewing these papers as the scope of our work is on the infrastructure level. However, 3 of the requirements stated by Riegelsberger *et al* that are relevant to our work include: stable identity, which we refer to as identity persistence in this thesis, traceability accountability, that involves the capability of tracing outcomes from actions, group membership and group identity.

We are specifically searching for the way design decisions are made in infrastructure related models. The design decisions include: 1) susceptibility to collusion, 2) predictive capability, and 3) architecture of the system which serves as the taxonomy for comparison. A number of literature have focused on the problem collusion and one that is noteworthy amongst them is the work of Liu and Issarny [LI04]; they proposed a (reputation-based) trust framework that is robust to both defamation and collusion attacks. [MM02, KSGM03, BLB02] are some other models that are susceptible to the attack as a result of the inherent properties of their approach and assumptions in their work. In terms of prediction, there are only a few literatures on trust management that focus on predictive accuracy. Prominent amongst them is the lightweight distributed trust propagation [Que09] that shows high predictive accuracy on a large real-world dataset, and, in contrast to existing approaches, it is fully decentralised. Other relevant work on reputation predictions include: [HCH08, LC10]. The architecture of trust models determines how information can be gathered, processed and disseminated in the domain; these and related literatures are discussed in Sections 3.10 and 3.11.3 of this chapter.

The selection of the RTMs in subsequent sections is based on the objective that each system is applicable in a one unique network domain and/or work that has been built on by other research. The architectural design is another factor that was put into consideration for our selection. Each system is chosen to provide insights into the assortment of reputation and trust management applications and to show the adaptation of the components of reputation management systems. The models reviewed are those that are infrastructure based with the following design decisions and issues: Information gathering, Monitoring, Information representation, Recommendations and Information sharing, Scalability, Prediction & Dynamism, and Reputation metric. These serve as criteria for comparison between the RTMs; the extent to which the systems meets or fails to meet each criterion is discussed.

Additionally, this chapter describes the problem of the corruption of trust decisions resulting from recommendations made by members with interest in the domain. A gap analysis of pending problems and comparative analysis are given and these serve as the motivation of this research.

3.2 Node Cooperation Enforcement

3.2.1 Collaborative Reputation Mechanism to Enforce Node Cooperation

In the COllaborative REputation mechanism to enforce node cooperation (CORE) [MM02], reputation is formed and updated with time through direct observations and information provided by network nodes. Nodes have to contribute continuously to the community to remain trusted or their reputation will be degraded until they are eventually excluded.

Each node monitors the behaviour of its neighbours with respect to a particular function, observing the execution of the function. These observations are recorded on a *Reputation Table* (RT) stored by every node. The table contains a list of *Reputation Values* (RVs) (referred to as reputation values in this research) representing the behaviour of each node in specific functions and a global RT is used to combine the RVs.

CORE suggests the use of *subjective*, *indirect* and *functional* reputation types. The subjective reputation $[-1, 1]$ refers to the reputation calculated directly through a neighbouring node's observation. Indirect reputation is influenced by information from other members in the network and the functional reputation is the combination of both subjective and indirect reputation with respect to an evaluation/observation criteria.

Pros

- Only positive information is shared in the CORE protocol, therefore preventing the distribution of false negative information about other entities.
- CORE determines the overall reputation of a specific node by considering its reputation in different functions.

Cons

- CORE suffers from collusion attack where misbehaving malicious nodes extend each other's survival time through false praise reports;
- This model unifies the reputation of a node for various functions such as packet forwarding function and routing function into a global reputation value. This approach may not be effective as it allows a malicious node to cover its misbehaviour with respect to one function by being well behaved in others [STW$^+$08].
- The CORE mechanism can only enforce cooperation effectively in non-mobile networks [CN06]. It is ineffective in a dynamic network or with high mobility because the watchdog requires a node to have information about its neighbourhood that may not be available to the node.

3.2.2 Cooperation of Nodes: Fairness in Dynamic Ad-Hoc Networks

Each node maintains a *reputation rating* and a *trust rating* about every other node of interest in the Cooperation of Nodes: Fairness in Dynamic Ad-Hoc NeTworks (CONFIDANT) [BLB02] protocol. The *reputation rating* represents the opinion formed by a trustor about a trustee in the base system (such as the ad-hoc network) while the *trust rating* represents a node's opinion about the honesty of another node. Nodes monitor and detect misbehaviour in their neighbourhood by means of an enhanced *packet acknowledgment* (PACK) mechanism where confirmation of acknowledgment comes indirectly by overhearing the next node's transmissions.

A *cooperation factor* which is the frequency of misbehaviour relative to the cumulative activity of a node is used. Every node i keeps a cooperation factor (R_{ij}) of every other node j. This factor is expressed as a function of the number of misbehaviours and expected behaviours. Second-hand recommendations from other nodes are accepted only if they are compatible, thus reducing the impact of false accusations.

Pros

- Only recent observations are propagated and more weight is given to these than past observations. This prevents the possibility of a node obtaining a good reputation initially and then misbehaving.

Cons

- There is heavy reliance on information provided by other nodes, making it difficult to detect problems when there is a collusion among nodes.
- Only negative information is propagated in the model resulting in false accusation

vulnerability. This makes it simple for a misbehaving node to distribute false information about other nodes, in order to initiate a denial of service attack. Also, a malicious node will realise it has been discovered and change its attack strategy as a result of the propagated information.

- CONFIDANT does not take into account the context in order to determine the overall trust and reputation of network members.
- The model assumes that nodes are authenticated and no node can pretend to be another in order to get rid of a bad reputation.
- Nodes that are excluded from the network recover after a certain period and this gives malicious nodes the opportunity to resume attack unless revoked after a threshold of re-entrance. This model allows malicious nodes to redeem themselves quickly [AD05, STW$^+$08].

3.3 Incentive Based Scheme

In another related study, Secure and Objective-Reputation based Incentive (SORI) [HWK04] scheme, a promiscuous mode of operation is assumed where a node overhears the transmissions of its neighbours (even if the packet is not intended for the node) and maintains a neighbour list. Each node keeps counters of the number of packets requested for forwarding and those actually forwarded, and a reputation rating is calculated from these counts along with the derived confidence metric [BLB05]. The confidence metric relies on two parameters: *Request-for-Forwarding* ($RF_N(X)$) and *Has-Forwarded* ($HF_N(X)$). The $RF_N(X)$ indicates the total number of packets that node N has transmitted to X

for forwarding and $HF_N(X)$ corresponds to the total number of packets that has been forwarded by X and noticed by N.

Given $RF_N(X)$ and $HF_N(X)$, node N creates a local evaluation record denoted by $LER_N(X)$, which contains a confidence metric that is used to depict how confident N is for its judgement on the reputation of X. $LER_N(X)$ consists of two entries, $G_N(X)$ and $C_N(X)$, where $G_N(X) = \frac{RF_N(X)}{HF_N(X)}$ and $C_N(X) = RF_N(X)$.

An incentive scheme is proposed to stimulate packet forwarding, consisting of *neighbour monitoring*, *reputation propagation* and *punishment*. Neighbour monitoring is the collection information about misbehaviour of neighbours and objective quantification of the neighbours" reputation while reputation propagation is aimed at sharing information among neighbouring nodes. Punishment encourages packet forwarding and disciplines selfish nodes.

Pros

- The propagation of reputation is secured by a one-way hash function which makes it difficult for a selfish node with a bad reputation to send packets or fake broadcast information to influence its reputation by impersonating a trusted node.

Cons

- This approach assumes that nodes are not malicious and there is no conspiracy amongst nodes.
- Each node operates in promiscuous mode and even listens to packets not intended for the node.

3.4 TrustGuard

A study by Srivatsa *et al.* in [SXL05, SL06] uses a strategic oscillation guard based on a Proportional-Integral-Derivative controller to combat malicious oscillatory behaviour:

$$TV_n(t) = \alpha * R_n(t) + \beta * \frac{1}{t} * \int_0^t R_n(x)\mathrm{d}x + \gamma * R_n^{'}(t) \qquad (3.1)$$

$R_n(t)$ denotes the rating of a node n at time t and this value can simply be an average of ratings over a recent time period. $R_n^{'}(t)$ denotes the derivative of $R_n(x)$ at $x = t$. α, β and γ are weightings of the three components. $TV_n(t)$, the dependable rating of n at time t is computed using the equation 3.1 above.

A large value for α biases the rating of node n to recent observations and that of β gives a higher weight to past behaviour. A lager value of γ amplifies sudden changes in behaviour of the node (as indicated by the derivative of the rating). The strategic oscillation guard takes as input, the raw reputation values computed from some other reputation system and formulates the output value as the sum of three weighted components:

1. The first component is a node's current performance and is the raw ratings as computed by the underlying reputation system.

2. The second component is the past history of a node's actions formulated as the integral of the function representing all prior reputation values divided by the current point in time.

3. The third component reflects sudden changes in a node's performance and is formulated by the derivative of the above mentioned function.

The averaging nature of the proportional and integral components enables the model to tolerate errors in raw ratings and reflect consistent node behaviour [SXL05].

Pros

- TrustGuard allows for flexibility by giving different trust components varying weights.
- In order to efficiently store and calculate the historical components specified by the strategic oscillation guard's formulation, the concept of fading memories is used. Instead of storing all previous values, TrustGuard represents the values by using only log_2t values, where t represents system time intervals, with exponentially more details stored about recent events. This technique allows the strategic oscillation guard calculations to be deterministically performed with an efficiency of $O(logt)$ instead of $O(t)$ [HZNR09].
- The system can be implemented with different degrees of centralisation and can also be fully distributed.

Cons

- A Trusted Third Party (TTP) is included in TrustGuard's architecture, which could become a performance bottleneck and a single point of failure. The TTP does not become a bottleneck with up to 1024 nodes in the system.
- An assumption in TrustGuard is that the framework is built on top of a secure overlay network which is not always the case.

3.5 Secure MANET Routing with Trust Intrigue

Capturing evidence of nodes' behaviour in a more efficient manner that eliminates the recommender's bias (the recommender is the observing network member) was the focus of Secure MANET Routing with Trust Intrigue [BVLT07] (SMRTI). Evidence of trustworthiness is captured from a broad perspective including direct interaction with neighbours, observing interaction of neighbours and through recommendations.

SMRTI obtains evidence from direct interactions with neighbours in order to identify their benign and malicious behaviours. The evidence captured from recommendations is used to predict whether a node is misbehaving or not. It is claimed that the model is able to address attacks they named: *honest-elicitation* and *free-riding* because nodes do not exchange ratings for recommendation. A malicious node may exhibit honest-elicitation by forwarding high recommendations for colluding malicious nodes. When a node accepts recommendations from other nodes, but fails to reciprocate with recommendations when requested by them, then the node is subject to free-riding.

There is no additional overhead incurred from communicating recommendations but collusion attack is possible in SMRTI because there is reliance on the opinion of domain members to maintain the reputation system.

3.6 High Integrity Networks Framework

Each node maintains reputation metrics that represents past behaviour of other nodes in the Reputation-Based Framework for High Integrity Sensor Networks (RFSN) framework by Ganeriwal *et al.* [GBS08]. RSFN clearly differentiates the difference between *reputation* and *trust* which is similar to the definitions of both terms as given in Chapter 2. The metrics are used as an inherent aspect in predicting future events. RFSN is very similar to CORE and CONFIDANT except for its applicability to sensor networks and the presence of a middleware service that can counter faulty misbehaviour of nodes.

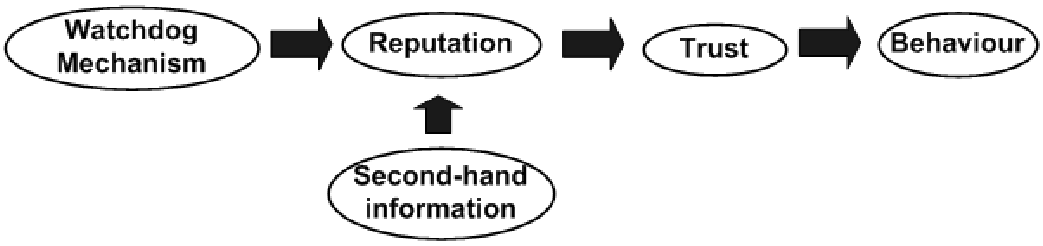


Figure 3.1: Architectural design of RFSN

As depicted in Figure 3.1, RFSN has two key building blocks, *watchdog* and *reputation* and direction of the arrows in the figure represent the flow of information. Each node has a watchdog component and maintains the RV of other nodes. There is a rating specifying the level of confidence associated with every observation made by the watchdog. This rating is any real number between $[0, 1]$. The reputation block manages the reputation representation, updates, integrates and ages reputation and also creates a trust metric output.

Pros

- RFSN offers real-time feedback on the behaviour of members. Only positive reputation information is propagated, as in CORE [MM02].

Cons

- Since nodes are used to validate each other, RFSN gives room for attacks like collusion, brainwashing, false praise and false accusation.

- The assumption that the watchdog mechanism will be present on all nodes is not feasible in real life networks.

- Packet loss influences the performance of a reputation system. In RFSN, it is assumed that there is no packet loss in the communication channel. This is a costly assumption in the applications of RFSN, because when there is a heavy load on the network for example, packet loss is inevitable.

- The model does not provide authentication and confidentiality for the readings made by the watchdog.

3.7 EigenTrust Algorithm

EigenTrust [KSGM03] was motivated from the need to decrease the number of downloads of inauthentic files in a P2P file-sharing network. The global reputation of each peer i is given by the local values assigned to i by other peers, weighted by the global reputation of the assigning peers. The final rating (which ranges between 0 and 1) for each peer j is computed by peer i initially computing a normalised local rating c_{ij} for j based on i's direct observations. Peer i then computes a value for k, by asking other peers, j, for their opinions of peer k. These opinions are weighted by i's opinion of j: $t_{ik} = \sum_j c_{ij}c_{jk}$, where t_{ik} represents the trust that peer i places in peer k based on response from other peers. Writing this in matrix notation, let C be the matrix $[c_{ij}]$ and $\overrightarrow{t_i}$ to be a vector containing the values t_{ik}, then the global rating is $\overrightarrow{t_i} = C^T\overrightarrow{c_i}$.

Pros

- There is minimal overhead in terms of message complexity. The problem of sudden behavioural changes is considered in this model.

Cons

- Collusion is possible in this model due to the reliance on domain members for maintaining the system.
- EigenTrust assumes that there are pre-trusted nodes in the system. This assumption is not always true.
- The algorithm does not consider effects of changes in behaviour over time.
- Using the normalised ratings does not differentiate between a peer that peer i did not collaborate with or another that is had a poor experience with.

3.8 Online Markets

The growth of online markets has led to the need for reducing the uncertainty of engaging in transactions with anonymous parties. In the computing overall score, eBay for example merely subtracts negatives from positives despite negatives being much rarer and hence more informative. In addition, there is no difference in buying from selling behaviour, just an overall reputation score [eBa12]. For example, a very high feedback score may not indicate that someone is a good seller if most of their positive feedback comes from buying, rather than selling activity.

Amazon calculates seller feedback scores using a $(1(worst) - 5(best))$ star system [Ama12]. Feedback percentage is calculated using the average of positive, negative and neutral feedbacks left in the last $30, 90, 365$ days, and lifetime, which is similar to the positive feedback rating used on eBay.

In these online communities, feedback is overwhelmingly positive and the systems do not predict performance. A similar example is the discrimination against a seller where someone who wants to falsely accuse a seller for example, gets his/her friends to purchase from the seller, and then rate the seller badly and also the reverse by promoting the seller to boost ratings. In addition to being linear and not capturing trust dynamics, these RTMs assume that past performance is an indication of future performance, which is not always true as illustrated in the intoxication attack described in Section 2.3.1.

3.9 Comparative Analysis

The components of RTMs are described in detail in this section, serving as criteria for comparison between the models discussed earlier and the extent to which each model meets or fails to meet each criterion.

3.9.1 Information Gathering

This feature of RTMs is the process by which an entity in a domain gathers information about other entities of interest. This can either be through first-hand or second-hand information gathering based on a direct observation, experience or recommendation. Therefore, an entity gathers information either through direct interaction with its neighbours, by direct observation of its neighbour's interaction with other domain entities, or by recommendations made by other entities of interest.

Only CONFIDANT and CORE have made the distinction between first and second-hand information gathering. CONFIDANT refers to this as personal experience and direct observation while CORE uses subjective and indirect reputation respectively. Similarly, SORI relies on direct observations through neighbour monitoring and SMTRI includes first and second-hand information recommendations from other members. TrustGuard collects feedback about nodes through the overlay protocol and this is aggregated to a rating. In EigenTrust, each peer i is assigned a unique global rating that reflects the experiences of all peers with i, also on eBay, feedback from direct observations are used.

Relying on information from others as opposed to first-hand observations has been shown in [BLB03] to offer no gain in reputation accuracy and also introduces additional vulnerabilities by creating a spiral of self-reinforcing information.

3.9.2 Monitoring

The monitoring function is the approach used to observe events that occur in the domain. The models [MM02, BLB02, HWK04, BVLT07, GBS08] rely on a *watchdog* mechanism, a promiscuous mode of monitoring where each node eavesdrops on the transmission of its immediate neighbours. By promiscuous monitoring, we mean that each entity overhears the transmission of neighbours to detect misbehaviour. The mechanism requires that every entity reports to the originator about the next entity. Once misbehaviour is detected, a negative rating is stored. The mechanism which is on every node collects first-hand observations about other nodes of interest.

The assumption that the watchdog mechanism will be present on all nodes is not feasible in real life networks. This is because nodes will require a considerable amount of energy in overhearing another node's transmission. In the WSNs community, privacy of individual nodes is emphasised making the approach unsuitable for such networks. The detection mechanism also has a weakness of failing to detect a misbehaving device in case of collusion.

In TrustGuard model, monitoring is dependent on the overlay protocol and monitoring in EigenTrust, and on eBay it is through the direct observation of a peer from other peers. The EigenTrust algorithm relies on the notion of transitive trust where a peer i will have a high opinion of peers that have provided it authentic files in a file sharing P2P. Peer i also trusts the opinion of those peers [HZNR09].

Table 3.1: Summary table of reputation and trust models

Criteria[a]	[MM02]	[BLB02]	[HWK04]	[SXL05]	[BVLT07]	[GBS08]	[KSGM03]	eBay
1	First and second-hand information from neighbouring nodes	First and second-hand information from neighbouring nodes	Direct observations	Feedback from overlay protocol	First and second-hand information, recommended reputation	Integration of direct and second hand observations	First and second-hand observations	First-hand experience
2	Watchdog	PACK, Watchdog	Watchdog like	Depends on overlay protocol	Watchdog like	Watchdog	-	-
3	Reputation table with each entry repres-enting a function	Bayesian approach[a]	Counters	Unspecified	Reputation ratings from captured evidence	Bayesian formulation[b]	Normalising and aggregating of ratings	Summation of ratings
4	Positive	Negative	Positive and negative recommendations	Unspecified	Positive and negative recommendations	Positive and negative recommendations	Positive and negative recommendations	Positive and negative recommendations

Continued on next page...

Table 3.1 – Continued

Criteria[a]	[MM02]	[BLB02]	[HWK04]	[SXL05]	[BVLT07]	[GBS08]	[KSGM03]	eBay
5	-	Performs well in the presence of high proportion of malicious nodes	Higher overhead as a result of increase in misbehaviour	-	-	Depends on the presence of trusted node along required path of communication	-	Tier-partitioning, horizontal scaling
6	Ratings are not constant, fading till a null value	Periodically updated (unspecified)	-	Feedbacks after transaction completion and rating is aggregated only when required	Reaction component predicts node behaviour and ratings continuous real values	Provides real time feedback	Unspecified	-
7	Weighted average of ratings ranging from [-1,1] (deterministic and continuous metrics)	Cooperation factor consisting of frequency of misbehaviour in relation to cumulative activity	Local record from the number of packet forwarded to and number forwarded by entity	Binary ratings	Continuous real values [-1, +1]	Probabilistic distribution	Continuous metrics converted into binary metrics via heuristics or statistical measures	Computed sum of feedbacks (+1, 0, -1), discrete values

[a] Key: 1. Information gathering, 2. Monitoring, 3. Information representation, 4. Recommendations and information sharing, 5. Scalability, 6. Prediction & Dynamism, 7. Reputation metric

[b] Baye's formula states how a conditional probability depends on its inverse. Let $P(a)$ be the probability that event **a** will occur. Also let $P(a|b)$ be the probability of a conditional on b, Bayes formula then states that $P(a|b) = P(b|a)P(a)/P(b)$. In a Bayesian network, unknown posterior probability is determined given a prior probability. That is, probability is updated given new information. If the prior probability is unknown, then recommendations are solicited to determine the posterior probability [HH07]. In [BLB02], the posterior probabilities of a node behaving in a good or bad manner and communicating correct or incorrect recommendations are computed from the prior probabilities by using Bayes formula.

[c] [GBS08] used a similar Bayesian formulation as Buchegger and Boudec in [BLB02].

3.9.3 Information Representation

This component of RTMs determines how information obtained about each node is converted to reputation ratings. Information representation deals with translating captured evidence meaningfully into a metric and how past and recent information affect the overall reputation of a node [BLB05, STW$^+$08].

A global reputation value is calculated in CORE by taking into account different observations or evaluation criteria such as packet forwarding or routing functions. The downside to the global reputation value is that a node may misbehave in certain functions and cover up by behaving as expected in other functions in order to gain a good global reputation. In the CONFIDANT protocol, a weighting scheme where nodes trust their own experiences and observations more than those of others is used. The rating is only changed when there is sufficient evidence of malicious behaviour and is changed according to a rate function namely: the greatest weight for own experience, a smaller weight from neighbours and the smallest weight for reported experience. CONFIDANT and RFSN both give a higher weight to past behaviour than recent behaviour in order to discourage intoxication.

SORI targets the non-forwarding misbehaviour and keeps count of packets forwarded by each node. The reputation rating consists of the ratio of these counts, taking into account the confidence in the rating proportional to the number of packets requested for forwarding. The first-hand and second hand information in RFSN are combined to get the reputation value of a node. The reputation of a node determines its rating which falls in the range of 0 and 1. In SMTRI, captured evidence is quantified and represented as reputation ratings. The quantified evidence is then represented as direct, observed, and recommended reputation ratings. Reputation is represented by continuous real values [-1, +1] in this model. SMTRI lays emphasis on context and events in determining ratings and this approach largely reduces its susceptibility to intoxication attacks.

Computation of trust in TrustGuard can be done using any existing trust evaluation mechanism. However in EigenTrust, each peer has a unique global rating assigned by other peers, weighted by the global ratings of the assigning peers. On eBay, participants rate each other in each transaction and the overall reputation is the sum of the ratings over a certain period of time.

3.9.4 Information Sharing

This component of RTMs is the way reputation information is propagated in the network to enable members decide whether to cooperate with others. Information that is propagated can either be negative, positive or both.

SORI proposes the use of a secure one-way hash function which makes it difficult for a selfish node with a bad reputation to send out its packets or broadcast fake observation information to affect the calculation of others. Only negative information is propagated in the CONFIDANT protocol, which prevents false praise attacks but results in false accusation vulnerability.

In CORE, only positive information is shared and the downside to this is false-praise vulnerability in the system. Similarly, only direct and positive reputation is propagated in RFSN and this results in an increase in memory overhead due to the need of maintaining a separate data structure for direct reputation [STW^{+}08]. Propagation of information in TrustGuard is dependent on the underlying overlay network while on eBay both positive and negative feedbacks are considered. EigenTrust uses a deterministic distributed dissemination framework relying on Distributed Hash Tables (DHTs) for rating storage and lookup. The efficiency of the dissemination corresponds to the efficiency of the underlying DHT in performing lookups, which is typically $O(logn)$.

3.9.5 Scalability

Scalability in this research refers to performance of an RTM with dynamic updates and feedback relative to the topology and/or number of misbehaving and normal members in the system. It is not evident to what extent some RTMs such as [MM02, SXL05, BVLT07, KSGM03] are scalable. However, the CONFIDANT protocol is scalable in terms of the total number of nodes in the network and performs well with a high proportion (60%) of malicious nodes. The overhead in the SORI scheme increases as number of connections increases, this is due to a higher probability of collision in the network leading to reputation miscalculation and hence, a larger overhead.

RFSN runs on each node and functions in a distributed manner, making it unfeasible for node to maintain the reputation of all nodes. It is worth noting that the assumption is that the subset of nodes with which a node interacts remains almost the same throughout the network lifetime. This makes RFSN relatively scalable. Not only is it sufficient for nodes to maintain reputation for only a few nodes, they can establish these metrics though simple local interactions [GS04]. In order to interact with distant nodes, a node can use a chain of trusted nodes to communicate. However, level of scalability RFSN provides is dependent on the presence of trusted nodes along the required path, a requirement which cannot always be guaranteed.

The resource usage on online communities should increase linearly (or better) with load (where load may be measured in user traffic, data volume, etc). With hundreds of millions of users worldwide, billions of page views a day, and petrabytes of data on such platforms, scalability is essential. Scalability in Amazon is handled by a fully distributed and decentralised services platform, serving many different applications. The reputation system on eBay is based on a centralised architecture with expected scalability problems such as overloading, availability etc. However, partitioning is used in every tier (code level, application and database) of eBay's architecture, dividing the workload into manageable units. In addition, horizontal scaling is present at every tier, therefore scaling out and not up [O'H06].

3.9.6 Prediction and Dynamism

The prediction property of RTMs is necessary to anticipate RVs of domain members. Likely future behaviours can be anticipated by considering historical data available and recent behaviour through a learning process. The dynamic property is in terms of the provision of current ratings by the models and feedback for decisions to be made in the system. These features help in making informed and security aware decisions, such as exclusion of misbehaving members from the domain or avoiding malicious members by others that need to collaborate.

The focus of models discussed here is not to carry out prediction except for RFSN that offers a form of prediction and real time feedback. In RFSN, the authors argue that reputation can only be used to statistically predict the future behaviour of other nodes and it cannot deterministically define the action performed by them.

3.10 Semi-Distributed Models with Trusted Third Party components

In order to manage trust, some reputation systems rely on a TTP in their architecture. In these literatures [BZ04, BZ05, LLYT05, GB07, SL06], the authors claim that they are distributed system. However due to the presence of the TTP, in this thesis, the models are regarded as semi-distributed models in this thesis.

The reputation management scheme proposed by Bamasak and Zhang [BZ04, BZ05] makes use of a TTP to assist agents in signature generation and to store evidences for the non-repudiation of signature receipt service provision. The focus of this research is on the evaluation of the trustworthiness and selection of a TTP to ensure the level security that conventional security solutions and cryptographic methods can not sufficiently provide. The reputation management scheme approach is unlike other RTMs that focus on how domain members join, are penalised or rewarded and their interaction patterns. The model credits and penalises a TTP-host according to its transactional behaviour, the transaction value and the reputation of the source of the feedback. The authors claim that this approach will deter TTP-host from misbehaving. A TTP's host reputation is in terms of a trust level and a reliability level, both of which are aggregated over a specific past period. This implies that the host may behave as expected for a lengthy period of time in order to become a TTP-host, and then later misbehave resulting in intoxication attack on the system.

A framework is proposed by [LLYT05] where every domain member (user) is associated with a *broker*; a form of centralisation. The broker collects distributed reputation ratings for members and in return the member provides the broker with ratings after every transaction. In this framework, there is an overhead on members on having to share feedback with brokers after each transaction. A problem with this also is how ratings provided by members can be trusted as the system assumes that users are diligent in providing honest feedbacks. What if members decide to cover up from one another, how does the system detect this?

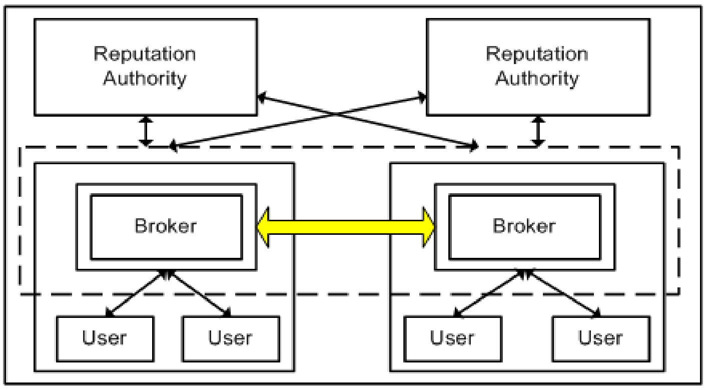


Figure 3.2: System Model by Lin *et al.* [LLYT05]

The framework comprises 3 components: *users*, *brokers* and *reputation authorities* as depicted in Figure 3.2. 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 user. The reliance on a TTP such as the broker and the reputation authority therefore, implies that the RTM is not completely distributed in terms of architecture as claimed, but semi-distributed.

A major problem of centralised systems is the scalability and the propagation of trust ratings across the domains. This is because reputation gathering in large systems by directly querying many members may result in significant communication overhead. The TRAVOS [Tea06] reputation model suggests that within each domain, there is a reputation broker agent. The agent is responsible for aggregating the opinions of all other agents within its domain; that is, the opinion of a reputation broker about a domain member is an aggregation of the opinions of all other members within its domain. Figure 3.3 illustrates the semi-distributed architecture proposed.

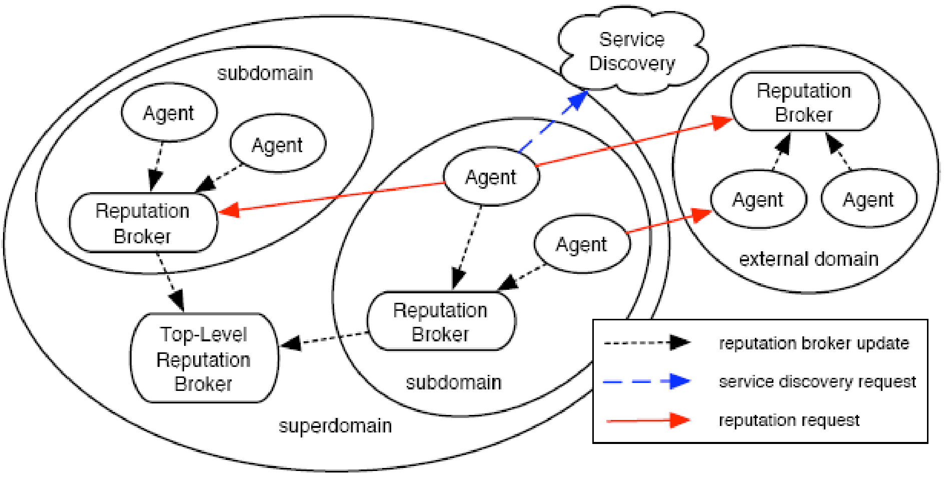


Figure 3.3: Reputation brokering system in TRAVOS [Tea06]

Following these examples of semi-distributed systems, it can be said that there is a need for a form of centralisation for an RTM to be adequately managed. This therefore justifies the argument that for a reputation and trust management system to be effective, there is a need of some form of central control. This fact is especially true in very critical domains such as traffic management systems, military warfare application for monitoring etc, where a TTP is required for quick decision making about the domain of application.

3.11 Gap Analysis

Traditional RTMs rely on recommendations provided by entities in the domain to determine the reputation of others. Each of the models addresses some of the trust issues though some new issues are sometimes introduced during the process. This section discusses currents issues of reputation management and motivates the use of a dynamic data-driven and predictive model for reputation management. This also serves as the motivation for the work done in this thesis.

3.11.1 Collusion

A common problem identified in the models is the vulnerability to collusion attacks [HB06]. 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 means each entity overhears the transmission of neighbours to detect misbehaviour. The mechanism requires that every entity reports to the originator about the next entity. Once misbehaviour is detected, a negative RV is stored. This detection mechanism also has a weakness of failing to detect a misbehaving device in case of collusion attack [MGLB00].

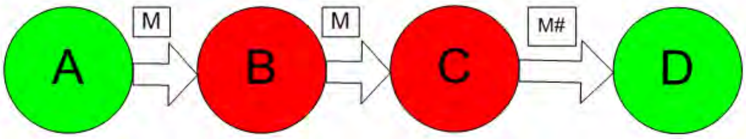


Figure 3.4: Domain members can misbehave by colluding and covering up for one another to deceive the system

An example of this attack can be described by having a set of sensor nodes that are deployed monitor vehicular movement, as introduced in Chapter 1 of this thesis. The nodes collaborate to collect and process data that generate information about traffic conditions. When a sensor node receives information from another, it is combined and fused with local information before being sent to a server, in order to control traffic. Figure 3.4 depicts collusion attack showing a downside of the watchdog mechanism. Knowing that sensor networks are vulnerable to attacks due to their nature, an adversary compromises a sensor node, which in turn compromises other nodes. Consider a normal situation, where for example, sensor node A forwards a message to node B and B forwards the message to C. Node C then forwards the message to node D. However, node C may decide to alter the message before sending it to D. With the watchdog mechanism, it is possible that B colludes with C and does not report to A when C alters message M, before forwarding the message. Misbehaving nodes do not only have the chance to collude but can also propagate false information. Therefore, trust decisions can be corrupted through recommendations made by such nodes.

A similar example is considering two nodes A and B that are controlled by an attacker. If node A tells B all of its secrets, then node B can masquerade as A to all of B's neighbours that node A shares pair-wise keys with and vice versa. The keys from each node that is subsequently obtained node can be reused by other attacker-controlled nodes, cascading the impact of the compromise. Therefore, an attacker can control a node undetectably by physically compromising the node and the node in turn, compromising other nodes within the network.

The assumption that all network nodes will operate in promiscuous mode is one that is not feasible in a real life network. This is because nodes will require a considerable amount of energy in overhearing another node's transmission. Thus, a model that prevents members' bias from influencing trust decisions is required.

3.11.2 Prediction

It is important that reputation is predictable for any reputation and trust-based system to be effective. The system should indicate members' trustworthiness and reliably predict future cooperation [CLB10]. One major drawback of some existing models [WV03b, WV03a, BLB04, TPJL05] of trust and reputation is that they do not focus on predictive accuracy. These approaches do not propose any method for predicting the future RVs of members in order to aid future trust-based decisions [HH07].

Intoxication attack occurs because the effect of past good behaviour outweighs the effect of current actions on reputation. This research does not consider past observations only, but also current and anticipated future events in the domain. Emphasis in D3-FRT (described in Chapter 5) is placed on past histories, recent behaviour and the future behaviour of members. Therefore giving the framework the desirable feature of prediction and also resulting in a proactive approach to reputation management.

3.11.3 Architecture

The challenge of distributed reputation systems is how to aggregate RVs without a centralised storage and management facility. Either that the system aggregates the ratings of only a few members and does not have a wide view of the members' reputation or it is able to aggregate the RVs of all members, but leads to congestion in the system. Some other problems of the distributed approach include:

- Extra traffic generation by information exchange [TKAS06];

- Extra computation in accepting observed reputation information;
- Corruption of trust decisions through recommendations made by domain members. Thus, decentralised approach exposes the system to false praise and accusation attacks;
- Low computational power of some devices in the network, which does not allow for complex analysis;
- Collusion attack may arise because the correct functioning of the system relies on members that may have been compromised.

A centralised element can result in a bottleneck, a single point of failure or result in lack of scalability. Despite these downsides, a centralised authority often leads to a simple solution with less potential for manipulation by malicious outsiders.

Apart from the reasons listed above, a semi-distributed approach is most desirable. The approach takes the responsibility of monitoring and determining the trustworthiness from individual members and combines the advantages of the other two structures.

3.11.4 Other Gaps

Some of the existing models lack the high level of dynamism required for spontaneous and ever changing networks. Dynamism is referred to, in terms of the provision of runtime trust rating by the models, and prediction of future behaviour of each member of the network. We argue that trust calls for a dynamic approach for its computation. Dynamics of trust is also reflected by its timeliness; reputation is aggregated in time by taking into account recent behaviour and past histories [LI04]. Hence, time is also a necessary dimension for reputation. This makes the framework useful as it considers a node in different future scenarios and provides a more holistic view of domain events. In addition to being dynamic in making predictions, the D3-FRT provides information about the system in good time.

Existing models such as eBay use a combination of average recommendations and the number of transactions performed by an entity as indicators of its rating. However, from [SL06] using a simple average does not guard against fluctuating behaviour or false praise and accusations. A dependable RTM should be able to identify and punish misbehaviour for such fluctuations because malicious entities may strategically alter their behaviour for their benefits. Therefore, an adaptive framework that can handle strategic fluctuations in behaviour is required for trust management.

The small size of nodes limits their storage and computational power and prevents them from carrying out complex analyses. However, in the models described in this chapter, each node has to monitor, calculate and maintain reputation information about other nodes in network. Since each node maintains reputation values of every other node, storing such information requires more storage at each node. For example, in [BLB05, BLB02, MM02], every node has to maintain *O(N)* reputation information, where N is the number of nodes in the network [TKAS06].

Ensuring *identity persistence* is another problem that remains despite the trust management solutions that have been proposed. If the reputation of each node is tied to its identity within the network, it will prevent excluded or isolated malicious nodes from gaining entrance again into the network, under another identity (identity spoofing). In the models described, there is a chance for a badly rated node to disappear and reappear with a different identity.

Additionally, through theoretical analysis, it has been confirmed that direct (first-hand) observations improve accuracy and are better than second-hand information [BMLB08]. The performance of first-hand information coincide on some range (if and only if the $\theta > 2\Delta$, where θ is the probability that a well behaving member is indeed observed as behaving well).Δ is the threshold for which the difference between first-hand and second-hand information should not exceed.

Furthermore, in the case of P2P file sharing P2P networks, some research suggested that 50% of shared files exchanged on Kazaa were infected. On Gnutella as well, 70% of peers free-riding other members by not sharing files but instead downloading from others [LYG+07]. This implies that there is a requirement to decrease these types of misbehaviours among peers; a need well suited for a dynamic trust management framework.

3.12 Summary

In summary, RTMs have gained popularity because they have shown to be promising in the area of trust management. In this chapter, we discussed the interlink between reputation and trust, and models that have contributed significantly to this area of research are reviewed. In the qualitative analysis done, the conclusion is that as these models try to solve the problems of reputation and trust management, other problems are introduced such as the susceptibility to collusion attacks.

A comparative analysis of these models and a gap analysis are discussed extensively in this chapter. Intoxication occurs due to the use of histories alone in determining the reputation of domain entities. This chapter discusses how relying on historical information only is inadequate using this approach as members can deceive the system.

The semi-distributed architecture to trust management is described in this chapter. This architecture combines the advantages of both centralised and fully distributed architectures. Furthermore, dynamism is a gap that has not been the focus of previous work and we have described how this gap can result in a reactive approach. With these research problems in mind, the provision of recent information about domain activities and corresponding computations of reputation/trust will make such models more proactive giving the system adequate time to make preventive measures.

This thesis aims to contribute towards the provision of reliable reputation management for critical domains that require the cooperation of members to be effective. This involves running tests and simulations on the proposed approach in a variety of scenarios and with different input parameters. It is hoped that the insight gained from the results presented will be effective in making predictions and providing useful information for stakeholders that will result in a proactive approach to reputation management.

Chapter 4

Reputation and Trust Dynamics: A Data-Driven Simulation Approach

4.1 Introduction

This chapter is divided into two main parts. The first part describes in detail the DDDAS paradigm, an approach of a symbiotic relation between applications and simulations. In the second part of this chapter, we argue on the usefulness and applicability of DDDAS paradigm in the domain of reputation management. The paradigm's primitives and how they fit for the trust dynamics issues are also discussed in the second part.

In DDDAS, an application can accept and respond dynamically to new data injected into the executing application, and in reverse, such an application has the ability to dynamically control itself. The data is fed online or from a data archive and will then be used to influence the measurements for additional data it may require. Such capabilities provide more accurate analysis and prediction, more precise controls, and more reliable outcomes [Dar04, Dou08].

Several requirements and recent technological advances render DDDAS approaches more opportune than ever. These new realities call for more advanced methods of systems analysis and management. The methods required, go beyond the static modelling and simulation methods of the past, to new methods, such as DDDAS which augment and enhance the system models through continually updated information from monitoring, feedback and control aspects of the system. Together with these driving needs of emerging systems, several technological and methodological advances have produced added opportunities and impetus for DDDAS approaches [Dar10].

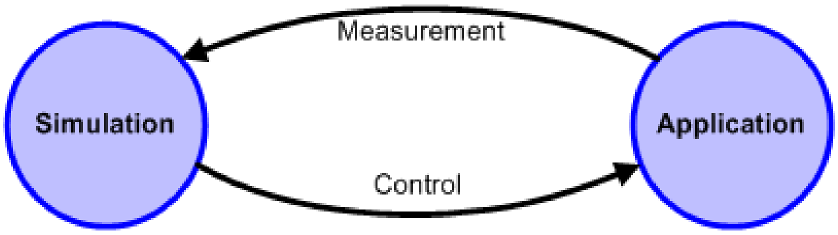


Figure 4.1: DDDAS feedback-loop between systems and simulation

DDDAS involves dynamic information exchange and control between the application and the measurement systems, where each dynamically affects the behaviour of the other with potential improvement in their effectiveness. The relationship between a system and simulation as suggested in DDDAS is given in Figure 4.1. Such tight and dynamic integration presents special challenges because it addresses application domains in which highly sophisticated but segregated simulation and measurement approaches diverge and often fail to predict real system behaviour. DDDAS requires algorithms with guaranteed stability properties under dynamic data inputs [DDD06].

Current researches in DDDAS focus on simulations of physical, artificial and social entities [KTS+07, MSB06]. The simulation can make predictions about an entity, regarding how it will change and what its future state will be. The simulation is then continuously adjusted with data gathered from the entity. The predictions made by the simulation can then influence (control) how and where future data will be gathered from the entity in order to focus on areas of uncertainty [KT06]. Intelligent agents will be required to make decisions on which data to incorporate, when it should be incorporated, and how it should be incorporated. In addition, the predictions can be useful for updating the simulation parameters in order to make more precise predictions in the future. Many of these simulations are inspired by older models used to make predictions about physical systems [Dar04] such as in sensor networks, weather forecasting, traffic management etc.

4.2 Dynamic Data-Driven Application Systems

In this section, a review of existing DDDAS is provided. The systems are selected not only for their significant contribution but also for their capabilities and diversity in terms of benefiting from the paradigm. The selection is also based on their methodology for data handling, adaptive input parameters and predictive capability, which are attributes that are important to this research. The use of agent-based modelling as a simulation component and a similar domain of applications are factors that influenced our DDDAS selection in this section. The approach, strength(s), weakness(es) of each of the DDDAS and gaps are identified and discussed here. The objective of this review is to learn from existing work, new insights that can influence our decisions with dealing with the dynamic nature of reputation and trust. We consider the characteristics of the systems and highlight useful methods that are a fit for the purpose of this research.

4.2.1 Event Correlations in Sensor Networks

Reducing the number of sensors, unnecessary communication and energy consumption are the reasons for discovering salient correlation between the events in a sensor network. For example, if the relations between factors such as temperature and humidity are known, one factor can be determined from knowing the value of the other.

The event correlations research [NWC09] extracts correlations from a large number of sensors instead of using a traditional method based on an apriori algorithm and pattern growth. The method is event-driven and discovers specific valuable patterns instead of a complete pattern set. Algorithms are incorporated to improve efficiency for discovering concise and accurately correlated patterns.

Through experiments, Ni *et al.* [NWC09] show that the method is both highly effective and efficient. The goal is to discover anomalous events in a large sensor network where the structure is unknown. The algorithm proposed enables users to select the correlation confidence level and only display the significant event correlations.

4.2.2 Leveraging the Cell Phone Network

During a disaster, emergency response managers require timely alerts and quality information about the location and movement of the entire affected population. Reports from on-scene coordinators, first responders, public safety officials, the news media, and the affected population are often inaccurate, conflicting and incomplete with gaps in geographical and temporal coverage. Additionally, those reports must be merged into a coherent evolving picture of the entire affected area to enable emergency managers to effectively respond [MSB06].

Integrated Wireless Phone Based Emergency Response System (WIPER) [MSB06], uses wireless call data, including call volume, who calls whom, call duration, services in use, and cell phone location information for emergencies. WIPER selects from data streams in the physical vicinity of a communication or any traffic anomaly and dynamically injects it into an Agent-Based Modelling and Simulation (ABMS) system to classify and predict the unfolding of the emergency in real time. This approach that is adopted in this research as detailed in Chapter 5. The ABMS dynamically steers local data collection in the vicinity of the anomaly. Multiple distributed data collection, monitoring, analysis, simulation and decision support modules are integrated using a Service Oriented Architecture to generate traffic forecasts and emergency alerts for stakeholders. Both the reliability of the cellular phone system during a disaster, and privacy concerns present potential limitations to the WIPER system.

4.2.3 Rail Monitorring

The complex dynamics and the ever changing environment in which rail systems are exposed limit the use of classical simulation. Changing environmental conditions and second order dynamics challenge the validity of the rail system models and seriously reduce model (re)usability. Several challenges in rail transport have been identified such as handling large volume of data, classification of data, the selection of data, the implementation of different data analysis methods, and the difficulties of data interpretation [HSV10].

The research into automated model calibration and validation to rail transit simulation [HV09] describes the potential benefits of the DDDAS in the domain. Emphasis is placed on automated model reconfiguration, calibration, and validation through the use of data

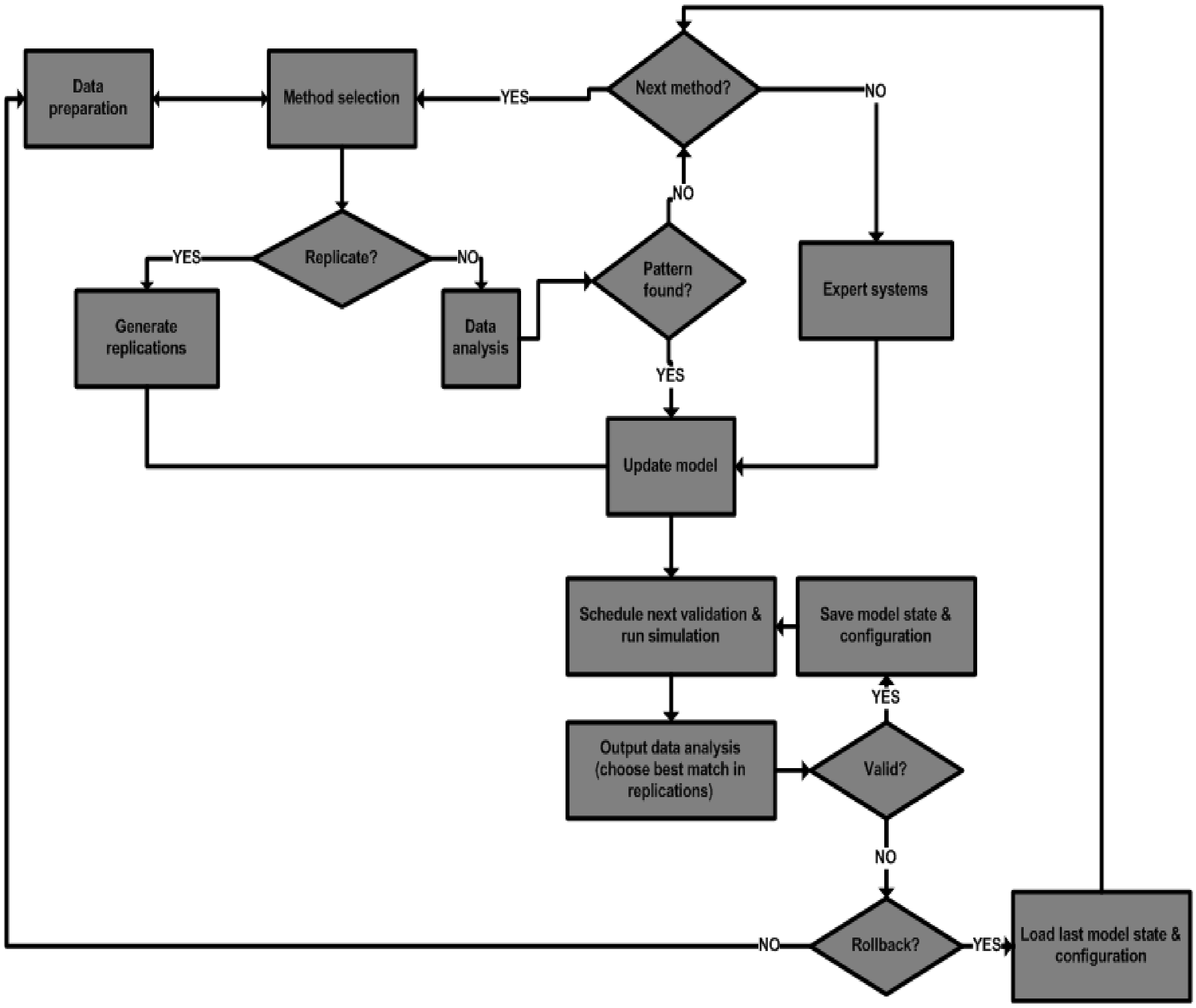


Figure 4.2: The Simulation Process [HSV10]

analysis methods. The work aims at capturing both first and second-order dynamics of the system from the onset. The main effort is to employ computational logic that continually performs model validation and calibration using the extensive available data during a simulation run.

This research gives a generic data-driven simulation process (in Figure 4.2) that can be used with other tools in other applications, which is a step towards a generic library for dynamic data-driven model calibration and validation. In this process, the model output and measurements from the system are continually compared. Once the deviation exceeds a predefined threshold, the model parameters should be updated based on data analysis results. Different analysis methods need to be available to discover useful hidden patterns in the data sets, and to estimate the parameters. Mechanisms should be defined to enable saving model states and parameter configurations as a reference for ulterior comparison or model state rollback [HV09].

4.2.4 Injecting Dynamic Real-Time Data into a DDDAS

There are several tools for mitigating damages caused by fires such as fire propagation simulators that use physical or mathematical models. Most simulators of natural phenomena however, demand high computing resources and require as inputs, a wide set of variables whose values are either not well known or estimated prior to execution including a considerable degree of uncertainty. Setting up inputs at the beginning of a simulation process is a major drawback because as the simulation time continues, variables previously initialised can dramatically change and thus, produce inaccurate results [RCM10].

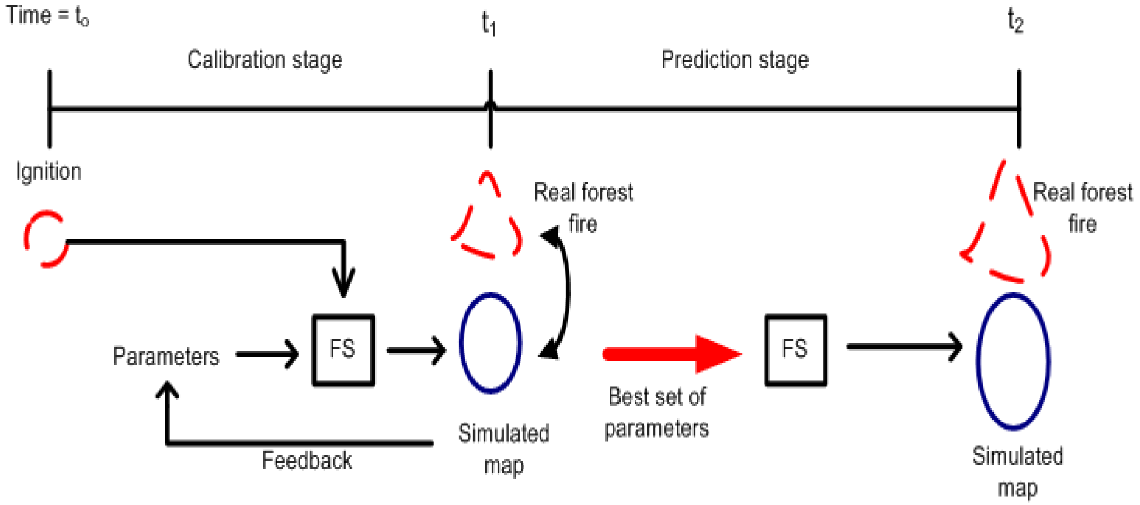


Figure 4.3: Prediction methods by Rodriguez *et al.* [RCM10]

The fire propagation prediction research (depicted in Figure 4.3) uses DDDAS methodologies to predict wild fire propagation. The focus is on reducing the execution time and improving the prediction results of a DDDAS for fire spread prediction. This method improves dynamic adaptation to sudden changes in environmental conditions. The prediction uses fixed environmental conditions as the Fire Simulator's (FS) input variables at time instant t_0 where the fire simulator is used to predict fire evolution at instant t_1. This time interval $(t_0 - t_1)$ is called the *prediction stage*. Another stage is included before the prediction stage, where the simulator's input parameters are calibrated, depending on the observed fire behaviour. The two-stage fire prediction methodology reduces the negative impact of input parameters' uncertainty. Though the execution time with this approach was reduced by 20 times, the prediction improvements were quite similar to the results obtained applying the previous prediction schemes.

Following the review above of the applications that have benefited from DDDAS, it can be seen that DDDAS is a useful paradigm:

- In domains where there are uncertainties (dynamism),
- Where there is a need to draw conclusions (predictions/forecast) from the captured events,
- In the presence of large volumes of data that needs to be collected, processed and interpreted.

Furthermore, the paradigm is capable of providing valuable information about how a domain can evolve over time and the components that stakeholders need to focus on in order to maintain a trusted system.

Knowing that these are requirements of an effective trust model, we can conclde that DDDAS is fit for the reputation and trust management landscape. In addition, adopting the paradigm approach opens a whole new approach to managing trust and reputation.

4.3 Dynamic Data-Driven Application Systems and Reputation Management

4.3.1 Trust Dynamics

Trust dynamics can be described as the influence of changes in a domain, computations and prediction on the same domain. The consideration of trust dynamics issues in the methodology adopted by any reputation and trust management system is critical to the success of the system. Govindan and Mohapatra in [GM12b] described trust dynamics as the evolution of reputation over time, i.e., the changes in behavioural patterns of an entity with time. The dynamic and changing nature of reputation and trust has been established earlier in this thesis. This nature may arise as a result of different factors in the application domain; factors such as: collaboration, attacks in the domain or on the reputation system, mobility among members such as wireless sensors, changing topology etc. For example, factors that may influence trust dynamics in a network scenario can be the mobility of the nodes and the number of nodes that are collaborating in the network. In addition, the dynamic nature of trust makes RTMs vulnerable to attacks such as intoxication as described in the publications arising from this thesis [OBT10, OTB11].

The dynamics of trust can be characterised by trust *propagation* or information sharing, *prediction* and *aggregation* [GM12b]. Trust propagation is based on the transitive property of trust. Propagation of trust deals with the approach with which trust information is communicated across the domain. Different approaches to trust propagation have been proposed in literature and these are given in Chapter 3 of this thesis.

Some models in literature focus majorly on trust dynamics and acknowledging that existing models still suffer from sudden changes and dynamic domain events. Chang *et* *al.* [CWG06], suggested the use of a fading factor, which discounts older evidences, to capture the oscillations in peer behaviour in a P2P network. In the fading scheme, a forgetting factor is used; a function of recent observations in the domain. Relying on the forgetting factor only, can therefore result in the exploitation of the reputation system where an entity may decide to wait for a certain time interval for histories to fade before misbehaving again.

SecuredTrust [DI12] on the other hand, handles changes in behaviour by using a *deviation reliability* function. In the absence of interactions with other agents, an agent's trust should degrade gradually using a decaying (forgetting) factor. The predicted trust in SecuredTrust however, relies on recent and historical observations. As stated in previous chapters, relying on recent observations alone is inadequate for effective trust management. In addition, none of these approaches anticipate possible future events such as member behaviour in the domain in making trust predictions and decisions; which our DDDAS-inspired approach can provide.

The approach presented in this thesis focuses on trust dynamics in the areas of *prediction* and *aggregation* only. *Propagation* has been dealt with extensively in literature: [GBS08, MM02, BLB02]. In this research, we give a definition to trust *prediction* by describing the concept as the anticipation of the possible events that can occur in the domain in the future and computing a representing rating (RV) for each involved entity. In order to aggregate the RVs, observed events are converted to useful metrics and combined to obtain a single value (referred to as reputation value so far in this thesis). In addition, the *aggregation* of trust is the technique with which ratings are gathered from different sources and how the RV is computed from the same ratings. Several approaches to trust *aggregation* have been adopted in literature as detailed in the review of Chapter 3, while the methodology adopted in D3-FRT is discussed in Chapter 5 of this thesis.

4.3.2 Gap Analysis

In this research, the dynamic nature of trust and reputation is controlled by exploiting the primitives of the DDDAS paradigm; which includes *measurement*, *simulation*, *feedback* and *control* mechanisms (which were described in Chapter 4). The dynamic nature of trust makes it difficult to detect sudden and changing behaviour that is actually considered as misbehaviour in an application domain. For this reason the DDDAS concept, which is an equally dynamic approach, is fit for identifying and isolating misbehaving members.

The pending issue of collusion attack shows that there is a need of identifying, monitoring and predicting misbehaviour of domain members. Therefore, a *feedback* system that will aid making dynamic, informed and security-aware decisions is required for RTM.

The missing elements of traditional RTMs that have been identified so far are:

1. Provision of dynamic RVs to identify malicious members

2. Prediction of future RVs to prevent misbehaviour

3. Reliance on members for trust decision resulting in collusion

4. Segmentation into regions of risk to focus on regions of high-risk

In the next sub-sections, we describe how the DDDAS inspired framework can fill these elements.

4.3.3 The Paradigm in this Thesis

This research adopts the concept of the DDDAS paradigm in order to address the issue of dynamism of trust and collusion attack amongst domain members. The recent measurements (data about observations and behaviour) are simulated to gain a better understanding and a more accurate prediction of the level of trust for each member. The simulation dynamically measures trust levels, and continually incorporates new measurements immediately (the simulation cycle is illustrated in Figure 4.4). This will enable the simulation to determine and feedback the reputation of each node into the system. The output of the simulation will help control the domain in terms of decisions to be made as depicted in Figure 4.5.

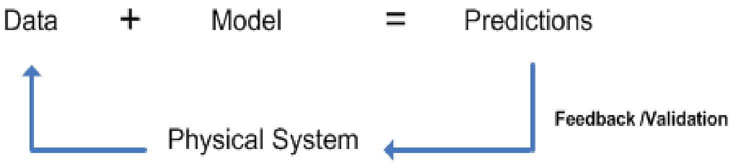


Figure 4.4: Simulation Cycle

One of the missing elements of traditional RTMs is the reliable prediction of future RVs of members to proactively prevent misbehaviour. The classification of members into different levels of risk is also an important missing element. This classification can potentially help the RTM to focus on members that are of high-risk in the domain.

Hence, we propose a framework that:

1. Predicts the future RVs using past events, recent events and possible future interactions

2. Provides information about members that are classified as high-risk

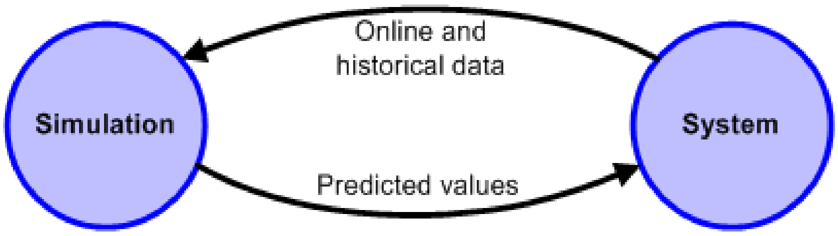


Figure 4.5: The feedback loop between the physical system and simulation

3. Prevents members' bias from influencing trust decisions

4. Provides dynamic RVs of domain members.

The manner in which the DDDAS approach fits with the RTM approach considered in this thesis is given in Figure 4.6. The application of DDDAS for trust management, which is the novelty of this research, provides dynamism in the detection of misbehaving members and prediction of future ratings.

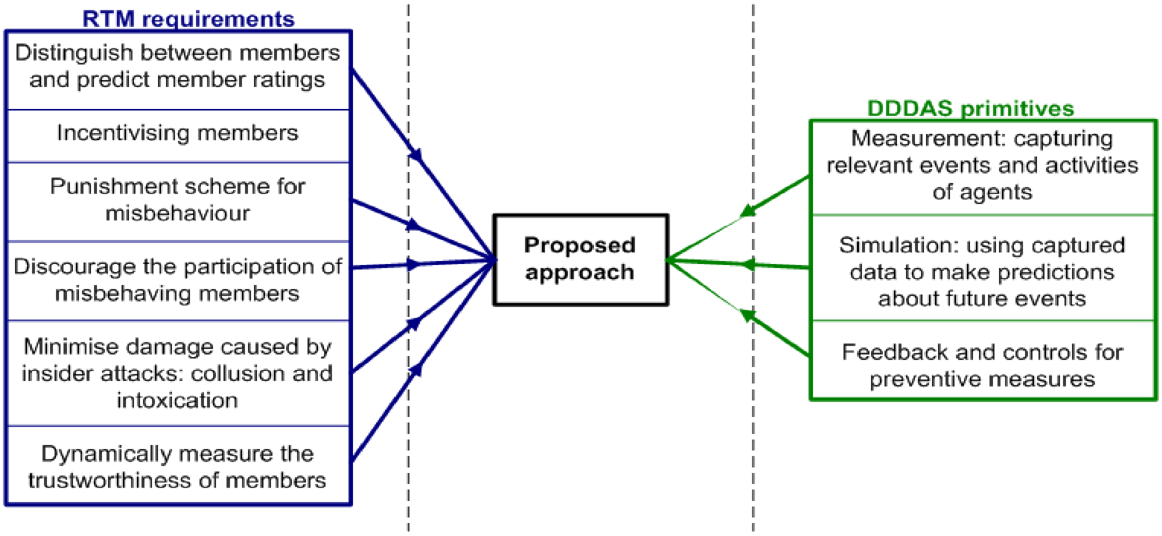

RTM requirements
Distinguish between members and predict member ratings
Incentivising members
Punishment scheme for misbehaviour
Discourage the participation of misbehaving members
Minimise damage caused by insider attacks: collusion and intoxication
Dynamically measure the trustworthiness of members
Proposed approach
DDDAS primitives
Measurement: capturing relevant events and activities of agents
Simulation: using captured data to make predictions about future events
Feedback and controls for preventive measures

Figure 4.6: The proposed approach in connection with both DDDAS and requirements for reputation management systems

4.4 Summary

This chapter discusses the DDDAS computational model in the context of examples of novel capabilities enabled through its implementation in different application areas. The models have given insights to how DDDAS can be applied in our case which is mainly directed towards trust dynamics issues. We considered the salient characteristics of the DDDAS and highlight useful methods that are fit for our purpose. From our study, the DDDAS *feedback* loop comes with high potential pay-off by providing timely information for controls within the domain. Therefore, the creation of a framework with predictive capabilities; a new methodology for more efficient and effective measurement processes are the benefits of adopting the paradigm in this research.

Dynamism and predictive capabilities are some of the outstanding properties of some existing reputation models (described in Chapter 3) and the *simulation* and *feedback* system proposed is useful for making predictions about how a domain may evolve with time. The predictive capability of DDDAS is exploited by D3-FRT for providing dynamic recent ratings and predictions of possible future ratings in a system of participants. In addition, making components of trust configurable is advantageous for effectiveness in dynamic systems.

Finally, this chapter focuses on the usefulness of DDDAS for providing more reliable reputation management in trust-reliant application domains. We described how the concept of DDDAS can feed into reputation management through D3-FRT making it a solution for trust and reputation management issues. The adoption of a dynamic data-driven approach motivated by DDDAS is the first of its kind and serves as the novelty of this research. For the reasons listed in this chapter, we can conclude that the paradigm is a useful approach for reputation management because of its capability in addressing the inherent dynamic issues of trust and reputation.

Part III

It is bad enough that so many people

believe things without any evidence.

What is worse is that some people

have no conception of evidence

and regard facts as just someone else's opinion.

Thomas Sowell

Chapter 5

A Predictive Framework for Modelling Reputation

5.1 Introduction

This chapter introduces the Dynamic Data-Driven Framework for Reputation management. The D3-FRT framework benefits from the outcomes of literature reviews in Chapters 3 and 4 of this thesis. From these reviews, we have identified some requirements that are useful for the effectiveness of any trust-based and dynamic reputation system. The identified requirements are highlighted below and they serve as a part of the contribution of this research to the body of knowledge:

- Having a predictive capability is an essential criterion for any RTM. Generally, RTMs make use of past events as an indication for future behaviour and this has been shown to be inadequate in [OTB11, OBT11]. For an RTM to be reliable and effective in trust management, trust has to be predictable. We propose that predictions should be based on *past events, recent events* and possible *future interactions.*

- The *dynamic* computation of RVs is important because trust and reputation are *dynamic*, i.e they constantly change as a result of interactions and other environmental/domain factors. The dynamic property of reputation and trust was described in Chapter 2 of this thesis.

- Preventing of members' bias that may influence trust decisions, and which can result into collusion attacks in the system.

- Providing an *online decision support system* that allows stakeholders to make informed decisions about the domain in question.

This chapter describes D3-FRT as it captures these requirements and how the requirements are implemented. The components of the proposed framework are described individually in the early part of the chapter and the inter-relationships between the components are detailed in later sections. We discuss the relevance of agent-based modelling to our DDDAS-inspired framework initially in this chapter. Thereafter, the initial version of the D3-FRT, its downsides and the improved version are then described in detail.

5.2 Overview of the Realisation of the D3-FRT Model

The development of D3-FRT is based on the outcomes from the survey and literature review of the preceding chapters. An initial high-level architecture for the D3-FRT framework was designed. By keeping in mind the requirements of trust management we have identified and the DDDAS primitives that make the requirements realisable. Without going into much detail at this point, the following diagram (Figure 5.1) depicts the initial version of the D3-FRT. A more detailed description of the framework and its components are in Sections 5.7 and 5.8 of this Chapter.

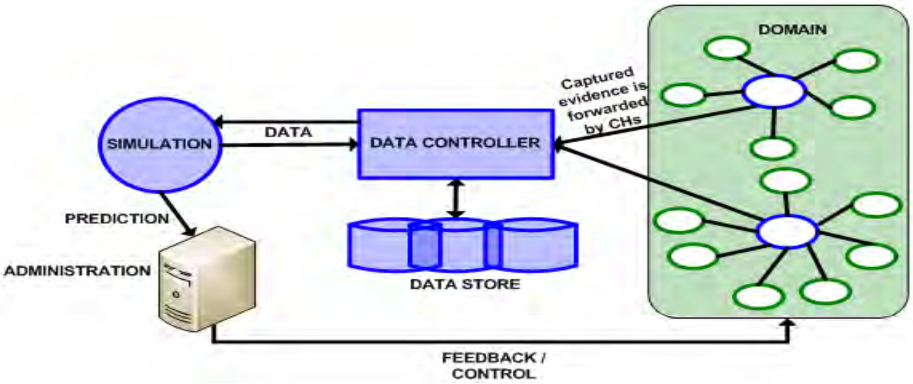


Figure 5.1: High-level diagram of initial version of D3-FRT

In order to implement the framework, an ABMS approach is adopted and this is justified by its capability to provide insightful views of the past, present and future state of a system. This capability aids the fulfilment of D3-FRT's property of anticipating and making predictions about the future state of the system. Recursive Porous Agent Simulation Toolkit (Repast) ABMS is selected to be used as a proof of concept for designing a refined version of the framework. We extended the Rapast [Rep11] for modelling the D3-FRT framework in this research. Figure 5.2 depicts the implementation architecture of D3-FRT, which is an extension of the toolkit for the purpose of this research. To the best of our knowledge, this is the first time the toolkit is being extended to the problem of reputation and trust. With this extension, D3-FRT becomes a flexible model of interacting agents that provide facilities to store, display and activate agent behaviour.

An iterative design approach was followed in designing the D3-FRT model, as it is considered the most effective for practical model development [MN10, NM07]. This process started with an initial description of D3-FRT components and agent behaviours, including

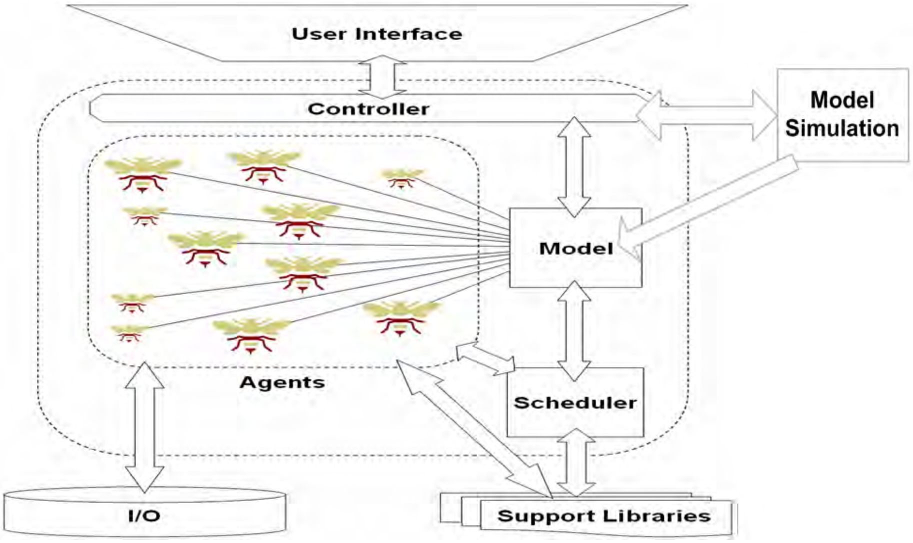


Figure 5.2: Implementation Architecture Extending Repast

the initial state of the system. This description was then converted to an initial functional model that was executed to generate some results. On examining the results, the model is refined and re-run until the expected behaviours and results were obtained. Some important model design questions [MN10] were answered and these answers aided the development of the model design. These questions include:

1. How can the issues of trust dynamics be solved by the model?

2. What added-value would agent-based modelling in addition to the DDDAS approach provide to address the problem that other approaches cannot add?

3. What should the agents be in the model? Who are the decision makers in the system? What are the characteristic behaviours of the entities involved? What data on agents are simply descriptive (static attributes)? What agent attributes would be computed endogenously by the model and updated in the agents (dynamic attributes)?

4. What is the agents' environment? How do the agents interact with the environment?

5. What agent behaviours are of interest? What decisions do the agents make? What behaviours are being acted upon? What actions are being taken by the agents?
6. How do the agents interact with each other and the environment? How expansive or focused are agent interactions?
7. Where might the data come from, especially on agent behaviours, for our model?

Answers to these questions are discussed in the remainder of this chapter.

5.3 Postulations of Trust Framework

D3-FRT is potentially more proactive compared to other RTMs (described in detail in Chapter 2) because the framework aims to anticipate misbehaviour before an attack occurs [OBT11]. This is done with the use of past interactions among domain members, their recent and anticipated future RVs for trust management. The advantage of this proactive approach is that informed decisions can be made before misbehaviour occurs as illustrated in Figures 5.3 and 5.4. We refer to proactivity in terms of providing control such as downgrading of RV of suspect members that are predicted to be malicious before they can carry out an attack. The premise is that a member who has been compromised by an adversary exhibits a sequence of steps in order to misbehave. Proactivity in this research is the provision of useful information to stakeholders for countermeasures that can effectively reduce the effect of misbehaviour.

The hypothetical example in Figures 5.3 and 5.4 shows the difference in response times between the proposed approach and others. Figure 5.3 shows that the RV is only downgraded at time t_5 after misbehaviour. D3-FRT can potentially predict the

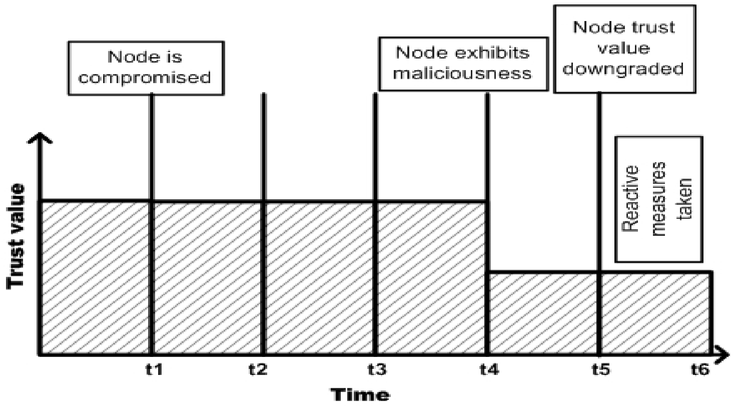


Figure 5.3: Reactive approaches that compute ratings after each member interaction

misbehaviour between time interval t_1 and t_2 and the RV is downgraded at time t_3 in Figure 5.4. This is different to how other approaches work; that only downgrade RVs as a reaction to misbehaviour. Therefore, misbehaving members continue to attack until the reputation system identifies the misbehaviour.

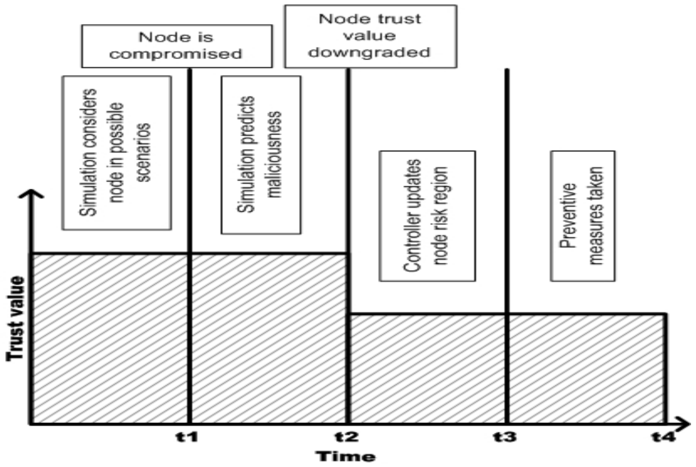

Node is compromised
Node trust value downgraded
Simulation considers node in possible scenarios
Simulation predicts maliciousness
Controller updates node risk region
Preventive measures taken
Trust value
t1
t2
t3
t4
Time

Figure 5.4: Proactive approach that predicts misbehaviour

This capability fits within the DDDAS paradigm and it enables D3-FRT to perform better by making predictions about the possible future RVs of members. The prediction gives the system adequate time for preventive measures.

Furthermore, D3-FRT aims to rate agents and predict agents' RVs in a manner that represents exhibited behaviours. The graph in Figure 5.5 depicts the expectations of the output from the framework which are changes to the RVs of agents depending on the behaviours that are exhibited.

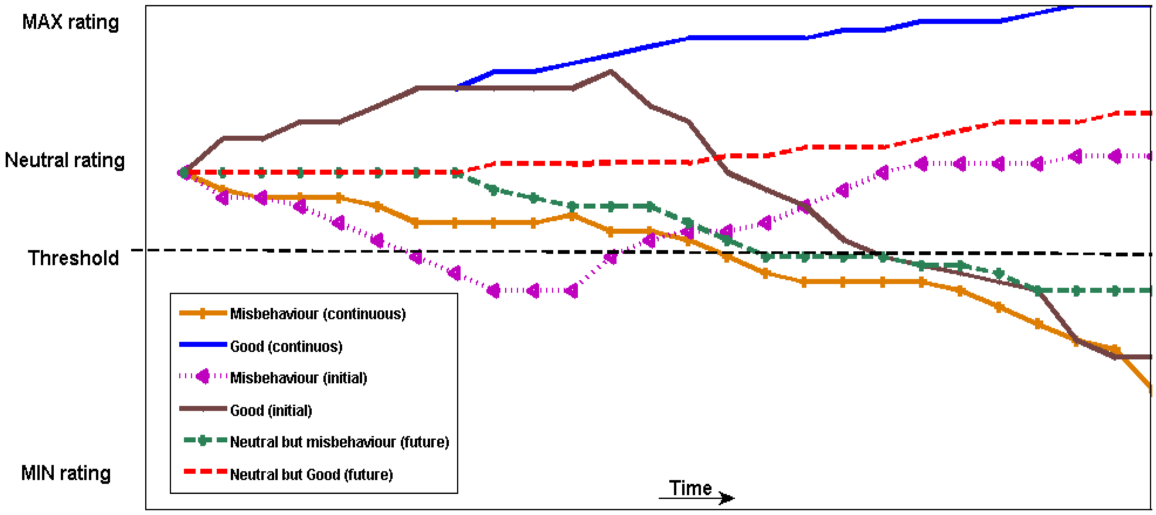


Figure 5.5: Expected changes to RVs as agents exhibit different behaviours as time progresses

In the figure, *MAX* and *MIN* ratings are the maximum and minimum allowable RVs in the system respectively while *Neural rating* is the default RV for all domain members. The threshold is the minimum rating an agent can have to remain reputable; any value below the threshold is punishable depending on the nature of the application.

5.4 An Agent-Based Model Approach

ABMS is an approach to modelling systems composed of autonomous, interacting agents. In addition, agent-based modelling is a way to model the dynamics of complex systems and complex adaptive systems. Such systems often self-organise themselves and create an emergent order. Agent-based models also include models of behaviour (human or otherwise) which are used to observe the collective effects of agent behaviours and interactions. The development of agent modelling tools, the availability of micro-data, and advances in computation have made possible a growing number of agent-based applications across a variety of domains and disciplines [MN10].

ABMS is widely used to understand systems composed of interacting individuals [NM07] (agent). These agents are the peers, nodes, participants, users and members that *collaborate* for a purpose in a domain. The term collaborate in this context is the interaction between agents in a specific domain. For example, consider mobile sensor nodes in a network monitoring vehicular movement as described in Chapter 1. The act of exchanging captured evidence between these nodes is referred to as collaboration. In this research, we use an ABMS approach that takes the agents and their interactions and embed them into our computational framework.

Due to the complexity (interactions and interdependencies) of the systems in use these days, a dynamic approach to prediction is required to capture all requirements, to have a close representation of reality. ABMS is suitable as evolving and dynamic behavioural changes in the domain are a major consideration for this research.

Provision of useful information to control the systems is another justification for the use of ABMS. For example, when agents optimise their collective behaviour through simple exchanges of information as is done in an ant colony optimisation [MN09], the purpose is to achieve a desired end-state, that is, an optimised system, rather than to simulate a dynamic process for its own sake.

Therefore, the choice of ABMS in this research resulted from the following requirements and properties:

- The problem consists of interacting agents.
- Scaling up is an important consideration of this research as the framework has to be

tested in various scenarios.

- Agents can change their behaviours with time and there is dynamism in agent interaction patterns.
- There are well-defined and outlined decisions and behaviours.
- As stated in Sections 1.2 and 3.11.2, relying on past histories only is not effective for trust management. ABMS is useful for when the past is a poor predictor of the future.

The adoption of ABMS approach in D3-FRT allows for prediction from the knowledge of the behaviours of the agents, i.e. modelling how the system would mostly likely evolve over time.

Finally, ABMS allows for testing to identify how the configurations and rules make the system evolve and their effects on the agent interaction patterns.

5.5 Agent-Based Modelling in D3-FRT

5.5.1 Modelling and Simulation

As stated by Macal and North [MN06], ABMS can be described as the act of simulating the actions and interactions of agents repeatedly over time in order to assess their effects on the system. There is no universal agreement on the precise definition of the term *agent* in the context of ABMS, but this research has adopted a computer science perspective which is a discrete entity with its own goals and behaviours with a capability to adapt and modify its behaviours. This requires agents to be responders and planners rather than purely passive components [NM07]. Agents are diverse, heterogeneous, and dynamic in their attributes and behavioural rules, as shown in Figure 5.6.

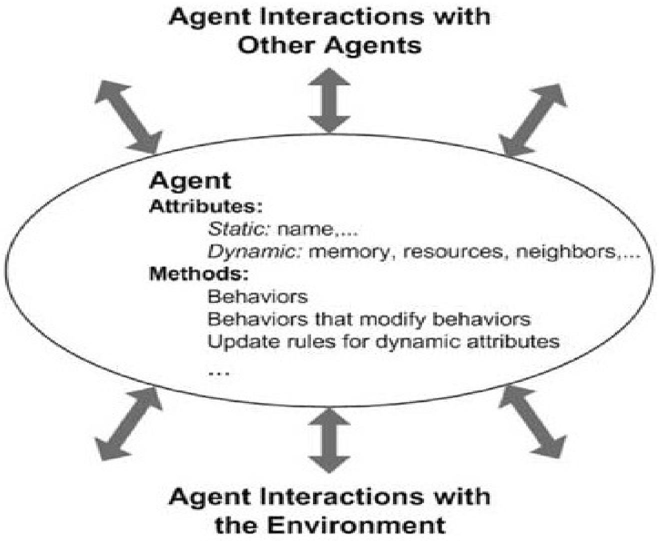

Agent Interactions with Other Agents
Agent
Attributes:
Static: name,...
Dynamic: memory, resources, neighbors,...
Methods:
Behaviors
Behaviors that modify behaviors
Update rules for dynamic attributes
...
Agent Interactions with the Environment

Figure 5.6: A typical agent by Macal and North in [MN10]

Agents have internal and external properties referred to as *state variables*. Some variables such as Agent identifier (AgentID), Agent type (trusted, misbehaving etc) are static while others such as RVs may change with time. Changes in the state variables may either be as a result of the influence of the internal in-built logic in the agent or as a result of cooperating with another entity as shown in Figure 5.7.

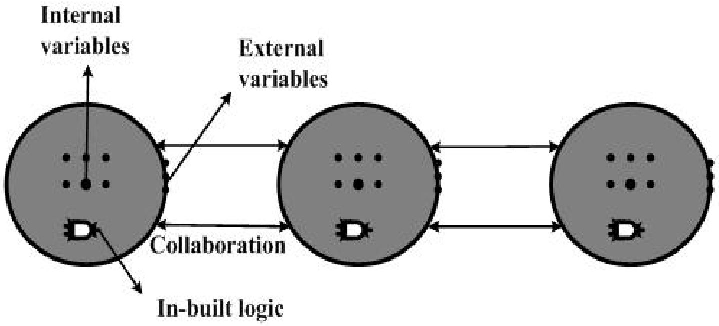


Figure 5.7: Properties of agents in simulation

In the modelling of the D3-FRT and its components, a mixture of Java, Groovy, and flowcharts are used. In the model, agents are organised into their space through the use of *contexts* and *projections*. From a modelling perspective, the context represents an abstract population while objects in a context are the population of a model. Projections take the population as specified in a context and impose a new structure on it. This structure defines relationships on the population by using the semantics in the projection. In other words, an agent population is realised once a projection is applied to it. From a practical point of view, this implies that projections are added to a context to allow the agents to interact with one another. Projections have many-to-one relationship with contexts, while contexts can have arbitrary number of projections associated, which implies that within each context, the agents can create an arbitrary number or types of relationships with each other [Rep11].

The context-projection relationship adopted in this research is depicted in Figure 5.8.

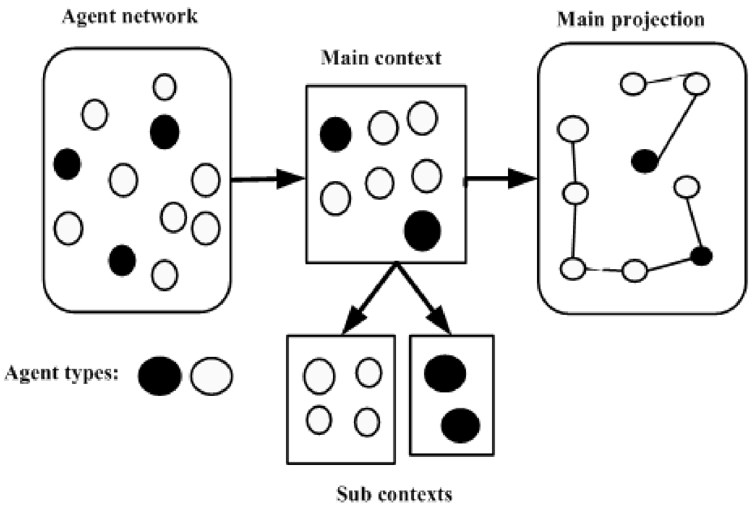


Figure 5.8: The relationships between agents in the system, contexts and projections

In the D3-FRT model, agents belong to a context, which is an abstract population of the system. The message exchange between these agents is defined within the context. Context-sensitive behaviour is implemented in the system by triggers created in the agents.

For example, it is until the RV of an agent passes a predefined threshold that the agent type is identified as a misbehaving agent.

Agents in D3-FRT are connected by dynamic links which unlike static links that are predefined and do not change. The dynamic links imply that our framework allows for dynamic interaction between agents in the system.

Therefore in summary, the structure of D3-FRT model as recommended in [MN10] has three elements:

1. A set of agents, their attributes and behaviours.

2. A set of agent relationships and methods of interaction: An underlying topology of connectedness defines how and with whom agents interact.

3. The agents' environment: Agents interact with their environment in addition to other agents.

5.5.2 Agent Behaviours, Properties and Rules

Simulation agents have behaviours, often described by simple rules, and interactions with other agents, which in turn influence their behaviours [MN09]. D3-FRT agents have behavioural features that include decision rules to select their actions of deciding on which other agent to interact with. The decision rules on which agents collaborate govern agent behaviours by employing the Bayesian theory [Bay63, HU93]. For an agent to collaborate with another, conditional probabilities are used. In order to determine the posterior probability that two agents collaborate, given the prior probability that both agents have close ratings in the system is considered. The RVs of the agents are maintained by the controller, which is the TTP in the system. The values are dependent on the behaviours exhibited by the agents overtime. From these values, the posterior probability of both agents collaborating can be determined.

An example is the hypothesis that agents A_1 and A_2 exchange a message given that they are rated almost equally or their RVs are within a close range.

$$P(A|x) = \frac{P(x|A)P(A)}{P(x)} = \frac{P(x|A)P(A)}{P(x|A)P(A) + P(x|B)P(B)} \tag{5.1}$$

$P(A)$ is the prior probability of the hypothesis regardless of any other information in Equation 5.1. That is, the probability that A_1 and A_2 collaborate while $P(B)$ is the probability there is no collaboration. The difference in ratings of the agents is denoted as x; it is the evidence that their RVs are in range and this value depends on the implementation of the application domain. $P(x)$ refers the probability of x. $P(x|A)$ is the conditional probability of seeing the evidence, if the hypothesis above is true. This is referred to as the likelihood function. That is, the probability that the agents have similar ratings given that they collaborate. While $P(x|B)$ is the probability that they have similar ratings but do not collaborate. The posterior probability $P(A|x)$ of A given x; is the estimate of the probability that the agents exchanging a message is true taking the evidence of their RVs into account. In D3-FRT's simulation component, the probability that the agents collaborate or not are both 0.5, that is, $P(A) = P(B) = 0.5$ and $P(x|A) >> P(x|B)$.

The following are the other rules that change the state of agents and these rules represent agent behaviour.

Rule 1: RV ranges between [0...5], and an agent will have a neutral state at the onset. This implies that the agent has a RV of 2.5.

Rule 2: An agent's reputation value will be decremented if it interacts with agents that are not trusted.

Rule 3: An agent's reputation value will be incremented if it interacts with other trusted agents.

Rule 4: An agent will be in a high-risk region if its computed value is 1 and below. This implies that the state of the agent is not trusted.

Rule 5: An agent will be in a medium-risk region if its computed value is above 1 but below 4.

Rule 6: An agent will be in a low-risk region if its computed value is above 4. This implies that the agent is at a trusted state.

For the modelling purposes of this research, the following properties and attributes [MN10] of agents are applied:

- Agents are social and they interact with other agents using application specific mechanism of communicating causing agent states to change.

- The agents are self-contained as they are uniquely identifiable using AgentIDs with a set of characteristics, behaviours and attributes and belong to a region of trust. RV is a metric encapsulating the reputation of each agent and determines the trust region it will automatically belong to. The trust region concept is further discussed in Section 5.7 of this chapter.

- Autonomy and self-directness: The agents can function independently and interact with other agents and controller by exchanging of messages and influence.

- Agents have behaviours that relate information sensed by each agent to its decisions and actions. An agent's information comes through interactions with other agents and with the environment.

- Each agent has a state that changes over time. That is, a state that represents the essential variables associated with the agent's current situation.

5.5.3 The Concept of Time

Timing is an important consideration of this research as it has a great impact on how the agents and the system in general evolve. Agent-based models can either be synchronous or asynchronous. In synchronous models, all agents are assumed to change simultaneously. The calling order of the objects has no influence in this mode but conflicts can arise when agents compete over limited resources. With asynchronous updating, agents change in turn, each observing the reality left by the previous agent and conflicts between agents are therefore resolved. In fact, the order of updating (often, but not necessarily, random) is critical as it may influence model results [Cro07]. Therefore, the asynchronous method of time is adopted in D3-FRT, simulating a chronological sequence of events.

In D3-FRT, the schedule function used combines *Time-Step* and *Discrete-Event* simulation schedule that uses an integer counter to track the progression of time. Time steps (referred to as *ticks*) from 0 to 1 and so on, i.e. $\{0, 1, 2, ..., t-1\}$ are used to prioritise events, rather than a simulation clock, where t is the current time. Agent behaviour occurs at one of these time steps. In addition, this approach allows for multiple agent behaviours to occur at the same time which is fit for the purpose of the D3-FRT framework.

5.6 Semi-Distributed Approach

In Chapter 2, the benefits and downsides of purely distributed and centralised architectures for RTMs are discussed. The justification for a semi-distributed framework is also detailed in the chapter. D3-FRT is a data-driven framework whose goal is to use data obtained from agent interactions for making informed trust decisions through the reputation system [OBT09, OBT10, OTB11, OBT11, OBT12]. In order to achieve this goal, we propose a semi-distributed system architecture. This is because the ultimate aim of any RTM is to achieve trusted communication 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 RTM. Secondly, prediction of agents' RVs is necessary and having a more proactive approach to the detection of malicious members is another requirement.

Application of primitives: dynamic *measurement*, *simulation*, *feedback* and *control* in D3-FRT provides dynamism in the detection of malicious agents and prediction of future behaviour of each agent. The runtime *measurements* (behaviour of agents which is converted to RVs) are *simulated* to gain a better understanding and a more accurate prediction of the level of trust for each agent. The simulation dynamically obtains RVs, and continuously incorporates new data online. This will enable the simulation to determine and *feedback* the reputation of each agent into the system. The output of the simulation will help *control* the system in terms of decisions to be made, to maintain a trusted communications among agents. The DDDAS paradigm requirement of a TTP is to monitor, simulate, feedback measurements in the system.

In modelling D3-FRT, the reputation of other agents is not entirely determined and managed by individual agents in the system but by a middle layer of trusted superagents (*Cluster Heads* (CHs)) with higher computational power (similar to the hierarchical intrusion detection systems [RMK08]) and through simulation. Figure 5.9 shows the hierarchy of the agents in the system as described in Onolaja *et al.* [OBT11].

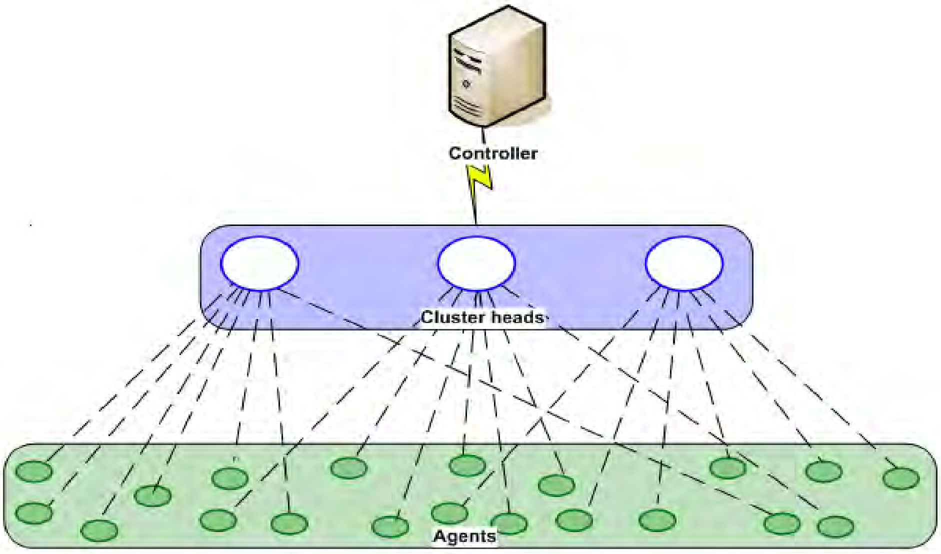


Figure 5.9: Hierarchical topology

5.6.1 Postulations

In modelling the D3-FRT, the following postulations were applied:

- The controller and CHs are assumed to be secure since a compromise of these components would render the reputation system useless. The functionality of the controller can be spread across multiple hosts for fault-tolerance and robustness; this implies that most if not all of the hosts have to be compromised before the whole system can be compromised.

- Generally, agents such as sensor nodes are constrained by limited resources such as low computational power and energy resources. This limitation does not allow for complex cryptographic requirement, algorithms or computations. Therefore, no trust requirements are placed on the domain agents.

- In order to prevent agents from having multiple identities, it is assumed that each agent is bound to an identity and cannot spoof different identities. This will ensure that agents have a verifiable and non-repudiable identity. Public Key Infrastructure (PKI) is a method by which IP can be provided with the controller acting as the *Certificate Authority.*

5.7 Initial Version of D3-FRT

At initialisation, a cynical approach is adopted where each agent has a neutral rating of 2.5; note that RVs range from [05]], where a score of 0 means an agent is completely untrusted, 5 means an agent is absolutely trusted and if $0 < RV < 5$, then it implies that the agent is trusted to a certain extent. Each agent has three global variables: θ_h, θ_o and θ_f which are the historical, recent and predicted future ratings respectively. This agent property remains the same value until the agent demonstrates its trustworthiness or maliciousness through collaboration with other agents. The behaviour of each agent will determine the adjustment of its RV accordingly.

Figure 5.10 below shows the sequence of events that occur when an agent wishes to collaborate with another. Agent A registers its presence with a CH and obtains a unique AgentID and a θ_o of 2.5. The network of agents is partitioned into clusters, where each cluster has a head: CH, which is a super-agent and has a direct connection with every other member of the cluster. The idea of clustering is a physical structure of the system.

CHs have a higher capacity and enhanced capabilities in terms of computational power, energy and storage in comparison to other agents.

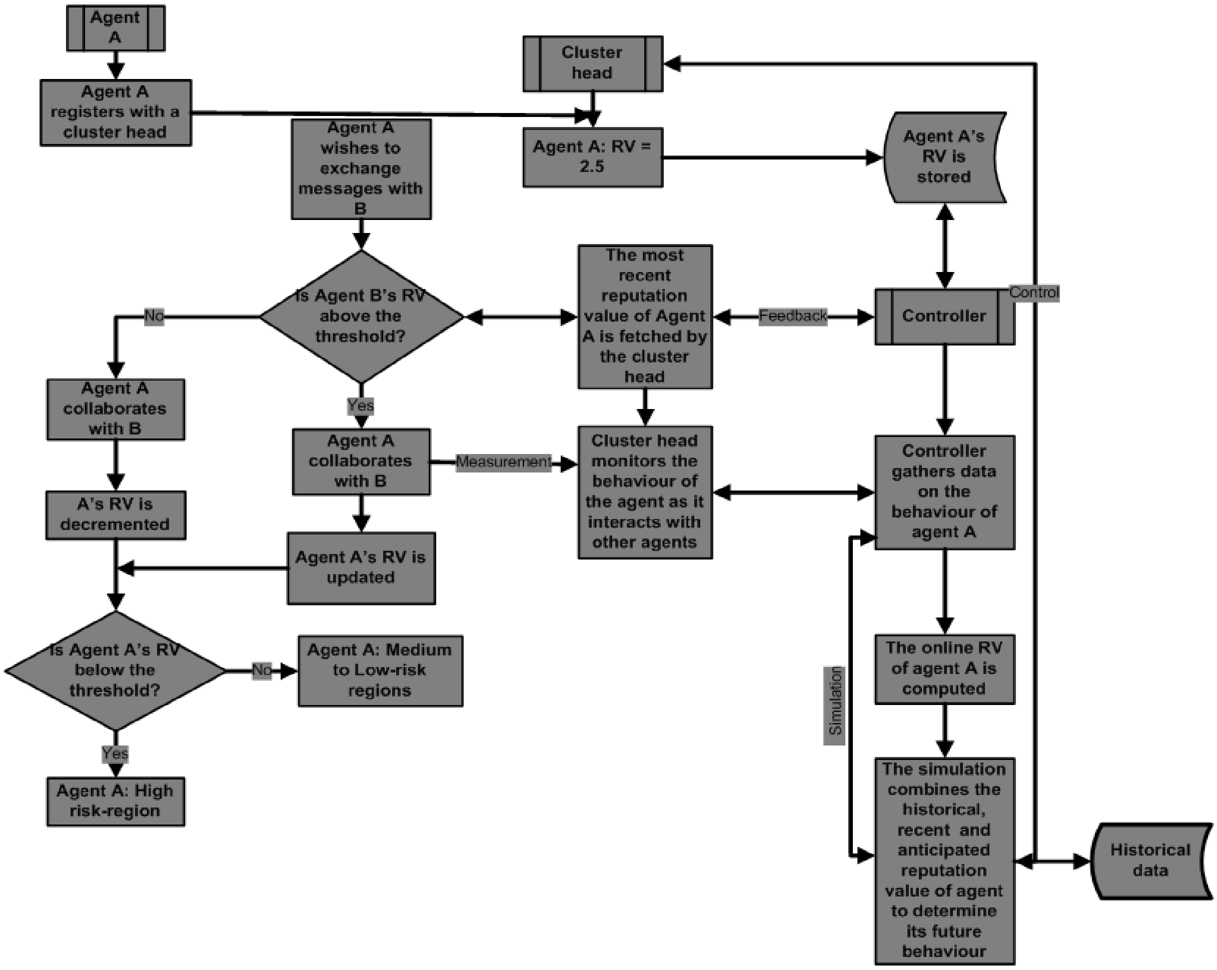

Agent A
Agent A registers with a cluster head
Cluster head
Agent A: RV = 2.5
Agent A's RV is stored
Agent A wishes to exchange messages with B
Is Agent B's RV above the threshold?
No
Yes
The most recent reputation value of Agent A is fetched by the cluster head
Feedback
Controller
Control
Agent A collaborates with B
Agent A collaborates with B
Measurement
Cluster head monitors the behaviour of the agent as it interacts with other agents
Controller gathers data on the behaviour of agent A
A's RV is decremented
Agent A's RV is updated
Is Agent A's RV below the threshold?
No
Yes
Agent A: Medium to Low-risk regions
Agent A: High risk-region
Simulation
The online RV of agent A is computed
The simulation combines the historical, recent and anticipated reputation value of agent to determine its future behaviour
Historical data

Figure 5.10: Flowchart of agent interaction and trust simulation and computation

Typically, nodes such as mobile wireless sensor nodes are limited in the computation and analysis they can carry out due to their size and power. Therefore, having every member of the network for example, running the watchdog mechanism is unfeasible. The approach of deploying CHs that will carry out the monitoring function is ideal for such networks. Instead of each agent operating in a promiscuous mode, the CH is responsible for *monitoring* and obtaining all information. A CH overhears all traffic to and from all members within its cluster. Apart from reducing the computation, this approach takes the responsibility of monitoring from the agents and does not require agents to operate in a promiscuous mode. From our review, RTMs lack effective mechanisms for monitoring reputation information. They rely on the promiscuous mode of monitoring (an assumption that is not always true in a real network). Consequently, they inherit the watchdog's detection weakness [DKB05]. In the process of capturing information about nodes, these RTMs introduce additional problems such as collusion attacks. Therefore, there is a need for an effective approach to monitoring the behaviour of nodes. This is addressed in D3-FRT by the introduction of a monitoring function by CHs. The CH is responsible for *information gathering*; that is, the collection of data. By delegating to the CHs, agents only need to request from the CHs about potential nodes to collaborate with before any transactions. Therefore in D3-FRT reputation of other agents is not entirely determined and managed by individual agents in the system, but by a group of CHs and the controller.

Agents are encouraged to collaborate with reputable agents through the incentive of increase in RVs. Therefore, if agent A wishes to collaborate with another agent B for example, agent B's reputation will affect the A's RV. Agent A requests the RV of B from the nearest CH. The CH which stores the RV provides the current RV of the agent B from its table. The *controller* obtains data about events in the domain to compute an updated RV of each relevant agent and the simulation component determines the future RVs of the agents, using historical data and recent events in the system. That is, 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 RVs of all agents. The *simulation* of the entire system utilises the processed data to predict the future behaviour of members.

Considering the computation of ratings, the model described by [MM02] gives a higher weight to past behaviour. The authors argue that a more recent sporadic misbehaviour should have minimal influence on an agents' reputation, which has been built over a long period of time. In contrast, in the approach of [BLB02, BLB05], the current behaviour of an agent carries more weight. This is to prevent agents from obtaining a good reputation with high RVs and subsequently misbehaving. The later approach is adopted in this research, with a higher weight given to recent behaviour and this is described further in Section 5.8.2.

By comparing the historical behaviour of an agent with its recent behaviour, runtime dynamic changes in rating are incorporated in trust ratings and misbehaving agents are detected. This is based on the assumption that the behaviour of a malicious agent is different from expected behaviour in the domain.

5.7.1 Computing Ratings

RVs are expressed in a continuous manner rather than in a discrete manner ranging from 0 to 5, where 5 means an agent is absolutely trusted and a score of 0 means an agent is completely not trusted. If $0 < RV < 5$, then it implies that the agent is trusted to a certain extent. The identity of each entity is used in storing its reputation information within the system and RVs are assigned according to identities.

In the initial version of D3-FRT, each agent has certain properties that are denoted by the tuple:

$$(c, a, e, \theta_o, t) \tag{5.2}$$

That is, cluster head c trusts agent a for event e with a reputation value θ_o at time t. The time is divided into time intervals $1, 2....t-1$, that is, up to the immediately previous time stamp. After an agent identifies with the nearest CH, the agent begins to build a reputation through its actions. The CH constantly monitors the behaviour of each agent in its cluster and updates their RVs at intervals. Misbehaviour threshold should not be exceeded for an agent to have an acceptable RV and allowed to participate in system activities. For example, packet drop count should not exceed a certain threshold. If the RV is within the acceptable range, then the requesting node forwards the packet through the node. Else if the RV is below the expected value, the node is avoided in the transmission.

RV's are dynamically updated by the CHs in the system as many times as an agent interacts with other agents and at specified time intervals j. The new RV (rv) is determined by the CH by monitoring agents as the collaborate. The value is computed using the formula:

$$rv = (\theta_h + (\mu * \theta_o)) \tag{5.3}$$

where μ_o is a weight introduced to reduce the effect of historical behaviour of agents on their new RVs, thereby placing more emphasis on recent behaviour. To keep the resultant value in [0...5], we consider θ_h, θ_o and μ_o at their maximum values in the following inequality, i.e. $0 \leq \frac{(\theta_h+(\mu_o*\theta_o))}{\delta} \leq 5$. The value of the inequality is 30, making $\delta \geq 6$. The value of the inequality is 0, if we consider the variables at their minimum values. Therefore, the equation can be written as:

$$rv = \frac{(\theta_h + (\mu_o * \theta_o))}{\delta} \tag{5.4}$$

One of the aims of this framework is to anticipate future RV of agents. In order to achieve this, the historical ratings of each agent are considered with their recent behaviour. To predict the future RV (θ_f) of the agents, the aggregate historical RV (θ_h) at different time intervals $j = 1, 2, 3, ...t - 1$ is combined with the current recent RV (θ_o) with respect to time, where n is the count of the number of time stamps.

$$\theta_f = \frac{\sum_{j=1}^{t-1}(\theta_h)}{n} + \theta_o \tag{5.5}$$

The property list of an agent following the prediction is given as the tuple:

$$(a, e, \theta_o, \theta_f, j) \tag{5.6}$$

Which implies that agent a in event e has an recent value θ_o and a predicted value θ_f at time j.

An aging factor a is introduced to place more emphasis on recent behaviour, which is a value to reduce the effect of previous behaviour at the specified time intervals j.

$$\theta_f = a * \frac{\sum(\theta_h)}{n}\theta_o \tag{5.7}$$

In summary, in this section we introduced the initial version of the D3-FRT framework that is capable of providing dynamic trust ratings of agents at runtime and predicting the future behaviour. This version constitutes a first step in exploiting the DDDAS paradigm to aid reputation and trust management.

5.8 Improved Version of D3-FRT

In Chapter 1, we stated that a general assumption of some RTMs is that past behaviour is an indication of future behaviour [Kol99]. This assumption might not be true with intoxication, described in Chapter 2 of this thesis.

Therefore, we argue that using historic (or past) interactions as the only basis for predicting the future RVs of identities in a domain is inadequate to provide a trusted system. The extended D3-FRT develops the supposition further by not only considering past interactions but also anticipating possible future behaviour of domain members.

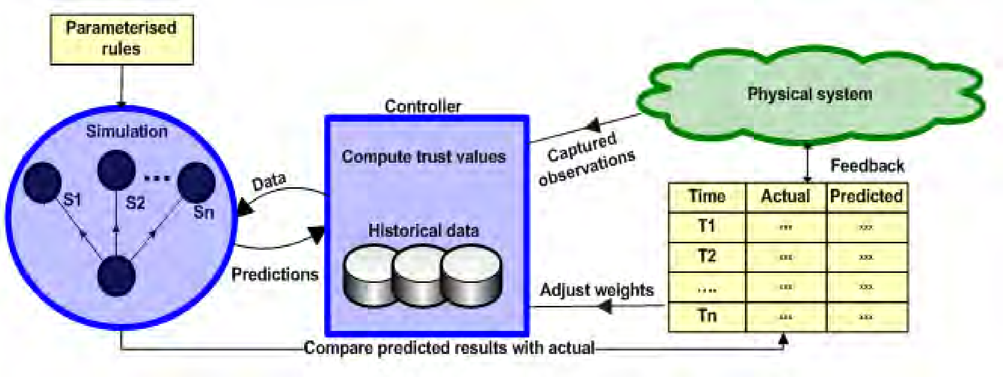


Figure 5.11: High-level diagram showing the components of the framework

In this section, an extension of the original design is presented that makes predictions of RVs with the use of historical, recent and anticipated data [OBT11]. The extended version of the framework is depicted in Figure 5.11 and a detailed version in Figure 5.14. Figure 5.12 [OTB11] illustrates the steps required for reputation computation, prediction and countermeasures in D3-FRT and the following sub-sections describe the components of the framework.

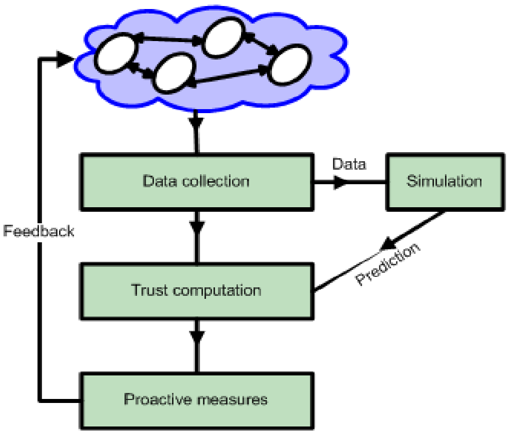


Figure 5.12: Steps to reputation computation and prediction

5.8.1 Physical System

The *physical system* may be a network such as a P2P network, MANET, a network of Wireless Sensor Nodes (WSNs) deployed at a major junction to monitor vehicular movement or a social network, which comprises of collaborating agents and agents in these domains can be in the form of network nodes, mobile phones, payment cards or other tokens, each with a unique identity. Each agent in the system has attributes and behaviours associated with it, shown in Figure 5.13, and is uniquely identifiable with its AgentID attribute. The historical (physical system and simulation), recent and future RVs, denoted θ_h^R, θ_h^S, θ_o^R and θ_f^S respectively are also attributes for each agent.

The reputation of a given agent is determined by the ratings maintained by the *controller* while the reputation of a given *region* is determined by the collective ratings of the agents that belong to the region. The *regions* of trust changes (in terms of the agents that constitute the region, the size of the region and the overall reputation of the region) as the system evolves over time. Each agent belongs to a region of trust which may either be of high-risk, medium-risk or low-risk depending on their RVs.

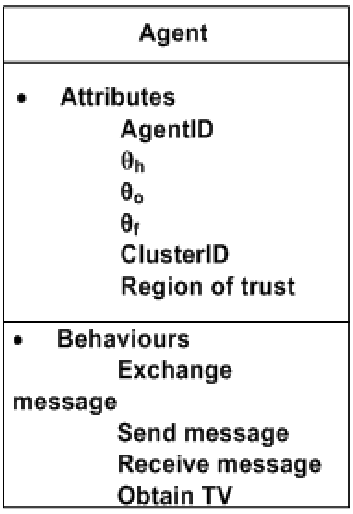


Figure 5.13: Agent attributes and behaviours

5.8.2 Controller

The controller is assumed to secure from attacks because it is more powerful device and can protect itself; it is considered as trustworthy. This controller however, may become a performance bottleneck and/or a single point of failure. However, depending on the domain of application, it is possible to have back-up controllers to provide redundancy and scalability in the framework.

The data controller obtains stream of data and filters out the useful portion for further processing. The raw data about an agent interaction is converted to a value. That is, each interacting agent earns a value depending on its behaviour in the interaction. The value obtained is added to the average θ_h^Rs for each agent in order to obtain the agent's current RV (θ_o^R) which is then stored in the database. In order to carry out these functions, the controller has the following components an *Aggregator*, a *Data Transformer*, a *Reputation Value Calculator* and a *Data Repository* that are described below.

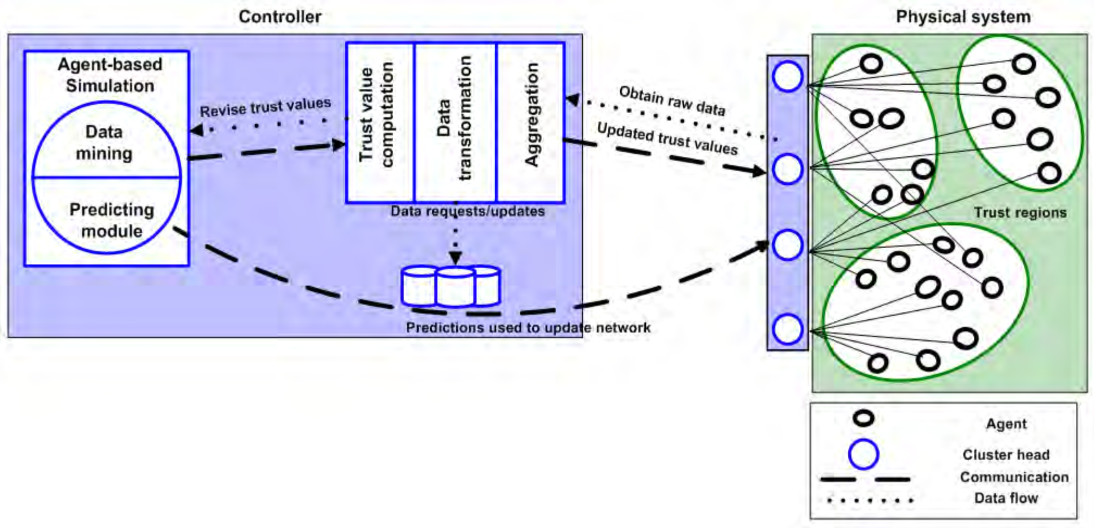
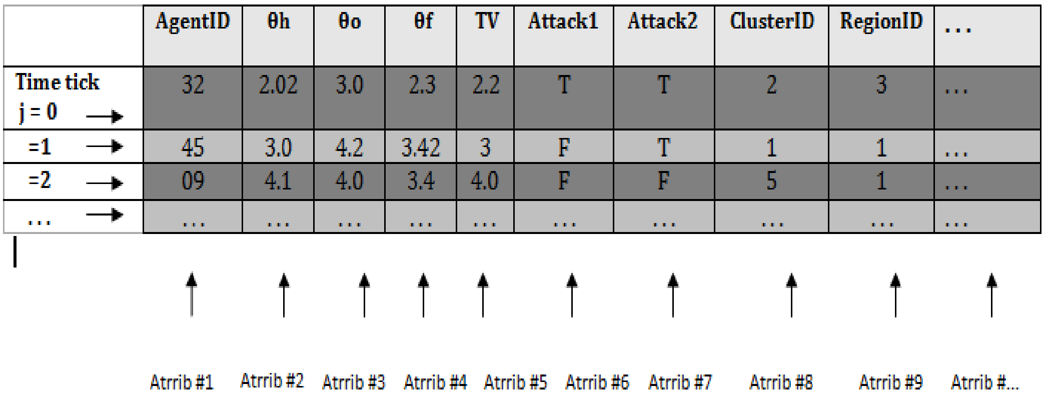


Figure 5.14: A more detailed diagram that depicts the components of the D3-FRT

5.8.2.1 Data Aggregation

Data which is a representation of the behaviour and collaboration among agents is collected from the CHs. The requirement for data to be admitted into the controller is to meet the specified criteria and any other data is simply discarded. To meet the criteria the data has to be relevant. Relevant data may include data on events about agents with RVs below the threshold and about agents that have been identified as high-risk in the system. In addition, the output of interaction with misbehaving agents is relevant and filtered as part of the useful data. Sudden changes in agent behaviour are also captured as relevant. All captured relevant data are stored in the data store and Figure 5.15 depicts sample content of the data store. Each column in the results table is a distribution of values of one variable and each row contains the record of an agent, including its RVs, trust regions and other attributes.

Each domain event has a unique identifier referred to as TransID. Data is filtered using the TransID of the events and duplicate entries are automatically discarded.



	AgentID	θh	θo	θf	TV	Attack1	Attack2	ClusterID	RegionID	...
Time tick j = 0 →	32	2.02	3.0	2.3	2.2	T	T	2	3	...
=1 →	45	3.0	4.2	3.42	3	F	T	1	1	...
=2 →	09	4.1	4.0	3.4	4.0	F	F	5	1	...
... →	...	...	...	...	...	...	...	...	...	...

Atrrib #1
Atrrib #2
Atrrib #3
Atrrib #4
Atrrib #5
Atrrib #6
Atrrib #7
Atrrib #8
Atrrib #9
Atrrib #...

Figure 5.15: A sample dataset from D3-FRT

5.8.2.2 Data Transformation

In order for a reputation system to function as it should, observations and experiences need to be captured and represented numerically. The qualitative data about the interactions that are captured is to be accurately represented as a quantitative value. That is, agent behaviour is captured, quantified and measured by a representative numerical rating - RV.

Every behavioural expectation in the system has a corresponding value and the collection of these values eventually determines an agent's current RV, as described earlier in this section. Agents are scored after each transaction and either earn or loose ratings. The Algorithm 1 shows the process by which scores and regions are determined for each transaction in D3-FRT. The value of these points is domain dependent. The act of transacting with a suspected malicious individual, with a RV of 1 (which is below the threshold of allowable RVs) for example, will result in a downgrade of the RV of a trusted agent.

A set of continuous RVs is assumed in D3-FRT and each value represents a degree of reputation as detailed in Table 5.1. This introduces flexibility into any application of D3-FRT, because different behaviour corresponds to different levels of trust [OBT09]. The

Algorithm 1 Scoring of agent behaviour and risk assessment

```
while agent collaboration = true do
    for each agent $a_i \in (a_1, a_2, \dots, a_n)$
    return agentID
    if $behaviour = good$ then
        $score > 0$
    else if $behaviour = bad$ then
        if $intox = true$ then
            $score \leq 0$
        else if $coll = true$ then
            $score \leq 0$
        else
            $score = 0$
        end if
    end if
    add score to $a_i$'s $\theta_o$
    return $\theta_o$
    compute $a_i$'s RV
    if $RV \geq 4$ then
        region ⇒ low
    else if $RV \geq 2$ and$\leq 3.9$ then
        region ⇒ medium
    else
        region ⇒ high
    end if
    return $regionID$
end while
```

reputation degrees are similar to the value stratification idea (a form of fuzzy logic) in the work of Marsh [Mar94]. Another advantage of having such degrees is that a rating designated as excellent is acknowledged across the domain as excellent. Thus, we avoid the problem of what does a trust of 0.5, or 50% mean? Is it high or low?, for example [Mar94].

5.8.2.3 Data Repository

Agents in D3-FRT have a verifiable and persistent identity attached to their behaviour in the system. IP is an important factor to be considered in any RTM as action and reaction

Table 5.1: Table showing the degrees of reputation and corresponding regions of risk

Reputation Value	**Meaning**	**Description**	**Region**
5	Complete trust level	Trusted agent with an excellent reputation	Low-risk
4	Good trust level	Very reliable agent	Low-risk
3	Average trust level	Average value and somewhat reliable agent	Medium-risk
2	Average trust level	Average value but questionable agent	Medium-risk
1	Poor trust level	A questionable agent	High-risk
0	Complete distrust level	Malicious agent with a bad reputation	High-risk

events have to be accounted for and stored for future reference in a repository. D3-FRT automatically attaches an identity to each agent on their entrance to the system. This allows for easy auditing of events that occur in the system.

The repository acts as a historical data source and archive. The data transformation function is performed and the RVs of all agents are stored on a reputation table in the repository, with each row of the table containing the RVs uniquely identifiable agents. These historical values of the agents that are stored as RVs and fetched from the data repository are used as evidence in order to predict future possible values.

5.8.2.4 Computation of Reputation

Computation of reputation and trust is very difficult, as it has to be defined in a very precise way. This is crucial to the fulfilment of the functions of any trust-based framework. Computing reputation and/or trust in RTMs has been described as an abstract mathematical specification of the transformation of available information to a usable metric [HZNR09]. In this framework, the specification is made through explicit equations that are discussed in this sub-section.

The RV of domain agents are dynamically updated at specified time intervals j in our framework, where $j = \{0, 1, 2, ..., t-1\}$. Note that $t-1$ is the last time a snapshot of the physical system was taken. Using the notation of θ_h^R to represent the historical RV and θ_n^R to be the newly computed RV of an agent after each interaction, we define the reputation value θ_h^R of an agent with time to be

$$\theta_h^{R(i)} = \frac{1}{t-1} \sum_{j=1}^{t-1} (\theta_n^{R(j)}) \tag{5.8}$$

That is,

$$\theta_h^{R(i)} = \frac{\theta_n^{R(1)} + \theta_n^{R(2)} + \theta_n^{R(3)} + ... + \theta_n^{R(t-1)}}{c(\Delta t)} \tag{5.9}$$

where the new RV of an agent is denoted as θ_n^R. θ_h^R is the average of the sum of the previous RVs up until the time $t-1$. $c(\Delta t)$ is the count of the computed θ_n^R. The current RV, derived from agents' recent activity is denoted θ_o^R. This is the sum of observations captured and represented numerically for each agent. The new RV θ_n^R of an agent at time i is defined as

$$\theta_n^R = \frac{\mu_h \theta_h^{R(i)} + \mu_o \theta_o^{R(i)}}{\mu_h \mu_o} \tag{5.10}$$

In order to determine the best range of values for the scaling factors μ_o and μ_h (which are factors for the current and historical RVs respectively), we prove that:

$$\theta_n^R = \frac{\mu_h(\theta_h^{R(i)})}{\mu_h \mu_o} + \frac{\mu_o(\theta_o^{R(i)})}{\mu_h \mu_o}$$

$$\theta_n^R = \frac{(\theta_h^{R(i)})}{\mu_o} + \frac{(\theta_o^{R(i)})}{\mu_h}$$

We can consider $1/\mu_o$ as a coefficient of θ_h and $1/\mu_h$ as that of θ_o. This is equivalent to $a\theta_h + b\theta_o$ where $a = 1/\mu_h$ and $b = 1/\mu_o$. This applies even if the number of components is greater than 2.

Now to find the coefficient to ensure that the result is below maximum reputation value and above the minimum: $0 \leq a\theta_h^R + b\theta_o^R \leq max$ in the case of our framework, $max = 5$ and $min = 0$. In the worst-case scenario, if θ_h^R and θ_o^R are both at a maximum value: $\Rightarrow 0 \leq amax + bmax \leq max$. We have an inequality, as $max > 0$.

$$\Rightarrow 0 \leq a + b \leq 1 \tag{5.11}$$

$$\Rightarrow 0 \leq \frac{1}{\mu_o} + \frac{1}{\mu_h} \leq 1 \tag{5.12}$$

Therefore, Equation 5.10 can be rewritten as: $\theta_n = \mu_h\theta_h^R + \mu_o\theta_o^R$ as long as $0 \leq 1/\mu_o + 1/\mu_h \leq 1$. This inequality equation specifies the range for the scaling factors used.

5.8.3 Simulation

The simulation of the system and the real system both run concurrently. However, this component of the framework works ahead in time of the system. The aim of the simulation is to predict RV of members by using past interactions, current events and possible future scenarios. At specific time slots, the current state of the system is obtained and adapted to the simulation.

Dynamic data (recent behaviour) is incorporated in the simulation, which helps with analysis and prediction of the reputation of each agent. We are interested in modelling the application level behaviour of the agents rather than the low level protocols involved in a network. To this end, D3-FRT utilises Repast (which we extended) to implement its predictive capabilities. Depending on the behavioural rules (discussed in Section 5.5.2) incorporated into the agents in the simulation, the predicted RVs of each agent change using probabilities of collaboration among the agents. These changes are indicators of possible expectations in the physical system.

Data collected are the recent RVs (θ_o^R) that represent the current rating of an agent and the other computed RVs. These values from the system are injected into the simulation at the start. The simulation runs for more time steps and considers different '*what-if*' scenarios in which an agent may be in the near future. Possible outcomes of the '*what-if*' scenarios are simulated to anticipate possible fluctuations in agent behaviour. This is because the behaviour of members generally in any network, domain or context is dynamic and changes with time. The predefined scenarios that are considered in this research are collusion: message dropping and message altering, intoxication and normal expected behaviour.

The simulation considers the probability of misbehaviour of an agent and that of the collaboration between the two or more agents through the predefined behavioural rules incorporated in the simulation agents. The assumption here is that agents in the same risk domain are more likely to collaborate. The resulting RV for an agent in each scenario is considered and with this information, it is possible to average the ratings and obtain the θ_f^S of the agent. In addition, the scenarios can be given different scaling factors.

In formalising the simulation process, the following model is used:

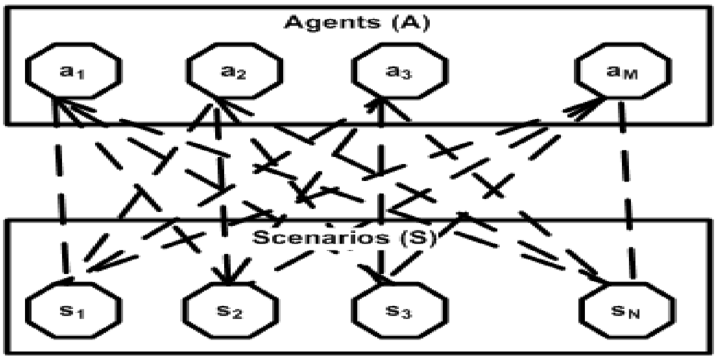


Figure 5.16: Model for the simulation rating process, where arrows indicate the scenarios S that Agent A is considered

$Pr(S)$ is the probability of a scenario S occurring, R is the rating in a particular scenario and $\overline{x}$ is the mean value. f is the factor of importance that is chosen relative to the probability of the occurrence of a scenario; an example is for $f = high$ to be set to 1 and $f = default$ set to 0.2. These values including the probability *limit* depend on the domain of application and can be determined by the model designer. The simulation stability in Algorithm 2 indicates when the simulation closely reflects the physical system; the simulation can be passed through iterations of learning process until the system stability is obtained depending on the design decisions of the modeller. The results from the simulation is then combined with recent and historical RVs in order to obtain an overall RV by computing the future θ_f^S. The overall RV is computed as:

$$RV = \mu_h \theta_h^R + \mu_o \theta_o^R + \mu_f \theta_f^S \tag{5.13}$$

where μ_f is a scaling factor for the predicted value which can be adapted depending on the domain requirements. The scaling factors are used to control the effect of historical behaviour of agents on their recent activities. For example, if $(\mu_o, \mu_h) > 0$ and $\mu_o > \mu_h$, this places more emphasis on recent behaviour as opposed to historical. μ_o should be

Algorithm 2 Simulation algorithm inspired by DDDAS paradigm

```
start simulation
input data feed from source
while simulation state is not stable do
    adjust parameters
end while
select a set $\{a_1, a_2, a_3, ..., a_M\}$ of high-risk agents
for each agent A $\in \{a_1, a_2, a_3, ..., a_M\}$
in scenarios $S \in \{s_1, s_2, s_3, ..., s_N\}$ generate rating R
return R
if $Pr(S) \geq limit$ then
    f is high
else if $Pr(S) < limit$ then
    f is default
end if
compute $\theta_f^S := \overline{x}(R \times f)$
return $\theta_f^S$ for each A
output feedback data to destination
end simulation
```

selected from the interval [0...1] and μh from the interval $[0...\texttt{min}(\mu_o, 1 - \mu_o)]$ based on the preceding condition and $0 \leq$ sum of factors ≤ 1. Therefore, if for example $\mu_h = 0.3$ and $\mu_o = 0.5$, then $\mu_f \leq 0.2$. In selecting the initial or default values of the factors, a modeller may run several iterations of the model in order to establish the most optimal values in their environment.

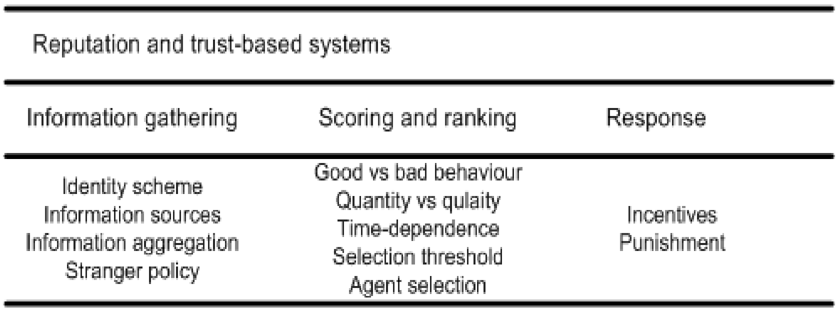
After some specified time intervals $t_1, t_2, ..., t_n$, the simulation state is observed and compared with the actual state; this comparison is done automatically in the controller. The framework can be self-adaptive by re-tuning and optimising the values of the scaling factors (μ_h, μ_o, μ_f), such that if there are any differences in the predicted values and the reality, the factors can be continually adjusted to reflect reality. Each instance of the adjustment always ensures that the condition $\mu_o > \mu_h$ holds. This implies that an entity's most recent action has more impact on its RV than past actions.

Also, if there are any discrepancies between the predicted value and the RVs in reality, questions are raised and the role of the simulation is to find answers to the questions. Assuming an agent A (in a low-risk region) collaborates with an agent B (in a high-risk region), what happens to the RV of agent A? Will A be moved to a high-risk region? What could make A and B collude? How will the regions evolve over time?

The output obtained from the simulation helps in identifying regions of high-risk, medium-risk and low-risk in the domain. The grouping helps in the management of the system to focus on critical group of agents that require more attention. This will equally aid future security-aware decisions in the domain.

5.9 D3-FRT and Other Reputation Systems

In [MGM06] the breakdown of the functionalities of RTMs are divided into: information gathering, scoring and ranking, and response. A RTM collects information on the transactional behaviour of each agent (information gathering), scores and ranks the agents based on expected reliability (scoring and ranking), and allows the system to take action against malicious agents while rewarding contributors (response). Each component requires separate system mechanisms (listed in Figure 5.17).



Reputation and trust-based systems		
Information gathering	Scoring and ranking	Response
Identity scheme Information sources Information aggregation Stranger policy	Good vs bad behaviour Quantity vs qulaity Time-dependence Selection threshold Agent selection	Incentives Punishment

Figure 5.17: Breakdown of reputation system components [MGM06]

Matching these functionalities to D3-FRT's components, the information gathering function is performed by the agents and CHs in the domain. Also, cooperative and uncooperative behaviours are used to score and rank agents. The ranking of an agent determines how other agents will want to collaborate with it. Agents receive incentives of increase in RVs and risk levels or are punished by decrease in RV, which could eventually result in exclusion from the system. Table 5.2 gives a comparison of the extended version of D3-FRT with the some existing RTMs that have similar trust management functionalities as the framework.

Table 5.2: Summary table comparing existing RTMs with key functions of D3-FRT

	CORE	**EigenTrust**	**CONFIDANT**	**RFSN**	**D3-FRT**
Monitoring	Watchdog mechanism	Peer recommendation	Watchdog mechanism	Watchdog mechanism	Controller & CHs
Simulation	n/a	n/a	n/a	n/a	Simulation of possible future states
Dynamism	Ratings are not constant	Periodic iterations to compute global ratings	Periodically updated	Provides real time feedback	Current ratings and control at specific intervals
Prediction	n/a	Past interactions serve as an indication of ratings	n/a	Trust metric that is representative of a nodes' future behaviour	Prediction of ratings using data from historical, current and possible future RVs

It is worth noting from Table 5.2 that whilst RFSN (in comparison with D3-FRT and unlike the CORE, CONFIDANT and EigenTrust) has implemented the functionalities that we focus on in this thesis, the protocol still relies on the watchdog mechanism for monitoring and observing domain events.

5.10 Summary

In this chapter, we introduced the two versions of D3-FRT. The framework is capable of providing dynamic trust ratings of agents at runtime and predicting the future reputation values through the simulation of historical and recent behaviour. The framework does not rely on collective opinion and ratings to determine the reputation of system entities as it has been shown that such an approach can result in collusion. Instead, the framework predicts a potential compromise before the attack occurs so that informed decisions can be made, which may include isolating misbehaving agents or even denying them service.

The initial version of D3-FRT constituted a first step in exploiting the DDDAS paradigm to aid prediction in reputation management. Furthermore, the extended version of D3-FRT does not rely on past histories alone but also on anticipated events in the domain. This makes the framework proactive as a more representative picture of the framework is made available through simulation. The simulation component of D3-FRT considers the probability of an agent misbehaving and the probability of collaboration between the two or more agents. This is done using the predefined behavioural rules incorporated in the agents in simulation. Through a continuous learning process, knowledge converted from captured evidence is used to predict future possible behaviour of agents. This component improves the predictive capability of the framework and provides adequate feedback that enables for example, the administrator to manage and control the network by making security-aware decisions.

Chapter 6

Empirical Study

6.1 Introduction

So far this thesis has discussed reputation and trust management and related issues in detail. In Chapter 3, relevant approaches that have contributed significantly to this area of research and the issues they encounter were discussed. Chapter 5 introduced D3-FRT's approach that exploits the DDDAS paradigm primitives with an agent-based simulation approach, to make predictions about domain members' ratings. This chapter builds up on the previous chapters, and investigates D3-FRT framework through empirical and qualitative studies. As a proof of concept, a P2P network case study with file-sharing peers is implemented and is evaluated through different scenarios.

The significance of this research lies in the fact that trust models have, in recent times been considered for use in more critical domains such as in vehicle and traffic monitoring [DJLZ10, GM12a]. Security in these domains is vital because attacks potentially have a great impact on the society, and could result in loss of human life and property. D3-FRT is valuable enough to be adopted because of its applicability in these domains. What is being investigated in this research is different from previous trust models particularly because expectations are anticipated in the domain to influence predictions and decisions. The research objectives listed below are reasonable and are achievable:

- Facilitate dynamic changes to ratings of agents as they collaborate. Provide an immediate metric that expresses agents' recent behaviour.

- Anticipate the future ratings of agents and provide possible RVs in the future. This results in a proactive trust management and allowing for informed decisions to be made by stakeholders about the domain.

The experiments in this chapter are carried out to illuminate and test the objectives listed above. More specifically, we present results from a series of experiments to show the effectiveness of D3-FRT in various domain conditions and scenarios. Section 6.2.1 examines the predictive capability of the framework in terms of the level of accuracy offered while Section 6.2.2 demonstrates the error of estimate by comparing actual reputation values with the predictions made by D3-FRT. Section 6.2.3 of this chapter illustrates the behaviour of the framework in the presence of pair-wise with n-wise collusion attack in the reputation system. The usefulness of the proposed framework in providing dynamism by being proactive, and anticipating future events is tested in Section 6.2.4. The computation of reputation values as the system evolves is tested to show the dynamic nature of the D3-FRT framework. As stated in previous chapters, dynamism is essential to having a reliable and trusted domain. The impact of collaboration among the domain members is assessed and relationship between the rate of interactions and computation is evaluated. The effect of varying network conditions on the framework's performance is investigated in Section 6.2.5 and in Section 6.2.6, the need for the DDDAS components for trust management is examined. An experimental comparison between TrustGuard trust model and D3-FRT is explored in Section 6.2.7 and a qualitative evaluation with other models is detailed in Section 6.3. Finally, Section 6.4 concludes this chapter.

6.2 Experimental Setup

D3-FRT has been implemented using the Microsoft.NET Framework version of Repast modelling framework, a free and open source tool, which provides a visual mode of development. Initially to model the framework, the point-and-click development environment was used to generate Java classes; however, this did not provide the flexibility required to build our model. In order to gain the higher modelling power required in this research, Microsoft.NET C# framework version of Repast is used. Furthermore, for the purpose of the experiments in this chapter, additional tools used are: Microsoft SQL server for storage of captured observations in the network and MATLAB for the analysis of the data captured.

The evaluation of D3-FRT is done using a P2P network as a test-bed as the domain is representative of the complex interactions found in domains that require trust management to function properly. The network consists of an interacting set ($p \in N$) of distinct peers, that are connected to a CH, where each peer has direct and indirect links with other peers in a cluster. N represents the entire network population.

In setting up the experiments and in order to obtain realistic results, we model the simulation to resemble a real life P2P as much as possible. The network is modelled with peers interacting with others using the communication mechanism found in a P2P network, causing peer states to change. Using the suggested local and peer-level parameters by Schlosser *et al.* in [SCK02], we model the network by focusing on the content distribution and peer behavioural parameters. We assume random interaction patterns among the peers and the peers function independently and interact by means of message exchange.

In modelling the network, the following tasks can be performed by each peer: a) request for files, b) respond to a request, c) transfer files and d) obtain the rating of a peer. The CHs maintain an index of files that available on the peers within a cluster and can make requests to other CHs. To determine the number of files shared we use the probability distribution of Napster P2P file-sharing network [SGG02], which is $40 - 60\%$ peers sharing $5 - 20\%$ of the files.

In determining peer behavioural parameters, peers in the network remain from the start of the simulation to the end. File requests are made randomly and sharing between peers are not necessarily symmetric; for example, a peer A may request a file from peer B whereas peer B might not request a file from A. All peers in a cluster are mutually connected and peers do not abort transfers; all interactions are completed. The peers are self-contained as they are uniquely identifiable with an AgentID and with a set of attributes: historical, current, predicted and overall RVs as described in Chapter 5. The peers exhibit different behaviours which include intoxication (`ITX`), collusion (`COL`), active in file upload and download (`NOR`).

In designing our experiments, we injected intoxication and collusive attacks using the Algorithms detailed in Appendix A of this thesis. It is worth noting here that, in our experiments in this chapter, we make a real-life assumption that misbehaving peers are always few compared to the entire network population.

Throughout the experiments, there is an *actual* and *predicted* RV for every interaction for each peer per time. The variances between actual and predicted values in the experiments are computed using the mean squared error in Equation 6.1.

$$MSE[predicted; actual] = E[(predicted - actual)^2] \tag{6.1}$$

Error bars are displayed using 95% confidence intervals, as is standard practice. The methods we use to compute the confidence intervals, along with analysis of variance are as described in [Coh95]. In this section, each experiment is repeated 10 times and average of the results is reported, unless otherwise stated. Time is measured in *ticks*, which is a compression of time in our simulation.

6.2.1 D3-FRT's Predictive Capability

The purpose of this experiment is to evaluate the performance of the D3-FRT, in terms of its predictive capability in the presence of collusion and intoxication attacks.

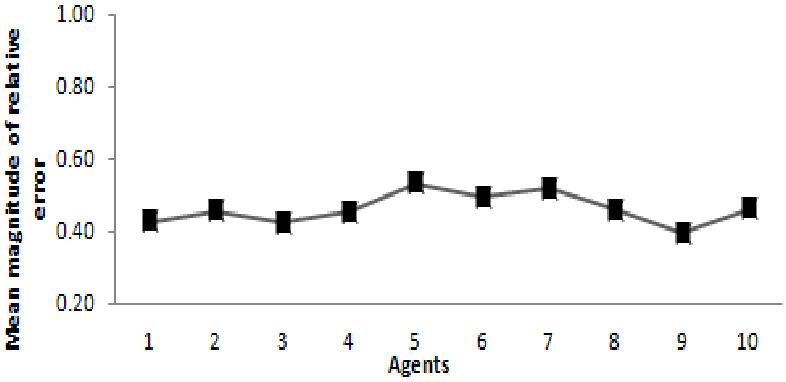


Figure 6.1: Average reputation value prediction error

The following parameters were used throughout the simulation: $N = 50$ agents with number of $CHs = 2$, each with an initial RV of 2.5. RV ranges between 0 and 5 and an agent with a RV of 1 and below is regarded as misbehaving. The simulation total time was 1000 ticks and 30% of the agents eventually misbehaved. The predicted RV is computed using Equation 5.13 and the following scaling factors were used: $\mu_o = 0.5, \mu_h = 0.3$ and $\mu_f = 0.17$. In the experiment, agents interact randomly as in P2P network and these interactions are captured at every 1 tick. Figure 6.1 provides an indication of the predictive capability of D3-FRT.

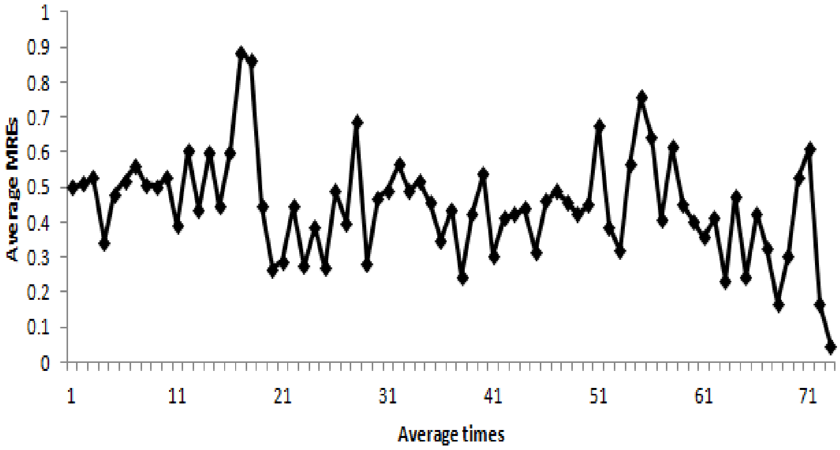


Figure 6.2: Mean magnitude of relative error per agent.

The predictive accuracy of D3-FRT is measured by comparing the agents' RV in the network with the predicted RVs. An average Magnitude Relative Error (MRE) [CDS86]: $\left|\frac{RV_{actual}-RV_{predicted}}{RV_{atual}}\right|$ is computed for all agents for the duration of the simulation averaged over a set of randomly selected agents. Figure 6.2 shows the Mean Magnitude Relative Error (MMRE): $\frac{1}{N}\sum_{i=1}^{N}\left|\frac{RV_{actual}-RV_{predicted}}{RV_{actual}}\right|$ where $N = 10$ agents in the previous experiment and the result ranging between 0.39 and 0.5 and the overall MMRE, averaged at ≈ 0.46. The MMRE remained below 1.0 throughout the simulation runs. A possible explanation for this might be that the simulation does not have any prior knowledge of the network initially. However as the system evolves, the simulation converges as shown in the next set of experiments.

Consequently, given the MMRE, there is a difference between the actual and the predicted values in D3-FRT and this might account for possible false-positives and false-negatives in the system.

6.2.2 Error Rate in Predictions

In order to test the reliability of the framework, the error rate in predictions made is tested using different parameters and scaling factors to compute RVs. The acceptable error rate of D3-FRT will depend on the criticality of the application domain, and the risk appetite in the domain. With a penalty of 0.4 and 0.5 for intoxication and collusive behaviour respectively, and a reward of 0.5 for normal behaviour, the following default parameters are used: $\mu_h = 0.3, \mu_o = 0.5, \mu_f = 0.17$. Initially, several iterations of the model were done and the observation was that these default values produced the optimal results. For every 100-tick cycle of the simulation, $N = 50$ with 12% and 10% of the agents exhibiting collusive and intoxicating behaviours and for the duration of 9000 ticks.

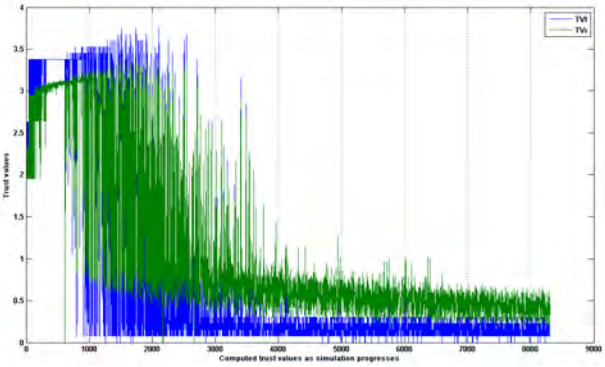


Figure 6.3: θ_r and θ_f, $\mu_o = 0.5$

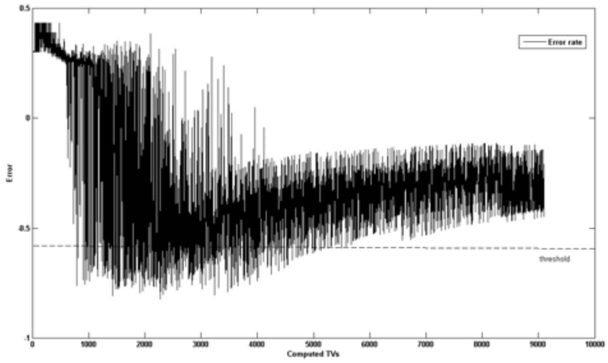


Figure 6.4: Estimation error, $\mu_o = 0.5$

Figures 6.3 and 6.4 show the actual RVs versus the predicted values during the simulation. In these experiments, there is a 5.1% prediction error rate at an error threshold of ±0.6.

In order to reduce the possibility for intoxication attacks in the experiments, μ_o has to be greater than μ_h; reducing the effect of historical behaviour on predictions and all reputation computations. This is because choosing a larger value for μ_h biases the reputation value of an agent a to the observations currently received about a. A larger value of μ_o gives heavier weight to the performance of the agent in the past. With μ_o set to 0.6 and retaining the error threshold value from the previous experiments, D3-FRT made some inaccurate predictions about agent reputation values at the rate of 15%; this is much higher than error rate of when $\mu_o = 0.5$. The graph in Figure 6.5 shows the overall RVs of the peers compared with the predicted values. From the figure, as the simulation progresses the actual and predicted values seem to deviate. In Figure 6.6, from approximately 5500 ticks in the simulation, the error rate reduced considerably and remained above the lower bound -0.6 of the threshold.

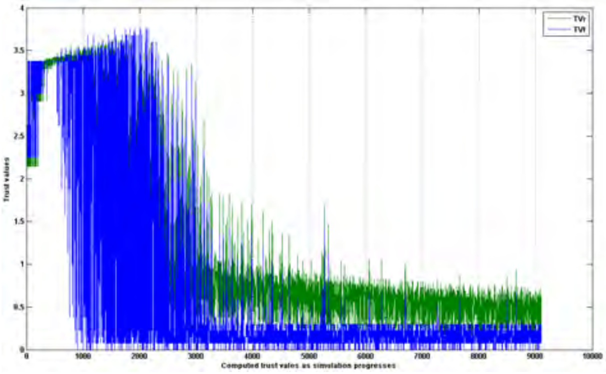


Figure 6.5: θ_r and θ_f, $\mu_o = 0.6$

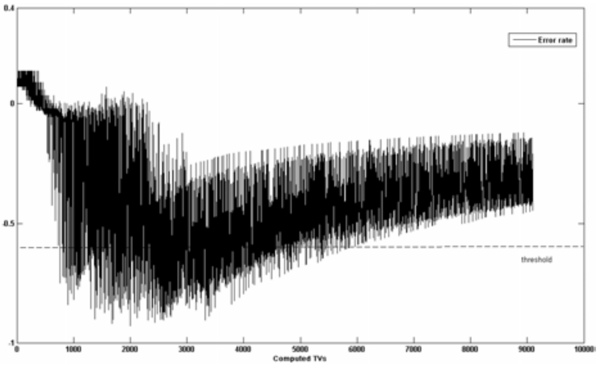


Figure 6.6: Estimation error, $\mu_o = 0.6$

We tested the behaviour of D3-FRT when μ_o is the same value as μ_h which is 0.3 and the results are given in the Figures 6.7 and 6.8 below. The error rate in these experiments is 21% and Figure 6.8 shows that the errors in prediction initially fluctuated above and within the threshold boundary but later remained within the threshold boundary from around 3800 and for the rest of the simulation.

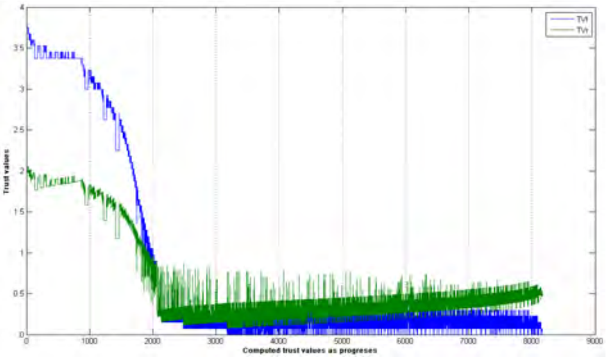


Figure 6.7: θ_r and θ_f, $\mu_h = \mu_o = 0.3$

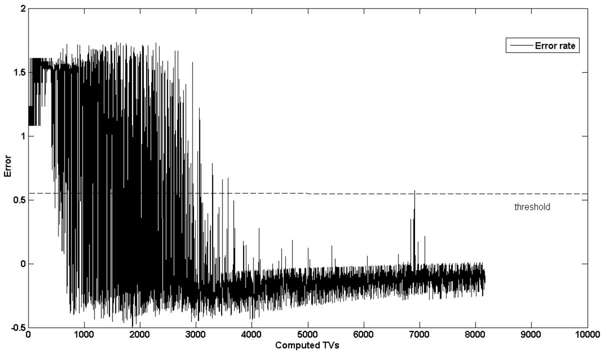


Figure 6.8: Estimation error, $\mu_h = \mu_o = 0.3$

From these, one might remark that the best value for the scaling factors are $\mu_h = 0.3, \mu_o = 0.5$ as the best results were obtained with these values and in addition reducing the possibility of intoxication attacks.

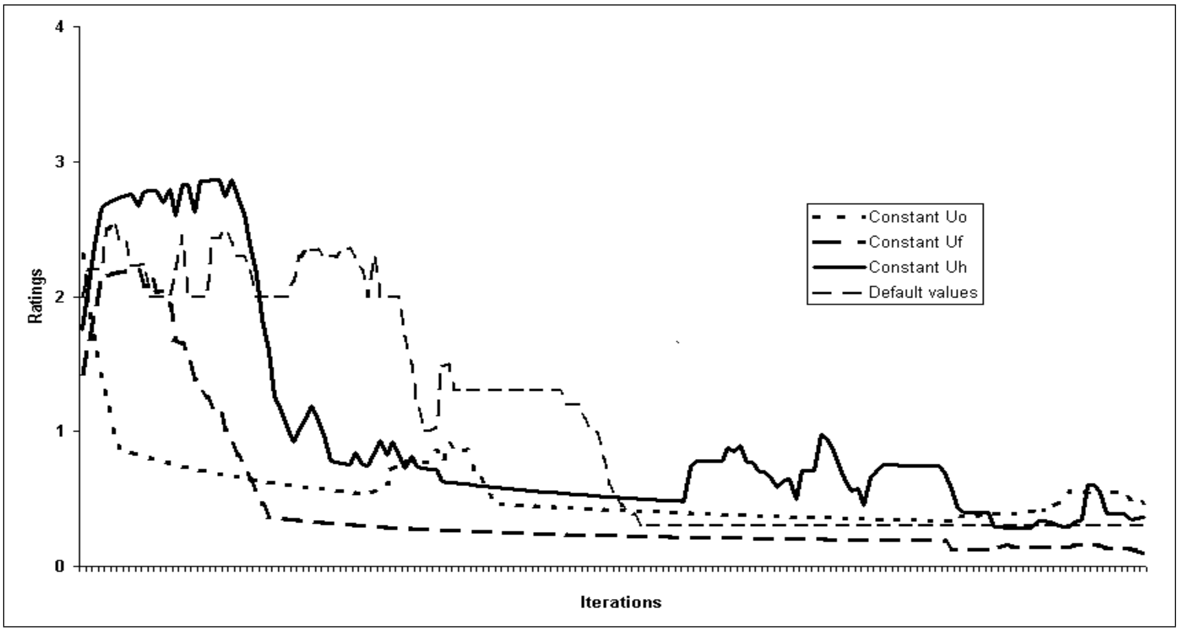


Figure 6.9: Comparison of ratings in different values of the scaling factors for collusive agents

Furthermore, when using quantitative data, it is possible that small differences in individual values produce relatively large differences in the overall ratings [Mar94]. Since the mathematical model is defined by a series of equations, parameters, and variables, it is subject to many sources of uncertainty including errors of measurement, absence of information and poor or partial understanding of the driving forces and mechanisms. This uncertainty imposes a limit on our confidence in the response or output of the model [Cen12]. In our case, we try to gain insight on the whether our model is sensitive to the scaling factors used in computing agent ratings. The aim of the analysis is to determine the effect of the scaling factors on computed values.

The sensitivity analysis is performed by running different simulations of the algorithm with modification of parameters in order to evaluate the behaviour of the algorithm. When conducting the experiments, we run 10 simulations of the algorithm for each parameter, to make statistically correct conclusions. We analysed 3 parameters: μ_h, μ_o, μ_f and 3 different values were used for each parameter and for each iteration, μ_h is always greater than μ_o. These parameters are important to our model because they influence the accuracy of the model output. The simulations were carried out using the default values for the parameters chosen ($\mu_h = 0.3, \mu_o = 0.5$ and $\mu_f = 0.17$). The results from using default values are used as a benchmark for the other values tested, because with default values our model is optimal, as shown in the the preceding experiments of this subsection.

In Figures 6.9-6.12 we show the sensitivity of D3-FRT with respect to changes in its input parameters based on the type of behaviour exhibited. Constant μh in the graph legend is the trend line obtained from keeping the factor constant at its default value whilst varying the values of the other factors and this is the case for the trend lines of constant μo and constant μf as well (using the values in Table 6.1).

From these, it can be observed that the values of the scaling factors affect the resultant

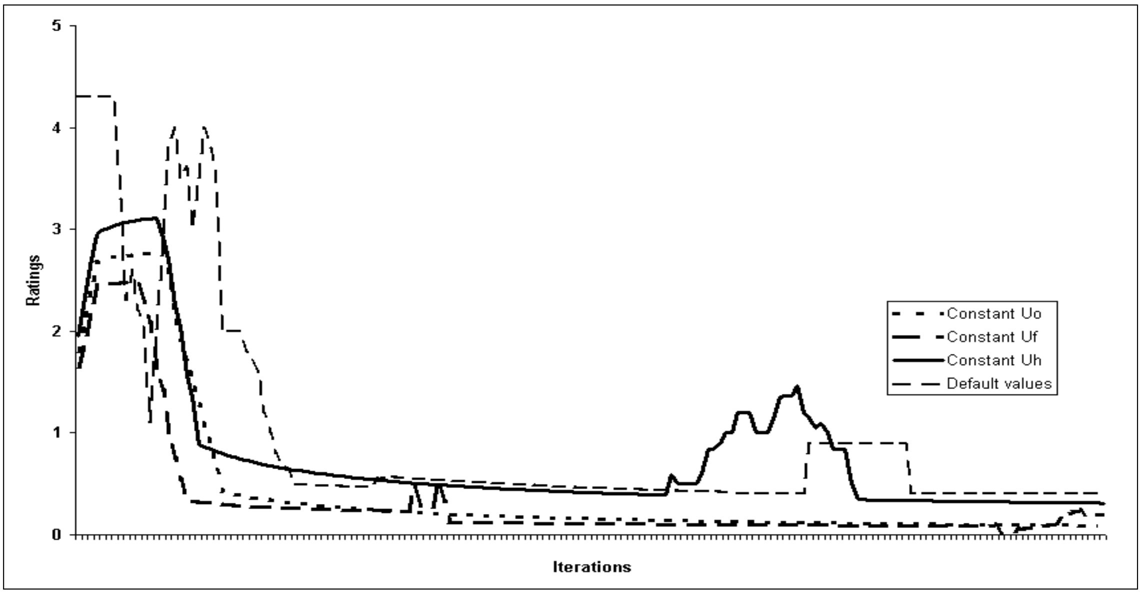


Figure 6.10: Comparison of ratings in different values of the scaling factors for intoxicating agents

Parameter values for sensitivity test		
Parameter 1	Parameter 2	Parameter 3
$\mu_o = 0.5$	$\mu_f = 0.1$ $\mu_f = 0.17$ $\mu_f = 0.2$	$\mu_h = 0.4$ $\mu_h = 0.3$ $\mu_h = 0.2$
$\mu_f = 0.17$	$\mu_h = 0.2$ $\mu_h = 0.3$ $\mu_h = 0.4$	$\mu_o = 0.3$ $\mu_o = 0.4$ $\mu_o = 0.5$
$\mu_h = 0.3$	$\mu_f = 0.1$ $\mu_f = 0.17$ $\mu_f = 0.2$	$\mu_o = 0.4$ $\mu_o = 0.5$ $\mu_o = 0.6$

Table 6.1: Parameter list for sensitivity test

ratings. Although, the graphs show similar trend lines for the different cases, slight changes to the input values have resulted in noticeable variations in the output. In each of the graphs, one of the variables is kept constant at its default value while the other factors change.

The experiments done have shown that there is a noticeable change on the behaviour of the model depending on the values of the parameters analysed. The motivation behind the sensitivity analysis of the D3-FRT algorithm was to identify the influence of the variability

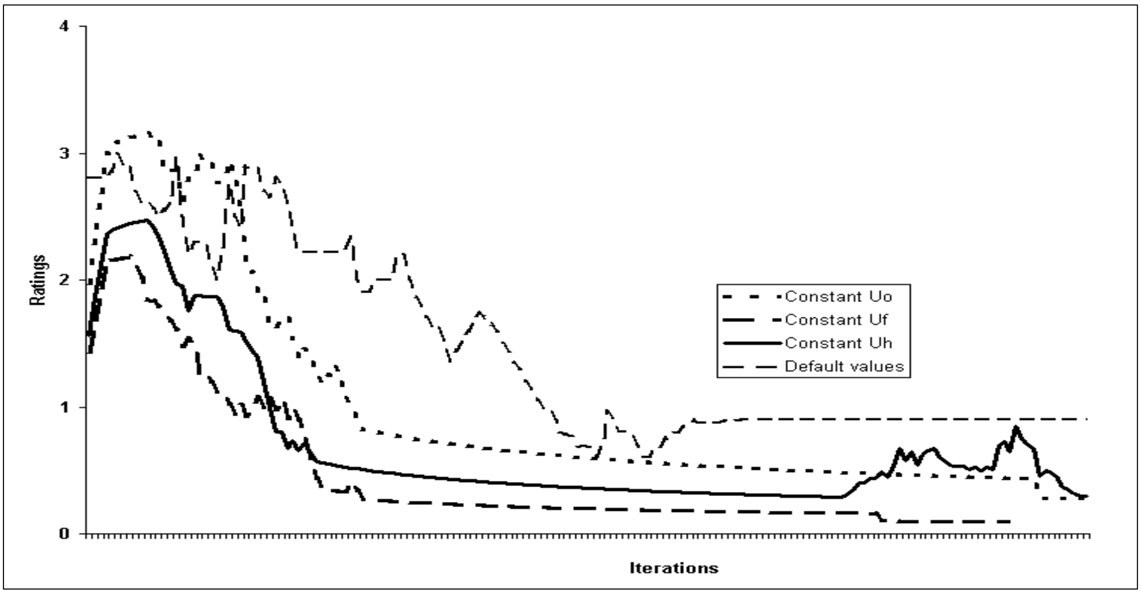


Figure 6.11: Comparison of ratings in different values of the scaling factors for collusive and intoxicating agents

of the parameters of the algorithm in order to know which at which point the parameters can improve the performance of the algorithm. The ratings obtained from running the experiments using default values are significantly different from the other ratings. In particular, we have proven that the use of adaptive factors in computing ratings potentially influences the behaviour of our model.

6.2.3 Pair-Wise Collusion versus n-wise Collusion in D3-FRT

The performance of the framework in the presence of pair-wise collusion versus arbitrary number of collusive agents is tested in this section. This is to analyse the behaviour of D3-FRT in the presence of different numbers of collusion nodes and to assess how the framework adapts in the presence of misbehaviour.

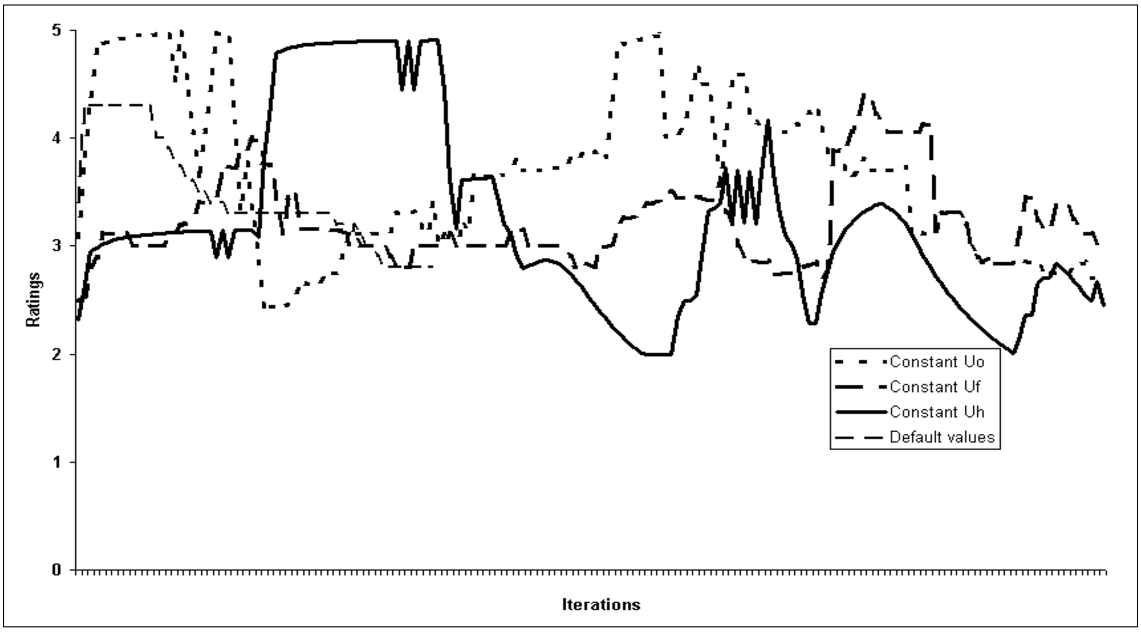


Figure 6.12: Comparison of ratings in different values of the scaling factors for agents that exhibit good behaviour

The peer selection strategy for the collusive peers is as follows:

$$\begin{aligned} I_{Arr}[0] &= x_1 \\ I_{Arr}[1] &= x_2 \\ &\vdots \\ I_{Arr}[n-1] &= x_n \end{aligned} \tag{6.2}$$

C is the total number of collusive agents, n is the number of intermediate agents that are randomly selected by the model at runtime, where $n \in C$. Agents in positions $(0-(n-1))$ are passed through a random selection function that selects a pair-wise list of sequential collusive agents or n-wise arbitrary list of collusive agents in the array positions $x[0], x[1], ..., x[n-1]$. Agents in a pair-wise collusion have direct one-hop links from each other.

The results of the variance between current and predicted ratings in both forms of

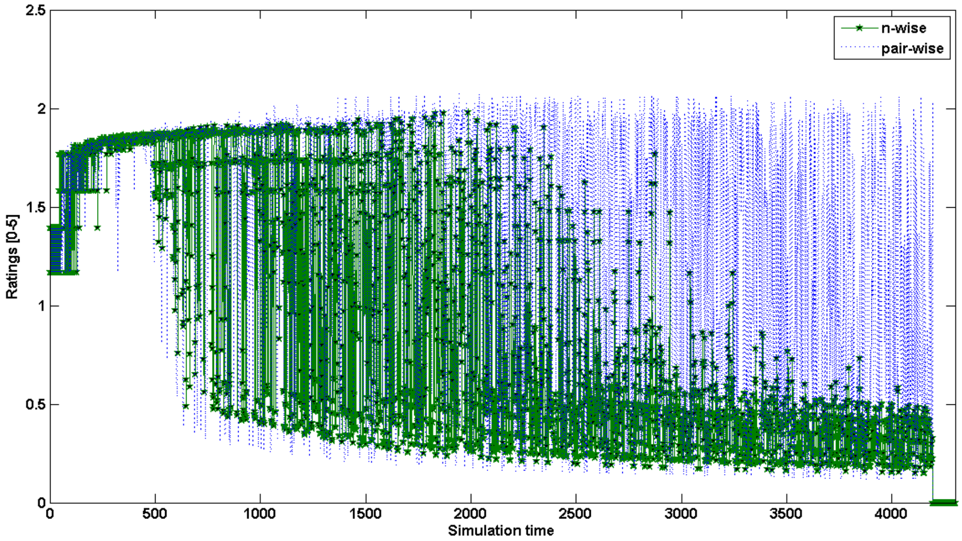


Figure 6.13: Current ratings

collusion scenarios are depicted in Figures 6.13 and 6.14 respectively, where ratings are ranked along the vertical axis and the simulation time progression is represented on the horizontal. With 40 peers collaborating in the network, we assess if the results are from normal distributions with the same variance, against the alternative that they come from normal distributions with different variances. Given the null hypothesis (H_o) in both cases, the same results are produced. That is $H_o : \sigma_1^2 = \sigma_2^2$ where σ_1^2 and σ_2^2 are the standard deviations of pair-wise and n-wise RVs. The graph in Figure 6.13 depicts the current ratings of peers in the presence of pair-wise and n-wise collusion attacks while Figure 6.14 shows the predicted ratings in both forms of the attack. To test the null hypothesis, an F test is used to analyse the variances in the ratings, where ($C \geq n \geq 2$) throughout the experiments. According to Cohen [Coh95], for a hypothesis test the sample size used in these experiments is no larger than that required to show a constant result (as depicted from simulation time 2500 in Figures 6.13 and 6.14); this aided the sample size selection for this experiment. The test on current RVs conducted resulted in a value of 0.7 and that of the predicted values is 0.8, where $H = 1$ in both instances.

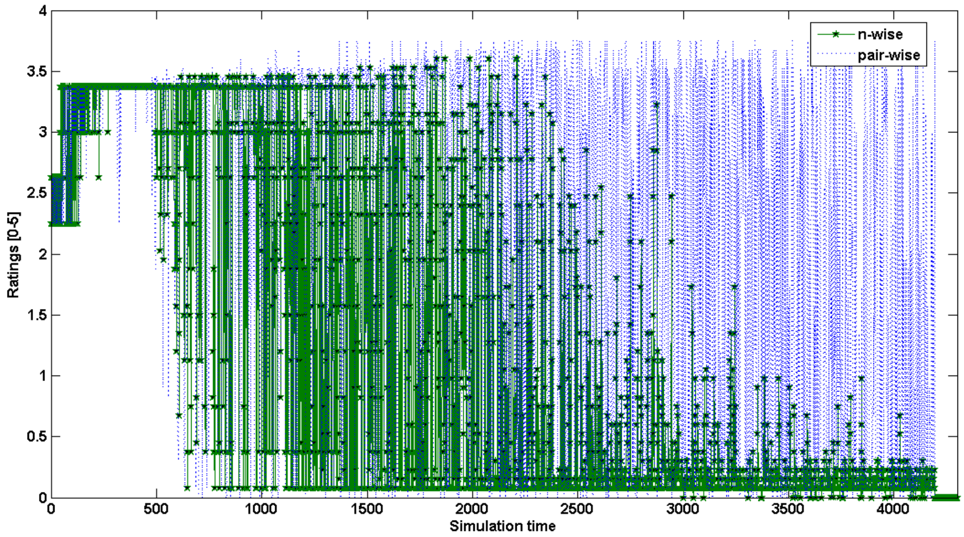


Figure 6.14: Predicted ratings

The null hypothesis that the standard deviations are equal is accepted at 95% significance level and the variances are assumed equal, implying that there is a positive significant relationship between the performance of D3-FRT in both forms of the attack. Note that this applies in both the computation of current and predicted values. We accept the null hypothesis that the two standard deviations are equal, and any difference is due to random error. Following these, it can be concluded that the performance of the framework is independent of the collusion attack agent selection, or the number of colluding agents.

6.2.4 Demonstrating the Dynamism of D3-FRT

In chapter 2, we described the dynamic nature of reputation and trust and how RTMs have to be dynamic as well in order to effectively fulfil their functions. In this sub-section we demonstrate D3-FRT's dynamic property. The simulation parameters for every 100-tick cycle of the experiment include: $N = 20$, with 30% and 10% of the peers exhibiting collusive and intoxicating behaviours respectively and throughout the simulation. Peers are penalised with a value of 1 for misbehaviour and 0.9 reward for good behaviour and all agents have an initial neutral RV of 2.5.

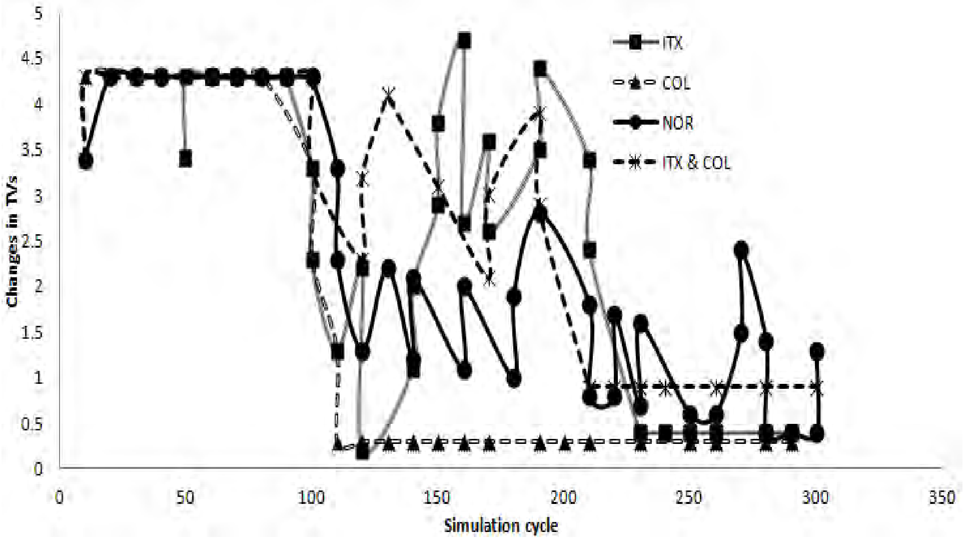


Figure 6.15: Changes in current RVs as simulation progresses

Selected representative agents' RVs from the results obtained are illustrated in Figure 6.15. The selection is based on the behaviours exhibited by the agents that are representative of the entire population. The graph depicts a range of varying changes to the current RVs of the agents that exhibited different behaviours throughout the simulation.

The first 100 ticks was an initialisation period, when the entire system was in the process of evolving. ITX in Figure 6.15 shows the changes to an agent that exhibited intoxication misbehaviour, COL depicts that of a collusive agent while the line NOR in the chart shows the changes to the RV of a normal agent.

To evaluate the impact of collaboration, we compare the rate of change in RVs with the number of interactions among peers. The graph in Figure 6.16 shows that in general, as the simulation progresses and peers collaborate with each other, corresponding RVs are computed per peer. This implies that the framework dynamically computes new ratings after each peer interaction.

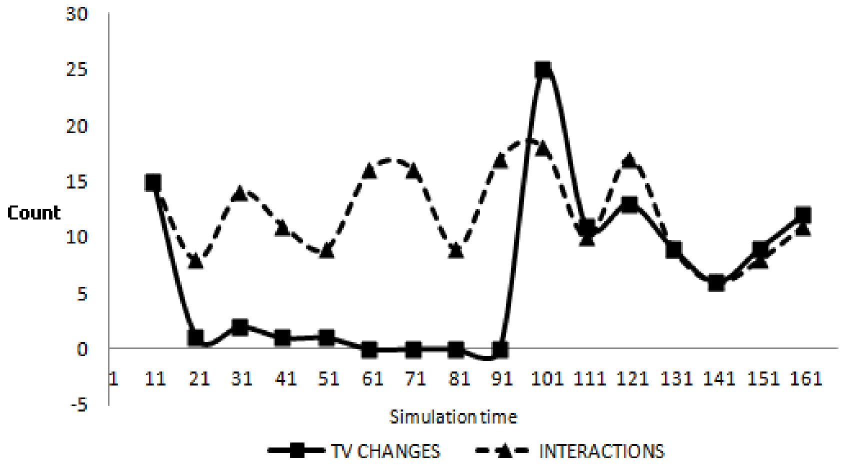


Figure 6.16: Changes in current RVs as simulation progresses

In order to test the viability of the framework, a linear regression between observed and predicted values is performed. Also, in consideration of allowing for more interactions among the agents, the penalties for intoxication and collusive behaviour are updated to 0.4 and 0.5 respectively. The reward for good behaviour is 0.5 and the minimum threshold value for RVs in the network is 0.3. For every 100-tick cycle of the simulation, $N = 50$ with 12% and 10% of the peers exhibiting collusive and intoxicating behaviours, for the duration of 1500 ticks.

Using regression analysis to show the dynamic property of D3-FRT, we compare the rate of interaction in the system with the corresponding changes to θ_os of the collaborating agents. The chart in Figure 6.17 shows this comparison and a power trend-line is used to forecast for another $\approx$ 300 ticks, to clearly show the progression in the system. As collaboration occurs, there should be corresponding change to current RVs to show the dynamic nature of D3-FRT.

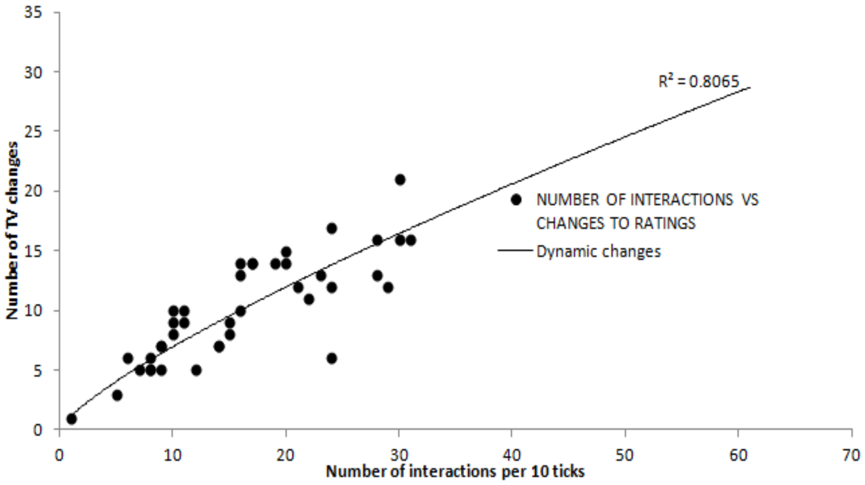


Figure 6.17: Rate of interaction versus rate of changes to RVs

In the chart, the pattern of changes to RVs as the simulation progresses is moving up and to the right; therefore there is a positive relationship. This strongly suggests that as collaboration increases, current RVs are dynamically computed and also increase. The R^2 value of the trend-line is ≈ 0.8, showing the reliability of the trend-line. Figure 6.17 clearly indicates that as collaboration continues in the system, so does changes to the RVs.

6.2.5 Performance in Varying Conditions

Performance is a major requirement for any reputation management system which necessitates the need to verify the performance of D3-FRT under different network sizes, in terms of the Time-To-Detect (TTD) of misbehaviours. Experimental variations with increasing total numbers of peers and misbehaving peers are carried out in order to test the framework performance with respect to the provision of timely information in different network sizes.

We evaluate the effect that network size (in terms of the total number of agents and population of misbehaving agents) has on the performance of D3-FRT. We noted that as

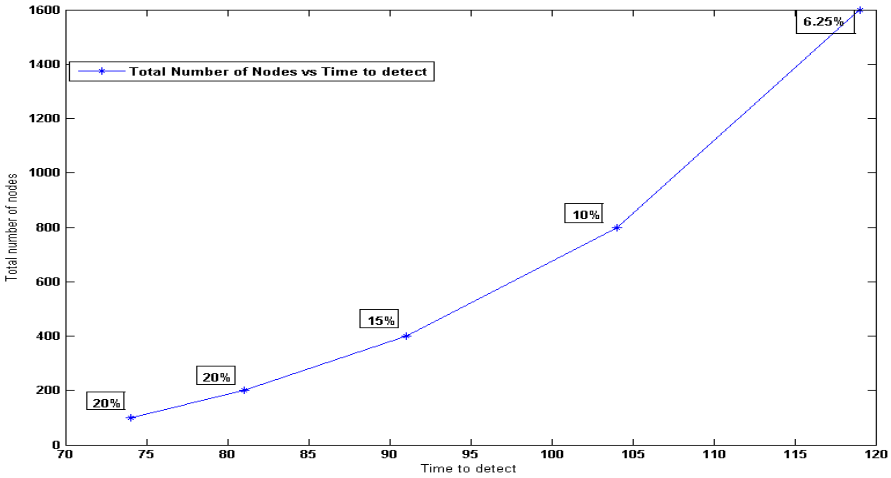


Figure 6.18: The number of misbehaving agents versus TTD

the network size increases with an increase in the number of misbehaving agents, so does the TTD the agents; this is illustrated in Figure 6.18. With a network size of 400 agents out of which 15% are misbehaving peers, the TTD is ≈ 90 ticks in a best-case scenario with 1600 agents, and having 6.25% proportion of attackers, D3-FRT's average TTD is ≈ 120 ticks. This result shows that the number of interacting agents has an impact on performance of the D3-FRT as there was a significant positive correlation between the network size and performance.

Furthermore, we considered the situation where the total number of misbehaving nodes are constant at 20% of varying network sizes. The average TTD across 10 runs of the experiment is depicted in Figure 6.19. We can see from the results that the TTD of all misbehaving nodes in each experiment increases with the network size.

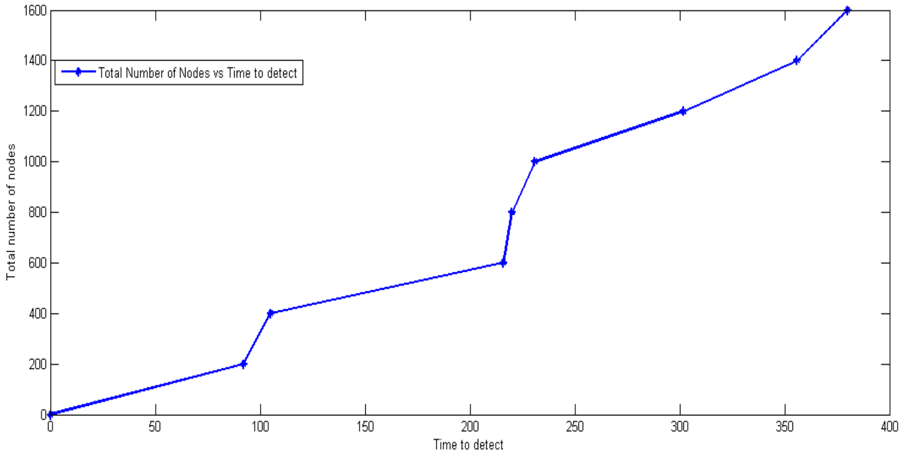


Figure 6.19: The number of misbehaving agents versus TTD with 20% misbehaving in the population

6.2.6 Verifying the Requirement of the Simulation and Controller Components

This experiment is to evaluate the usefulness of the DDDAS components of D3-FRT. We test D3-FRT in two scenarios and these are: (a) the case of where reputation computation is based only on recent data and past interactions with no predictions from the simulation and (b) where future behaviours are anticipated. These scenarios are depicted in Figure 6.20. The simulation parameters used in these experiments are in Table 6.2 and simulation component which runs concurrently with the network contains a snapshot of the network and is 20 *ticks* ahead.

The RV derived from peers recent activities θ_o^R is updated every 5 ticks. The θ_o^R from the last update replaces the value of θ_h^R every 5 ticks. The set of past θ_h^Rs is stored in a database for records of historical RVs. With every observation k in the experiment, we compute θ_o^R with the equation $(\theta_o^R)k^{th} = ((\theta_o^R)k - 1^{th}) - \alpha$ and $(\theta_o^R)k^{th} = ((\theta_o^R)k - 1^{th}) + (\alpha + 0.5)$ for observed bad and good behaviour respectively, where α is set to 0.5. For these experiments, the weights were kept at constant values of 0.5, 0.3 and 0.17 for μ_o, μ_h and μ_f respectively.

Table 6.2: Simulation parameters

Parameter	**Value**
Total simulation time (in ticks)	100
Total number of agents	100
Percentage of malicious agents	4
Default reputation values θ_o^R, θ_h^R	2.5
Current weight μ_o	0.5
Historical weight μ_h	0.3
Prediction weight μ_f	0.17

The simulation component in the second experiment considered three possible what-if scenarios (collusion, intoxication and failure to cooperate in forwarding of files) and the corresponding RVs for each scenario were obtained. A scenario where the peer is active and behaves as expected is also considered. The average θ_f^S from the scenarios was used and combined with θ_o^R and θ_h^R (of each peer) to compute the overall RV in the second experiment.

In the absence of prediction, the misbehaving agents colluded and sent inauthentic files through the network at 60 ticks. With prediction, the framework detected and flagged the peer as malicious at 40 ticks and with a downgrade of its RV immediately. Figure 6.20 shows the scenario of RVs of one of the misbehaving peers, with and without the use of prediction. The figure shows the time gained with the use of prediction with a downgrade of the peer's RV immediately.

Ultimately in the experiment with prediction, the peer is isolated because its overall RV is below the threshold for other peers to want to cooperate with the peer. This averts the misbehaviour, unlike in the experiment without the prediction (similar to the models that do not anticipate future behaviour by simulation), where the RV was downgraded as a response to the attack. Figure 6.21 compares the predicted trust with actual RV for some peers. The graph shows the changes in the value of a peer exhibiting intoxication,

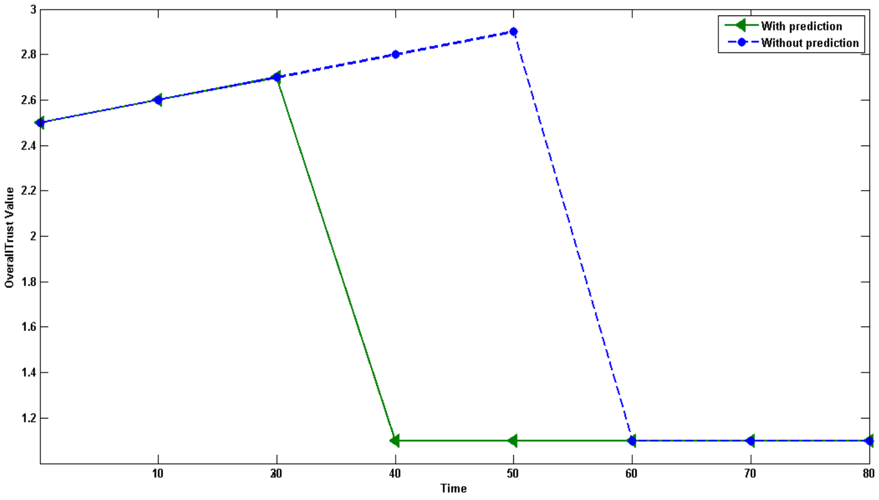


Figure 6.20: P2P file-sharing network result (with and without prediction of RVs)

an untrusted peer whose RV continues to drop and a trusted peer that is active with a high value.

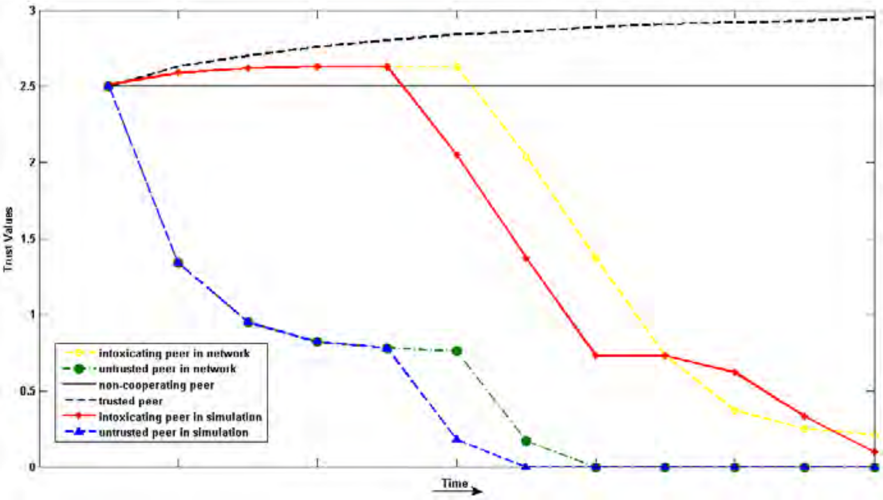


Figure 6.21: RVs of a peer and the comparison of the values in the network and simulation

Although, the simulation time for these experiments was for a 100 ticks each, it is worth noting that the volume of data generated by the framework per tick is large enough to justify the need for a data-driven approach. A snapshot of the sample data set generated that shows detailed output obtained within a time frame is given in Appendix B of this thesis.

6.2.7 An Experimental Comparison of TrustGuard and D3-FRT

A related protocol to D3-FRT is TrustGuard which has been discussed in detail in Chapter 3. Similar to D3-FRT, TrustGuard introduces flexibility into trust management by giving different trust components varying weights and considers sudden changes in behaviour. However, the models differ significantly in their approach in computing trust and in making predictions.

The purpose of experiments in this section is to test the predictive accuracy of both D3-FRT and the TrustGuard models. Both models are implemented using agent-based simulation approach. Since TrustGuard assumes being built upon an overlay secure network, the experiments are performed using the Repast C# platform simulating agents as peers in a P2P network. For simplicity, we assume that the RVs of agents are updated periodically within each time period T. Let successive time periods be numbered with consecutive integers starting from zero. We call TV_i the dependable reputation value of agent n in the interval i. TV_i can be viewed as a function of three parameters:

1. The feedback reports received at interval i,
2. The integral over the set of the past reputation values of agent n, and
3. The current derivative of the reputation value of agent n.

The simplified equation used for computing ratings in the TrustGuard model is given in equation 6.3.

$$RV = \alpha\theta_o + \beta\theta_h + (\gamma(\theta_o - \theta_h) * (\theta_o - \theta_h)) \tag{6.3}$$

According to the TrustGuard's algorithm, $\alpha = 0.1$, $\beta = 0.9$ and to determine the value of γ,

$$\begin{aligned} \gamma_1 = 0.05 \quad if \quad \theta_o - \theta_h \geq 0, \quad then \quad \gamma = \gamma_1 \\ \gamma_2 = 0.2 \quad if \quad \theta_o - \theta_h < 0, \quad then \quad \gamma = \gamma_2 \end{aligned} \tag{6.4}$$

The value of γ changes as the differences between current and historical ratings are computed. Assigning a higher value to the scaling factor (β) of historical ratings results in TrustGuard placing more emphasis on past events, which defeats the purpose of reducing the possibility of intoxication. Also, in order to compute the final ratings of collaborating agents using the TrustGuard algorithm, more recent observations were considered compared to historical ones.

Using regression analysis, we describe the relationship between predicted and actual values. In each case the total population of peers $N = 500$, having 10% exhibiting intoxication and 15% collusive agents in 1000 ticks. Also the correlation coefficient R is also computed to show if the predictions fit the domain events. Figures 6.22 and 6.23 show the regression analysis while comparing the predicted versus the actual ratings and estimation error rate for each computed RV in D3-FRT while Figures 6.24 and 6.25 show that of TrustGuard protocol.

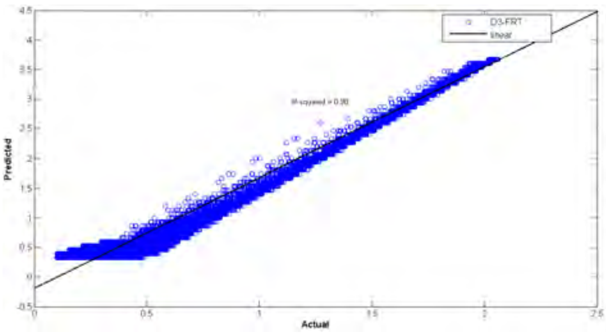


Figure 6.22: Predicted versus actual

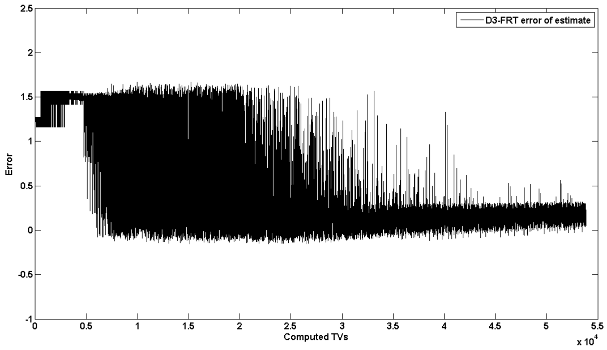


Figure 6.23: D3-FRT's error of estimate

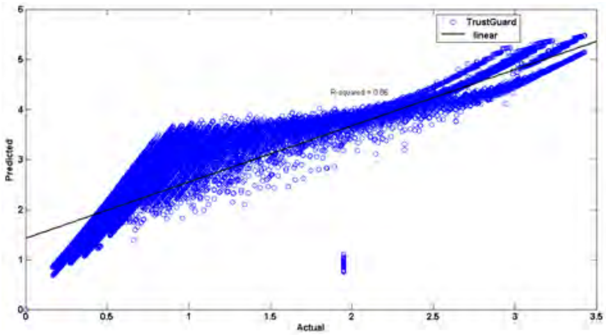


Figure 6.24: Predicted versus actual

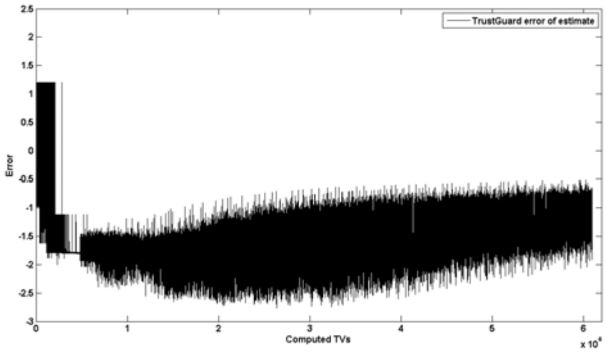


Figure 6.25: TrustGuard's error of estimate

The correlation coefficient R is 0.99 for D3-FRT, showing that the correlation is not perfect but is strong between the actual and predicted RVs. Also, $R^2 = 0.98$ implies that 98% of the total variation in the predictions of D3-FRT can be explained by the linear relationship between actual and predicted values. The observed high value of the coefficient of determination could be attributed to a closer relationship between actual and predicted ratings. The other 2% of the total variation in the predictions remains unexplained. In the experiments carried out using the TrustGuard algorithm however, R = 0.93 and $R^2 = 0.86$, this indicates that 86% of the total variation of TrustGuard's prediction can be explained by the linear relationship between actual and predicted values. We note that while TrustGuard error remained at a constant rate, D3-FRT adapted and the error rate reduced with time.

To show the degree of certainty of the results generated above, standard error bars were used to plot the variance in ratings as the simulation progressed for both models. Selected results from these experiments are illustrated further in Figure 6.26, in which simulation progression of both models is varied along the horizontal axis of the graph, while the variance between actual and predicted ratings are along the vertical axis.

Finding an average of errors is not meaningful here and since the variation there is in

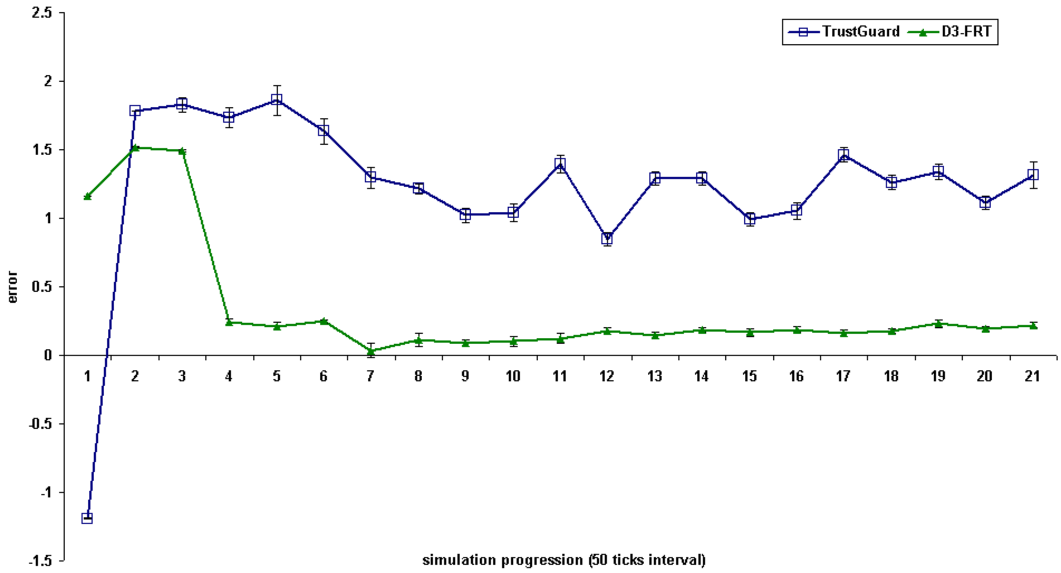


Figure 6.26: Mean error values with corresponding error bars

the measurement is unknown, error bars are used. The tight range above and beneath each means shows the degree of certainty in the data values. It shows that the risk of an incorrect conclusion in Figures 6.23 and 6.25 is minimal. From the confidence interval, it can be seen that the variance in predictions of TrustGuard is constantly higher than that of D-FRT.

The quick computation of current RVs is necessary for the effectiveness of any trust management system. To show the performances in terms of the timeliness in misbehaviour detection, we conducted experiments in the presence of colluding (20%), intoxicating (10%) peers and kept the total population size at 20 in the network. Figures 6.27 and 6.28 show the performances of TrustGuard and D3-FRT in the presence of collusion attack only, while Figures 6.29 and 6.30 illustrate their performances with intoxication attacks only. The vertical axis in Figures 6.27 - 6.30 show the observed TTD of the misbehaviour by both models of the peers that are varied along the vertical axis. In each case, the TTD is the time when the agent's RV drops below 1.5, i.e. the time taken to detect the agent as malicious and then considered to be of high-risk in the network while AT stands for the simulation time when the attack actually occurred.

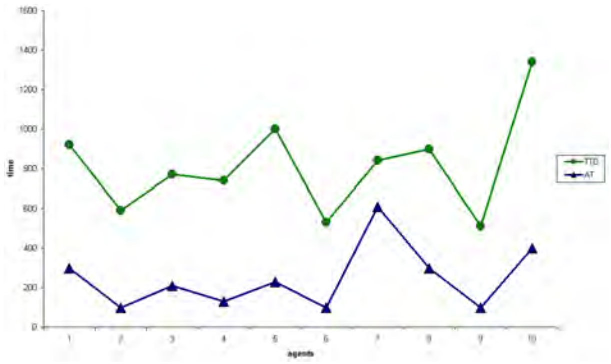


Figure 6.27: TTD versus the time of misbehaviour in the presence of collusion attack only in TrustGuard

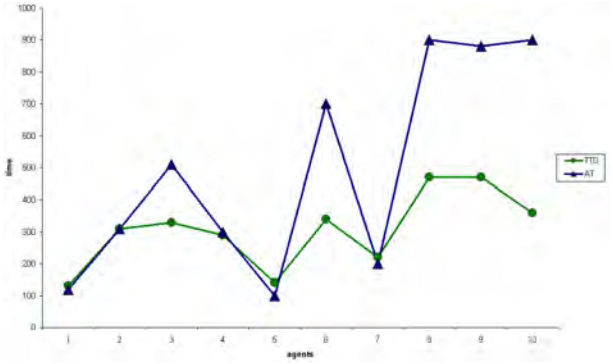


Figure 6.28: TTD versus the time of misbehaviour in the presence of collusion attack only in D3-FRT

As can be seen from the figures, TrustGuard's results in presence of intoxication only and collusion attack only are not significantly different. Similar results apply to D3-FRT, although in Figure 6.28, D3-FRT has performed better than in the case of intoxication only. Generally D3-FRT consistently provide timely reputation computations compared to TrustGuard but both models performed better in presence of collusion only compared to that of intoxication attack (Figures 6.29 and 6.30) only. D3-FRT suffered less from this attack by having lower TTD values overall as can be seen from the time variation between the vertical axis of Figures 6.29 and 6.30.

From a series of experiments, we compare the models using a more symmetric measure with their respective average Mean Variation from Estimates (MVREs) [HCYM98]:

$$MVRE(RV_{actual}, RV_{predicted}) = 1/N \sum_{i=1}^{N} (RV_{actual}(i) - RV_{predicted}(i))/RV_{predicted}(i) \quad (6.5)$$

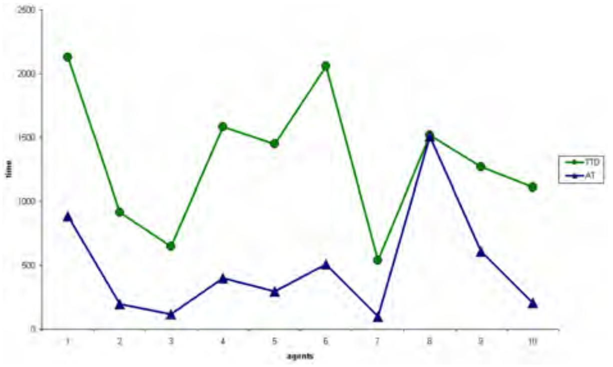


Figure 6.29: TTD versus the time of misbehaviour in the presence of intoxication attack only in TrustGuard

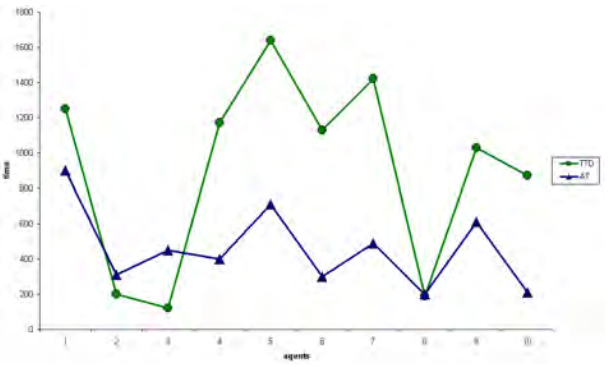


Figure 6.30: TTD versus the time of misbehaviour in the presence of intoxication attack only in D3-FRT

The goal is to initially compare the results by providing a quantitative score that describes the degree of similarities between the actual and predicted values in both models. In comparing the models, the MVRE of the overall and predicted ratings is computed from the average of five test trials as shown in Table 6.3. Every trial involves using the same simulation parameters as in the previous experiments for a period of 2000 ticks.

Table 6.3: Comparison of D3-FRT with TrustGuard using the mean variation estimate approach

Trial Number	**TrustGuard**	**D3-FRT**
1	0.65	0.47
2	0.68	0.4
3	0.68	0.5
4	0.68	0.5
5	0.67	0.48
Average MVRE	0.67	0.49

The result in Table 6.3 shows that based on our implementation of the models, D3-FRT has a MVRE of 0.49 compared to that of TrustGuard which has a higher error rate, giving a significant difference of 0.19 between both models. This may have significant effect on the domain's performance. As can be seen, in the presence of collusion and intoxication attacks, the mean variation of estimate of D3-FRT is consistently less than that of TrustGuard.

The results of this investigation show that D3-FRT outperforms TrustGuard in the presence of known attacks, in varying network conditions and in making predictions about domain events.

6.3 Qualitative Evaluation

Table 6.4 gives a qualitative and comparative analysis that is extended to include D3-FRT from the work of Govindan and Mohapatra [GM12b], on relevant models that focus on prediction. We analyse the models' methodology for making predictions and the information that influences the trust decisions. The analysis focuses on the prediction approach of each RTM, where the context in use is the underlying principle of the functioning of the models. The trust metric in the comparison is the method with which the observations are captured, represented, and the reward system in the RTM. The models are also evaluated on the basis of their advantages and limitations over others.

The prediction approach in [JT99] is based on a formal analysis of trust dynamics and evolution from sequences of experiences among domain members. In this approach, a trust update mathematical function is used to relate current experiences and ratings to the next trust rating. Capra and Musolesi in [CM06] proposed the use of a basic Kalman filter based on a set of direct observations, a prediction model is derived and used to makes predictions. Here, new observations are fed in by means of a set of recursive mathematical equations that can be efficiently computed in order to increase the accuracy of the prediction. The level of confidence in the predictions is also dependent on the number and frequency of interactions that occur. In the model, predictions about the discrepancy between a claimed trustee's own ratings and what the trustor's experience will be; where the higher the discrepancy, the lower the RV. Similarly, in RLM [WLS12], the Kalman filter theory is adopted for feedback aggregation, but through theoretical analysis, the robustness of the model's design is tested against collusion and false accusation attacks.

Table 6.4: Comparison of D3-FRT with other predictive approaches

Approach	**Context**	**Metric**	**Prediction**	**Limitations**
C. Jonker *et al* [JT99]	Formal framework that relies on past histories	Trust is represented by fuzzy type of descriptions	Accuracy can be a achieved with large data-sets	Performance is dependent on sample size
Capra and Musolesi [CM06]	Kalman filter theory used for predictions through recursive mathematical expressions	Trust value range: [0...1]	Accuracy can be achieved with the well established Kalman filter approach	Can be implemented at the expense of additional complexity
RLM [WLS12]	Linear Markov model for trust representation and aggregation	Continuous variable bounded in an interval	Reputation prediction variance that serves as a quality measure of the reputation value computed from feedbacks aggregation	Relies on subjective and insufficient feedbacks
D3-FRT [OBT12]	Dynamic data-driven approach that captures fluctuations of trust and reputation	Fuzzy-like representation and continuous real values	More accuracy can be achieved by the adaptability of trust components	Cost of simulation

From the quantitative analysis done so far in this chapter, coupled with this comparative analysis, D3-FRT's approach has shown to be useful in making predictions about how the system can evolve at a future time. The approach considers trust dynamics in terms of fluctuations and sudden changes in behaviour in drawing conclusions about the domain.

6.4 Discussions and Summary

This thesis considered, for the first time a data-driven simulation approach to reputation management and to establish the validity and practicability of the proposed approach. This chapter gives experimental evidence which demonstrates the usefulness of D3-FRT.

Furthermore, the main research questions that this chapter answers are:

- How useful is the DDDAS paradigm in aiding trusted communication in reputation management systems? To what extent will the framework support dynamism? How dynamic is agent rating?
- How accurate is the predictive capability of D3-FRT? How reliable is the framework as the network size grows?

Ratings are computed after each peer interaction in the network, and the newly computed values are made available across the network for other peers that wish to collaborate with the peer under consideration. Throughout the experiments, RV is computed after every peer interaction in D3-FRT. This approach proactively assigns ratings to peers as they collaborate. Therefore the framework captures the dynamic nature of reputation and trust making it useful in critical and trust-reliant domain. In validating our model, we have shown through the experiments in this chapter that an agents' RV cannot be outside of the bound [0...5]. In addition, using one or a combination of the attack scenarios and simulation progressed, a good peer exhibiting consistently good behaviour cannot have a $RV < 4.0$, and a bad agent exhibiting consistently bad behaviour cannot have a $RV > 1.0$ in the model.

The estimation error rate of the D3-FRT is observed when actual ratings of domain members are compared with the predicted values. Generally, the pass criterion for predictions is largely dependent on the domain of application, the criticality and risk appetite in that domain. Though there were some variances between the actual and predicted values of the framework in the experiments, this is likely to be as a result of the initialisation period, when the simulation does not have any prior knowledge of the network. However as shown in this chapter, D3-FRT has the capability to converge to make predictions that are closer to reality. Also, varying the value of the trust components using scaling factors have shown to have a great impact on the outcome of the predictions. Hence, making D3-FRT adaptable will allow for better results. The factors should therefore be chosen with care with respect to the intentions of the modeller and the application domain. It is worth noting here that the observations may vary with respect to the modelling approach, the time window of the observations and any other factors that may influence the behaviour of the model.

Scalability in the experiments refers to the extent to which the network can grow before there is degradation in performance. The scalability and reliability of D3-FRT in different network scenarios is examined in this chapter. The framework is tested by gradual increase of domain members and the results indicate that it is able to support increase in the number of network agents both normal and misbehaving agents in different network scenarios. Although the framework eventually detected all misbehaving agents in the different scenarios, it was observed that as the network increased in size the TTD also increased.

Adopting the DDDAS approach to trust management has shown to be useful with the advantage of providing trust and dynamism in the domain. The network is simulated with and without the presence of the predictive DDDAS component. By excluding this component we were able to isolate the individual effect of prediction. The usefulness of anticipating domain events, and making predictions was tested. In its absence, peers successfully misbehaved (which is unlike the case with the prediction component), preventive measures were taken. The peers RVs were downgraded before successfully carrying out an attack. This therefore makes D3-FRT proactive, reducing the overall negative impact of misbehaviour in the network.

D3-FRT has shown to be useful as a result of anticipating domain events leading to making predictions that are provided earlier than the actual network. This allows for preventive measures to be taken against agents that have been identified to have a high potential to misbehave. The agents are also grouped to regions depending on their RVs and agents that have ratings below the threshold are considered as high-risk and are excluded from the network. It is worth noting that although the predicted values are not 100% accurate, they differentiate between good agents from misbehaving ones by their relative RV ranking.

The experimental evidence in this chapter shows not only that reputation information encourages good behaviour and helps in the exclusion of misbehaving agents under dynamic scenarios, but a reputation approach that allows domain members to collaborate with trusted agents.

It can therefore be concluded that the DDDAS paradigm and its simulation primitive coupled with agent-based modelling can consequently reduce misbehaviour in a trust-reliant domain and the reliability of reputation management systems increases.

Part IV

We shall not cease from exploration

And the end of all our exploring

Will be to arrive where we started

And know the place for the first time.

T.S. Eliot

Chapter 7

Conclusion and Future Work

7.1 Introduction

The main goal of the work presented in this thesis is a dynamic and predictive approach to reputation management among a network of participants who have no prior knowledge of each other. From our aims and objectives earlier described in this thesis, the issue of trust dynamics has been studied and a novel approach for managing trust has been proposed. In addition to our study, the prediction of futuristic events has shown to be useful in providing timely information about domain events. Through qualitative evaluation, the use of a semi-distributed architecture has been justified.

More specifically, we identified the problems that arise as a result of trust dynamics and proposed a DDDAS-inspired framework using an agent based approach to solving the problems. The framework does not rely on only domain members for making trust decisions, but makes predictions by anticipating possible future states of the system. We demonstrated the effectiveness of our approach through qualitative and quantitative, simulation-based experiments. The framework introduces flexibility into reputation management by allowing scaling factors for different inputs, and trust components that determine the ratings of domain members. The success of this approach was illustrated in Chapter 6 against a range of reputation management objectives: dynamism, performance and predictive capability.

In D3-FRT, the controller is a trusted party. This is a sensible assumption when considering critical domains, where a form of central control is required. This includes domains such as that in the traffic management scenario of the example in Chapter 1 or in military networks where motes (tiny sensors) are deployed in a network to monitor enemy intrusion. Unlike other models, this framework does not only use historical data but also current and anticipated future events for prediction. D3-FRT logically groups the collaborating agents into regions of trust based on their reputation and ratings in the system. Our approach allows for placing more attention on groups of agents that pose a higher risk in the domain.

The D3-FRT approach satisfies some of the properties desirable in reputation and trust based system (as discussed in Section 2.3). Specifically, these properties are listed below:

- The framework provides predictions and ratings that help to distinguish members relative to their behaviours in the system.

- The framework is robust to known forms of misbehaviour and attacks which include intoxication and collusion, by both independent and collective misbehaving domain members.

- Behavioural changes are captured and are reflected in the timely current reputation computations of collaborating domain members. With the help of the DDDAS simulation component ratings converge to reflect the true changes in behaviours.

Also, the feedback loop allows for updates to be made to the simulation in order to reflect reality more effectively.

- D3-FRT is potentially adaptive because of the allowance of scaling for different components of trust, making the framework flexible to changes in domain activities.

7.2 Contributions

This thesis resulted in D3-FRT (Part III) that exploits the dynamic data-driven simulation (Part II) paradigm for trust management and predictions. The main contributions of this thesis are summarised here.

In this thesis, we extracted the requirements for addressing the issues of dynamics of trust. From these requirements, a novel framework for reputation management is presented. The framework is inspired by the DDDAS paradigm and adopts simulation, feedback and control primitives. The framework is a reliable semi-distributed approach that allows domain members to collaborate, and provides control. This prevents the spread of biased information and known attacks on the system. Furthermore, a predictive methodology using agent-based simulation approach is adopted. The approach anticipates possible future states of the network for effective trust management. In order to make predictions, the approach then uses historical, current and the anticipated future states with more emphasis placed on more recent events than dated ones. Although the predictions are not 100% accurate, D3-FRT is able to differentiate good members from bad ones in most cases by their relative ranking.

We take into consideration the dynamic nature of reputation and trust, this work provides the computation of current ratings of members as they collaborate with each other. Therefore, acting as an online decision support system that provides timely information for stakeholders to make informed decisions about the domain. As a result of the online computation and anticipation of behaviours, the framework makes reputation and trust management proactive in the domain it is applied. That is, it has the capability to react quickly to changes in the behaviour of domain members.

7.3 Limitations and Future Work

This section discusses the limitations and possible future directions arising from this thesis. The future directions identified for this work are in specific areas. D3-FRT is purposefully generic, with the objective of being generally applicable to different domains that possess dynamic characteristics and for critical environments that require recent and reliable information about participants. As a result, specific technical requirements and capabilities are not the main considerations of this thesis.

7.3.1 Improvements to the Current Framework

To accommodate adaptability, the scaling factors used in the computation of ratings can be automatically adjusted as participants collaborate. Even though constant values have been used, the simulation has a potential to be adaptive in a way that the feedback gathered from the system can help in the adjustments of the factors for future rounds. The adaptability of D3-FRT in terms of updating the scaling factors online will allow the simulation to reflect the physical system more closely.

We considered the file sharing context of a P2P network in this thesis. However, as trust can be described as being context dependent (detailed in chapter 2), it will be useful in the future to investigate the effect of multiple contexts on D3-FRT. In this case, agents will have different RVs for different events that occur in the system depending on the domain of application and expected action of members in the domain. A typical example of this is in a MANET where a node can have a RV for forwarding packets and may have another for its behaviour in route discovery. The results of this investigation will useful for preventing another form of intoxication attack.

In addition, it will be interesting to explore ways of improving the correlation of ratings (i.e. simulated and actual) in the future. Techniques to parameterise simulation rules for more dynamism in the framework will also be considered. Furthermore, the validation of computational models and simulation results is a critical issue in agent-based simulation [TKK08]. Therefore the presented results may be sensitive to how the network and the agents have been modelled.

In this thesis, we focused on predicting misbehaving members as misbehaving because of the criticality of applicable domains. Redemption methodologies to cater for false-positives in the framework will be interesting to researched on. Also, we considered only known attacks: intoxication and collusion attacks (false praise and false accusations). Some unknown attacks might go undetected and may constitute new threats. Future studies should assess how the reputation system copes with previously unknown pattern of misbehaviour, and unknown attacks in the system. The unknown attacks can be configured by the system administrator using parameterised rules or making the reputation system to 'self-learn' and capture such undocumented attacks. Knowing how D3-FRT adapts to, identifies and acts on unknown misbehaviour will be useful to better understand the usefulness of the DDDAS primitives for trust management.

7.3.2 Identity Persistence

The importance of IP cannot be overlooked for reputation management as it impacts the effectiveness of the reputation system. This is because domain entities have to be uniquely identifiable in order to avoid the issue of non-repudiation and impersonation. In the case studied in this thesis, peers have unique identifications which might not be true in other domains such as in a WSN.

To solve the issue of IP, some previous researches have adopted the distributed PKI for identity persistence which requires a high computational power and other resources. A problem with this technique is how to make public keys available to others in such a way that its authenticity is verifiable [CBH00]. Although the distributed PKI approach has been ineffective because of the inherent properties (limited computation, and battery and storage capabilities) of nodes in mobile and ad-hoc networks, such as the use of private (*symmetric cryptography*) and public key (*asymmetric cryptography*) infrastructure, it is not totally impossible. Symmetric cryptography offers low computational overhead but lacks scalability, however, asymmetric PKI has a better authentication technique but with more overhead in terms of being computationally intensive. Therefore, it would be interesting to assess the use of a less intensive approach for identity persistence, especially for domains where members join and leave the system dynamically.

7.3.3 Cost

In our experiments we show that the performance of D3-FRT with up to 1600 agents in a P2P network situation. However, cost of simulation and prediction has not been the focus of this research; it will however be useful to study the cost, and scalability of the framework for larger systems. Future research can therefore concentrate on the investigation of distributed simulation for trust management.

The role of back-up servers in providing *fault tolerance* will be assessed. There is a need to verify the usefulness of introducing extra servers that are synchronised with the main servers. The controller for example, can be simulated to have a downtime of a few seconds and the ability of the back-up controller to support the network will be assessed. The expected outcome is for the backup controller to take over the responsibility with no obvious degradation in network performance. The controllers are then expected to synchronise once the original controller is back online.

In critical domains, running a simulation system concurrently with the real system is essential. However, in resource constrained environments where having a simulation is not feasible; there is a need to slice down. The decision of how resources should be managed will depend largely on the criticality, nature of the application domain and the risk appetite of stakeholders coupled with the trade-off between cost and performance. Future work can be based on the risk of having no simulation component.

The feedback between the system and the simulation adds an overhead to the use of D3-FRT. Although the simulation component provides information to make the framework proactive, in comparison with other models and allows for preventive measures, it can be costly. The cost of accepting this overhead relative to the usefulness of the framework needs to be considered. Also, approaches to reducing the overhead is another area of research for further work.

We do not claim that the approach presented in this thesis is the best approach but it is a suitable one for managing reputation. The methodology given in Chapter 5 could be extended to accommodate or reduce the communication overhead in terms of message exchanges across the domain in order to reduce the possibility of congestion especially in the network domain. Potentially, a future direction for this is to design an approach of reducing the volume of data exchanged but maintains the precision and efficiency required in a trust management system.

As reputation and trust are dynamic and vary over time, having fast convergence speed is an area that should be explored further. This is essential because reputation aggregation should converge fast enough to reflect the true changes of members' behaviour [ZH07].

The trade-off of running different simulations that are working on different rules may be considered. This can potentially lead to learning about the domain that can be beneficial in the next cycle and may involve the use of artificial neutral networks to learn about how the system evolves. Furthermore, it would be useful to know the outcome of reducing the computation complexity introduced with having a simulation component.

7.3.4 Applications and Extensions

Globalisation and technological change have created a strongly coupled and interdependent world. The place of reputation and trust management as we move into the era of *Big Data*, is yet to be defined. Considering that the volume of data that could be generated from sensor networks, social and military networks etc in the next decade can be significantly greater than that generated in the last 100 years, there is a need for a trusted Web, where models can simulate, predict and turn information into useful knowledge [Con12]. The approach described in this thesis, which fits perfectly with the requirements of this evolution, is timely and would have a place in the advancement.

A P2P network case study was used to test our approach in this thesis. It will be useful to reflect on how the proposed approach can be useful in other domains. D3-FRT is likely to be sensible in other domains apart from mobile or sensor network situations. Further investigation and experimentation into possible approaches that can extend this work is recommended. Bayesian probability is used in this research for developing rules of collaboration among participants as it is an effective approach for arbitrary behaviours [GBS08]. The approaches that can be considered include online learning, Data Mining, Game Theory and fuzzy techniques. Although Game Theory has been used for trust management, the main goal was just to obtain a *nash equilibrium* [Buc07] and where cooperation is the optimal strategy for each member. A further study of reputation and trust using fuzzy computational approach is required where both factors are not considered as crisp attributes of members as this does not reflect their meanings [BAS09].

Group trust [Cap05] is an area of trust management that can be useful for coordinating large number of devices in a sensor network for example. The idea can aid the reasoning and making of trust decisions about groups of devices. In the near future, we may leverage on the group trust approach as an extension of D3-FRT's grouping into regions of trust (as described in this thesis). Cloud computing can also benefit from the framework in scenarios where auto agents that are used in Cloud computing to monitor Service Level Agreements violations (such as latency in responding to requests) by users to ensure that only reputable providers are sought. The Cloud is a dynamic domain where new services are introduced and removed randomly. It is an open environment where the number of users joining or leaving the domain cannot be predicted. Vehicular Ad-hoc Networks (VANETs) can also benefit from the DDDAS approach presented here. This can serve as future approach in VANETs because such networks are critical and dynamic, making trust management a necessary requirement. It would be beneficial to know how these extensions can introduce more intelligence to the framework's current approach to trust management.

7.4 Conclusion

We have presented a novel framework, D3-FRT for dynamic reputation management with countermeasures against two vulnerabilities, namely, collusion and intoxication attacks. D3-FRT adopts an agent-based simulation and a dynamic data-driven approach with an adaptive potential. The framework enables the anticipation and therefore, the prediction of domain events and ratings.

While other related approaches to that presented in this thesis do exist, specifically those which use trusted third parties, the proposed method does not rely on biased domain members for reputation and trust decisions. It is worth noting that the framework is most suitable for critical domains that rely on feedback and recommendations with some control.

The effectiveness of the framework has been demonstrated through simulation based experiments. Following the study carried out in this research, it can be concluded that the D3-FRT approach has the potential to efficiently and effectively guard trust-reliant systems, making it a dependable decision support system.

As we move into Smart Cities, the Cloud and the era of Big Data, there is a need for more sophisticated frameworks for monitoring complex interactions among a group of anonymous participants in these evolving domains. This applicable piece of work demonstrates the feasibility of our idea with the potential to be improved and extended. There is therefore a future for D3-FRT because it brings solutions to the problem of trust dynamics in such domains.

Bibliography

[AD05] W. Adams and N. Davis. Toward a Decentralized Trust-based Access Control System for Dynamic Collaboration. In *Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop*, pages 317 – 324, 2005.

[AEKHES08] M. Azer, S. El-Kassas, A. Hassan, and M. El-Soudani. A Survey on Trust and Reputation Schemes in Ad hoc Networks. In *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, pages 881 – 886. IEEE Computer Society, 2008.

[Ama12] Amazon. Amazon. Online, September 2012. `http://www.amazon.co.uk/gp/help/customer/display.html?nodeId=13841791`.

[ARH00] A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities, 2000.

[ASSC02] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[BAS09] K. Bharadwaj and M. Al-Shamri. Fuzzy Computational Models for Trust and Reputation Systems. *Electronic Commerce Research and Applications*, 8(1):37 – 47, 2009.

[Bay63] T. Bayes. An Essay Towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London*, 53:370 418, 1763.

[BLB02] S. Buchegger and 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*, pages 226 – 236, 2002.

[BLB03] S. Buchegger and J. Le Boudec. The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks. In *WiOpt 03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[BLB04] S. Buchegger and J. Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*, 2004.

[BLB05] S. Buchegger and J. Le Boudec. Self-Policing Mobile Ad hoc Networks by Reputation Systems. *IEEE Communications Magazine*, 43(7):101 – 107, 2005.

[BMLB08] S. Buchegger, J. Mundinger, and J. Le Boudec. Reputation Systems for Self-Organized Networks. *Technology and Society Magazine, IEEE*, 27(1):41 – 47, 2008.

[Buc07] E. Buchmann. Trust Mechanisms and Reputation Systems. In *Algorithms* *for Sensor and Ad Hoc Networks*, volume 4621 of *Lecture Notes in Computer Science*, pages 325–336. Springer Berlin Heidelberg, 2007.

[BVLT07] V. Balakrishnan, V. Varadharajan, P. Lucs, and U. Tupakula. Trust Enhanced Secure Mobile Ad-hoc Network Routing. In *Advanced Information Networking and Applications Workshops, AINAW'07*, volume 1, pages 27 – 33, 2007.

[BZ04] O. Bamasak and N. Zhang. A Secure Proxy Signature Protocol for Agent-based M-commerce Applications. In *International Symposium on Computers and Communications (IEEE Cat. No.04TH8769)*, volume 1, pages 399 – 406, 2004.

[BZ05] O. Bamasak and 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, EEE-05*, pages 270 – 275, 2005.

[Cap05] L. Capra. Reasoning about trust groups to coordinate mobile ad-hoc systems. In *International Conference on Security and Privacy for Emerging Areas in Communication Networks - SecureComm*, pages 142 – 152, 2005.

[CBH00] S. Capkun, L. Buttyan, and J. Hubaux. Self-Organized Public-Key Management for Mobile Ad hoc Networks. *IEEE Transactions on Mobile Computing*, 2(1):52 – 64, 2000.

[CDS86] S. Conte, H. Dunsmore, and V. Shen. *Software engineering metrics and models.* Benjamin-Cummings Publishing Co., Inc., 1986.

[Cen12] European Commission Joint Reseach Centre. Sensitivity Analysis. Online, 2012.

[CLB10] J. Chen, H. Lu, and S. Bruda. A Reputation-based Approach for Countering Vulnerabilities in P2P Networks. In *2nd International Conference on E-Business and Information System Security, EBISS2010*, pages 263 – 266, 2010.

[CM06] L. Capra and M. Musolesi. Autonomic Trust Prediction for Pervasive Systems. In *Advanced Information Networking and Applications*, volume 2, 2006.

[CN06] R. Carruthers and I. Nikolaidis. Certain Limitations of Reputation-Based Schemes in Mobile Environments. In *ACM MSWiM 2005 - Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 2 – 11, 2006.

[Coh95] P. Cohen. *Empirical Methods for Artificial Intelligence.* MIT Press, 1995.

[Con12] FuturICT Consortium. Futurict. Online, June 2012. `http://www.futurict.eu/sites/default/files/docs/newsletters/FuturICT%20-%20What%20FuturICT%20Will%20Do.pdf`.

[Cro07] A. Crooks. The Repast Simulation/Modelling System for Geospatial Simulation. Centre for Advanced Spatial Analysis (University College London): Working Paper 123, 2007.

[CWG06] J. Chang, H. Wang, and Y. Gang. A Dynamic Trust Metric for P2P Systems.

In *Fifth International Conference on Grid and Cooperative Computing*, pages 117 – 120, 2006.

[CWHG08] H. Chen, H. Wu, J. Hu, and 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*, pages 359 – 364. IEEE Computer Society, 2008.

[Dar04] F. Darema. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In *International Conference on Computational Science*, pages 662 – 669, 2004.

[Dar10] F. Darema. InfoSymbiotics/DDDAS: The Power of Dynamic Data Driven Applications Systems. In *Multi-Agency Workshop*, 2010.

[DDD06] DDDAS. Dynamic Data Driven Applications Systems Workshop Report. Online, 2006.

[DI12] A. Das and M. Islam. Secured Trust: A Dynamic Trust Computation Model for Secured Communication in Multiagent Systems. *IEEE Transactions on Dependable and Secure Computing*, 9(2):261 – 274, 2012.

[DJLZ10] Q. Ding, M. Jiang, X. Li, and X. Zhou. Reputation Management in Vehicular Ad hoc Networks. In *International Conference on Wireless Communications and Signal Processing, WCSP*, 2010.

[DKB05] D. Djenouri, L. Khelladi, and A. Badache. A Survey of Security Issues in

Mobile Ad hoc and Sensor Networks. *Communications Surveys & Tutorials, IEEE*, 7(4):2 – 28, 2005.

[Dou08] C. Douglas. Dynamic Data Driven Applications Systems. In *International Conference on Computational Science ICCS (3)*, volume 5103 LNCS, pages 3 – 4, 2008.

[eBa12] eBay. eBay. Online, September 2012. `http://pages.ebay.co.uk/services/forum/feedback.html`.

[GB07] A. Gutowska and K. Bechkoum. A Distributed Agent-based Reputation Framework Enhancing Trust in e-commerce. In *11th IASTED International conference on Artificial Intelligence and Soft Computing*, pages 92 – 101, 2007.

[GBS08] S. Ganeriwal, L. Balzano, and M. Srivastava. Reputation-based Framework for High Integrity Sensor Networks. *ACM Transactions on Sensor Networks*, 4(3):15:1 – 37, 2008.

[GM12a] F. Gomez and G. Martinez. TRIP, A Trust and Reputation Infrastructure-based Proposal for Vehicular Ad hoc Networks. *Journal of Network and Computer Applications*, 35(3):934 – 941, 2012.

[GM12b] K. Govindan and P. Mohapatra. Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. *IEEE Communications Surveys and Tutorials*, 14(2):279 – 298, 2012.

[GS04] S. Ganeriwal and M. Srivastava. Trustworthy Sensor Networks: Issues, Challenges & Solutions. Technical report, University of California, 2004.

[HB06] J. Hu and M. Burmester. LARS - A locally aware reputation system for mobile ad hoc networks. In *Proceedings of the ACM SE Regional Conference*, volume 2006, pages 119 – 123, 2006.

[HCH08] F. Hussain, E. Chang, and O. Hussain. A robust methodology for prediction of trust and reputation values. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 97 – 108, 2008.

[HCYM98] R. Hughes, A. Cunliffe, and F. Young-Martos. Evaluating Software Development Effort Model-Building Techniques for Application in a Real-time Telecommunications Environment. *IEEE Proceedings-Software*, 145(1):29 – 33, 1998.

[HH07] E. Hussain, F.and Chang and O. Hussain. State of the Art Review of the Existing Bayesian-network Based Approaches to Trust and Reputation Computation. In *2nd International Conference on Internet Monitoring and Protection*, pages 154 – 158, 2007.

[HSV10] Y. Huang, M. Seck, and A. Verbraeck. Towards automated model calibration and validation in rail transit simulation. *Procedia Computer Science*, 1(1):1259 – 1265, 2010.

[HU93] C. Howson and P. Urbach. *Scientific Reasoning: The Bayesian Approach*, volume 48. Open Court Publishing Company, 1993.

[HV09] Y. Huang and A. Verbraeck. A Dynamic Data-Driven Approach for Rail Transport System Simulation. In *Winter Simulation Conference (WSC)*, pages 2553 – 2562, 2009.

[HWK04] Q. He, D. Wu, and P. Khosla. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. In *Proceedings of WCNC Wireless Communications and Networking Conference*, volume 2 of *IEEE Wireless Communications and Networking Conference*, pages 825 – 830, 2004.

[HZNR09] K. Hoffman, D. Zage, and C. Nita-Rotaru. A Survey of Attack and Defense Techniques for Reputation Systems. *ACM Computing Surveys*, 42(1):1 – 31, 2009.

[IFI11] IFIPTM2011. First ifip wg 11.11 summer school on trust management. Online, 2011.

[JIB07] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618 – 644, 2007.

[JT99] C. Jonker and J. Treur. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: MultiAgent System Engineering*, MAAMAW '99, pages 221 – 231. Springer-Verlag, 1999.

[Kol99] P. Kollock. The Production of Trust in Online Markets. In *Advances in Group Processes*, volume 16, pages 99 – 123, 1999.

[KSGM03] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640 – 651. ACM, 2003.

[KT06] C. Kennedy and G. Theodoropoulos. Intelligent management of data driven simulations to support model building in the social sciences. *International Conference on Computational Science-ICCS 2006. 6th. Proceedings, Part III (Lecture Notes in Computer Science)*, 3993:562 – 569, 2006.

[KTS$^+$07] C. Kennedy, G. Theodoropoulos, V. Sorge, E. Ferrari, P. Lee, and C. Skelcher. AIMSS: An architecture for data driven simulations in the social sciences. In *Lecture Notes in Computer Science*, volume 4487, pages 1098 – 1105, 2007.

[LC10] T. Liqin and L. Chuang. Computation and analysis of node intending trust in WSNs. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, volume 2, pages 496 – 499, 2010.

[LI04] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In *Trust Management. Second International Conference, iTrust 2004. Proceedings.(Lecture Notes in Comput. Sci. Vol.2995)*, pages 48 – 62, 2004.

[LJT04] Z. Liu, A. Joy, and R. Thompson. A Dynamic Trust Model for Mobile Ad hoc Networks. In *10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 80 – 85, 2004.

[LLYT05] K. Lin, H. Lu, T. Yu, and C. Tai. A reputation and trust management broker framework for web applications. In *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, EEE-05*, pages 262 – 269, 2005.

[LS10] L. Lui and W. Shi. Trust and Reputation Management. *Internet Computing, IEEE*, 14(5):10 –13, 2010.

[LYG$^+$07] Y. Liu, S. Yang, L. Guo, W. Chen, and L. Guo. A distributed trust-based reputation model in P2P system. In *SNPD 2007: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, volume 1, pages 294 – 299, 2007.

[LZY$^+$07] Q. Lian, Z. Zhang, M. Yang, B. Zhao, Y. Dai, and 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*, 0:56, 2007.

[Mar94] S. Marsh. *Formalising Trust as a Computational Concept.* PhD thesis, University of Stirling, 1994.

[MGLB00] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad hoc Networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pages 255 – 265, 2000.

[MGM06] S. Marti and H. Garcia-Molina. Taxonomy of Trust: Categorizing P2P Reputation Systems. *Computer Networks*, 50(4):472 – 484, 2006.

[MM02] P. Michiardi and 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*, volume 100, pages 107 – 121, 2002.

[MN06] C. Macal and M. North. Introduction to Agent-based Modeling and Simulation. Online, November 2006. MCS LANS Informal Seminar.

[MN09] C. Macal and M. North. Agent-based Modeling and Simulation. In *Proceedings of the 2009 Winter Simulation Conference (WSC 2009)*, pages 86 – 98, 2009.

[MN10] C. Macal and M. North. Tutorial on Agent-based Modelling and Simulation. *Journal of Simulation*, 4:151 – 162, 2010.

[Mom10] S. Momani, M. Challa. Survey of trust models in different network domains. *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, 1(3), 2010.

[MSB06] G. Madey, G. Szabo, and A. Barabasi. WIPER: the integrated wireless phone based emergency response system. In *Computational Science-ICCS 2006. 6th International Conference. Proceedings, Part III (Lecture Notes in Computer Science)*, volume 3993, pages 417 – 424, 2006.

[NM07] M. North and C. Macal. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation.* Oxford University Press, 2007.

[NWC09] P. Ni, L. Wan, and Y. Cai. Event Correlations in Sensor Networks. In *Lecture Notes in Computer Science*, volume 5545, pages 500 – 509, 2009.

[OBT09] O. Onolaja, R. Bahsoon, and G. Theodoropoulos. An Architecture for Dynamic Trust Monitoring in Mobile Networks. In *Lecture Notes in Computer Science (LNCS)*, volume 5872, pages 494 – 503, 2009.

[OBT10] O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Conceptual Framework for Dynamic Trust Monitoring and Prediction. *Procedia Computer Science*, 1(1):1235 – 1244, 2010.

[OBT11] O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Trust Dynamics: A Data-Driven Simulation Approach. In *IFIP International Federation for Information Processing*, pages 323 – 334, 2011.

[OBT12] O. Onolaja, R. Bahsoon, and G. Theodoropoulos. Agent-based Trust Management and Prediction using D3-FRT. *Procedia Computer Science*, 9(0):1119 – 1128, 2012.

[O'H06] C. O'Hanlon. A conversation with Werner Vogels. *Queue*, 4(4):14 – 22, 2006.

[OTB11] O. Onolaja, G. Theodoropoulos, and R. Bahsoon. A Data-Driven Framework for Dynamic Trust Management. *Procedia Computer Science*, 4:1751 – 1760, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.

[QHC06] D. Quercia, S. Hailes, and L. Capra. B-trust: Bayesian Trust Framework for

Pervasive Computing. In *Proceedings of the 4th International Conference on Trust Management, LNCS*, pages 298 – 312. Springer-Verlag, 2006.

[Que09] D. Quercia. *Trust Models for Mobile Content-Sharing Applications.* PhD thesis, University College London, 2009.

[RCK99] E. Royer and T. Chai-Keong. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2):46 –55, 1999.

[RCM10] R. Rodríguez, A. Cortés, and T. Margalef. Data Injection at Execution Time in Grid Environments Using Dynamic Data Driven Application System for Wildland Fire Spread Prediction. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, pages 565 – 568, 2010.

[Rep11] Repast. Recursive Porus Agent Simulation Toolkit. Online, December 2011. `http://repast.sourceforge.net`.

[RMK08] M. Rafsanjani, A. Moveghar, and 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*, volume 34, pages 2070 – 3740, 2008.

[RSM05] J. Riegelsberger, M. Sasse, and J. McCarthy. The mechanics of trust: a framework for research and design. *International Journal of Human-Computer Studies*, 62(3):381 – 422, 2005.

[RZFK00] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation Systems. *Communications of the ACM*, 43(12):45 – 48, 2000.

[SAB10] G. Swamynathan, K. Almeroth, and Z. Ben. The Design of a Reliable Reputation System. *Electronic Commerce Research*, 10(3-4):239 – 270, 2010.

[SCK02] M. Schlosser, T. Condie, and S. Kamvar. Simulating a File-Sharing P2P Network. Technical report, Department of Computer Science, Stanford University, 2002.

[SGG02] S. Saroiu, P. Gummadi, and S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Multimedia Computing and Networking*, 2002.

[SL06] M. Srivatsa and L. Ling. Securing Decentralized Reputation Management using TrustGuard. *Journal of Parallel and Distributed Computing*, 66(9):1217 – 1232, 2006.

[SRIT11] R. Saadi, M. Rahaman, V. Issarny, and A. Toninelli. Composing Trust Models Towards Interoperable Trust Management. In *IFIP Advances in Information and Communication Technology*, volume 358 AICT, pages 51 – 66, 2011.

[STW$^+$08] A. Srinivasan, J. Teitelbaum, J. Wu, M. Cardei, and H. Liang. *Reputation and Trust Based Systems for Ad Hoc Networks*. Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks. Wiley, algorithms and protocols for wireless, mobile ad hoc networks edition, 2008.

[SXL05] M. Srivatsa, L. Xiong, and L. Liu. TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks. In *Proceedings of the 14th international conference on World Wide Web*, pages 422 – 431. ACM, 2005.

[Tea06] W. Teasy. *Agent-Based Trust and Reputation in the Context of Inaccurate Information Sources.* PhD thesis, University of Southampton, 2006.

[TKAS06] A. Trivedi, R. Kapoor, A. Arora, and S. Sanyal. RISM - Reputation Based Intrusion Detection System for Mobile Adhoc Networks. In *3rd International Conference on Computers and Devices for Communication CODEC-06*, 2006.

[TKK08] K. Takadama, T. Kawai, and Y. Koyama. Micro- and Macro-Level Validation in Agent-Based Simulation: Reproduction of Human-Like Behaviors and Thinking in a Sequential Bargaining Game. *Journal of Artificial Societies and Social Simulation*, 11(2):9, 2008.

[TPJL05] W. Teacy, J. Patel, N. Jennings, and M. Luck. Coping with Inaccurate Reputation Sources: Experimental Analysis of a Probabilistic Trust Model. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 997 – 1004. ACM, 2005.

[WLS12] X. Wang, L. Liu, and J. Su. RLM: A General Model for Trust Representation and Aggregation. *IEEE Transactions on Services Computing*, 5:131 – 143, 2012.

[WV03a] Y. Wang and J. Vassileva. Bayesian Network-Based Trust Model. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, pages 372 – 378. IEEE Computer Society, 2003.

[WV03b] Y. Wang and J. Vassileva. Trust and Reputation Model in Peer-to-Peer Networks. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, P2P '03, pages 150 – 157. IEEE Computer Society, 2003.

[YMG08] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[ZH07] R. Zhou and K. Hwang. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):460 – 473, 2007.

Appendices

Appendix A

Attack Modelling

A.1 Introduction

Here we present the model for attacks in a mobile sensor network deployed to monitor traffic and vehicular movement, that was described in the example of Chapter 1. In this section, how collusion and intoxication attacks have been implemented in this research are described, including the attacker's goals and capabilities.

A.1.1 Collusion

The mobile sensor nodes in the network may decide to collectively deceive the RTM by having corrupt portions of a file and form a swarm and carry out a collusion attack. This type of attack has been described extensively in Section 3.11.1. The nodes capture vehicular movement around a junction and forward the captured evidence through neigbouring nodes to a central server; where decisions are made about traffic redirections or for further investigation about the cause of an accident. Assuming captured evidence is in form of a message, collusion proceeds according to Algorithm 3.

Algorithm 3 Algorithm for a simple case of collusion attack in a mobile sensor network scenario

```
while node collaboration = true do
  for for nodes $a, b, c, d$ do
    with message transmission $a \rightarrow b \rightarrow c \rightarrow d$
    $a(M) \rightarrow b$
    $a$ listens to $b$'s transmission
    $b(M) \rightarrow c$
    $b$ listens to $c's$ transmission
    $c(M \rightarrow M\#)$
    $c(M\#) \rightarrow d$
    $b$ reports to $a$ on $c's$ transmission as $M$
  end for
end while
```

A.1.2 Intoxication

Generally, intoxication attack is deceiving the system by being cooperative for an extensive period of time to gain the trust if others and suddenly begin to misbehave. This thereby, makes it difficult for the reputation system to identify the misbehaving node. In the network, a node that has achieved a 'trusted' status by cooperating in sending accurate reports across the network, receives a message about captured evidence from a next hop neigbour. The node replaces the message with misleading, corrupt or malicious content and sends it for onward journey to the server. Algorithm 4 gives the steps to a successful intoxication attack.

A.1.3 Network Goals

The goals of the network in the example above are:

- Preserve message integrity across the network.

Algorithm 4 Algorithm for a simple case of intoxication attack in a WSN

```
while node collaboration = true do
    for a trusted node $a$ in a network
    with direct links with nodes $(b, c)$
    $a(M) \rightarrow b, a(M) \rightarrow c$
    $a(RV) + +$
end while
$a(RV) \geq 3$ max threshold
$a(M\#) \rightarrow b$ or $c$
```

- Dynamically compute new ratings for nodes as they collaborate.
- Encourage good behaviour through a reward system of higher ratings.
- Ensure that misbehaving sensor nodes are punished or excluded.
- Minimise the spread of misleading or inaccurate message.

A.1.4 Attackers Goals and Capabilities

From all of the above, the attackers goals are:

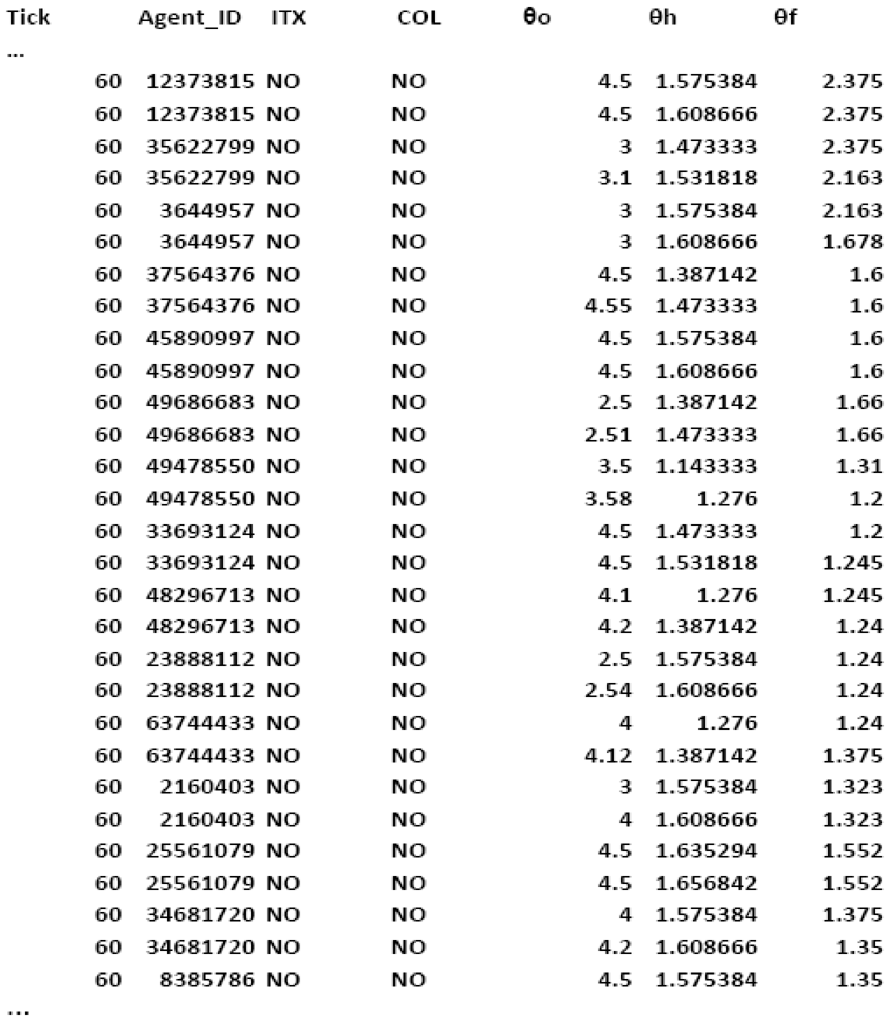
- To broadcast corrupt messages or exchange the message with others.
- To collude with other nodes in order to flood the network with corrupt messages and eventually result in wrong judgments from the server.
- To gain a 'trusted' status by sending true observations and later spreading corrupt files across the network.
- To constantly send bogus messages and flood the network with messages to deplete the resources of other nodes

To achieve the goals listed above, a misbehaving node with a new unique identity and authenticated to participate in network activities sends corrupt messages by generating bogus observations and sending to its next hop. The can alter messages received from other nodes because it is one hop away from the nodes.

Appendix B

Dataset

This section provides a sample data-set showing the observed evidences of interactions among agents in a time tick. Figure B.1 details the fluctuations in trust values and ratings of agents in the domain. The definitions of *tick*, *AgentID*, *ITX*, *COL*, θ_h and θ_o in the table have been given in Chapter 6. The table depicts the substantial amount of data (in one tick) that can be generated from D3-FRT and saved to the database before any analysis or processing, making our approach a fit for the data intensiveness of DDDAS.



Tick	Agent_ID	ITX	COL	θo	θh	θf
...						
60	12373815	NO	NO	4.5	1.575384	2.375
60	12373815	NO	NO	4.5	1.608666	2.375
60	35622799	NO	NO	3	1.473333	2.375
60	35622799	NO	NO	3.1	1.531818	2.163
60	3644957	NO	NO	3	1.575384	2.163
60	3644957	NO	NO	3	1.608666	1.678
60	37564376	NO	NO	4.5	1.387142	1.6
60	37564376	NO	NO	4.55	1.473333	1.6
60	45890997	NO	NO	4.5	1.575384	1.6
60	45890997	NO	NO	4.5	1.608666	1.6
60	49686683	NO	NO	2.5	1.387142	1.66
60	49686683	NO	NO	2.51	1.473333	1.66
60	49478550	NO	NO	3.5	1.143333	1.31
60	49478550	NO	NO	3.58	1.276	1.2
60	33693124	NO	NO	4.5	1.473333	1.2
60	33693124	NO	NO	4.5	1.531818	1.245
60	48296713	NO	NO	4.1	1.276	1.245
60	48296713	NO	NO	4.2	1.387142	1.24
60	23888112	NO	NO	2.5	1.575384	1.24
60	23888112	NO	NO	2.54	1.608666	1.24
60	63744433	NO	NO	4	1.276	1.24
60	63744433	NO	NO	4.12	1.387142	1.375
60	2160403	NO	NO	3	1.575384	1.323
60	2160403	NO	NO	4	1.608666	1.323
60	25561079	NO	NO	4.5	1.635294	1.552
60	25561079	NO	NO	4.5	1.656842	1.552
60	34681720	NO	NO	4	1.575384	1.375
60	34681720	NO	NO	4.2	1.608666	1.35
60	8385786	NO	NO	4.5	1.575384	1.35
...						

Figure B.1: A data set of captured evidences within a time tick from D3-FRT