Dynamic Trust Management for Mobile Networks and Its Applications

Fenye Bao

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Computer Science and Applications

Ing-Ray Chen (Chair)
Chang-Tien Lu
Wenjing Lou
Konstantinos P. Triantis
Ananthram Swami

May 27, 2013
Falls Church, Virginia

Keywords: mobile networks, trust management, adaptive control, optimization, secure routing, intrusion detection, performance analysis.

Dynamic Trust Management for Mobile Networks and Its Applications

Fenye Bao

Abstract

Trust management in mobile networks is challenging due to dynamically changing network environments and the lack of a centralized trusted authority. In this dissertation research, we *design* and *validate* a class of dynamic trust management protocols for mobile networks, and demonstrate the utility of dynamic trust management with trust-based applications. Unlike existing work, we consider *social trust* derived from social networks in addition to traditional *quality-of-service* (QoS) *trust* derived from communication networks to obtain a composite trust metric as a basis for evaluating trust of nodes in mobile network applications. Untreated in the literature, we design and validate trust composition, aggregation, propagation, and formation protocols for dynamic trust management that can learn from past experiences and adapt to changing environment conditions to maximize application performance and enhance operation agility. Furthermore, we propose, explore and validate the design concept of application-level trust optimization in response to changing conditions to maximize application performance or best satisfy application requirements. We provide formal proof for the convergence, accuracy, and resiliency properties of our trust management protocols. To achieve the goals of identifying the best trust protocol setting and optimizing the use of trust for trust-based applications, we develop a novel model-based analysis methodology with simulation validation for analyzing and validating our dynamic trust management protocol design.

The dissertation research provides new understanding of *dynamic trust management* for mobile wireless networks. We gain insight on the best trust composition and trust formation out of social and QoS trust components, as well as the best trust aggregation and propagation protocols for optimizing application performance. We gain insight on how a modeling and analysis tool can be built, allowing trust composition, aggregation, propagation, and formation designs to be incorporated, tested and validated. We demonstrate the utility of dynamic trust management protocol for mobile networks including mobile ad-hoc networks, delay tolerant networks, wireless sensor networks, and Internet of things systems with practical applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition. Through model-based analysis with simulation validation, we show that our dynamic trust management based protocols outperform non-trust-based and Bayesian trust-based protocols in the presence of malicious, erroneous, partly trusted, uncertain and incomplete information, and are resilient to trust related attacks.

Acknowledgements

The dissertation would not have even been possible without the help of so many people in so many ways. I owe big thanks to them all and would really like to express my deepest appreciations here.

First of all, I would like to express my sincere gratitude to my advisor, Dr. Ing-Ray Chen. Thank you so much for providing valuable guidance and persistent help during my four-year PhD study. It is your training, encouragement, and support that make me myself today. I believe the skills, enthusiasm and earnest working attitude that I have learnt from you will carry me further in my future career.

I would also like to thank my committee members, Drs. Chang-Tien Lu, Wenjing Lou, Konstantinos P. Triantis, and Ananthram Swami. Your valuable advices and comments have greatly improved the quality and readability of the dissertation.

Drs. Jin-Hee Cho and MoonJeong Chang, thank you so much for your collaboration on this research topic and your insightful suggestions on my dissertation work. Drs. Jason J. Xuan and Xinghua Li, I really appreciate the opportunities to work with you. This experience not only was enjoyable but also expanded my vision of the research area.

Most importantly, I would like to thank my fiancée Lijuan Xu. Your support, encouragement, patience and unwavering love were undeniably the bedrock of my life in the past five years. I would like to thank my mother Xueyun Qiu, my father Chenggan Bao, and my brother Yeqing Bao for their constant love, support, patience, and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Trust Management in Mobile Networks

A mobile network typically comprises heterogeneous nodes performing peer-to-peer wireless communications to achieve the system functionality. There are various types of mobile networks, including mobile ad-hoc networks (MANETs) [59], delay/disruption tolerant networks (DTNs) [38, 66], mobile wireless sensor networks (WSNs) [8, 70, 151], Internet of things (IoT) systems [11, 79] , etc. The key features of mobile networks are reconfigurability (low dependency on infrastructure), distributed control (no centralized entity needed for managing the network), and dynamicity (change of network topology, population size, etc.). Because of these features, mobile networks have been widely deployed in many civil and military applications. For example, without relying on any wireless infrastructure, like access points or wireless routers, conference attendees can set up an ad-hoc network using their laptops for instant messaging and discussion. In battlefield situations, a commander can dynamically assemble and manage a mobile network consisting of trustworthy group members to achieve a critical mission assigned. In zoology research, sensors are attached to wild animals to form a delay tolerant WSN in order to track long term animal migration behaviors [99, 198].

In mobile networks, a node could be an autonomous or human operated device working cooperatively with others. A mobile network is vulnerable to many attacks. In addition to attacks to wireless networks such as eavesdropping, tampering, jamming, and denial of service attacks [148], mobile networks face more mobility-induced attacks [101, 112, 151], such as black-hole, wormhole, Sybil, and slandering attacks. A node in a mobile network can be compromised and launch insider attacks which are hard to defend with traditional cryptography techniques. Because very frequently there is no trusted third party available in mobile networks, each participating node may need to assess the trustworthiness of others based on direct observations and/or indirect recommendations. The primary objective of our dissertation research is to design and validate a trust management protocol that can provide a subjective yet accurate assessment

of trust of mobile nodes in the presence of malicious, erroneous, partly trusted, uncertain and incomplete information, and to demonstrate the utility of the trust management protocol designed for mobile networks with practical applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition.

A challenge to trust management protocol design in mobile networks is that very often nodes exhibit a wide range of heterogeneous QoS characteristics (e.g., energy level, bandwidth, moving speed, etc.) and social behaviors (e.g., selfishness, honesty, social connections, etc.) especially for sensing-capable devices carried by human operators such as smartphones and digital personal assistants [65, 66]. Hence, a trust protocol management must take both QoS and social metrics into consideration in trust composition and trust formation, allowing the best trust components to be selected and composed based on application performance and security requirements. Moreover, application-level trust optimization techniques must be an integral part of protocol design, allowing applications to identify and deploy the best way to use trust to classify nodes to maximize application performance.

Another challenge to trust management in mobile networks is the dynamically changing network environment. The network environment of a mobile network can be characterized by a set of variables with each defining one aspect of network conditions, such as node density, number of misbehaving nodes, etc. The dynamicity may be described by an evolving set of variables as a function of time, and can be categorized into three levels: *node dynamics*, *locality dynamics*, and *network dynamics*. The node dynamics includes the changing status of each node, such as energy level, available bandwidth, cooperativeness, compromising rate, etc. The locality dynamics is reflected by the change of each node's local environment (consisting of neighbors or nodes that interact most often), such as the number of cooperative/compromised neighbors, average energy level of neighbors, social connection with others, etc. The network dynamics is reflected by the evolving global network status, such as network topology, mobility pattern, population size, etc. In order to maximize the application-level performance throughout the network lifetime, it is critical to consider dynamicity and incorporate adaptability in protocol design. For example, some nodes may have low energy due to intensive message transmission at the early stage. To save energy, at the later stage these nodes may become uncooperative to others. A node may change its forwarding policy when it is surrounded by many uncooperative/compromised nodes. Applying one protocol setting optimized for a static network condition to a dynamic network environment will degrade the application performance. A trust management protocol must adapt to the dynamically changing network environment, such that the trust assessment is accurate and performance of an application built on top of the trust management protocol is maximized.

In large-scale mobile networks, it might be impractical for a node to obtain and store trust information towards all other nodes due to its limited computation and storage resources. Instead, a node may keep trust values of a small set of nodes with higher trust values or with which it shares interests. The challenge is to design a trust management system for large-scale mobile networks with mobile nodes equipped with limited resources, satisfying the requirements in trust convergence, accuracy, and resiliency, and trust-based application requirements.

Lastly, very often there is no centralized trust entity in a mobile network from which a node can refer trust of another node with complete confidence. Consequently a node will have to rely on direct observations when it comes in contact with another node or recommendations from other nodes in the mobile network. A challenge to trust management in mobile environments is to provide subjective yet accurate trust assessment to cope with rapidly changing environments and the presence of malicious, erroneous, partly trusted, uncertain and incomplete information.

1.2 Research Goal

The dissertation research is motivated by the two important but unaddressed problems in the literature: (a) diverse social and QoS trust characteristics of mobile nodes; this calls for a trust management protocol to identify the best way to compose and form trust out of social and QoS trust properties so as to satisfy application performance or security requirements; and (b) difficulty of providing subjective yet accurate trust assessment because of rapidly changing environments and the presence of malicious, erroneous, partly trusted, uncertain and incomplete information; this calls for dynamic trust management that can dynamically adjust trust settings at runtime to maximize application performance. In this dissertation research, we aim to design, analyze and validate a dynamic trust management protocol for mobile networks, including identifying the best way to perform trust composition, trust aggregation, trust propagation, and trust formation to satisfy application requirements, and the best way to adjust trust settings dynamically in response to environment changes to maximize application performance.

For the design part, we propose to combine social trust properties derived from social networks with QoS trust properties derived from communication networks into a composite trust metric, taking into account both social relationships and functional competencies of network participants. We propose to explore various social trust properties (e.g., honesty, intimacy, etc.) and QoS trust properties (e.g., connectivity, competence, etc.) and demonstrate their effectiveness for trust-based applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition. For each individual trust property, we

explore the best design to aggregate and propagate trust information in the presence of malicious, erroneous, partly trusted, uncertain and incomplete information such that between two nodes the trust assessment toward each other is close to actual status.

We propose to investigate a new design notion of application-level optimization for trust composition, and trust formation to maximize application performance. For example, for secure routing applications in mobile networks, delivery ratio and delay are considered the most important performance metrics. This calls for the use of novel component trust metrics such as connectivity for reducing delay, and honesty and cooperativeness for enhancing delivery ratio. Application-level optimization in this case concerns not only trust composition (to select novel trust components, such as connectivity, honesty and cooperativeness for compositing trust), but also trust formation (to find the best way to form trust out of these trust components).

We propose to investigate the design notion of dynamic trust management by which a trust-based application built on top of our trust protocol may dynamically adjust trust parameters (e.g., trust weights associated with trust components for trust aggregation, propagation and formation, and application-level trust thresholds for classifying and selecting nodes) in response to the dynamically changing network environment, so that throughout the network lifetime, participating nodes can have subjective yet accurate assessment of trust toward each other and the trust-based application as a whole can use trust at its best to maximize application performance.

For validation, we propose to develop a novel model-based analysis methodology by which, the "subjective" trust evaluation obtained from protocol execution can be compared with the "objective" trust evaluation generated from actual network status, thus theoretically validating our trust protocol design. We further experimentally validate analytical results with extensive simulation using ns-3 [1]. Our goal is to demonstrate the utility of our trust management protocol with practical applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition for a wide array of mobile networks including MANETs, DTNs, WSNs, and IoT systems both analytically and by simulation. For each application, we will conduct extensive comparative studies with conventional trust-based and non-trust-based solutions to further validate our dynamic trust management protocol design.

1.3 Research Contribution

This dissertation research addresses all aspects of trust management for mobile networks: trust composition, trust aggregation, trust propagation, and trust formation. The overarching contribution is that we *design* and *validate* a dynamic trust management protocol that can provide a subjective yet accurate assessment of trust of mobile nodes

in the presence of malicious, erroneous, partly trusted, uncertain and incomplete information, and we demonstrate the utility of the dynamic trust management protocol for mobile networks including MANETs, WSNs, DTNs, and IoT systems with practical applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition.

We envision the following original contributions from the dissertation research with high impacts:

1. We propose the notion of using not only traditional ***QoS trust*** derived from communication networks, but also ***social trust*** derived from social networks to obtain a composite trust metric as a basis for evaluating trust of nodes in mobile network applications [19, 20, 44, 61]. The intention is to take into account both social relationships and functional competence and obtain a comprehensive trust assessment toward each node, such that the performance of a trust-based application can be maximized.
2. Untreated in the literature, we design and validate dynamic trust management protocols that can learn from past experiences and adapt to changing environment conditions (e.g., increasing/decreasing hostility, increasing misbehaving node population, etc.) to maximize application performance and enhance operation agility. This is achieved by addressing critical issues of trust management for mobile applications, namely, trust composition, aggregation, propagation, and formation. The learning process and adaptive designs are reflected in trust aggregation, trust propagation, and trust formulation. For trust composition, aggregation and propagation, we explore novel social and QoS trust components and then devise trust aggregation and propagation protocols for trust data collection, quality of information (QoI) dissemination and analysis for peer-to-peer subjective trust evaluation of individual social and QoS trust components, and prove that the convergence, accuracy, and resiliency properties are preserved by means of theoretical analysis and simulation validation. For trust formation, we identify the best way to form trust out of social and QoS trust components depending on application characteristics so as to maximize application performance. Dynamic trust management is achieved by first determining the best trust protocol settings under which application performance is maximized, given a set of model parameters specifying the environment conditions (e.g., percentage of malicious nodes), and then at runtime the system learns and adapts to changing environment conditions by using the best protocol settings identified from static analysis.
3. To achieve the goals of identifying the best trust composition and trust formation for trust-based applications, we develop a novel model-based analysis methodology for analyzing and validating our trust management protocol design. The model-based

analysis methodology can be applied to a wide variety of applications in mobile networks including MANETs, WSNs, DTNs, and IoT systems. The novelty lies in the new design notion of objective trust derived from global knowledge or ground truth derived from the mathematical model describing a node against which subjective trust obtained as a result of executing our trust management protocol may be compared and validated. The research requires a semi-Markov mathematical model and an iteration solution technique be developed to faithfully describe a large number of heterogeneous mobile entities with a variety of QoS and social behaviors to yield global knowledge or ground truth of node status, thus providing objective trust against which subjective trust from protocol execution can be validated. The end product is a model-based analysis tool for design, testing and evaluation of our trust management protocol applicable to a wide range of trust-based applications in mobile networks, allowing trust composition, trust aggregation, trust propagation, and trust formation designs to be incorporated, tested and validated.

We propose a two-step process of optimization. The first optimization problem is to identify the best parameter setting for trust aggregation and propagation to minimize trust bias (i.e., a node's trust evaluation value vs. a node's ground truth status). This addresses trust accuracy, convergence, and resiliency issues. The second optimization problem is to identify the best parameter setting for trust formation (i.e., weights of trust components) to maximize application performance. The global optimum to maximize application performance is not unique. As long as the trust evaluation value is consistent with ground truth status in terms of trust ranking, one can find the global optimum by adjusting the trust formation parameter setting. However, the global optimum cannot be guaranteed if the trust evaluation value is inconsistent with ground truth status. Our proposed two-step process can significantly reduce the search space. Moreover, the best parameter setting for trust aggregation and propagation once identified could be reused and applied to different applications.

4. Untreated in the literature, we propose, explore and validate the design concept of application-level trust optimization in response to changing conditions to maximize application performance or best satisfy application requirements. For the misbehaving node detection application, we identify the best application-level drop-dead trust threshold below which a node is considered misbehaving, and that the minimum trust threshold can be adjusted in response to changing conditions to minimize the false alarm probability. For the trust-based survivability management application, we identify the best minimum trust level required for successful mission completion and the drop dead trust level to maximize the system reliability of mission execution with dynamic team membership. For the trust-based secure routing application, we identify the best forwarding trust threshold and recommender trust

threshold to ensure quality of information (QoI) routing, and that the trust thresholds can be adjusted dynamically to best satisfy application requirements in delivery ratio or message delay in the presence of misbehaving nodes.

5. The dissertation research provides new understanding of dynamic trust management for mobile network applications. We gain insight on the best trust composition and trust formation out of social and QoS trust components, as well as the best trust aggregation protocol, when given application mission characteristics and trustee properties as input. We gain insight on how a tool described in contribution #3 above can be built, allowing trust composition, aggregation, and formation designs to be incorporated, tested and validated. We gain insight on dynamic trust evaluation for optimizing application performance including misbehaving node detection, trust-based survivability management, secure routing, and service composition. Lastly but not the least, we gain insight on how to prove that a dynamic trust management protocol can be made resilient to a class of malicious attacks that aim to disrupt the trust of the system.

1.4 Thesis Organization

The rest of the dissertation is organized as follows. Chapter 2 provides a comprehensive survey of trust management in mobile networks and its applications. Chapter 3 presents the network environment and assumptions for dynamic trust management in mobile networks. In Chapter 4 we describe the general design principles of dynamic trust management in mobile networks. In Chapter 5 we discuss our model-based analysis methodology with simulation to validate protocol designs. In Chapter 6, Chapter 7, Chapter 8, and Chapter 9, we apply dynamic trust management design principles to MANETs, DTNs, WSNs, and IoT systems, respectively, with specific dynamic trust management protocols being developed, and a comprehensive comparative performance analysis of trust-based applications build on top of dynamic trust management being performed. In Chapter 10, we investigate the applicability and implementation issues of our proposed dynamic trust management designs. Finally, Chapter 11 summarizes research milestones achieved, work to be completed, and the schedule to complete the proposed dissertation research.

Chapter 2

Related Work

2.1 Concept of Trust

2.1.1 The Meaning of Trust

Trust exists everywhere in daily life. It is involved in our decision making process whenever there is a cooperation among different entities. The concept of trust is originally discussed in social sciences. Since trust is a multidisciplinary concept [128], people from different fields take diverse views of trust. For example, in sociology [63, 75, 123, 124, 166, 176], trust is considered as one element of the social constructs [166] and reflects the relationship between social actors (individuals or groups). Psychologists take a cognitive view of trust [37] and believe that trust is built on top of social influence [132]. In philosophy [15, 94, 129], trust is considered as an attitude towards people. Some philosophers differentiate trust from reliance since reliance can only be disappointed while trust can be betrayed or at least let down, not just disappointed [15, 129]. In organizational management [161, 164] and economics [29, 195], trust is used for risk analysis and incentives are used very often for trust enforcement and trust establishment.

There is no consensus on the definition of trust. Here we give several definitions of trust from the literature.

(1) "*Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.*" [75]

Diego Gambetta, 2000

(2) "*Trust basically is a mental state, a complex mental attitude of an agent 'x' towards another agent 'y' about the behaviour/action 'a' relevant for the result (goal) 'g'.*" [37]

Cristiano Castelfranchi and Rino Falcone, 2000

(3) "*Trust is an attitude that we have towards people whom we hope will be trustworthy, where trustworthiness is a property, not an attitude.*" [129]

Carolyn McLeod, 2011

Gambetta's definition of trust is one of those that are widely cited, where trust can be seen as a threshold point of a probabilistic distribution and only concerns those future actions that will affect our decisions. However, Castelfranchi and Falcone pointed out that trust is more complex than a subjective probability. They argued for a cognitive view of trust and claimed that only a cognitive agent can trust another agent. McLeod's definition of trust emphasizes that trust is an expectation that someone is competent in certain respects and, trust and trustworthiness are distinct.

In this dissertation, we adopt a trust definition similar with Gambetta's, particularly for mobile network environments. We use trust to represent a degree of belief of an agent based on its own knowledge (direct or indirect) whether or not another agent will perform an action or will be in a state (e.g., whether a node is cooperative or malicious, whether a node has close connectivity with the designation node and can quickly deliver the message). We are concerned with the following trust dimensions:

(1) *Trustor*: This is the agent which is doing the trust assessment.

(2) *Trustee*: This is the agent which is being assessed.

(3) *Trust Property/Component*: This represents the particular action that the trustee might perform or might be in a particular state. Specifically, we consider both social trust properties derived from social networks and quality of service (QoS) trust properties derived from communication networks.

(4) *Knowledge*: This includes all information that the trustor has. It could be direct observations or indirection recommendations from other parties.

(5) *Time*: Trust evolves over time since the states of both trustor and trustee may change, and the trustor may obtain new information. The importance of the *time* dimension in trust management for mobile networks is manifested by dynamically changing environments, which is one research goal in this dissertation.

(6) *Space*: Trust also evolves over space since direct-observation trust ("seeing is believing") tends to be more trustable than indirect recommendations, especially trust is propagated over space from one entity to another. A research goal in this dissertation is to devise trust aggregation and propagation protocols such as peer-to-peer trust evaluation is close to actual status.

2.1.2 Trust versus Reputation

In the literature, trust and reputation are often used together. Ostrom [146] and Mui *et al*. [139] provided a clarification for trust, reputation and other related concepts: *reputation* is the perception that an agent creates through past actions about its intentions and norms, and *trust* is a subjective expectation an agent has about another's future behavior based on the history of their encounters. Mui *et al*. [139] claimed that the increase in an agent's reputation should also increase the trust towards this agent, and the increase of trust will increase the reciprocating actions which will then increase the reputation. The levels of trust and reputation are positively reinforcing and a decrease in any of them can also lead to a downward spiral [146].

Josang *et al*. [97] used the following two statements to illustrate the difference between trust and reputation:

(1) "*I trust you because of your good reputation*."

(2) "*I trust you despite your bad reputation*."

The first statement is consistent with a general understanding that trust and reputation are positively reinforcing. The second statement reveals that in addition to reputation, there exists some other knowledge (e.g., intimate relationships [97]) involved in trust related decision making.

Despite the conceptual difference between trust and reputation, the difference between *trust system* and *reputation system* is blurred since both trust and reputation can be used for decision making. Trust systems can of course incorporate elements of reputation systems and vice versa [97]. In this dissertation research, we distinguish *trust* from *reputation* since we consider social trust in addition to QoS trust for trust composition and formation.

2.2 Computational Trust Model

Researchers and practitioners have applied the concept of trust in human society to networks and distributed systems, and proposed computational trust models [10, 97] to improve the system security, such as Web of Trust in PGP (Pretty Good Privacy) [172] and trust models in e-commerce [107, 142].

Marsh [126] is among the first who tried to develop a formalization for trust as a computational concept. In Marsh's formalization, trust is separated into three different categories: *basic*, *general*, and *situational* trust, with each represented by a value in [-1, 1). *Basic trust* represents the general trust disposition of the trustor, not in any specific situation or toward any specific trustee. It is derived from past experience with all other agents in all situations, through the entire life of experiences. *General trust* represents the trust toward a specific trustee, but not in any specific situation. *Situational trust* repre-

sents the trust toward a specific trustee in a specific situation. They formalized situational trust as the product of three parts: *utility* that can be gained from the situation, *importance* of the situation to the trustor, and *general trust*. They also introduced the temporal index into the formalization to represent evolving trust over time. The formalization provides a description of trust and is large in the sense that extensions are possible [126]. Nevertheless, the limitations of the formalization, as discussed in their work, are (a) the range value chosen for trust [-1, 1) is problematic (e.g., the product of two negative trust values is positive), and (b) the operators for the formalization are limited.

There are many computational trust models being proposed in the literature, including *weighted summation* [4, 162, 168, 180], *Bayesian* [76, 95, 186], *game theory based* [29, 56, 91], *belief based* [98], *routing algebra based* [196], *graph based* [177], *flow based* [13, 14] , *fuzzy logic based* [36], and *information theory-based* [175] models. Below we survey and contrast our work with the first three computational trust models which have been used most frequently in the literature.

Weighted Summation Models

One of most popular and straightforward computational trust models is the weighted summation or average model [3, 4, 30, 113, 120, 130, 159, 162, 168, 180, 188]. Models in this category aggregate trust using a weighed calculation on information collected from different sources (e.g., direct observation vs. indirect observation [4, 162, 168, 180], past experience vs. recent experience [162], etc.). The weight parameters are determined by factors such as the trustworthiness of the information provider, the rate of trust decay, etc. For example, eBay [159] employs this model to calculate the feedback score. The advantages of this kind of models are, first it is simple and easy to understand, and second the linear calculation is easy to implement and efficient. However, it is a challenge to find the best weight parameters to achieve an accurate trust evaluation. Our dissertation research considers weighted summation as one of the many possible ways for *trust formation* and it seeks the best trust composition and formation to maximize application performance.

Bayesian Models

In Bayesian trust models, the evidence of trust is considered as a stochastic process. First, a *prior* distribution of the trust value is assumed. Then, the evidence is observed and can be used as the *likelihood* to calculate the *posterior* distribution following Bayes' Theorem. After new evidence is observed, the previous *posterior* distribution obtained can be used as a new *prior* distribution to calculate the next *posterior* distribution iteratively. The new evidence could be from direct observations or indirect recommendations. Direct observations may be used to update the numbers of positive and negative interaction experiences, whereas indirect recommendations may be discounted by the confidence [68] or belief [96] of the trustor toward the recommenders. Since this is an

iterative computing process, it is desirable if both the *prior* and *posterior* distributions follow the same distribution and only the parameters are updated iteratively after new evidence is observed. Therefore, conjugate prior distributions, like Beta distribution [76, 96, 137, 149, 174, 194] and Dirichlet distribution [76, 95, 186], are usually used as the *prior* distribution to build trust models.

For example, in a Bayesian trust model we could use Beta distribution as the prior distribution for the trust variable $\theta \sim Beta(\alpha, \beta)$. Observations can be considered as instances of a Binomial (or Bernoulli) experiment $k \sim Binomial(n, \theta)$, where k and n are the number of positive observations and the total number of observations respectively. Since Beta distribution is the conjugate prior of Binomial distribution, the posterior distribution for θ will be $Beta(\alpha + k, \beta + n - k) = Beta(\alpha', \beta')$. Then the trust value can be calculated as the expected value of θ, which is $E(\theta) = \frac{\alpha'}{\alpha' + \beta'}$. If there is no prior knowledge available, usually the initial prior distribution is assumed to be $Beta(1, 1)$, the uniform distribution. In addition, discounting mechanisms can be provided for recommendations and trust decay over time [76, 96]. When the likelihood above is a Multinomial distribution instead of Binomial distribution, using Dirichlet distribution as the prior distribution can provide great flexibility [76, 95, 186].

Although the Bayesian trust model above provides a statistically sound basis for trust assessment, they do not consider the noise in the evidence or its observation. In real systems, especially for mobile networks, those noises are unavoidable. Capra [35] proposed another Bayesian estimation for trust evaluation based on the basic Kalman filter [100]. This model provides an optimal estimation given the presence of noises in both evidence and its observation. Essentially, it is an iterative process updating the *posterior* estimation with the combination of the *prior* estimation and the new observation. When the covariance of the observation noise is large, the *posterior* estimation approaches the *prior* estimation; when the covariance of the observation noise is small, the posterior estimation approaches the observation itself, in such a way that the trust estimation error is minimized. However, the covariance of the observation noise has to be an input to this model. It can be obtained if there is a sufficient amount of training data. In practice, it is difficult especially in the presence of malicious nodes performing slandering attacks.

Bayesian based computational trust models aim to accurately evaluate a single trust metric which is treated as a random variable following a probability distribution function. Then, based on evidences collected, the Bayesian estimates of the distribution function parameters are obtained (e.g., α, β of Beta distribution) to make the observed evidences the most probable so as to yield the average trust value. We note that the iterative evidence-based calibration of the distribution function parameters is designed for a single trust metric. Consequently, within the scope of trust composition, trust aggrega-

tion, trust propagation, and trust formation, a Bayesian based trust model addresses only trust aggregation protocol design. In our dissertation research, we address all aspects of trust management issues. We consider not just one but multiple distinct QoS and social trust properties to address the issue of trust composition. Furthermore, for each trust property we consider not only direct evidences but also indirect recommendations for trust aggregation and propagation protocol design. For Bayesian trust management, there is no direct trust and indirect trust weight parameters because the weight to indirect trust is determined by confidence or belief based on the positive and negative experiences/recommendations received. In our dynamic trust management, we consider direct trust vs. indirect trust weights as protocol parameters and dynamically adjust the parameter values in order to minimize trust bias and maximize application performance. Finally, we address the issue of trust formation to maximize application performance. In Chapter 7, we compare the performance of our protocol against Bayesian based trust models.

Game Theory Models

Game theory based trust models [29, 56, 91] usually use incentives to stimulate the cooperation between nodes, such that the system can reach a stable state where the overall utility is maximized. However, these models only consider selfish nodes and cannot deal with malicious nodes that intend to disrupt the system functionality. Staab, *et al*. [73] proposed a trust model by considering a *game* between normal nodes and attackers, given the knowledge of the strategies that attackers will use in each system configuration. Their model can be used to find the optimal parameters for an evidence-based trust model to maximize the expected utility. However, in reality, it is difficult to obtain a complete set of attacker strategies and the attacker behavior may change dynamically. In our dissertation research we do not make assumptions of the attacker strategies. Rather, we design dynamic trust management protocols that can learn from past experiences and adapt to changing environment conditions to maximize application performance and enhance operation agility.

Information Theory Models

In information theory models [175], trust is considered as a measure of certainty of whether the trustee will perform an action in the trustor's point of view. Depending on the way of aggregating trust, there are two trust models: *entropy-based* and *probability-based*. In the entropy-based trust model, trust is calculated as the entropy of information (recommendations) from others. In the probability-based model, trust is obtained by aggregating recommendations using conditional probability. Similar to Bayesian trust management, information theory models do not have direct trust vs. indirect trust as design parameters and only address trust aggregation protocol design. In our dynamic

trust management, we consider the design of trust composition, trust propagation, trust aggregation and trust formation protocols.

2.3 Trust Management in Mobile Networks

Blaze *et al*. [27] first introduced the term "Trust Management" and identified it as a separate component of security services in networks. They clarified that "trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships." Trust management in mobile networks is needed when participating nodes, without any previous interactions, desire to establish a network with an acceptable level of trust relationships among them, for example, for coalition operation without predefined trust. Thus, the concept of trust is attractive to communication and network protocol designers where trust relationships among participating nodes are critical to build collaborative environments to achieve system optimization. Many trust management protocols have been designed for mobile networks with different characteristics for a wide range of applications.

2.3.1 Trust Management in Mobile Ad-Hoc Networks

Trust Management Framework

Eschenauer *et al*. [72] discussed the properties of trust establishment in MANETs and illustrated the differences from the trust establishment in the Internet. Because of the special characteristics of MANETs, including a lack of a fixed infrastructure, rapid changes of network topology, and unreliable wireless transmission, trust evidences that can be collected in traditional networks or authorized by a trusted third-party are not prevalent in MANETs. They claimed that peer-to-peer communications are more suitable for trust evidence *generation*, *distribution*, and *discovery* in MANETs, and pointed out that a crucial aspect of trust establishment in MANETs about what specific trust metrics should be adopted for trust evidence evaluation. Michiardi and Molva [130] proposed a collaborative reputation mechanism to enforce node cooperation (CORE) in MANETs. The CORE scheme relies on two key designs: a reputation table stored by each node to maintain the reputation toward others and a watchdog mechanism for detecting cooperative behavior. The reputation table combines the reputation from both direct observations obtained from the watchdog and indirect recommendations from other nodes. Buchegger and Boudec proposed CONFIDANT [30] protocol and applied it to the Dynamic Source Routing (DSR) [93] in MANETs. They used a *neighborhood watch* (similar to the watchdog mechanism in CORE) to detect non-compliant behaviors of neighboring nodes. Once a node detects malicious evidence, it sends an *alarm* message to others to propagate the evidence. Theodorakopoulos and Baras [177] modeled the trust evaluation process in MANETs as a path finding problem on a directed graph, where nodes

represent entities and edges represent trust relations. Using the theory of semirings on an established direct graph, two nodes without previous direct interaction can establish indirect trust relation. Sun *et al*. [175] presented an information theoretic framework for modeling trust propagation and aggregation in ad hoc networks. The framework comprises four axioms as the basis for trust propagation and aggregation. Under this framework, *entropy-based* and *probability-based* trust models are proposed.

The above trust management protocols assume a flat structure in MANETs and have scalability issues when the network size increases. Verma *et al*. [182] and Davis [67] considered hierarchical trust management for MANETs. In their hierarchical trust management schemes, each node performs trust evaluation locally. However, their schemes heavily rely on the certificates issued off-line or by trusted third parties which typically are not available in MANET environments.

Our trust management protocol design when applying to MANETs can handle small, flat MANETs as well as large, hierarchically-structured MANETs. Moreover, a major distinction of our dissertation research from the above cited work is that our trust framework covers all aspects of trust management, namely, trust composition, trust aggregation, trust propagation, and trust formation. For MANET applications built on top of trust management, e.g., misbehaving node detection, we propose a new design notion of *application-level trust optimization* in using trust to classify nodes to maximize application performance.

Trust Metrics

Many QoS performance metrics have been used for trust evaluation in MANETs, such as control packet overhead, throughput, goodput, packet dropping rate and delay [78, 167, 183]. Dependability metrics such as availability [82], convergence time to reach a steady state in trustworthiness for all participating nodes [23], percentage of malicious nodes [28], result of intrusion detection [121] and fault tolerance based on reputation thresholds [136] also have been employed. The use of a "trust level" to associate with a node has received attention recently, considering general attributes such as confidence [203], trust level [175], trustworthiness [136], and opinion [177]. Social trust metrics have also been employed to deal with malicious and uncooperative behaviors in MANETs. Golbeck [80] introduced the concept of social trust by suggesting the use of social networks as a bridge to build trust relationships among entities. Yu *et al*. [192] used social networks to evaluate trust values in the presence of Sybil attacks. Cho, Swami and Chen [61] pioneered the use of both social and QoS trust metrics for trust management of mission-oriented group communication systems in MANETs. In this dissertation research, extending from prior work [61] we propose to combine the notions of social trust derived from social networks with quality-of-service (QoS) trust derived from communication networks to obtain a composite trust metric as a basis for evaluating

trust of mobile nodes in mobile ad hoc network (MANET) environments. We also investigate the best trust formation approach to combine multidimensional trust properties for application-level performance optimization.

Resiliency Analysis

Trust management aims to provide a secure mechanism for MANETs. However, trust management itself faces attacks from malicious nodes, including good-mouthing attacks (recommending a bad node as a good node), bad-mouthing attacks (recommending a good node as a bad node), and white-washing attacks (recommending itself as a good node). Mundinger and Boudec [140] performed a theoretical analysis on the robustness of a reputation system in the presence of liars (providing false recommendations). They claimed that there is a liar percentage threshold above which lying has an impact and can finally corrupt the reputation system. The reputation system needs to compromise between *fast-convergence* and *accurate trust evaluation*. These attacks can be alleviated by taking trust recommendation only from trusted recommenders or performing statistical analysis on the recommendation values to remove bias. Zouridaki *et al.* [203] proposed a robust cooperative trust scheme for secure routing in MANETs. In their scheme, recommenders are chosen in the order of: (1) good recommenders, (2) nodes with recommender trustworthiness higher than a threshold, and (3) all other recommenders. Balakrishnan *et al*. [16] proposed a trust protocol for MANETs to address similar issues (i.e., *recommender's bias*, *honest elicitation*, and *free riding*) in trust recommendations.

We address the issue of protocol resiliency by design and validation. For design, we explore new design concepts against good-mouth or bad-mouth attacks (see Chapter 4 Design Principles). For validation, we propose to demonstrate our protocol's resiliency in two ways. One way is to formally prove protocol resiliency. This can be done by proving that a bad node remains as a bad node despite good-mouthing attacks, and a good node remains as a good node despite bad-mouthing attacks. Another way is to perform model-based analysis with simulation validation (see Chapter 5 Design Validation).

2.3.2 Trust Management in Delay Tolerant Networks

Because of the sparse connection of DTNs, trust management proposed for traditional MANETs are not directly applicable to DTNs. Xu *et al.* [187] proposed a trust management scheme for secure routing in DTNs. Their protocol considers three sources to estimate trust: *cryptographic operation*, *node's behavior*, and *reputation*. For cryptographic operations, encryption and decryption mechanisms are used to provide authentication and confidentiality and to defend outside attackers. A watchdog mechanism is adopted to detect node's behavior, i.e., whether a neighbor node has successfully forwarded a

message or not. The information obtained from cryptographic operation and node's behavior is combined using weighted summation to generate a local trust value. Each node also exchanges its local trust evaluation as recommendation to others. A limitation of their work is that, they did not consider insider attacks from compromised nodes that already have the secret information for encryption and decryption. Another issue is that in DTNs, a node usually has little chance to observe the behavior of next message carrier because of the sparse connectivity and *store-and-forward* routing mechanism. Ayday *et al*. [13, 14] designed an iterative trust management scheme for DTNs. They employed the authentication technique as the underlying mechanism to evaluate a node. A node exchanges its trust evaluation with others and interactively updates its trust evaluation. Inconsistent trust evaluations are identified and removed iteratively until the trust evaluation converges. However, the iteration process has to be performed on each node every time trust is updated, which is inefficient and time-consuming for mobile networks with a large number of nodes.

There is very little research to date on the social aspect of trust management for DTNs. Social relationship and social networking were considered as criteria to select message carriers in a DTN [33, 65]. However, no consideration was given to the presence of malicious or selfish nodes. Li *et al*. [114] considered routing by socially selfish nodes in DTNs, taking into consideration the willingness of a socially selfish node to forward messages to the destination node because of social ties. However, their protocol assumes a social connection graph is known and uses this graph to facilitate trust evaluation. Such information may not be available as input especially for military operations.

Unlike existing work, this dissertation research considers both social trust properties (e.g., selfishness and honesty) as well as QoS properties (e.g., connectivity) for trust composition. Furthermore, our trust aggregation and propagation protocol design takes DTN's opportunistic connectivity into consideration to update and propagate trust without incurring high overhead. Lastly, untreated in the literature, we specifically address trust formation to maximize application performance of applications built on top of our protocol design, e.g., secure routing and intrusion detection, in DTNs.

2.3.3 Trust Management in Wireless Sensor Networks

In the literature, several trust management schemes have been proposed for WSNs. These schemes usually use *certificate-based* or *behavior-based* approaches to perform peer-to-peer trust evaluation. The challenges for trust management in WSNs are constrained resources and scalability. Thus, trust management for WSNs needs to be lightweight and highly scalable to deal with a large number of nodes.

Ganeriwal *et al*. [76] proposed a reputation-based framework for data integrity in WSNs. The proposed reputation system takes information collected by each node using a Watchdog mechanism (from direct monitoring and observations) to detect invalid data and uncooperative nodes. Yao *et al*. [188] proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node maintains highly abstract parameters to evaluate its neighbors. Aivaloglou and Gritzalis [4] proposed a hybrid trust and reputation management protocol for WSNs by combining certificate-based and behavior-based trust evaluations. However, [4, 76, 188] cited above only considered a node's QoS property in trust evaluation. Also the analysis was conducted based on a flat WSN architecture which is not scalable. Liu *et al*. [120] and Moraru *et al*. [137] proposed trust management protocols and applied them to geographic routing in WSNs. However, no hierarchical trust management was considered for managing clustered WSNs. Their work again evaluated trust based on QoS properties only such as the packet dropping rate and the degree of cooperativeness. Our dissertation research considers both QoS and social trust for trust evaluation of a sensor node. Furthermore, we consider hierarchical trust management protocol design for scalability.

Shaikh *et al*. [168] proposed a group-based trust management scheme for clustered WSNs in which each sensor node performs peer evaluation based on direct observations or recommendations, and each cluster head evaluates other cluster heads as well as sensor nodes under its own cluster. This work is similar to ours in that a hierarchical structure is employed for scalability. However, trust in their case is assessed only based on past interaction experiences in message delivery, which in our case is just one possible trust component along with other social and QoS trust components comprising the overall trust metric. Furthermore, we address the trust formation issue (i.e., how a peer-to-peer trust value is formed) to maximize application performance. Zhang *et al*. [197] followed the same hierarchical trust architecture and considered multi-attribute trust values instead of just one as in [168]. They also considered a decay function that captures the changing nature of trust in trust calculations. However, their work is theoretical in nature without addressing what trust attributes should be used (a trust composition issue), how trust can be aggregated accurately (a trust aggregation issue), or what weights should be put on trust attributes to form trust (a trust formation issue). On the contrary, our work addresses all three aspects of trust management. Moreover, we address protocol validation issues by devising a mathematical model yielding objective trust against which subjective trust from protocol execution may be compared for assessing its accuracy.

Capra *et al*. [35] discussed the notion of human trust which could be formed from three sources: direct experiences, credentials and recommendations. In particular, recommendations are trust information coming from other nodes in the social context. We

consider only two sources in our notion of trust, namely, direct experiences and recommendations, since it is hard for sensor node with limited resources to carry credentials. A significant difference of Capra's work from our work is that we specifically consider individual QoS and social trust property, say, X, and devise specific trust aggregation protocols using direct experiences and recommendations to form trust property X, while Capra used the three sources of information to form human trust. Moreover, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, we identify the best way for each trust property X to take in direct experiences and recommendations information so that the assessment of trust property X would be the most accurate against actual status in trust property X. Another significant difference is that we consider trust formation as the issue of forming the overall "trust" out of individual social and QoS trust properties, while Capra considered it as the issue of forming human trust out of the three sources of trust information. Lastly, we introduce new design concepts of dynamic trust management and application-level trust optimization in response to changing conditions to maximize application performance, and demonstrate the feasibility with trust-based applications, by identifying the best way to form trust as well as use trust out of individual social and QoS trust properties at runtime to optimize application performance.

2.3.4 Trust Management in Internet of Things

The Internet of Things (IoT) refers to uniquely identifiable objects (things) and their virtual representations in an Internet-like structure. Security has drawn the attention in IoT research [42, 43, 158, 160, 200]. Roman *et al*. [160] discussed threats to IoT, such as compromising botnets trying to hinder services and the domino effect between intertwined services and user profiling. Traditional approaches to network security, data and privacy management, identity management, and fault tolerance will not accommodate the requirements of IoT due to lack of scalability and not being able to cope with a high variety of identity and relationship types [160]. Possible solutions were proposed to each security problem, but no specific protocol or analysis was given. Ren [158] proposed a compromise-resilient key management scheme for heterogeneous wireless IoT. The proposed key management protocol includes key agreement schemes and key evolution policies (forward and backward secure key evolution). The author also designed a quality of service (QoS) aware enhancement to the proposed scheme. However, the proposed scheme does not take social relationships among IoT identities into consideration. Chen and Helal [42] proposed a device-centric approach to enhance the safety of IoT. They designed a device description language (DDL) in which each device can specify its safety concerns, constraints, and knowledge. Nevertheless, their approach is specifically designed for sensor and actuator devices, and does not consider social relationships among device owners. Zhou and Chao [200] proposed a media-aware traffic security

architecture for IoT. The authors first designed a multimedia traffic classification, and then developed this media-aware traffic security architecture to achieve a good trade-off between system flexibility and efficiency. The limitation of their work is that they only considered direct observations to traffic without considering indirect recommendations.

There is little work on trust management in IoT environments for security enhancement, especially for dealing with misbehaving nodes who are legit members of a social IoT community. Chen *et al*. [43] proposed a trust management model based on fuzzy reputation for IoT. However, their trust management model considers a specific IoT environment consisting of only wireless sensors with QoS trust metrics like packet forwarding/delivery ratio and energy consumption. In our dissertation research, we consider multiple trust properties for community-based social IoT environments. For each trust property, we reveal the design tradeoff between trust convergence vs. trust fluctuation and identify the best design parameters for trust propagation and aggregation. Furthermore, we provide a method to identify the best formation to maximize the performance of trust-based applications in IoT systems.

2.4 Applications of Trust Management in Mobile Networks

The ultimate goal of our dissertation research is to design and validate a novel dynamic trust management protocol and to demonstrate its wide applicability to mobile networks including MANETs, WSNs, DTNs and IoT systems. Below we survey the state of the art of practical trust-based applications including misbehaving node detection, trust-based survivability management, trust-based secure routing, and trust-based service composition in these mobile networks and contrast our approach with existing approaches.

2.4.1 Trust-based Applications in Mobile Ad-Hoc Networks

Secure Routing

Trust management is often used in MANETs as an extension to existing routing protocols, such as AODV [150], DSR [92], etc. to deal with malicious and selfish nodes [16, 78, 136, 167, 175, 183, 201]. Sen *et al*. [167], Sun *et al*. [175], Crepeau *et al*. [64], and Balakrishnan *et al*. [16] used trust management as an add-on to DSR routing protocol design in MANETs. In these protocols, a node's trustworthiness is estimated by monitoring its routing behaviors or through an authentication mechanism [64]. A selfish or malicious node having a trust value lower than a threshold will be excluded from routing activity. Moe *et al*. [136] introduced an incentive mechanism into the trust-based DSR routing design in MANETs to enforce cooperation among nodes. They modeled the packet for-

warding process with a stochastic model and used hidden Markov models (HMMs) for estimating the probability that a node is selfish. In their model, nodes identified as being selfish with a high probability are black-listed and excluded from the routing activity. In addition, an identified selfish node may be still monitored and given a second chance to rejoin the network.

Trust management has also been used in AODV routing protocol design [78, 183]. Ghosh *et al*. [78] designed a secure routing protocol to find an end-to-end routing path free of malicious nodes. In their protocol, a node collaboratively works with its neighbors to detect misbehaving nodes and is able to deal with colluding malicious nodes. Wang *et al*. [183] developed a method to distinguish selfish nodes from cooperative nodes. In their approach, each node locally observes the AODV routing actions of its neighbors and builds up a statistical description of their behavior to estimate the trustworthiness of each neighbor.

All MANET secure routing protocols cited above considered only protocol-specific misbehavior as opposed to our dissertation research considering both (social) misbehavior and QoS trust properties. Protocol compliance depending on the specific routing protocol used (e.g., DSR vs. AODV) would be just a QoS metric. Furthermore, our dissertation research addresses the issue of application level optimization with trustee-dependent and mission-dependent design concepts.

Intrusion Detection

Trust management has also been used for intrusion detection in MANETs [71, 156, 184]. Wang *et al*. [184] proposed an intrusion detection mechanism based on trust for MANETs. They employed the concepts of evidence chain and trust fluctuation to evaluate a node in the network, with the evidence chain detecting misbehavior of a node, and the trust fluctuation reflecting the high variability of a node's trust value over a time window. Ebinger *et al*. [71] introduced a cooperative intrusion detection method for MANETs based on trust evaluation and reputation exchange. They split the reputation information into trust and confidence for reputation exchanges and then combine them into a trustworthiness parameter for intrusion detection. Researchers have also proposed to use trust management in MANETs for many other applications, such as *trust-based key management* [40, 82], *trust-based authentication* [145], and *trust-based access control* [2], etc. However, they only considered QoS routing performance or employed authentication as a mechanism to estimate trust without considering social connections or interactions among nodes as the criteria. Our dissertation research considers both social and QoS trust properties. Also, our trust management protocol can adapt to dynamically changing network environments (such as changing population of misbehaving nodes) to maximize intrusion detection performance.

2.4.2 Trust-based Applications in Delay Tolerant Networks

Secure Routing

In the literature, DTN routing protocols based on encounter patterns have been investigated [34, 89, 144]. However, if the predicted encounter does not happen, then messages would be lost for single-copy routing, or flooded for multi-copy routing. Moreover, these approaches could not guarantee reliable message delivery due to the presence of selfish or malicious nodes. The vulnerability of DTN routing to node selfishness was well studied by Karaliopoulos [104]. Several recent studies [169, 187, 202] considered using reputation in selecting message carriers among encountered nodes for DTNs. Nevertheless, [169, 202] assumed that a centralized entity exists for credit management, and [187] merely used reputation to judge if the system should switch from reputation-based routing to multipath routing when many selfish nodes exist.

Adversary Detection

Ayday *et al*. [13, 14] used trust management for adversary detection in DTNs. They assumed malicious nodes provide false recommendation (bad-mouthing and ballot-stuffing) to others in order to disrupt the reputation system. An iterative procedure was developed to detect these malicious nodes until the trust evaluation on each node reaches a stable state.

The trust management protocols used in these DTN applications do not consider social trust properties of the nodes in a DTN. Unlike prior work cited above, in this dissertation research, we integrate social trust and QoS trust into a composite trust metric to estimate a node's trustworthiness in DTNs based on the trustee and application characteristics and properties to maximize application performance.

2.4.3 Trust-based Applications in Wireless Sensor Networks

Trust-Based Geographic Routing

In the literature, several trust-based geographic routing protocols have been proposed in WSNs [120, 137]. In geographic routing [106], neighboring nodes exchange their location information to facilitate route discovery. Liu *et al*. [120] considered the possibility that a misbehaving node may falsify its location information and a malicious node can selectively drop packets. They developed a location verification algorithm to detect false location information and used overhearing techniques to check whether a node has actually forwarded a message in the correct direction. Once the misbehavior is detected, a node's trust level will be downgraded until reaching a threshold level below which it will be excluded from routing activity. Moraru *et al*. [137] also applied trust management to geographic routing in WSNs. Instead of considering malicious or selfish

nodes, their protocol aims to deal with obstacles or low local density areas in WSNs. In their protocol, each node considers several paths generated from different routing strategies and ranks them according to path performance. Gradually, the non-optimal paths with low performance will be ignored and optimal paths can be identified. The first work cited above essentially uses misbehavior as the metric, while the second work uses path performance as the metric. Our work considers both as a possibility and seeks the best way to form trust out of these possible social and QoS trust metrics to maximize geographic routing performance in WSNs.

Data Integrity, Data Privacy, and Malicious Node Detection

In WSNs, data integrity is vulnerable to both malicious nodes and faulty nodes. Ganeriwal *et al*. [76] used trust to ensure data integrity in WSNs. In their approach, a node rates the trust value of others by detecting whether their sensor readings are outliers. Finally, the trust value is used as criteria to fuse the data to ensure data integrity. Aivaloglou and Gritzalis [3] used trust to ensure in-network data privacy in WSNs. Their method exploits pre-deployment knowledge of the network topology and information flows in a WSN to assign role-based trust to nodes in the system and ensure that privacy data are disclosed only to trusted nodes. In the literature, there is little research on trust-based malicious node detection for WSNs. Existing work for malicious node detection was mostly based on anomaly detection techniques [154] to discover deviations from expected behaviors, including rule-based [25, 170], weighted summation [85], data clustering [122], Support Vector Machine (SVM) [155]. Moreover, rule-based and weighted summation techniques result in a high false positive probability especially when novel attacks appear. The effectiveness of data clustering and SVM based techniques hinges on the accuracy of the underlying algorithms achievable only through heavy learning and computation which may impede their use for real time operation in WSNs.

In the above cited work, either pre-deployment role-based knowledge or QoS performance metrics was used for trust evaluation in these WSN applications. Furthermore, neither approach is scalable to a large number of sensors. In this dissertation research, we develop a highly scalable dynamic hierarchical trust management protocol for WSNs, allowing application-specific social and QoS trust metrics to be incorporated into design to maximize application performance. We demonstrate its effectiveness by applying to trust-based geographic routing and malicious node detection, compared with existing approaches.

2.4.4 Trust-based Applications in Internet of Things

Trust-Based Service Composition

IoT as one of emerging computation and networking paradigms has attracted a variety of applications running on top of it, including e-health [32, 90], smart-home, and smart-community [115]. Trust management protocols designed for IoT systems can be applied to these novel applications to further improve the performance and enhance the security of IoT applications. In this dissertation research, we consider a service composition application scenario where nodes in IoT are service requesters and/or providers, to demonstrate the utilization of our proposed trust management protocol for IoT systems. We consider a community-based social IoT environment in the presence of misbehaving node including malicious and uncooperative nodes. We demonstrate the effectiveness of our dynamic trust management protocol by applying it to trust-based service composition and comparing the performance with baseline service composition approaches, including random service composition and ideal service composition.

Chapter 3

System Model

Our dynamic trust management protocol can be applied to a wide array of mobile networks, including MANETs, DTNs, WSNs, and IoT systems. System models for these mobile network systems will be stated in Chapter 6, Chapter 7, Chapter 8, and Chapter 9 respectively, when trust management in each specific network system is discussed. In this chapter, we discuss the attacker model, including both inside and outside attackers, to a mobile network and how we deal with these attacks in our protocol design. We also discuss the notion of social and QoS trust and how we measure social and QoS trust in mobile networks.

The most fundamental task of information security is to ensure *confidentiality*, *integrity*, *availability* and *authenticity* of an information system. This task becomes more challenging in mobile networks due to unreliable wireless transmission, a lack of trusted third party, constrained resources, and dynamic network environments. An outside attacker is one outside of the community. An inside attacker is one inside of the community who shares secret keys of the system. Usually, outsider attacks can be prevented using cryptographic techniques. In contrast, insider attacks are much harder to deal with and intrusion *detection* schemes are usually employed to identify insider attacks. In this dissertation research, we are primarily concerned with inside attackers although cryptography-based outside attacker detection mechanisms adopted by a mobile network can be used to derive evidence for trust evaluation. We consider the following attack models in mobile networks:

1. *Eavesdropping*. Eavesdropping is a passive reconnaissance activity which is hard to detect, and is often the preceding activity of many other attacks [102]. Due to the exposure of wireless transmission and multi-hop forwarding approach in mobile networks, an adversarial node in the transmission range can easily eavesdrop or intercept packets, hence leaking confidential information. We assume that cryptographic techniques such as encryption/decryption, authentication, and key management are used by the system to ensure confidentiality in the presence of eavesdropping activities. In addition, *adaptive power control* [102] and *smart an-*

tennas [110] techniques can reduce successful eavesdropping activities with low computation cost.

2. *Tampering*. An adversarial intermediate node in mobile networks can intercept a packet and forward a modified packet to the destination. For example, by changing the routing table information, an attacker can create a loop in the network. We assume that cryptography-based authentication mechanisms, such as digital signature and HMAC (keyed-hashing for message authentication) [109] are used by the system to detect tampering attacks, in order to achieve data integrity.
3. *Jamming*. Wireless communications are vulnerable to jamming attacks. An adversarial node can emit strong noise signals on the channel of legitimate nodes to disrupt the communication, leading to denial of service. We assume that spread spectrum techniques [138, 153], such as frequency hopping spread spectrum (FHSS), direct-sequence spread spectrum (DSSS), etc., are used by the system to detect jamming and provide jamming resistance.
4. *Packet drop attacks*. In a mobile network, a compromised node can report itself on a short path to the destination during a route discovery phase, but then drop the data packet it receives. Specifically, a compromised node can simply drop all packets to perform *black-hole attack* or adopt a more sophisticated strategy by selectively or randomly dropping packets (*gray-hole attack*). We use overhearing (a node overhears the transmission of next message carrier after forwarding message to it) and monitoring techniques to detect packet drop attacks.
5. *Sinkhole attacks*. In a sink-hole attack, packets in the network (or in a particular region) are attracted to a compromised node (sinkhole). Packets are then simply dropped, selectively forwarded to others, or used to reveal confidential information in the network. The success of a sinkhole attack relies on other collaborative attacks (e.g., wormhole attacks and routing information falsification) to lure the network traffic. We use overhearing and monitoring techniques to detect abnormal traffic toward a node as evidence against sinkhole attacks.
6. *Wormhole attacks*. In a wormhole attack, the attacker creates a tunnel between two distant nodes by intercepting the packet and replaying it at another location. Wormhole attacks can severely degrade the routing performance by deceiving the route discovery protocol, but are hard to prevent or detect using traditional cryptographic techniques (the attacker can simply record the transmission at one place and replay it at another place without knowing the context). In order to detect wormhole attacks in mobile networks, we assume that the system will attach geo-location or temporal information to packets for verification [86, 105], or construct a virtual topology of the network and detect the anomaly [152, 185].

7. *Flood attacks*. Similar to the jamming attack at physical layer, a compromised node can launch flood attack at the network or higher layer by propagating bogus messages in the network. Flood attack is a type of Denial of Service (DoS) attacks and can block the communication between normal nodes, and even deplete network resources. We employ monitoring and snooping techniques to detect flood attacks by analyzing the activity and traffic flow of each neighbor node.
8. *Routing information falsification*. Because of the distributed nature of mobile networks, route discovery is usually based on the path report from intermediate nodes, and thus is vulnerable to routing information falsification. We assume that the system uses certificate-based techniques for validating routing information. For example, *encounter tickets* [112] can be used as the proof of encounter events in DTNs.
9. *ACK counterfeiting*. A compromised node in mobile networks can send a fake acknowledgement to the source and other nodes such that the message is discarded by all nodes before delivery. We assume that an end-to-end acknowledgment signed by the destination node is used to detect such attacks.

We assume that these counterattack techniques or detection mechanisms employed by the system provide evidence as input to our dynamic trust management protocol for trust evaluation for each social or QoS trust property. These detection techniques also provide information about network environments to our dynamic trust management protocol for adjusting parameter settings such that the application-level performance can be maximized. Here, we note that some detection mechanisms require cryptography techniques for authentication. For MANETs, DTNs, and IoT systems, we assume that the system uses a *pre-generated private-public key pair pool* to support key management. Each new node joining the network is assigned an unused key pair in the pool and stores all public keys. For WSNs, a pairwise key is pre-generated between two nodes within a k-hop range at the system deployment time.

Since we consider indirect recommendations in addition to direct observations, dynamic trust management itself also faces many attacks. We consider the following trust-related attack models:

10. *Sybil attacks*. In a Sybil attack, the adversarial node creates a large number of *pseudo* entities. Then by launching Byzantine attacks [111, 193], these pseudo entities can disrupt the trust management system. We assume that the system uses authentication techniques to detect Sybil attacks.
11. *Slandering attacks* (*good-mouthing* and *bad-mouthing/ballot stuffing*). A compromised node could recommend a bad node as a good node with a high trust value (good-mouthing) and recommend a good node as a bad node with a low trust

value (bad-mouthing/ballot stuffing). Also, a compromised node could perform random attacks to evade detection. Our dynamic trust management protocol is resilient to slandering attacks since in our trust aggregation protocol, a recommendation is filtered out if the recommender's trust is below the recommender trust threshold and, if it passes the trust threshold, is weighted by the trust value of the recommender (*referral trust*) [98]. Therefore, the recommendation provided by a compromised node will be discounted or discarded. In addition, we apply outlier detection techniques on recommendations to counter slandering attacks.

12. *White-washing/self-promoting*. A compromised node may recommend itself as a trustworthy node to raise its trust value. Our dynamic trust management protocol is resilient to white-washing attacks because we only consider recommendations provided by a third party node and a node never has a chance to do self-promotion.

We consider a mobile network with a large number of nodes exhibiting heterogeneous social behaviors and QoS characteristics. To take into account of both aspects, our dynamic trust management maintains a composite trust metric to evaluate the trust of each node by integrating social trust and QoS trust. Social trust properties are derived from social network to reflect social relationship among nodes. QoS trust properties are derived from communication network to reflect the capability or competence to accomplish a task.

In a mobile network, a node could exhibit social behaviors in many ways, such as *mobility patterns*, *interaction patterns*, and *misbehaving models*. Multiple social trust metrics can be obtained from mobility or encounter patterns (e.g., *closeness*, *centrality*, *betweenness*, and *similarity*) used to improve the performance of a mobile network [65]. In this dissertation research, we consider both synthetic mobility patterns (e.g., random waypoint mobility and group mobility) [77, 87] and real mobility from trace data [39, 69]. Mobility patterns can derive physical proximity among nodes. However, nodes physically close to each other may not necessarily socially close to each other. A social trust metric like *closeness* or *intimacy* may be obtained by analyzing the interaction patterns and responses or using a social connection graph as an input [26]. A socially *selfish* node may respond to routing requests only if it is close to the source or destination node. A social trust metric like honesty may be obtained by analyzing the dishonesty model and designing specific detection mechanisms to gather evidence for trust assessment.

Very frequently nodes in a mobile network are small portable devices with limited resources in terms of *energy*, *buffer size*, *bandwidth*, and *computation capability*. These resources are critical to the QoS performance in mobile networks. QoS trust properties represent the availability of these resources and reflect whether a node is competent to accomplish an assigned task (e.g., forwarding a packet). For example, a node may be-

have *uncooperatively* in protocol execution if its energy level is low or its buffer is near full to conserve resources to serve urgent packets. A node with a low bandwidth or a low processing speed may also incur a long delay in message delivery. Most QoS trust values can be estimated using snooping and monitoring techniques.

Whether or not a social or QoS trust property is selected depends on application requirements and network characteristics. In DTN secure routing applications, for example, "*social connectivity*" is considered an important social trust metric. In WSN query applications, "*energy*" is considered an important QoS trust metric. Each trust property *X* (social or QoS) is evaluated by *direct trust evaluation* and *indirect trust evaluation*. For direct trust evaluation, a detection mechanism is required to rate a node's trust level in trust property *X*. For example, we assess the "social connectivity" property between two nodes by analyzing their encounter pattern in a DTN. We apply honesty detection techniques to assess the "honesty" property. We monitor transmission activities of a node to assess the node's "energy" property in a WSN [131]. For indirect trust evaluation, each node propagates its trust evaluation as recommendations to help trust convergence. We say that detection mechanism designs for direct trust evaluation, and trust propagation/aggregation protocol designs for indirect trust evaluation of each trust property *X* are *validated* when the status of trust property *X* assessed as a result of applying these protocols is close to actual status or ground truth.

Chapter 4

Design Principles

Recognizing that nodes in a mobile network very likely will involve human operators controlling communication devices (e.g., device-carried soldiers, and vehicles operated by human operations), we propose to explore, compose and measure trust in a way humans estimate with their cognition, e.g., competence is about task performance, intimacy is about comfortableness of having close nodes in the same mission, and honesty is about integrity rather than competence, to properly characterize trust for mobile network applications.

We design trust management for mobile networks to be dynamically reconfigurable and capable of adjusting trust parameters for protocol execution in response to dynamically changing environments (e.g., in response to increasing misbehaving node populations or evolving node density because of node failure, eviction, mobility or disconnection/reconnection) to maximize application performance. Finally we design a modeling and analysis tool to facilitate application of dynamic trust management in mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), delay tolerate networks (DTNs), and Internet of Things (IoT) systems in which mobile nodes collaborate to accomplish a mission despite the presence of malicious, erroneous, partly trusted, uncertain and incomplete information.

Our research on dynamic trust management has two major parts: *design* and *validation*. The design part addresses all core functions of trust management, namely, *trust composition*, *trust aggregation*, *trust propagation*, and *trust formation*. The learning process and adaptive design of dynamic trust management are reflected in trust aggregation, propagation, and formation. In addition, for an application built on top of trust management, e.g., misbehaving node detection, there is an *application-level trust optimization* component in using *trust* to classify nodes to maximize application performance or to satisfy application requirements. Finally, our dynamic trust management is designed to be *resilient* to trust-related attacks. Below we describe dynamic trust management design principles developed in this dissertation research.

4.1 Trust Composition

Taking into consideration that communication devices in future mobile networks may be carried mostly by human operators, our trust protocol design incorporates both social trust properties deriving from *social networks* [65, 125, 192] in addition to the conventional QoS trust properties deriving from *communication networks*. *Social trust* includes honesty, intimacy, selfishness, betweenness centrality, and social reputation. A mobile network would consist of heterogeneous mobile devices carried by soldiers, robotic vehicles, or ground vehicles operated by humans. Therefore, unlike traditional network research, *social trust* must be considered between these mobile agents. We use social networks to evaluate the *social trust* value of a node in terms of the degree of personal or social trends, rather than the *capability* of executing a mission based on past collaborative interactions. The latter belongs to *QoS trust* by which a node is judged if it is capable of completing an assigned mission as evaluated by *communication networks*. More specifically, *QoS trust* represents competence, dependability, reliability, successful experiences, and reputation or positive recommendations on task performance forwarded from direct or indirect interactions with others. We use the term *QoS trust* [61] to refer to trust evaluation in terms of task performance capability. We aim to design dynamic trust management to allow a variety of social and QoS trust metrics to be explored and tested for their effectiveness.

4.2 Trust Aggregation

For each social and QoS trust property explored, calling it X for notational convenience, we devise and validate a trust aggregation protocol to be executed by a trustor node at runtime to compute its trust toward a trustee node at time t, with the design goal that the trust value computed is close to *actual status* of the trustee node in X. This is achieved by means of a novel modeling and analysis methodology which provides a node's actual status at time t (this is the validation part). For the trust aggregation protocol for X, in addition to using both direct observations and indirect recommendations for trust aggregation, we incorporate learning and adaptive designs to react to changing environment conditions. For example, when nodes in a mobile network travel in high speed because they move to an unconstrained terrain, it would be ineffective to do direct observation based trust evaluation because of a very short contact time. The trust aggregation protocol can then dynamically decrease the weight associated with direct trust and contrarily increase the weight associated with indirect trust to gain a more accurate assessment of trust properties. Another novelty is that we explore the concept of *credential-based trust* (certified by authority) into our trust aggregation protocol design to more accurately assess certain trust properties. For example, when X=intimacy, the trust aggregation protocol could explore a prior knowledge of social friendship so that the

trustor node can more accurately assess the intimacy trust property of the trustee node. When *X*=connectivity in the context of DTN encounter-based routing, the trust aggregation protocol could explore authenticated encounter tickets as credential to verify the past encounter history of a trustee node so that the trustor node can more accurately assess the connectivity trust property of the trustee node. Our goal is to identify the best aggregation parameter setting for each trust property *X* such that the trust value computed is close to *actual status* of the trustee node in property *X*.

In the design of trust aggregation for dynamic trust management, we differentiate referral trust from functional trust [98]. When a recommender node, say, node *m*, provides its recommendation to node *i* for evaluating node *j*, node *i*'s referral trust on node *m* is multiplied with node *m*'s functional trust on node *j* to yield node *m*'s recommending trust value toward node *j* to account for trust decay in space. In general a node can have fairly accurate trust assessments toward its 1-hop neighbors within the same subtask group utilizing monitoring, overhearing and snooping techniques. For a node more than 1-hop away, a node will refer to a set of recommenders for trust assessment toward the remote node. Our trust aggregation of indirect recommendations is a form of weighted calculation based on trustworthiness (i.e. the referral trust value) of the recommender.

4.3 Trust Propagation

We propose a new design concept of *threshold-based trust propagation.* The basic idea is to enhance Quality of Information (QoI) by only allowing trustworthy trust propagation in the network. First, only trustworthy nodes can receive information. Second, only trustworthy information can be propagated in the network. This can be achieved if (1) the forwarding node is trustworthy; (2) the information received is trustworthy by checking the source of the information. We investigate two trust thresholds: a message forwarding trust threshold (FTT) specifying the minimum trust threshold above which a trustee node is considered trustworthy to forward a message to, and a recommender trust threshold (RTT) specifying the minimum trust threshold above which a trustee node is considered trustworthy as a recommender. Our dynamic trust management is made adaptive by adjusting these two thresholds in response to changing environment conditions (e.g., an increasing population of misbehaving nodes or an evolving node density) to maximize application performance.

In the design of trust aggregation and propagation protocols, there are two ways to update trust: *periodic update* and *event trigger*. The event trigger method is often selected for opportunistic network environments because of sparse connections. For example in delay tolerant networks, trust update is triggered by encounter events. In network environments where nodes can continuously monitor others, trust update can be performed

periodically. The trust update rate can greatly affect protocol performance. If the trust update rate is too low, some events happening between two trust update time points might not be captured in time, which results in a low trust convergence rate in response to dynamic network environment changes. If the trust update rate is too high, it may consume too much network resources (e.g., energy and storage) and results in a reduced network lifetime.

4.4 Trust Formation

Much of the learning process and adaptive design is incorporated in trust formation. After we validate that the measurement of trust property *X* obtained from executing its trust aggregation protocol is close to *actual status*, we investigate how individual trust properties may be combined to form an overall trust measure, properly reflecting the belief of a trustor node toward a trustee node in accomplishing a mission or satisfying an application requirement. Untreated in the literature, we explore various trust formation models including importance-weighted sum and nonlinear form to model the interplay relationship between QoS and social trust properties. Furthermore, we design and validate a trust formation protocol capable of learning from experiences and adapting to changing environment conditions by first selecting the most effective trust formation model, and then adjusting model parameter values of the selected trust formation model. An example is that in response to an increasing misbehaving node population, the protocol adaptively increases the weight associated with honesty (a social trust property) such that the overall trust formed can optimize the application performance (e.g., minimizing the false alarm probability for the misbehaving node detection application) or best satisfy the application requirement (e.g., satisfying the packet delivery ratio requirement for the secure routing application).

4.5 Application-Level Trust Optimization

We propose the design concept of *application-level trust optimization* allowing an application to optimize the use of trust to classify nodes to maximize application performance. We investigate three applications built on top of dynamic trust management to demonstrate the validity of the design. (1) For the *misbehaving node detection* application, we investigate an optimal application-level drop-dead trust threshold, say, T^{th}, below which a node is considered as misbehaving. By means of *runtime trust evaluation*, the dynamic trust management is made adaptive by adjusting the drop-dead trust threshold in response to changing environment conditions (e.g., an increasing population of misbehaving nodes) to minimize the false positive and false negative probabilities. (2) For the *survivability management* application, we identify the best minimum trust level required for successful mission completion and the drop dead trust level to maximize the system

reliability of mission execution with dynamic team membership. (3) For the *secure routing* application, we dynamically control a forwarding node trust threshold (FTT) and a recommender trust threshold (RTT) to classify nodes to optimize application requirements such as message delivery ratio and message delay.

4.6 Resiliency to Attacks

A malicious node may perform various attacks to disrupt the operation of a mission. Our main goal is to make dynamic trust management resilient to *trust-related attacks*, including bad-mouthing attacks (i.e., bad-mouthing a good node as a bad node), good-mouthing attacks (i.e., good-mouthing a bad node as a good node), and whitewashing attacks (i.e., reporting false information about itself to improve its trust status). Except for credential-based trust, our dynamic trust management is based on monitoring, snooping and overhearing for direct trust evaluation. It does not take information passed to it from a neighbor node in its direct trust evaluation process toward the neighbor node, so it is resilient to whitewashing attacks. We investigate two designs for protocol resiliency against good-mouthing or bad-mouthing attacks. One design is that indirect recommendations will be weighted by the recommender's referral trust. Thus, if a bad node (while performing a good-mouthing attack) provides a good recommendation about a bad node, the good recommendation will be discounted by the recommender's bad referral trust. The second design is that only trustworthy nodes will be used as recommenders. This is achieved by using a recommender trust threshold (RTT) in trust propagation design.

We demonstrate our protocol's resiliency against good-mouthing and bad-mouthing attacks by malicious nodes in two ways. One way is to formally prove protocol resiliency. This can be done by proving that a bad node remains as a bad node despite good-mouthing attacks, and a good node remains as a good node despite bad-mouthing attacks. Specifically, we prove the accuracy, convergence and resiliency properties of our protocol are preserved such that the peer-to-peer trust evaluation performed by a trustor toward a trustee based on our protocol converges to a trust value that deviates from ground truth status by a bounded error margin determined by environment noises and random attack behavior, despite the presence of good-mouthing and bad-mouthing attacks performed by malicious nodes. Another way is to perform model-based analysis with simulation validation. In this dissertation research, we build a mathematical model to describe the actual behavior of nodes in a mobile network in the presence of malicious and selfish nodes, and then quantitatively demonstrate with simulation validation that subject trust evaluation results obtained from dynamic trust management are close to objective evaluation results obtained from actual knowledge based on the mathemat-

ical model built despite bad-mouthing and good-mouthing attacks performed by malicious nodes in the system.

4.7 Dynamic Trust Management

Rapidly changing environments and node behaviors in mobile networks call for an adaptive design. Our notion of *dynamic trust management* is to identify and apply the best parameter settings at runtime in response to dynamically changing network conditions to minimize trust bias and to maximize application performance. At design time, we identify the best parameter settings, given an operational profile describing the network environment (e.g., the population of misbehaving nodes, mobility patterns, etc.) as input. Then, at runtime, each node after sensing the network environmental condition (e.g., estimating the percentage of active misbehaving nodes based its trust evaluation) applies the best protocol settings. Such dynamic trust management design is reflected in trust aggregation, trust propagation, trust formation, and application-level trust optimization. Specifically, the dynamic trust aggregation and propagation protocols identify and apply the best parameter settings, including weights of direct trust vs. indirect recommendations, trust decay factor, and recommender trust threshold to minimize the trust bias. The dynamic trust formation and application-level trust optimization protocols identify and apply the best parameter settings, including weights of individual trust components forming overall trust metric, and application-level trust thresholds such as the trust threshold for the selection of the next message carrier for secure routing to maximize application performance. Alternatively, heuristic search methods can be applied to derive the best trust parameter settings at runtime. This can be achieved by first formulating the trust estimation or trust-based application as an optimization problem, then using heuristic search methods to efficiently obtain optimal solutions at runtime based on past observations.

Chapter 5

Design Validation

We validate *dynamic trust management* designs by two ways. The first way is to formally prove our protocols preserve convergence, accuracy, and resiliency properties despite the presence of misbehaving nodes. We develop formal proof for specific systems in Chapters 6-9. The second way is perform model-based analysis with extensive simulation validation.

5.1 Model-Based Analysis

We develop a novel model-based analysis methodology based on a mathematical model comprising continuous-time semi-Markov stochastic processes (for which the event time may follow any general distribution) to define a mobile network consisting of a large number of mobile nodes designed to achieve missions in the presence of malicious, erroneous, partly trusted, uncertain and incomplete information in heterogeneous mobile environments. Each node in the model exhibits heterogeneous social and QoS behaviors characterized by its mobility model (synthetic or real trace), compromise rate, selfish rate, hardware/software failure rate, initial energy, energy consumption rate.

We take the concept of "operational profiles" in software reliability engineering [141] as we build the mathematical model. An operational profile is what the system expects to see during its operational phase. During the testing and debugging phase, a system would be tested with its anticipated operational profile to reveal design faults. Failures are detected and design faults causing system failures are removed to improve the system reliability. Our mathematical model is built with a system's anticipated operational profile given as input. Typically this would include knowledge regarding (a) hostility such as the attack types and random attack probabilities, and the compromise rates providing information of how often nodes may be compromised and performing attacks; (b) mobility providing information of how often nodes meet and interact with each other; (c) behavior specifications providing information regarding secure and inse-

cure behaviors during protocol execution; (d) resources such as node energy providing information of how fast resources are consumed; (e) mission processes providing information of how fast missions arrive and depart, and what constitute acceptable system responses; and (f) system failure definitions including security failure conditions.

Once the mathematical model is developed and proven to faithfully describe the behaviors of nodes in a mobile network (validated through simulation), we can use the model output to provide a global view of the system, which serves as the basis for ***objective trust*** *evaluation*. For example, we will know exactly if a node is compromised at time *t* based on global knowledge. This is vastly different from ***subjective trust*** evaluation based on local and referral information obtained at time *t*. We compare *objective trust* against *subjective trust* as the basis for iteratively fine-tuning the algorithm design for dynamic trust management so that subjective trust obtained as a result of executing dynamic trust management protocols is close to objective trust (i.e., actual status).

5.2 Static-Followed-by-Dynamic Testing Strategy

We test *dynamic trust management* designs by a static-followed-by-dynamic (SFD) testing strategy described as follows. First we identify the best trust management protocol setting, when given as input a set of environment variable values defining a particular state of the operational environment. An example of an environment variable would be the percentage of misbehaving nodes in a mobile network. Another example of an environment variable is the average amount of contact time between two nodes.

By the best trust management protocol setting, we mean the best set of parameter values being used by dynamic trust management in trust aggregation (e.g., weights of direct trust vs. indirect recommendations vs. credential-based trust), trust formation (e.g., weights on social and QoS trust properties if any for forming trust), and optimal application-level trust settings (e.g., the drop-dead minimum threshold for the misbehaving node detection application) for achieving the best application performance. Such best protocol settings are identified and recorded in a table at static time using the modeling and analysis methodology developed. Then at runtime after learning environment condition changes (e.g., an increasing population of misbehaving nodes), dynamic trust management adapts to environment changes by applying the best trust management protocol setting by means of table lookup or extrapolation techniques.

The capability of dynamic trust management is then tested by setting up an operational environment with a number of environment variables changing their values dynamically as time progresses (e.g., a positive compromise rate for a node initially behaving at t=0). The SFD testing process is completed if we can demonstrate that dynamic trust management (changing the protocol setting dynamically) significantly outper-

forms static trust management (without changing the protocol setting dynamically in response to changing environment conditions).

5.3 Simulation Validation

We develop simulation programs based on network simulator ns-3 [1] for simulation of mobile nodes in mobile networks utilizing Virginia Tech's HokieSpeed supercomputer. The simulation code for each mobile node can mimic its trust and security behavior in the system. The system is setup according to the system model for four specific mobile networks (i.e., MANETs, DTNs, WSNs, IoT systems) and each node executes dynamic trust management for subjective trust evaluation, as mobile nodes must collaborate to execute missions with various levels of mission difficulty in mobile network environments.

The simulation also serves as a tool for verifying and validating trust aggregation protocol designs. That is, a trust aggregation protocol designed for computing the trust value for each trust property is incorporated into simulation code and trust value obtained from simulation are compared against the actual status of trustee at a particular time instant to assess the effectiveness of the trust aggregation protocol designed. If it does not pass the acceptance test (i.e., the discrepancy exceeds a threshold error percentage), the trust aggregation protocol will be redesigned and reevaluated until it passes the acceptance test. Another important function of simulation is that we can test the effect of the *node mobility model* reflecting how and when a node moves (random vs. group-based vs. social networking based vs. traces) on the robustness of dynamic trust management protocol designs. Lastly, based on the simulation results in comparison with analytical results, we iteratively refine dynamic trust management protocol designs for subjective trust evaluation so that subjective trust is close to objective trust for practical and effective mobile network trust management for mission critical applications.

Chapter 6

Dynamic Trust Management for Mobile Ad-hoc Networks and Its Applications

In this chapter we apply design and validation principles of dynamic trust management for managing mobile groups in mobile ad hoc networks (MANETs). We demonstrate the effectiveness of the composite social and QoS trust management protocol for mission-oriented mobile groups in MANETs for critical mission executions. We develop a novel model-based approach to identify the best protocol setting under which peer-to-peer *subjective trust* as a result of executing our distributed trust management protocol is accurate with respect to ground truth status over a wide range of operational and environment conditions with high resiliency to malicious attacks and misbehaving nodes. Furthermore, using mission-oriented mobile groups as an application, we identify the best trust formation model under which the application performance in terms of the system reliability of mission-oriented mobile groups in MANET environments is maximized.

6.1 System Model

6.1.1 Operational Profile

Recall that (see Section 5.2) an operational profile specifies the anticipated operational and environment conditions. An operational profile for MANETs in this chapter provides knowledge regarding environment hostility, node mobility, node behavior, environment resources, and system failure definitions. Later in Section 6.4 we will exemplify the input operational profile for a mobile group application in MANET environments.

6.1.2 Problem Definition and Desirable Output

Our dynamic trust management for MANETs is distributed in nature and is run by each mobile node to subjectively yet informatively assess the trust levels of other mobile

nodes. Further, our dynamic trust management protocol is resilient against misbehaving nodes. Given the operational profile as input covering a wide range of operational and environment conditions, we aim to solve two problems:

- Discover and apply the best trust aggregation protocol setting of dynamic trust management to make "subjective trust" accurate compared with "objective trust" despite the presence of misbehaving nodes. The desirable output is to achieve high accuracy in peer-to-peer subjective trust evaluation with high resiliency to malicious attacks.
- Discover and apply the best trust formation to maximize application performance. For the mission-oriented mobile group application, the desirable output is to maximize the system reliability given a system failure definition.

6.1.3 Node Behavior Assumptions

Node behavior is part of the operational profile. While our model-based analysis technique is generally applicable to any node behavior specification, for illustration we consider the following node behaviors:

- Every node conserves its resources (e.g., energy) as long as it does not jeopardize the global welfare (i.e., successful mission execution). Thus, when a node senses that it is surrounded by many uncooperative 1-hop neighbors, it will tend to become cooperative to ensure successfully mission execution. On the other hand, a node with many cooperative 1-hop neighbors around will tend to become uncooperative to conserve its resources, knowing that this will not jeopardize the global welfare.
- Every node has a different level of energy, speed and vulnerability reflecting node heterogeneity. The energy consumption rate of a node depends on its status. If a node is uncooperative, the speed of energy consumption is slowed down since an uncooperative node will not follow protocol execution. If a node becomes compromised, the speed of energy consumption increases since a compromised node will perform attacks which consume energy. A node's vulnerability is reflected by a compromised rate, e.g., a capture by attackers after which the node is compromised.
- A compromised node may perform slandering attacks, (e.g., good-mouthing bad nodes and bad-mouthing good nodes), identity attacks (e.g., Sybil) or Denial-of-Service (DoS) attacks (e.g., consuming resources unnecessarily by disseminating bogus packets). We assume that a compromised node will always perform attacks on good nodes and does not discriminate good nodes when performing attacks.

6.1.4 Mission-Oriented Mobile Groups

As an application of our dynamic trust management for MANETs, we apply it to mission-oriented mobile groups. A mission-oriented mobile group consists of a number of mobile nodes cooperating to complete a mission, with one or more being the commander nodes of the group. Nodes are allowed to join or leave the mobile group depending on the application requirements. Upon every membership change due to join or leave, individual rekeying (meaning the rekey operation is performed immediately) will be performed based on a distributed key agreement protocol such as the Group Diffie-Hellman (GDH) protocol [173]. Our dynamic trust management aims to increase the probability of successful mission execution by a mobile group in MANET environments [62]. For mission-critical applications, it is frequently required that nodes on a mission must have a minimum degree of trust for the mission to have a reasonable chance of success. On one hand, a mission may require a sufficient number of nodes to collaborate. On the other hand, the trust relationship may fade away between nodes both temporarily and spatially. Our dynamic trust management equips each node with the ability to subjectively assess the trust levels of other nodes in the system and thus upon a mission assignment allows the node to select highly trustworthy nodes for collaboration to maximize the probability of successful mission completion.

6.2 Protocol Design

In this section we first describe our trust protocol to be executed by every node at runtime. Then we discuss its application to reliability assessment of a mobile group in MANET environments.

6.2.1 Trust Composition

A node with a very low trust value is of little value to the system and depending on the application requirement may be evicted to prevent it from performing attacks to damage the system functionality. A node's trust value is assessed based on evidences such as direct observations as well as indirect recommendations. Our trust model is evidence-based. Thus we do not consider dispositional belief or cognitive characteristics of an entity in deriving trust. The trust assessment of one node toward another node is updated periodically.

Our trust metric consists of two trust types: *social trust* and *QoS trust*. Social trust is evaluated through interaction experiences in social networks to account for social relationships. Note that this work concerns mobile devices carried by human users as part of a social network. Among the many social trust metrics such as friendship, honesty, privacy, similarity, betweenness centrality, and social ties [65], we select social ties (measured by ***intimacy***) and honesty (measured by ***healthiness***) to measure the social trust level of a node as these social properties are considered critical for trustworthy

mission execution in group settings. *QoS trust* is evaluated through the communication and information networks by the *capability* of a node to complete a mission assigned. Among the many QoS metrics such as competence, cooperation, reliability, and task performance, we select competence (measured by ***energy***) and protocol compliance (measured by ***cooperativeness*** in protocol execution) to measure the QoS trust level of a node since competence and cooperation are considered the most critical QoS trust properties for mission execution in group settings. Quantitatively, let a node's trust level toward another node be a real number in the range of [0, 1], with 1 indicating complete trust, 0.5 ignorance, and 0 complete distrust. Let a node's trust level toward another node's particular trust component also be in the range of [0, 1] with the same physical meaning.

The rationale of selecting these social and QoS trust metrics is given as follows. The intimacy component (for measuring social ties) has a lot to do with if two nodes have a lot of direct or indirect interaction experiences with each other, for example, for packet routing and forwarding. The healthiness component (for measuring honesty) is essentially a belief of whether a node is malicious or not. We relate it to the probability that a node is not compromised. The energy component refers to the residual energy of a node, and for a MANET environment, energy is directly related to the survivability capability of a node to be able to execute a task completely, particularly when the current and future missions may require a long mission execution time. Finally, the cooperativeness component of a node is related to whether the node is cooperative in routing and forwarding packets. For mobile groups, we relate it to the trust to a node being able to faithfully follow the prescribed protocol such as relaying and responding to group communication packets.

Other than the healthiness trust component, we assert that a node can have fairly accurate trust assessments toward its 1-hop neighbors utilizing monitoring, overhearing and snooping techniques. For example, a node can monitor interaction experiences with a target node within radio range, and can overhear the transmission power and packet forwarding activities performed by the target node over a trust evaluation window Δt to assess the target node's energy and cooperativeness status. For a target node more than 1-hop away, a node will refer to a set of recommenders for its trust toward the remote target node.

6.2.2 Design against Slandering Attacks

Our dynamic trust management is resilient to good-mouthing and bad-mouthing attacks by two recommender selection criteria: (a) *threshold-based filtering* by which only trustworthy recommenders with trust higher than a minimum trust threshold are qualified as recommenders; and (b) *relevance-based trust* by which only recommenders with

high trust in trust component *X* are qualified as recommenders to provide recommendations about a trustee's trust component *X*.

6.2.3 Trust Protocol Description

The trust value of node *j* as evaluated by node *i* at time *t*, denoted as $T_{i,j}(t)$, is computed by node *i* as a weighted average of intimacy, healthiness, energy, and cooperativeness trust components. The assessment is done periodically in every Δt interval. Specifically node *i* will compute $T_{i,j}(t)$ by:

$$T_{i,j}(t) = w_1\, T_{i,j}^{intimacy}(t) + w_2\, T_{i,j}^{healthiness}(t) + w_3\, T_{i,j}^{energy}(t) + w_4\, T_{i,j}^{cooperativeness}(t) \tag{6.1}$$

where $T_{i,j}^{intimacy}(t)$, $T_{i,j}^{healthiness}(t)$, $T_{i,j}^{energy}(t)$ and $T_{i,j}^{cooperativeness}(t)$ are the trust beliefs of node *i* toward node *j* in intimacy, healthiness, energy and cooperativeness trust components, respectively, and $w_1 : w_2 : w_3 : w_4$ is the weight ratio for weighing intimacy: healthiness: energy: cooperativeness with $w_1 + w_2 + w_3 + w_4 = 1$.

Node *i* evaluates node *j* dynamically at time *t* by direct observations and indirect recommendations. Direct observations are direct evidences collected by node *i* toward node *j* over the time interval $[t - d\Delta t, t]$ when node *i* and node *j* are 1-hop neighbors at time *t*. Here Δt is the trust update interval and *d* is a design parameter specifying the extent to which recent interaction experiences would contribute to intimacy. We can go back as far as *t*=0, that is, $d = t/\Delta t$, if all interaction experiences are considered equally important. Indirect recommendations, on the other hand, are indirect evidences given to node *i* by a set of recommenders node *i* trusts most. Specifically, node *i* will compute $T_{i,j}^{X}(t)$ where *X* is a trust component in Equation (6.1) by:

$$T_{i,j}^{X}(t) = \beta_1\, T_{i,j}^{direct,\ X}(t) + \beta_2\, T_{i,j}^{indirect,\ X}(t) \tag{6.2}$$

In Equation (6.2), β_1 is a weight parameter to weigh node *i*'s own information toward node *j* at time *t*, i.e., "direct observations" or "self-information" and β_2 is a weight parameter to weigh indirect information from recommenders, i.e., "information from others," with $\beta_1 + \beta_2 = 1$ with the expectation that $\beta_1 > \beta_2$ because a node tends to trust its own direct observations more than indirect recommendations.

The direct trust part, $T_{i,j}^{direct,\ X}(t)$, in Equation (6.2) is evaluated by node *i* at time *t* depending on if node *i* is a 1-hop neighbor of node *j* at time *t*. If yes, node *i* uses its direct observations toward node *j* during $[t - d\Delta t, t]$ to update $T_{i,j}^{direct,\ X}(t)$ where Δt is the periodic trust evaluation interval. Otherwise, it uses its old direct trust assessment

at time $t-\Delta t$ multiplied with $e^{-\lambda_d \Delta t}$ (for exponential trust decay over time) to update $T_{i,j}^{direct,\ X}(t)$. Specifically, node *i* will compute $T_{i,j}^{direct,\ X}(t)$ by:

$$T_{i,j}^{direct,\ X}(t) = \begin{cases} T_{i,j}^{1-hop,\ X}(t) & \text{if } i \text{ is a neighbor to } j \text{ at } t \\ e^{-\lambda_d \Delta t} \times T_{i,j}^{direct,X}(t-\Delta t) & \text{otherwise} \end{cases} \quad (6.3)$$

To account for trust decay over time, we adopt an exponential time decay factor, $e^{-\lambda_d \Delta t}$, to satisfy the desirable property that trust decay must be invariable to the trust update frequency [96]. Depending on the trust evaluation interval Δt, we can fine tune the value of λ_d to test the effect of trust decay over time. The notation $T_{i,j}^{1-hop,\ X}(t)$ here refers to the new "direct" trust assessment at time *t*. Below we describe specific detection mechanisms by which node *i* collects direct observations to assess $T_{i,j}^{1-hop,\ X}(t)$ for the case in which *i* and *j* are 1-hop neighbors at time *t*.

- $T_{i,j}^{1-hop,\ intimacy}(t)$: This refers to the new assessment of node *i*'s direct interaction experience toward node *j*. It is computed by node *i* by the ratio of the amount of time nodes *i* and *j* are 1-hop neighbors directly interacting with each other during $[t-d\Delta t, t]$
- $T_{i,j}^{1-hop,\ healthiness}(t)$: This refers to the belief of node *i* that node *j* is healthy based on node *i*'s direct observations during $[\ t-d\Delta t, t\ ]$. Node *i* estimates $T_{i,j}^{1-hop,\ healthiness}(t)$ by the ratio of the number of suspicious interaction experiences observed during $[t-d\Delta t, t]$ to a system "healthiness" threshold to reduce false positives. Node *i* uses a set of anomaly detection rules including the interval rule (for detecting node *j*'s sending bogus messages), the retransmission rule (for detecting node *j*'s dropping messages), the integrity rule (for detecting node *j*'s modifying messages), the repetition/jamming rule (for detecting node *j*'s performing DOS attacks), and the delay rule (for detecting node *j*'s delaying message transmission) as in [170] to keep a count of suspicious experiences of node *j* during $[t-d\Delta t, t]$. If the count exceeds the "healthiness" threshold, node *i* considers node *j* as totally unhealthy, i.e., $T_{i,j}^{1-hop,\ healthiness}(t)$=0. Otherwise it is equal to 1 minus the ratio. We model the deficiencies in anomaly detection (e.g., imperfection of rules) by a false negative probability (P_{fn}^{H}) of misidentifying an unhealthy node as a healthy node, and a false positive probability (P_{fp}^{H}) of misidentifying a healthy node as an unhealthy node.
- $T_{i,j}^{1-hop,\ energy}(t)$: This refers to the belief of node *i* that node *j*'s energy is adequate and hence is competent at time *t*. Node *i* overhears node *j*'s packet transmission activities during $[t-d\Delta t, t]$ utilizing an energy consumption model [74]

to first compute the amount of energy consumed by node *j* during $[t - d\Delta t, t]$ and then deduce the residual energy left in node *j* at time *t* by extrapolation.

- $T_{i,j}^{1-hop,\ cooperativeness}(t)$: This provides the degree of cooperativeness of node *j* as evaluated by node *i* based on direct observations during $[t - d\Delta t, t]$. Node *i* estimates $T_{i,j}^{1-hop,\ cooperativeness}(t)$ by the ratio of the number of cooperative interaction experiences to the total number of protocol interaction experiences. Note that both counts are related to protocol execution except that the former count is for positive experiences when node *j*, as observed by node *i*, cooperatively follows the prescribed protocol execution.

The indirect trust part, $T_{i,j}^{indirect,\ X}(t)$ in Equation (6.2) is evaluated by node *i* at time *t* by taking in recommendations from a subset of 1-hop neighbors selected following the threshold-based filtering and relevance-based trust selection criteria. Specifically, node *i* will compute $T_{i,j}^{indirect,\ X}(t)$ by:

$$T_{i,j}^{indirect,\ X}(t) = \begin{cases} \dfrac{\sum_{m \epsilon V} \left(T_{i,m}^{X}(t) \times\ T_{m,j}^{X}(t) \right)}{n_r} & \text{if } n_r > 0 \\ e^{-\lambda_d \Delta t} \times T_{i,j}^{indirect,X}(t - \Delta t) & \text{if } n_r = 0 \end{cases} \tag{6.4}$$

In Equation (6.4), *m* is a recommender and *V* is a set of n_r recommenders chosen by node *i* from its 1-hop neighbors which satisfy the *threshold-based filtering* and *relevance-based trust* selection criteria. That is, these are the recommenders for which node *i*'s $T_{i,m}^{X}(t)$ in trust component *X* is higher than a minimum threshold denoted by T_t^X. Here we note that when a recommender node, say, node *m*, provides its recommendation to node *i* for evaluating node *j* in trust component *X*, node *i*'s trust in node *m* is also taken into consideration in the calculation as reflected in the product term on the right hand side of Equation (6.4). If n_r=0 then $T_{i,j}^{indirect,\ X}(t) = e^{-\lambda_d \Delta t} \times\ T_{i,j}^{indirect,\ X}(t - \Delta t)$ to account for trust decay over time.

Lastly, depending on the mobile application, nodes in a mobile group may join or leave the mobile group. For a non-member, say, node *j*, the trust level $T_{i,j}(t)$ is the same as its trust level at the last trust evaluation instant $t - \Delta t$ discounted by time decay, that is, $T_{i,j}(t) = e^{-\lambda_d \Delta t} \times T_{i,j}(t - \Delta t)$.

An interesting metric is the average "subjective" component *X* trust probability of node *j* at time *t*, $T_j^{sub,X}(t)$, as evaluated by all active nodes in the system. It can be calculated by a weighted average of component *X* trust beliefs from all nodes except *j*, i.e.,

$$T_j^{sub,X}(t) = \frac{\sum_{all\ i \neq j}(T_{i,j}^X(t))}{\sum_{all\ i \neq j} 1} \tag{6.5}$$

Another interesting metric is the overall average "subjective" trust level of node *j*, denoted by $T_j^{sub}(t)$, as evaluated by all active nodes. Once we obtain $T_{i,j}(t)$ from Equation (6.1), $T_j^{sub}(t)$ can be computed by:

$$T_j^{sub}(t) = \frac{\sum_{all\ i \neq j} T_{i,j}(t)}{\sum_{all\ i \neq j} 1} \tag{6.6}$$

We compare $T_j^{sub}(t)$ with the "objective" trust of node *j*, denoted by $T_j^{obj}(t)$, calculated based on actual, global information to see how much deviation subjective trust evaluation is from objective trust evaluation. Specifically, let $T_j^{obj,X}(t)$ denote the "objective" trust of node *j* in trust component *X* at time *t*, which we can obtain by a mathematical model (see Section 6.3 below). Then, following Equation (6.1), $T_j^{obj}(t)$ is calculated by:

$$T_j^{obj}(t) = \sum_X w^X \times T_j^{obj,X}(t) \tag{6.7}$$

By means of a novel mathematical model (discussed later in Section 6.3) describing node behaviors in a MANET, we can calculate the objective trust levels of all nodes in the system based on actual status of nodes. This serves as the basis for validating our dynamic trust management.

6.2.4 Mission-Oriented Mobile Group Applications

We consider mission-oriented mobile groups as an application. In military battlefield situations, very frequently a commander (a special node in a MANET) will need to assemble and dynamically manage a mobile task group to achieve a critical mission assigned despite failure, disconnection or compromise of member nodes. A commander node, say node *i*, can use $T_{i,j}(t)$ based on its own local view towards node *j* as an indicator to judge if node *j* satisfies the mission-specific trust requirements for successful mission execution. More importantly, the commander node could obtain the mission success probability (as a reliability metric) when given knowledge regarding the mission failure definition, member failure definition and mission time.

Let $R(t)$ be the mission reliability given that the mission time is t. Then, the mission success probability, denoted by $P_{mission}$, is simply $R(TR)$ when the commander is given TR as the mission time, i.e.,

$$P_{mission} = R(TR) \tag{6.8}$$

The mission failure definition is application dependent. Assume that the commander node is fault-free because of high integrity and high security protection. Also assume that the mission fails if at least *n-k+1* out of *n* members (trustees) fail. Let $R_j(t)$ be member *j*'s reliability at time *t*. Then,

$$R(t) = \sum_{|J|>k} \left(\prod_{j \in J} R_j(t) \prod_{j \notin J} \big(1 - R_j(t)\big) \right) \tag{6.9}$$

The member failure definition, on the other hand, hinges on trustworthiness of each individual member. Suppose there are two trust thresholds: M_1 is a trust threshold above which a member is considered completely trustworthy for successful mission completion and M_2 is a drop dead trust level below which a member is completely not trustworthy. Below we give a possible definition of member failure based on dual trust thresholds, M_1 and M_2, defined above. Specifically, if at any time *t*, node *j*'s trust level is above M_1 then node *j* is completely trustworthy, so its *instantaneous trustworthiness*, denoted by $X_j(t)$, is 1. If node *j*'s trust level is below M_2 then node *j* is completely untrustworthy, so $X_j(t)$ is 0. If node *j*'s trust level is in between M_1 and M_2 then node *j*'s instantaneous trustworthiness is calculated as the ratio of its trust level to M_1. The commander node, node *i*, computes member *j*'s reliability $R_j(t)$ based on node *j*'s instantaneous trustworthiness over [0, *t*]. If at any time $t' \le t$, $X_j(t') = 0$, then the trust level of node *j* is not acceptable, so $R_j(t)$ is 0; otherwise, $R_j(t)$ is the average trustworthiness of node *j* over [0, *t*]. Summarizing above, node *i* computes member *j*'s reliability $R_j(t)$ by:

$$R_j(t) = \begin{cases} 0, if\ X_j(t') = 0 \text{ for any } t' \le t \\ E\big[X_j(t')\big], t' \le t, \quad \text{otherwise} \end{cases}$$

$$\text{where } X_j(t') = \begin{cases} 1, & if\ T_j^{overall}(t') \ge M_1 \\ 0, & if\ T_j^{overall}(t') < M_2 \\ T_{i,j}(t')/M_1, & \text{otherwise} \end{cases} \tag{6.10}$$

Here $X_j(t'), t' \le t$, is the instantaneous reliability of nodes *j* at time *t'* and $E\big[X_j(t')\big]$ is the expected value of $X_j(t')$, $0 \le t' \le t$, over [*0, t*]. One can see that the knowledge of as

$T_{i,j}(t)$ is very desirable for the command node to compute $P_{mission}$ once it is given knowledge regarding mission execution time distribution, member failure definition, and mission failure definition.

6.3 Performance Model

Our analysis methodology is model-based and hinges on the use of a Stochastic Petri Net (SPN) [53, 54, 163] mathematical model to probabilistically estimate node status over time, given an anticipated operational profile as input. The SPN outputs provide ground truth node status and can serve as the basis for "objective" trust evaluation. Our goal is to compare "subjective" trust obtained through protocol execution with "objective" trust obtained through the SPN outputs to provide a sound theoretical basis for validating the algorithm design for dynamic trust management.

6.3.1 Node SPN for Modeling Node Behavior

Figure 6.1 shows the "node" SPN model developed for describing the lifetime behavior of a mobile node in the presence of other uncooperative and malicious nodes in a mobile group following the input operational profile. The system SPN model consists of *N* node SPN models where *N* is the number of nodes in the system. We utilize the node SPN model to obtain a single node's information (e.g., intimacy, healthiness, energy, and cooperativeness) and to derive its trust relationships with other nodes in the system. It also captures location information of a node as a function of time. We consider a square-shaped operational area consisting of *M*×*M* regions each with the width and height equal to radio radius *R*. The node mobility model is specified as part of the operational profile.

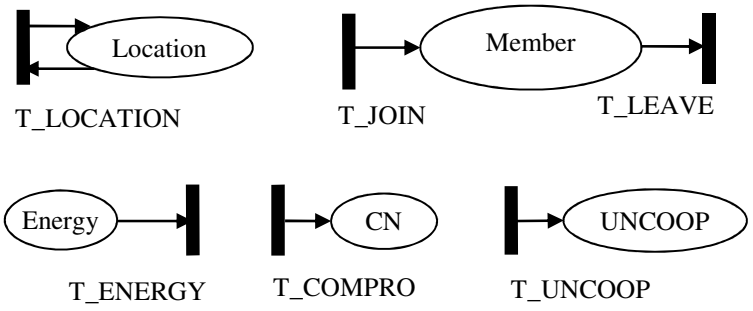


Figure 6.1: Node SPN Model.

The reason of using node SPN models is to yield a probability model (a semi-Markov chain [163, 178]) to model the stochastic behavior of nodes in the system, given the system's anticipated operational profile as input. The theoretical analysis yields *ob-*

jective trust based on ground truth of node status, against which *subjective trust* as a result of executing our proposed trust protocol is compared. This provides the theoretical foundation that subjective trust (from protocol execution) is accurate compared with ground truth. The underlying semi-Markov chain [163, 178] has a state representation comprising "places" in the SPN model. A node's status is indicated by a 5-component state representation (*Location*, *Member*, *Energy*, *CN*, *UNCOOP*) with "*Location*" (an integer) indicating the current region the node resides, "*Member*" (a boolean variable) indicating if the node is a member, "*Energy*" (an integer) indicating the current energy level, "*CN*" (a boolean variable) indicating if the node is compromised, and "UNCOOP" (a boolean variable) indicating if the node is cooperative. For example, place *Location* is a state component whose value is indicated by the number of "tokens" in place *Location*. A state transition happens in the semi-Markov chain when a move event occurs with the event occurrence time interval following a probabilistic time distribution such as exponential, Weibull, Pareto, and hyper-exponential distributions. This is modeled by a "transition" with the corresponding firing time in the SPN model. For example, when the node moves across a regional boundary after its residence time in the previous region elapses, transition T_LOCATION will be triggered, thus resulting in a location change. This is reflected by flushing all the tokens in place *Location* and replacing by a number of tokens corresponding to the id of the new region it moves into. After the move, the value of "*Location*" will be the id of the new region it moves into. Thus the three primary entities, i.e., places, tokens, and transitions, allow the node SPN model to be constructed to describe a node's lifetime behavior dynamically as time evolves. Below we explain how we construct the node SPN model.

Location: Transition T_LOCATION is triggered when the node moves to another region from its current location with the rate calculated as S_{init}/R (i.e., the node's mobility rate) based on an initial speed (S_{init}) and wireless radio range (R). Depending on the location a node moves into, the number of tokens in place *Location* is adjusted. Initially for simplicity nodes are randomly distributed over the operational area based on uniform distribution. Suppose that nodes move randomly. Then a node randomly moves to one of four locations in four directions (i.e., north, west, south, and east) in accordance with its mobility rate. To avoid end-effects, movement is wrapped around (i.e., a torus is assumed). The underlying semi-Markov model of the node SPN model when solved utilizing solution techniques such as SOR, Gauss Seidel, or Uniformization [178] gives the probability that a node is at a particular location at time t, e.g., the probability that node i is located in region j at time t. This information along with the location information of other nodes at time t provides global information if two nodes are 1-hop neighbors at time t.

Intimacy: Intimacy trust is an aggregation of *direct* interaction experience ($T_{i,j}^{intimacy}(t)$) and *indirect* interaction experience ($T_{i,j}^{indirect,\ intimacy}(t)$). Out of these

two, only new *direct* interaction experience ($T_{i,j}^{direct,\ intimacy}(t)$ via $T_{i,j}^{1-hop,\ intimacy}(t)$) is calculated based on if two nodes are 1-hop neighbors interacting with each other via packet forwarding and routing. Since the node SPN model gives us the probability that a node is in a particular location at time *t*, we can objectively compute direct interaction experience $T_{i,j}^{1-hop,\ intimacy}(t)$ (see Equation (6.3)) based on the probability of nodes *i* and *j* are in the same location at time *t* from the output of the two SPN models associated with nodes *i* and *j*.

Energy: Place *Energy* represents the current energy level of a node. An initial energy level of each node is assigned differently to reflect node heterogeneity. We randomly generate a number between 12 to 24 hours based on uniform distribution, representing a node's initial energy level E_{init}. Then we put a number of tokens in place *Energy* corresponding to this initial energy level. A token is taken out when transition T_ENERGY fires. The transition rate of T_ENERGY is adjusted on the fly based on a node's state: it is lower when a node becomes uncooperative to save energy and is higher when the node becomes compromised so that it performs attacks more and consumes energy more. Therefore, depending on the node's status, its energy consumption is dynamically changed.

Healthiness: A node is compromised when transition T_COMPRO fires. The rate to transition T_COMPRO is λ_{com} as the node compromising rate (or the capture rate) reflecting the hostility of the application. If the node is compromised, a token goes to *CN*, meaning that the node is already compromised and may perform good-mouthing and bad-mouthing attacks as a recommender by good-mouthing a bad node with a high trust recommendation and bad-mouthing a good node with a low trust recommendation.

Cooperativeness: Place *UNCOOP* represents whether a node is cooperative or not. If a node becomes uncooperative, a token goes to *UNCOOP* by triggering T_UNCOOP. We model a node's uncooperativeness behavior following the 'node behavior' model discussed in Section 6.1.3. Specifically, the rate to transition T_UNCOOP is modeled as a function of its remaining energy, the mission difficulty, and the neighborhood uncooperativeness degree as follows:

$$rate(T\_UNCOOP) = \frac{f_e(E_{remain}) f_m(M_{difficulty}) f_s(S_{degree})}{T_{gc}} \tag{6.11}$$

where E_{remain} represents the node's current energy level as given in *mark(Energy)*, $M_{difficulty}$ is the difficulty level of the given mission, S_{degree} is the degree of uncooperativeness computed based on the ratio of uncooperative nodes to cooperative nodes among 1-hop neighbors and T_{gc} is the group communication interval over which a node

may decide to become uncooperative in protocol execution and drop packets. The form $f(x) = \alpha x^{-\varepsilon}$ follows the demand-pricing relationship in Economics [9] to model the effect of its argument x on the uncooperative behavior, including:

- $f_e(E_{remain})$: If a node has a lower level of energy, it is less likely to be cooperative. This is to consider a node's individual utility in resource-constrained environments.
- $f_m(M_{difficulty})$: If a node is assigned to a more difficult mission, it is less likely to be uncooperative.
- $f_s(S_{degree})$: If a node's 1-hop neighbors are not very cooperative, the node is more likely to be cooperative to complete a given mission successfully.

A compromised node is necessarily uncooperative as it won't follow the protocol execution rules. So if place CN contains a token, place UNCOOP will also contain a token.

6.3.2 Objective Trust Evaluation

With the node behaviors modeled by a probability model (a semi-Markov chain) described above, the objective trust evaluation of node j in trust component X, i.e., $T_j^{obj,X}(t)$, can be obtained based on exact global knowledge about node j as modeled by its node SPN model that has met the convergence condition with the location information supplied. To calculate each of these objective trust probabilities of node j, one would assign a reward of r_s with state s of the underlying semi-Markov chain of the SPN model to obtain the probability weighed average reward as:

$$T_j^{obj,X}(t) = \sum_{s \in S} (r_s * P_s(t)) \tag{6.12}$$

For X = healthiness, energy or cooperativeness, and as:

$$T_j^{obj,X}(t) = \frac{\int_{t-d\Delta t}^{t} \sum_{s \in S} (r_s * P_s(t'))dt'}{d\Delta t} \tag{6.13}$$

For X = intimacy. Here S indicates the set of states in the underlying semi-Markov chain of our SPN model, r_s is the reward to be assigned to state s, and $P_s(t)$ is the probability that the system is in state s at time t, which can be obtained by solving the underlying semi-Markov model of our SPN model utilizing solution techniques such as SOR, Gauss Seidel, or Uniformization [178]. Table 6.1 summarizes specific reward assignments used to calculate $T_j^{obj,X}(t)$ for X=intimacy, healthiness, energy, or cooperativeness. In Table

6.1, E_T is the energy threshold below which the energy trust toward a node goes to 0. Once $T_j^{obj,X}(t)$ is obtained, we compute the average objective trust value of node j, $T_j^{obj}(t)$, based on Equation (6.7).

Here we note that in Table 6.1 we assign a binary trust value of 0 or 1 to a state in which it is clear in this particular state the trust value is either 0 or 1. Since the system evolves over time and there is a probability that it may stay at any state at time t with all state probabilities sum to 1, the expected value of a trust property (intimacy, healthiness, energy or cooperativeness) at time t based on a state-probability-weighted trust calculation is a real number between 0 and 1.

Table 6.1: Reward Assignment for Objective Trust Evaluation.

Component trust probability toward node j	**r_s: reward assignment to state s**
$T_j^{obj,\ intimacy}(t)$	1 $if\ mark(j's\ \ location)$ *is within 5-region neighbor area at time t*; 0 *otherwise*
$T_j^{obj,\ \ healthiness}(t)$	1 $if\ (mark(j's\ CN) = 0)$; 0 *otherwise*
$T_j^{obj,\ \ energy}(t)$	1 $if\ (mark(j's\ Energy) > E_T)$; 0 *otherwise*
$T_j^{obj,\ \ cooperativeness}(t)$	1 $if\ (mark(j's\ UNCOOP) = 0)$; 0 *otherwise*

6.3.3 Subjective Trust Evaluation

Unlike objective trust evaluation, subjective trust evaluation is based on Equation (6.1) to Equation (6.4) following the trust protocol execution. In particular, in Equation (6.3), a node must assess $T_{i,j}^{1-hop,\ X}(t)$ of its 1-hop neighbors using the detection mechanisms for trust property X described in Section 6.2.3. Because the assessment is direct, assuming that the detection mechanisms are effective, $T_{ij}^{1-hop,X}(t)$ computed by node i will be close to actual status of node j at time t, which can be obtained from the SPN model output. We assert that all detection mechanisms (discussed in Section 6.2.3) are effective and accurate, except for the anomaly detection mechanisms for detecting unhealthiness because of imperfection in anomaly detection, causing $T_{i,j}^{1-hop,\ \ healthiness}(t)$ to deviate from the actual healthiness status of node j. The imperfection is accounted for by considering the false alarm probabilities of anomaly detection mechanisms employed, i.e., a false negative probability (P_{fn}^H) and a false positive probability (P_{fp}^H), given as input to the system. Both P_{fn}^H and P_{fp}^H can be obtained from the provider of specific anomaly de-

tection mechanisms, e.g., [170]. Both must be sufficiently low (e.g., less than 5%) for the anomaly detection mechanisms to be considered as a valid design.

With these key observations, we leverage SPN outputs reflecting actual status of nodes to predict $T_{i,j}^{1-hop,\ X}(t)$ which would be obtained by node i at runtime. Table 6.2 gives specific reward assignments used to compute $T_{i,j}^{1-hop,\ X}(t)$. Here we note that when computing $T_{i,j}^{1-hop,\ healthiness}(t)$ in order to account for the imperfection of the anomaly detection mechanisms employed for detecting unhealthiness, instead of assigning a reward of 1 if node *j* is not compromised, i.e., *mark*(*j's CN*) = 0, we assign a reward of *1-*P_{fp}^{H} to account for the false positive probability. Also instead of assigning a reward of 0 if node *j* is compromised, i.e., *mark*(*j's CN*) = 1, we assign a reward of P_{fn}^{H} to account for the false negative probability. All other reward assignments for X=intimacy, energy, and cooperativeness simply yield the actual status of node *j* in property *X* at time *t*.

Table 6.2: Reward Assignments for Subjective Trust Evaluation.

Component trust probability of node *i* toward node *j*	**r_s: reward assignment to state *s***
$T_{i,j}^{1-hop,intimacy}(t)$	1 *if i and j are 1-hop neighbors within last* $d\Delta t$; 0 *otherwise*
$T_{i,j}^{1-hop,healthiness}(t)$	*1-*P_{fp}^{H} *if* (*mark*(*j's CN*) = 0); P_{fn}^{H} *otherwise*
$T_{i,j}^{1-hop,energy}(t)$	1 $if\ (mark(j's\ Energy) > E_T)$; 0 *otherwise*
$T_{i,j}^{1-hop,cooperativeness}(t)$	1 $if\ (mark(j's\ UNCOOP) = 0)$; 0 *otherwise*

When node *i* obtains $T_{i,j}^{1-hop,\ X}(t)$, it computes $T_{i,j}^{direct,\ X}(t)$ from Equation (6.3). Then node *i* computes $T_{i,j}^{indirect,\ X}(t)$ based on Equation (6.4), as well as $T_{i,j}^{X}(t)$ and $T_{i,j}(t)$ from Equations (6.2) and (6.1), respectively. Finally, the overall average subjective trust values of node *j*, $T_j^{sub,X}(t)$ and $T_j^{sub}(t)$, can be obtained through Equations (6.5) and (6.6), respectively. We compare $T_j^{sub}(t)$ with objective trust $T_j^{obj}(t)$ for validating dynamic trust management design.

6.4 Evaluation Results

6.4.1 Operational Profile as Input

Table 6.3 lists the parameter set and default values specifying the operational profile given as input for testing dynamic trust management for a mobile group application in MANET environments. We populate a MANET with 150 nodes moving randomly with speed S_{init} in the range of (0,2] m/s in a 6×6 operational region in a 1500m×1500m area, with each region covering R=250m radio radius. We use n_r 1-hop neighbors as the recommenders for indirect trust evaluation. The environment being considered is assumed hostile and insecure with the average compromising rate λ_{com} set to once per 18 hours. Each node's energy is in the range of [12, 24] hours. Further each node observes the node behavior model as specified in Section 6.1.3 and Section 6.3.1 with ε=1.2, α=0.8 and T_{gc}=120 sec. Initially all nodes are not compromised. When a node turns malicious, it performs good-mouthing and bad-mouthing attacks, i.e., it will provide the most positive recommendation (that is, 1) toward a bad node to facilitate collusion, and conversely the most negative recommendation (that is, 0) toward a good node to ruin the reputation of the good node. The initial trust level is set to 1 for healthiness, energy and cooperativeness because all nodes are considered trustworthy initially. The initial trust level of intimacy is set to the probability that a node is found to be in a 5-region neighbor area relative to 6x6 regions in accordance with the intimacy definition.

Table 6.3: Operational Profile for a Mobile Group Application.

Parameter	Value	Parameter	Value
λ_d	0.001	R	250m
α	0.8	n_r	5
ε	1.2	d	2
β_1:$\beta 2$	Variable	P_{fn}^H, P_{fp}^H	0.5%
w_1: w_2: w_3: w_4	Variable	$1/\lambda_{com}$	18 hrs
TR	Variable	Δt	1200s
S_{init}	(0, 2] m/s	E_{init}	[12, 24] hrs
T_{gc}	120s	E_T	0 hrs

Given this operational profile as input to the mobile group application, we aim to identify the best setting of β_1: β_2 (with higher β_1 meaning more direct observations or self-information being used for subjective trust evaluation) under which subjective trust is closest to objective trust. We also aim to identify the best setting of $w_1: w_2: w_3: w_4$ (the weight ratio for the 4 trust components considered), and M_1 and M_2 (the minimum trust level and drop-dead trust level) under which the application performance is maximized. For trust protocol execution, we set the decay coefficient $\lambda_d = 0.001$, and the trust evaluation interval $\Delta t = 20$ min, resulting in $e^{-\lambda_d \Delta t} = 0.98$ to model small trust decay

over time. Also the minimum recommender threshold T_t^X is set to 0.6, the trust evaluation window size d is set to 2, and the minimum energy trust threshold E_T is set to 0.

6.4.2 Identifying Trust Protocol Settings for Accurate Peer-to-Peer Subjective Trust Evaluation

To reveal which trust component might have a more dominant effect, we show individual trust component values, i.e., $T_j^{intimacy}(t)$, $T_j^{healthiness}(t)$, $T_j^{energy}(t)$ and $T_j^{cooperativeness}(t)$ for a node randomly picked. Other nodes exhibit similar trends and thus only one set of results is shown here. Figure 6.2 to Figure 6.5 show the node's trust values as a function of mission execution time for intimacy, healthiness, energy and cooperativeness, respectively, with β_1: β_2 varying from 0.3: 0.7 (30% direct evaluation: 70% indirect evaluation) to 0.9: 0.1 (90% direct evaluation: 10% indirect evaluation). We see that for all 4 trust components, subjective trust evaluation results are closer and closer to objective trust evaluation results as we use more conservative direct observations or self-information for subjective trust evaluation. However, there is a cutoff point (at about 80%) after which subjective trust evaluation overshoots. This implies that using too much direct observations for subjective trust evaluation may overestimate trust because there is little chance for a node to use indirect observations from trustworthy recommenders. Our analysis allows such a cutoff point to be determined given design considerations regarding trust decay over time ($e^{-\lambda_d \Delta t} = 0.98$ for direct trust decay in our case study).

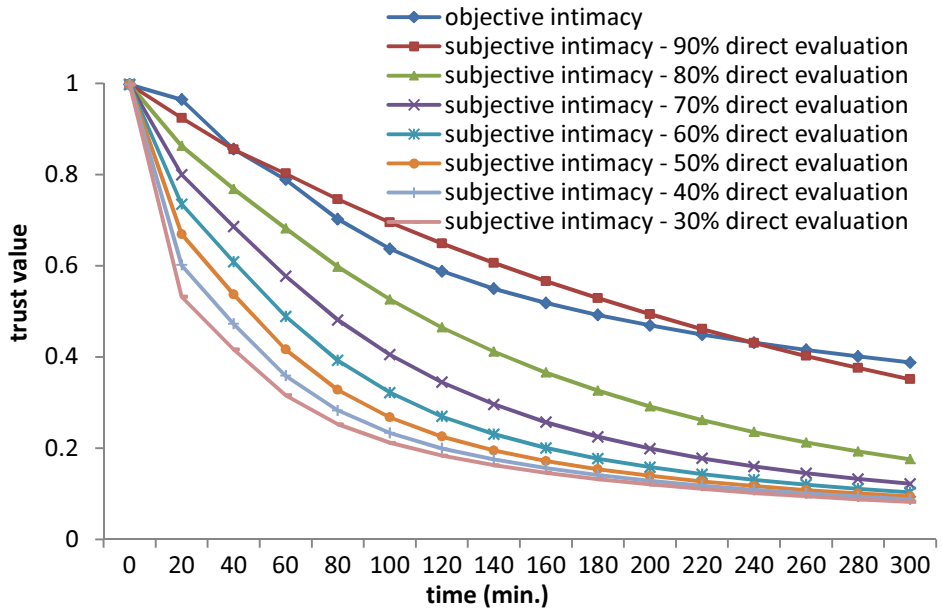


Figure 6.2: Intimacy Evaluation.

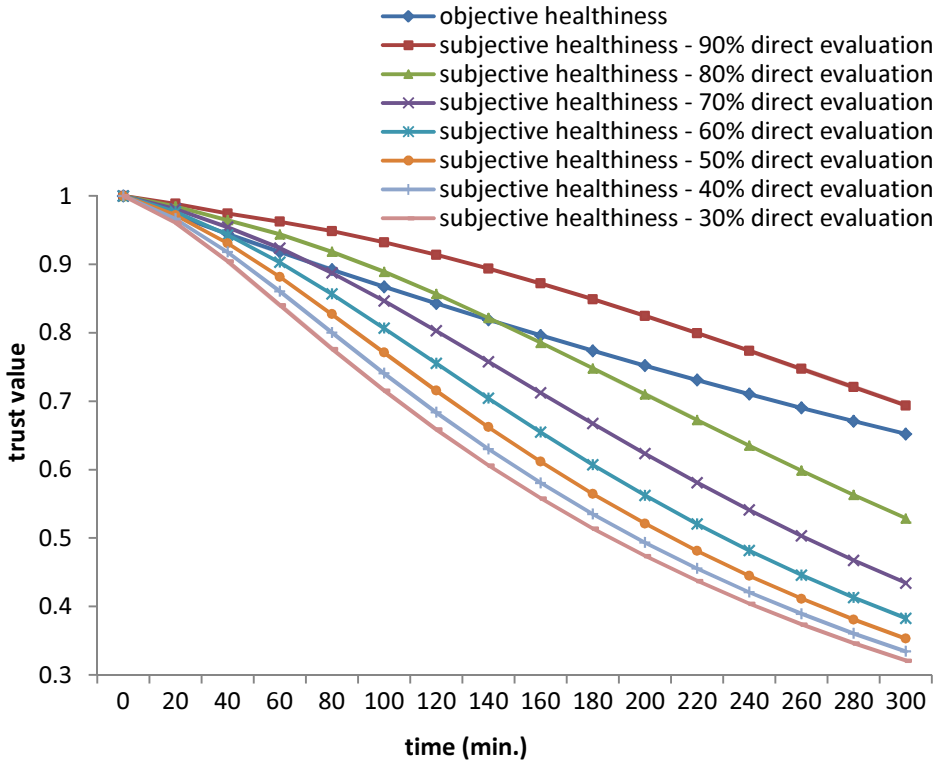


Figure 6.3: Healthiness Evaluation.

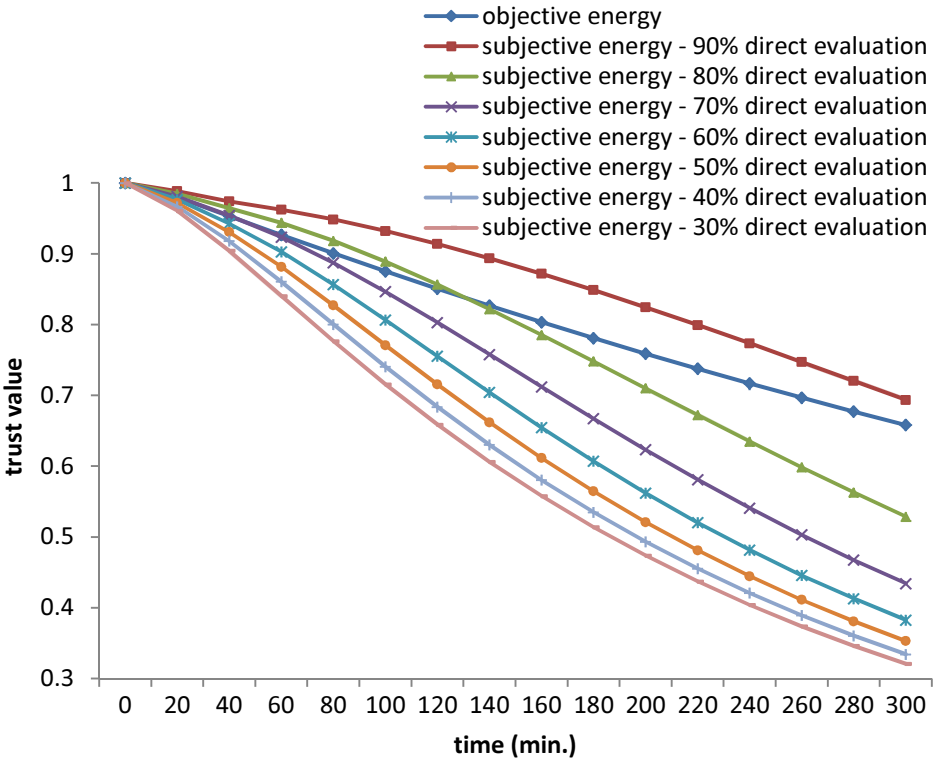


Figure 6.4: Energy Evaluation.

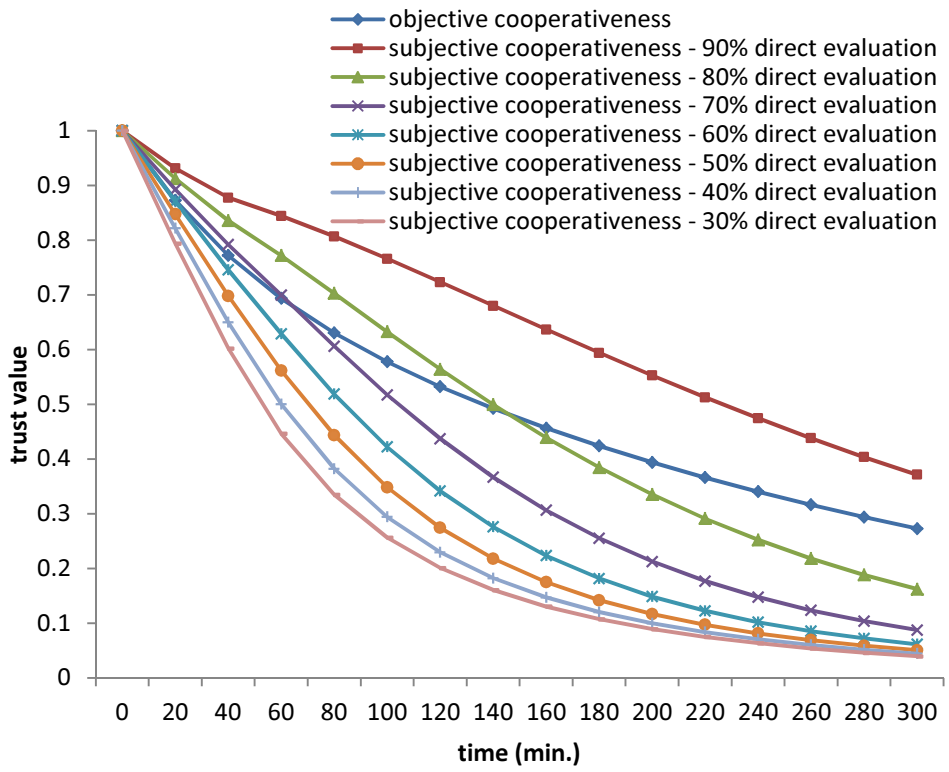


Figure 6.5: Cooperativeness Evaluation.

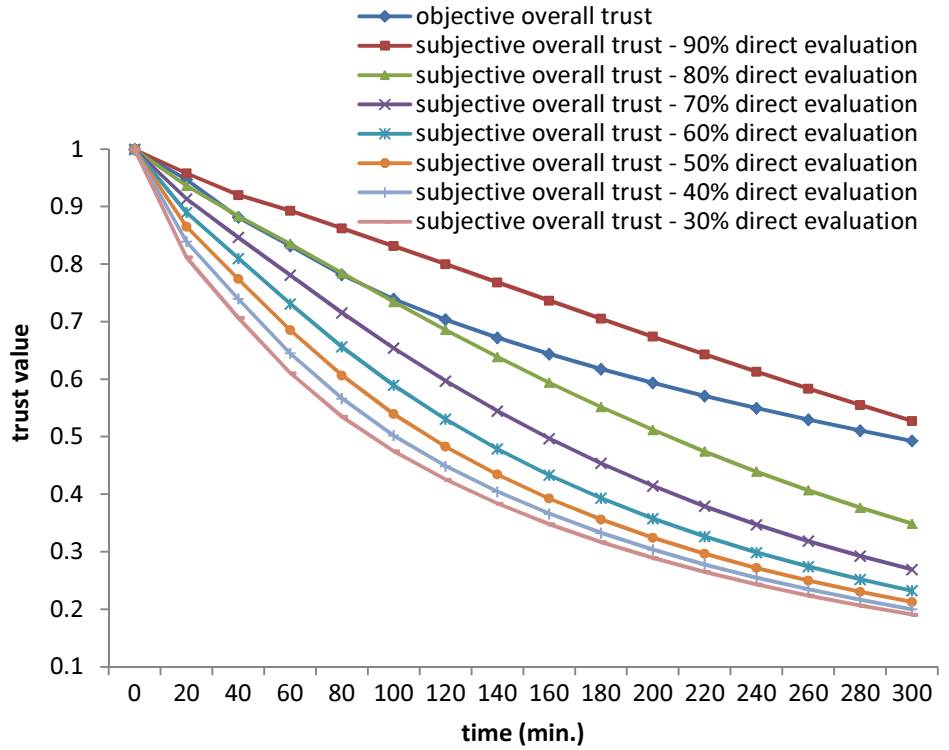


Figure 6.6: Overall Trust Evaluation.

Figure 6.6 shows the node's overall trust values obtained from subjective trust evaluation vs. objective trust evaluation, also as a function of time. We observe that the subjective trust evaluation curve is reasonably close to the objective trust evaluation curve, but again there is a cutoff point after which the trust value is overestimated compared to objective trust. Initially, subjective trust evaluation undershoots due to lack of observations. At the later stage, nodes might be compromised and consume much resources. Therefore, subjective trust evaluation could overshoot compared to objective trust. Nevertheless, Figure 6.2 to Figure 6.6 demonstrate that subjective trust evaluation results can be very close to objective trust evaluation results when the right amount of direct observations is used for subjective trust evaluation.

6.4.3 Identifying Best Trust Formation Settings to Maximize Application Performance

We consider a mission-oriented mobile group application scenario in which a commander node, say node *i*, dynamically selects *n* nodes (*n*=5 in the case study) which it trusts most out of active mobile group members for mission execution. We consider dynamic team membership such that after each trust evaluation window Δt the commander will reselect its most trusted nodes composing the team for mission executions based on its peer-to-peer subjective evaluation values $T_{i,j}(t)$ toward nodes *j's* as calculated from Equation (6.1). The rationale behind dynamic membership is that the commander may exercise its best judgment to select *n* most trusted nodes to increase the probability of successful mission execution. Assume that all *n* nodes selected at time *t* are critical for mission execution during [*t*, *t*+Δt] so that if any one node selected fails, the mission fails. We can then apply Equations (6.8) and (6.9) to compute $P_{mission}$ over an interval [*t*, *t*+Δt]. Since all time intervals are connected in a series structure, $P_{mission}$ over the overall mission period [0, *TR*] can be computed by the product of individual $P_{mission}$'s over intervals [0, Δt], [Δt, $2\Delta t$], …, [*TR*-Δt, *TR*].

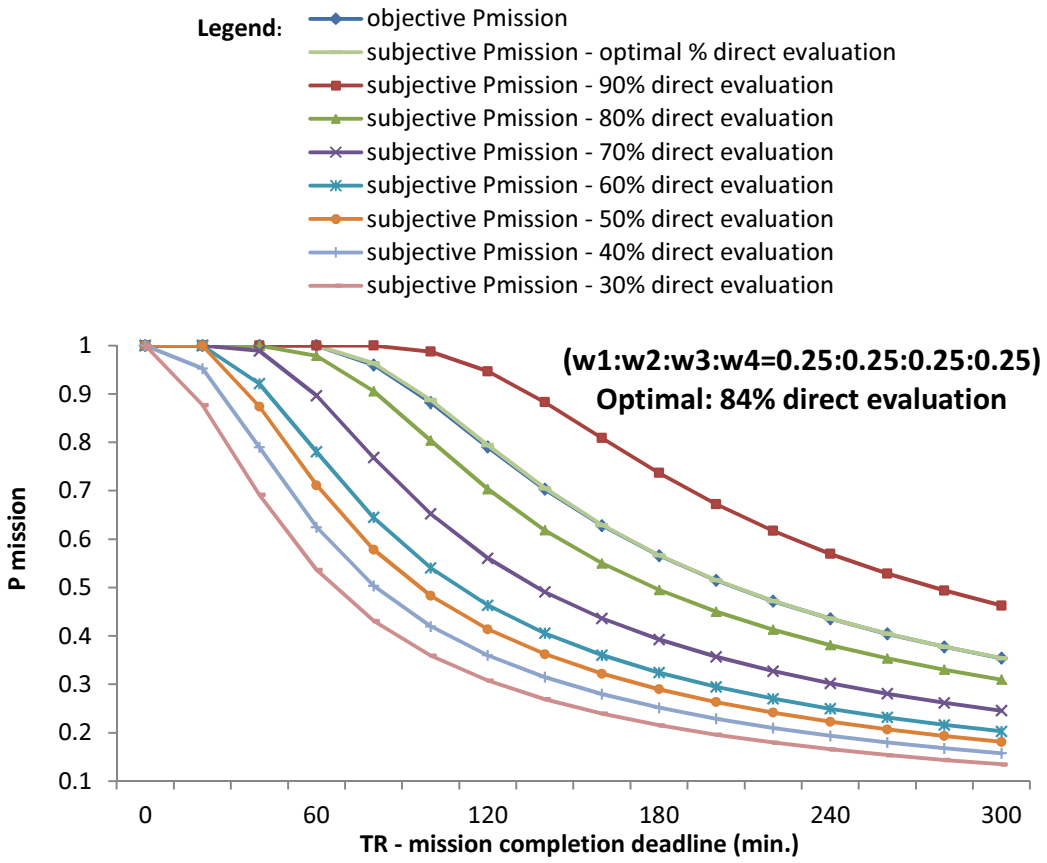
Figure 6.7 shows the mission success probability $P_{mission}$ as a function of *TR*. To examine the effect of $w_1 : w_2 : w_3 : w_4$ (the weight ratio for the 4 trust components considered), we consider 5 test cases: (a) *equal-weight*, (b) *social trust only*, (c) *QoS trust only*, (d) *more social trust*, and (e) *more QoS trust* as listed in Table 6.4 with (M_1, M_2) set to (0.85, 0.55) to isolate its effect.

For all test cases we see that as *TR* increases, the mission success probability decreases because a longer mission execution time increases the probability of low-trust nodes (whose population increases over time because of cooperativeness or healthiness trust decay) becoming members of the team for mission execution. For comparison, the mission success probability $P_{mission}$ based on objective trust evaluation results is also shown, representing the ideal case in which node *i* has global knowledge of status of all

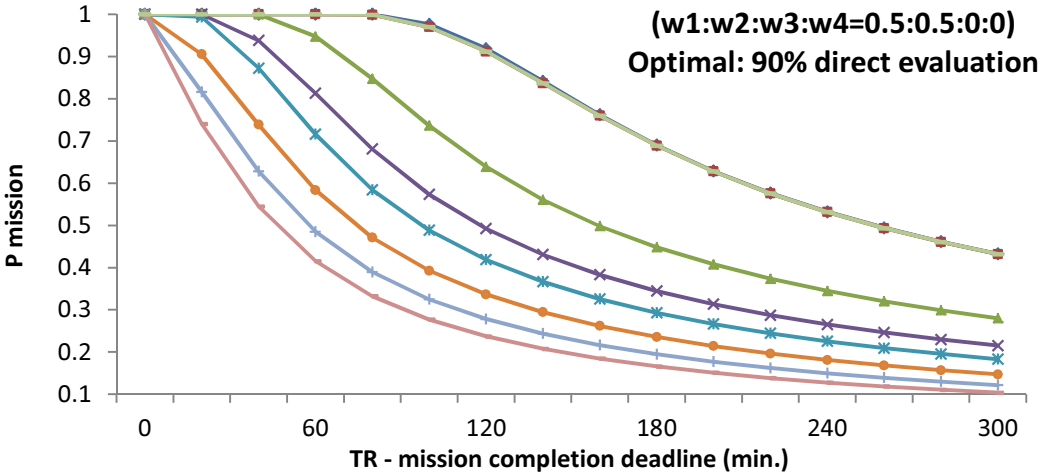
other nodes in the system and therefore it always picks n truly most trustworthy nodes in every Δt interval for mission execution. For each case, we also show the optimal β_1: β_2 ratio (with higher β_1 meaning more direct observations or self-information being used for subjective trust evaluation) at which $P_{mission}$ obtained based on subjective trust evaluation results is virtually identical to $P_{mission}$ obtained based on objective trust evaluations.

Table 6.4: Test Cases for Weight Ratio.

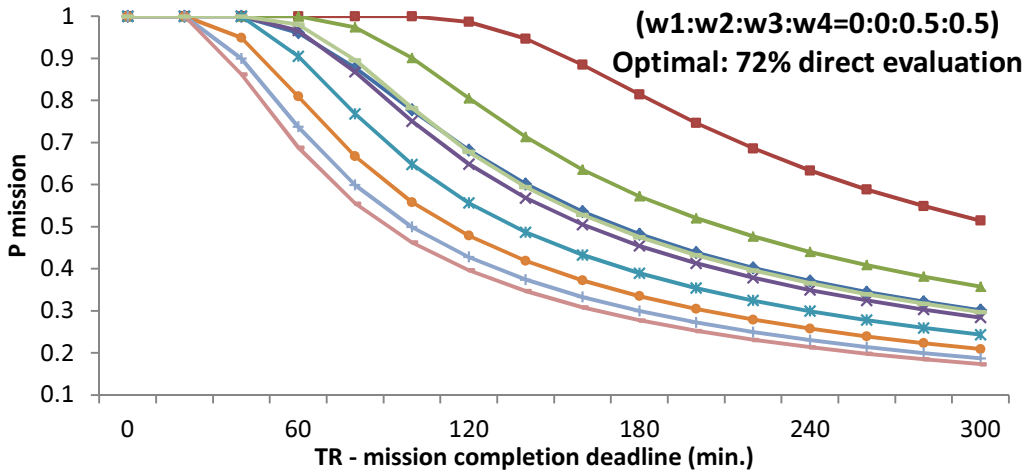
Test case	Weight ratio
Equal-weight	$w_1: w_2: w_3: w_4 = 0.25: 0.25: 0.25: 0.25$
Social trust only	$w_1: w_2: w_3: w_4 = 0.5: 0.5: 0: 0$
QoS trust only	$w_1: w_2: w_3: w_4 = 0: 0: 0.5: 0.5$
More social trust	$w_1: w_2: w_3: w_4 = 0.35: 0.35: 0.15: 0.15$
More QoS trust	$w_1: w_2: w_3: w_4 = 0.15: 0.15: 0.35: 0.35$



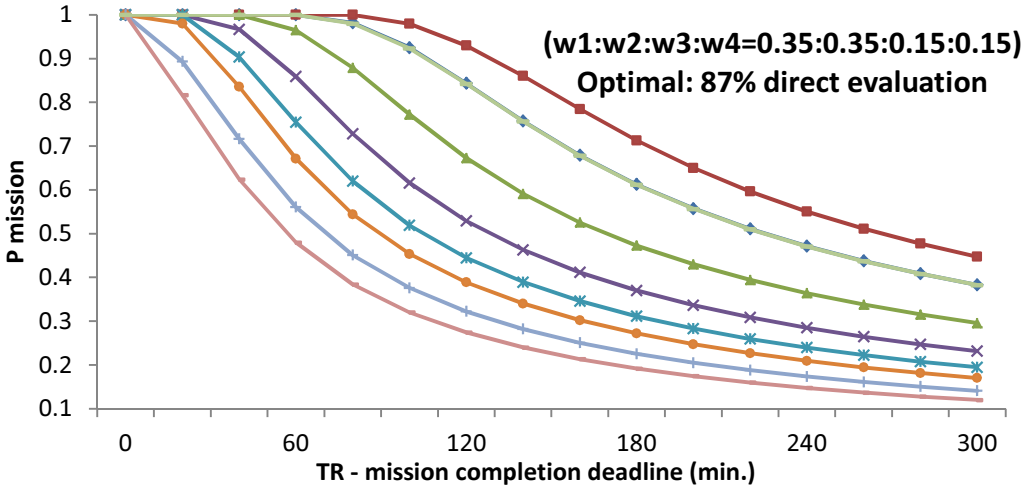
(a) Equal-Weight.



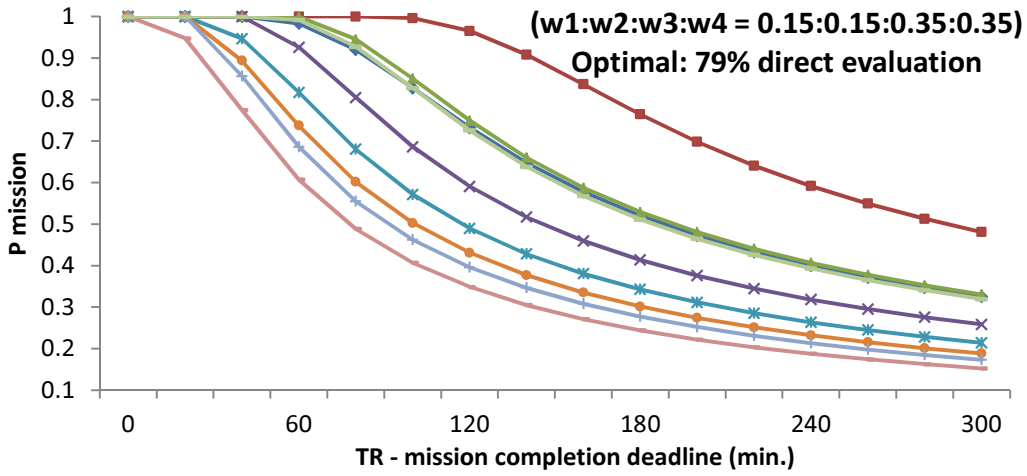
(b) Social Trust Only.



(c) QoS Trust Only.



(d) More Social Trust.



(e) More QoS Trust.

Figure 6.7: Mission Success Probability: Subjective vs. Objective Evaluation.

We observe that as more social trust is being used for subjective trust evaluation, the optimal β_1: β_2 ratio increases, suggesting that social trust evaluation is very subjective in nature and a node would rather trust its own interaction experiences more than recommendations provided from other peers, especially in the presence of malicious nodes that can perform good-mouthing and bad-mouthing attacks. Also again we observe that while using more conservative direct observations or self-information for subjective trust evaluation in general helps in bringing subjective $P_{mission}$ closer to objective $P_{mission}$, there is a cutoff point after which subjective trust evaluation overshoots.

Figure 6.7 demonstrates the effectiveness of our dynamic trust management for MANETs. We see that the mission success probability as a result of executing subjective trust evaluation is very close to that from objective trust evaluation, especially when we use more but not excessive direct observations for subjective trust evaluation. When given a mission context characterized by a set of model parameter values defined in Table 6.3, the analysis methodology that we develop helps identify the best weight of direct observations (i.e., β_1: β_2) to be used for subjective trust evaluation, so that our trust management protocol can be fine-tuned to yield results close to those by objective trust evaluation based on actual knowledge of node status.

In Figure 6.8 we compare $P_{mission}$ vs. TR for the mission group under various $w_1: w_2: w_3: w_4$ ratios, with each operating at its optimal β_1:β_2 ratio so that in each test case subjective $P_{mission}$ is virtually the same as objective $P_{mission}$. We see that "social trust only" produces the highest system reliability, while "QoS trust only" has the lowest system reliability among all, suggesting that in this case study social trust metrics used (intimacy and healthiness) are able to yield higher trust values than those of QoS trust metrics used (energy and cooperativeness). Certainly, this result should not be con-

strued as universal. When given a mission context characterized by a set of model parameter values defined in Table 6.3, the model-based analysis methodology that we develop helps identify the best $w_1 : w_2 : w_3 : w_4$ ratio to be used to maximum the system reliability.

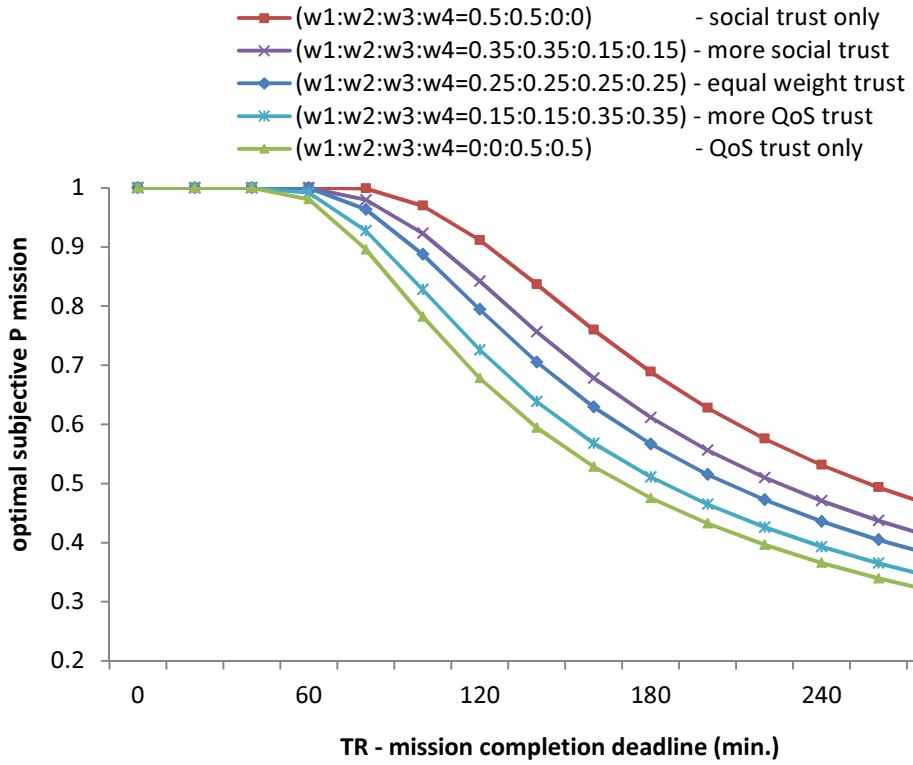


Figure 6.8: Effect of $w_1 : w_2 : w_3 : w_4$ on Mission Success Probability.

Lastly we analyze the effect of mission trust thresholds M_1 (the minimum trust level required for successful mission completion) and M_2 (the drop dead trust level). Figure 6.9 and Figure 6.10 show $P_{mission}$ vs. TR for the system operating under optimal β_1:β_2 settings in the equal-weight case for each (M_1, M_2) combination. Recall that M_1 and M_2 represent the belief if a node is considered trustworthy for mission execution. From Figure 6.9, we see that as M_1 increases, the system reliability decreases because there is a smaller chance for a node to satisfy the high threshold for it to be completely trustworthy for mission execution. Similarly from Figure 6.10, we see that as M_2 increases, the system reliability decreases because there is a higher chance for a node to be completely untrustworthy for mission execution. We also observe that the reliability is more sensitive to M_1 than M_2. A system designer can set proper M_1 and M_2 values based on the mission context such as the degree of difficulty and mission completion deadline, utilizing the model-based methodology that we develop to analyze the effect of M_1 and M_2 so as to improve the system reliability.

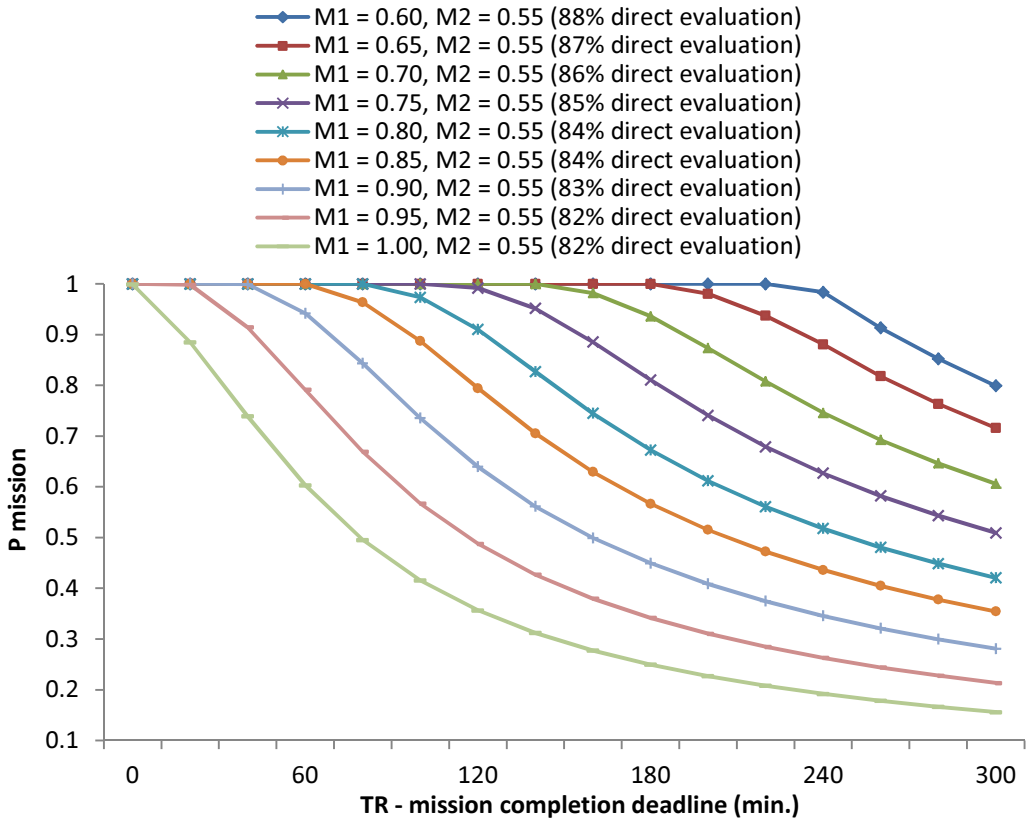


Figure 6.9: Effect of M_1 on Mission Success Probability.

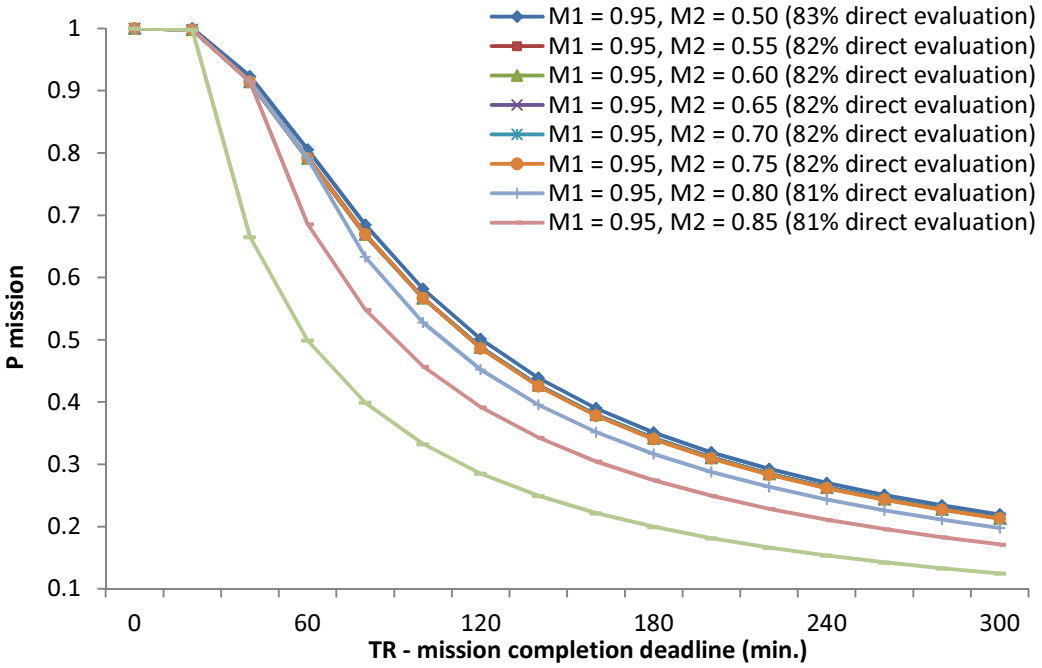


Figure 6.10: Effect of M_2 on Mission Success Probability.

6.5 Simulation Validation

We validate dynamic trust management and its application to mobile group reliability assessment through extensive simulation using ns-3 [1]. The simulated MANET environment is setup as described in Table 6.3. The network consists of 150 nodes following the random waypoint mobility model in a 1500 m × 1500 m operational area, with the speed in the range of (0, 2] m/s and pause time of zero. The initial node energy is in the range of [40, 80] joules, corresponding to [12, 24] hours of operational time in normal status. A node may be compromised with a per-node capture rate of λ_{com}. As time progresses, a node may become uncooperative, the rate of which is implemented according to Equation (6.12). When a node becomes uncooperative, it would not follow protocol execution and will drop packets to save energy. A compromised node will also drop packets. In addition, it will perform bogus message attacks, as well as good-mouthing and bad-mouthing attacks. All nodes execute dynamic trust management protocol as described in Section 6.2 to perform subjective trust evaluation.

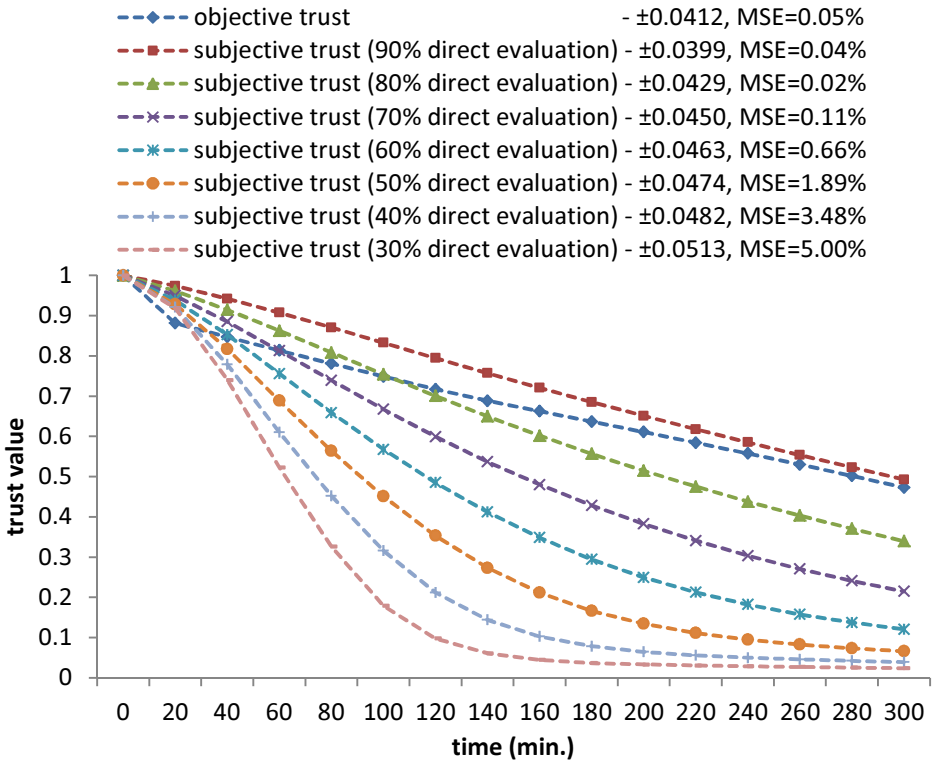


Figure 6.11: Simulation Results of Overall Trust Corresponding to Figure 6.6.

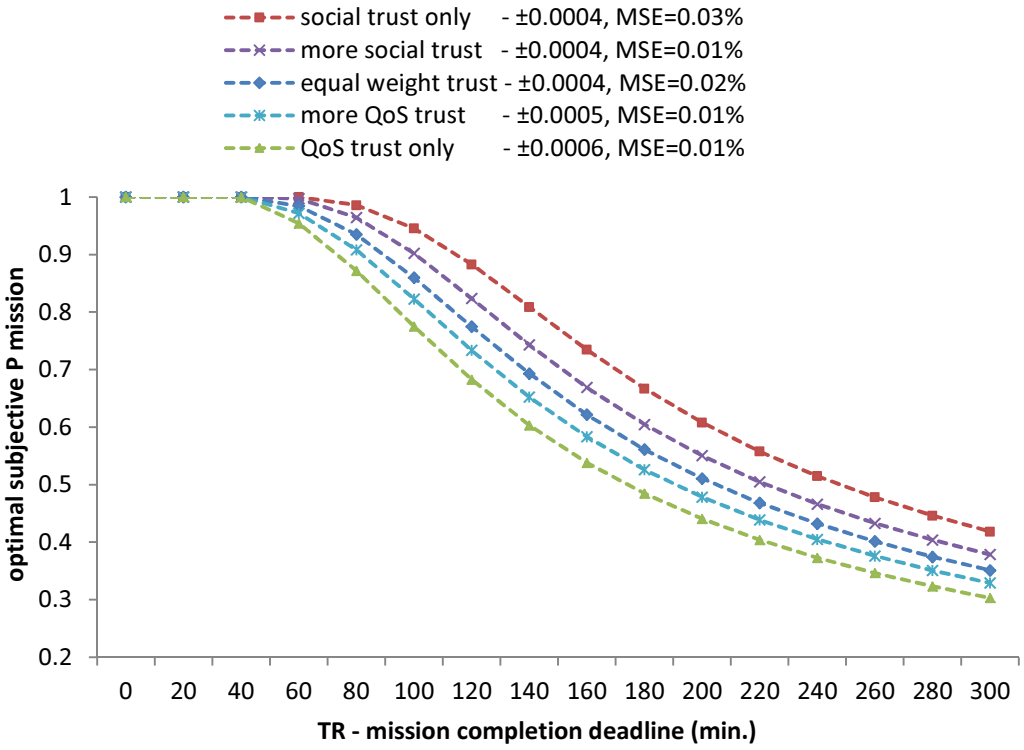


Figure 6.12: Simulation Results of Reliability Assessment Corresponding to Figure 6.8.

We collect simulation data to validate analytical results reported earlier. Figure 6.11 shows the simulation results for the overall subjective trust obtained under the equal-weight case, corresponding to Figure 6.6 obtained earlier from theoretical analysis. As in Figure 6.6, we simulate 7 cases with β_1:β_2 varying from 0.3: 0.7 to 0.9: 0.1. For each case, we collect observations from sufficient simulation runs with disjoint random number streams to achieve ±5% accuracy level with 95% confidence. For the clarity of presentation, we do not show the confidence interval in the figure. Instead, we indicate the confidence interval value on the legend of each line. The simulation results in Figure 6.11 are remarkably similar to the analytical results shown in Figure 6.6, with the average mean square error (MSE) between the simulation results vs. the analytical results less than 5%.

Figure 6.12 shows the simulation results for the effect of $w_1 : w_2 : w_3 : w_4$ on mission success probability $P_{mission}$, corresponding to Figure 6.8 obtained earlier from analytical calculations. As in Figure 6.8, we simulate 5 cases for the w_1: w_2: w_3: w_4 weight ratio (see Table 6.4). We observe that Figure 6.12 is virtually identical to Figure 6.8 in shape exhibiting the same trend that using more social trust would yield higher system reliability. The MSE is remarkably small (less than 0.03%) for all cases. We conclude that our analytical results reported in Figure 6.2 to Figure 6.12 are accurate and valid.

6.6 Summary

In this chapter, we proposed and analyzed a trust management protocol that incorporates both social and QoS trust metrics for subjective trust evaluation of mobile nodes in MANETs. The most salient feature of the proposed trust management protocol is that it is distributed and dynamic, only requiring each node to periodically estimate its degree of social and QoS trust toward its peers local or distance away. We developed a model-based methodology based on SPN techniques for describing the behavior of a mobile group consisting of well-behaved, malicious and uncooperative nodes following an anticipated system operational profile. By applying an iterative technique for solving the large SPN model, we allow the *objective* trust values of nodes to be calculated based on global knowledge regarding status of nodes as time progresses, which serves as the basis for performance evaluation. We demonstrated that our protocol is able to provide subjective trust evaluation results close to objective trust evaluation results, thus supporting its resiliency property to bad-mouthing and good-mouthing attacks by malicious nodes. We also demonstrated the effect of our trust management protocol on the reliability of mission-oriented mobile groups in MANETs, verified by the exact match between subjective mission success probability and objective mission success probability. Finally, we analyzed the effects of key design parameters such as β_1: β_2 (with higher β_1 meaning more direct observations or self-information being used for subjective trust evaluation), $w_1 : w_2 : w_3 : w_4$ (the weight ratio for the 4 trust components considered), M_1 and M_2 (the minimum trust level and drop-dead trust level), and TR (the mission completion deadline) on the system reliability of a mission-oriented mobile group and provided guidelines for fine-tuning these parameters so as to maximize the system reliability.

Chapter 7

Dynamic Trust Management for Delay Tolerant Networks and Its Applications

In this chapter, we apply design and validation principles of dynamic trust management to delay tolerant networks (DTNs). A delay tolerant network (DTN) comprises mobile nodes (e.g., humans in a social DTN) experiencing sparse connection, opportunistic communication, and frequently changing network topologies. Because of the lack of end-to-end connectivity, routing in DTN adopts a *store-carry-and-forward* scheme by which messages are forwarded through a number of intermediate nodes leveraging opportunistic encountering, hence resulting in a high end-to-end latency.

We design and validate a dynamic trust management protocol for secure routing optimization in DTN environments in the presence of well-behaved, selfish and malicious nodes. We develop a novel model-based methodology for the analysis of our trust protocol and validate it via extensive simulation. Moreover, we address dynamic trust management, i.e., determining and applying the best operational settings at runtime in response to dynamically changing network conditions to minimize trust bias and to maximize the routing application performance. We perform a comparative analysis of our proposed routing protocol against Bayesian trust-based and non-trust based (PROPHET and epidemic) routing protocols. The results demonstrate that our protocol is able to deal with selfish behaviors and is resilient against trust-related attacks. Furthermore, our trust-based routing protocol can effectively trade off message overhead and message delay for a significant gain in delivery ratio. Our trust-based routing protocol operating under identified best settings outperforms Bayesian trust-based routing and PROPHET, and approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message or protocol maintenance overhead.

7.1 System Model

We consider a DTN environment with no centralized trusted authority. Nodes communicate through multiple hops. When a node encounters another node, they exchange encounter histories certified by encounter tickets [112] so as to prevent black hole attacks to DTN routing. We differentiate socially selfish nodes from malicious nodes. A selfish node acts for its own interests including interests to its friends, groups, or communities. So it may drop packets arbitrarily just to save energy but it may decide to forward a packet if it has good social ties with the source, current carrier or destination node. We consider a friendship matrix [114] to represent the social ties among nodes. Each node keeps a list of friends in its local storage. When a node becomes selfish, it will only forward messages when it is a friend of the source, current carrier, or the destination node, while a well-behaved node performs altruistically regardless of the social ties. A malicious node aims to break the basic DTN routing functionality. In addition to dropping packets, a malicious node can perform the following trust-related attacks (see Chapter 3).

A malicious attacker can perform random attacks to evade detection. We introduce a random attack probability P_{rand} to reflect random attack behavior. When P_{rand} =1, the malicious attacker is a reckless attacker; when P_{rand} < 1 it is a random attacker. Collaborative attacks are possible through bad-mouthing attacks and ballot stuffing, which are mitigated in our protocol design by setting a trust recommender threshold T_{rec} to filter out less trustworthy recommenders.

A node's trust value is assessed based on direct trust evaluation and indirect trust information like recommendations. The trust of one node toward another node is updated upon encounter events. Each node will execute the trust protocol independently and will perform its direct trust assessment toward an encountered node based on specific detection mechanisms designed for assessing a trust property X. Later in Section 4 we will discuss these specific detection mechanisms employed in our protocol for trust aggregation.

7.2 Trust Management Protocol

Our trust protocol considers trust composition, trust aggregation, trust formation and application-level trust optimization designs. For trust composition, we consider two types of trust properties:

- QoS trust: QoS trust is evaluated through the communication network by the capability of a node to deliver messages to the destination node. We consider "connectivity" and "energy" to measure the QoS trust level of a node. The connectivity QoS trust is about the ability of a node to encounter other nodes due to its movement patterns. The energy QoS trust is about the battery energy of a node to perform the basic routing function.

- Social trust: Social trust is based on honesty or integrity in social relationships and friendship in social ties. We consider "healthiness" and social "unselfishness" to measure the social trust level of a node. The healthiness social trust is the belief of whether a node is malicious. The unselfishness social trust is the belief of whether a node is socially selfish. While social ties cover more than just friendship, we consider friendship as a major factor for determining a node's socially selfish behavior.

The selection of trust properties is application driven. In DTN routing, message delivery ratio and message delay are two important factors. We select "healthiness", "unselfishness", and "energy" in order to achieve high message delivery ratio, and we select "connectivity" to achieve low message delay.

We define a node's trust level as a real number in the range of [0, 1], with 1 indicating complete trust, 0.5 ignorance, and 0 complete distrust. We consider a trust formation model by which the trust value of node *j* evaluated by node *i* at time *t*, denoted as $T_{i,j}(t)$, is computed by a weighted average of healthiness, unselfishness, connectivity, and energy as follows:

$$T_{i,j}(t) = \sum_{X}^{all} w^X \times T_{i,j}^X(t) \tag{7.1}$$

where X represents a trust property explored (X = healthiness, unselfishness, connectivity or energy), $T_{i,j}^X(t)$ is node i's trust in trust property X toward node j, and w^X is the weight associated with trust property X with the sum equal to 1.

We aim to identify the best weight ratio under which the application performance (secure routing) is maximized, given an operational profile [141] as input. Before this can be achieved, however, one must address the accuracy issue of trust aggregation. That is, for each QoS or social trust property X, we must devise and validate a trust aggregation protocol executed by a trustor node to assess X of a trustee node such that the trust value computed is accurate with respect to actual status of the trustee node in X. This is achieved by devising a trust propagation protocol with tunable parameters which can be adjusted based on each trust property.

When evaluating $T_{i,j}(t)$, we adopt the following notations: node i is the trustor, node j is the trustee, node m is a newly encountered node, and node k is a recommender. Node i (trustor) updates its trust toward node j (trustee) in trust property X upon encountering a node at time t over an encounter interval $[t, t + \Delta t]$ as follows:

$$T_{i,j}^X(t + \Delta t) = \beta_1 \, T_{i,j}^{direct,\ X}(t + \Delta t) + \beta_2 \, T_{i,j}^{indirect,\ X}(t + \Delta t) \tag{7.2}$$

In Equation (7.2), $T_{i,j}^{direct,X}(t+\Delta t)$ and $T_{i,j}^{indirect,X}(t+\Delta t)$ are "direct trust" (based on direct observations) and "indirect trust" (based on recommendations) of node *i* toward node *j* in *X* at time $t+\Delta t$, respectively, and β in the range of [0, 1] is a parameter to weigh node *i*'s own direct trust assessment toward node *j*. Every trust property *X* has its own specific β value under which subjective $T_{i,j}^{X}(t)$ obtained is accurate, i.e., close to actual status of node *j* in *X* at time *t*.

7.2.1 Trust Update Upon Node *i* Encountering Node *j*

Upon encountering node *j* at time *t*, node *i* updates "direct trust" $T_{i,j}^{direct,X}(t+\Delta t)$ in Equation (7.2) based on "direct" observations or interaction experiences with node *j* over the encounter interval $[t, t+\Delta t]$. When a monitoring node (node *i*) cannot properly monitor a trustee node (node *j*) upon encounter because of a short contact time, it adapts to this situation by discarding the current monitoring result and instead updating direct trust by its past direct trust toward *j* decayed over the time interval Δt to model trust decay over time. Specifically, let $C_{i,j}^{direct,X}(t)$ be a boolean variable indicating if the needed data (discussed below) for assessing *X* is obtainable within Δt. Then, $T_{i,j}^{direct,X}(t+\Delta t)$, node *i*'s trust in *X* toward node *j* at time $t+\Delta t$ upon encounter at time *t* is calculated by:

$$T_{i,j}^{direct,X}(t+\Delta t) = \begin{cases} T_{i,j}^{encounter,X}(t+\Delta t), & if\ C_{i,j}^{direct,X}(t) = true \\ e^{-\lambda_d \Delta t} \times T_{i,j}^{direct,X}(t), & if\ C_{i,j}^{direct,X}(t) = false \end{cases} \quad (7.3)$$

In other words, node *i* will update $T_{i,j}^{direct,X}(t+\Delta t)$ with its new direct trust toward node *j* in property *X* only if node *i* directly encounters node *j* at time *t* and the data needed for assessing *X* is obtainable within the encounter interval Δt; otherwise, node *i* will simply update $T_{i,j}^{direct,X}(t+\Delta t)$ with its past experience $T_{i,j}^{direct,X}(t)$ decayed over Δt. We adopt an exponential time decay factor, $e^{-\lambda_d \Delta t}$ (with $0 < \lambda_d \leq 0.1$ to limit the decay to at most 50%).

Node *i* assesses $T_{i,j}^{encounter,X}(t+\Delta t)$ based on data collected from direct observations toward node *j* over the encounter interval $[t, t+\Delta t]$ as follows:

- $T_{i,j}^{encounter,healthiness}(t+\Delta t)$: Node *i* assesses node *j*'s unhealthiness based on evidences manifested due to malicious attacks including self-promoting, bad-mouthing and ballot stuffing attacks. Evidences of self-promoting attacks may be detected through the encounter history exchanged from node *j*. If the encounter history is not certified (e.g., using encounter tickets as in [34, 112]), or is certified but inconsistent with node *i*'s encounter history matrix accumulated, it is considered as a negative

experience. Evidences of bad-mouthing/ballot stuffing attacks may be detected by comparing node *j*'s recommendation toward node *q* with the trust value of node *i* toward node *q* itself. If the percentage difference is higher than a threshold, it is considered suspicious and thus a negative experience. These positive/negative experiences are collected over the new encounter period $[t, t+\Delta t]$ to assess $T_{i,j}^{encounter,healthiness}(t+\Delta t)$. It is computed by the number of positive experiences over the total experiences in healthiness-related behavior.

- $T_{i,j}^{encounter,unselfishness}(t+\Delta t)$: Our notion of social selfishness is that friends will be cooperative toward each other even if they are selfish. Hence, from node *i*'s perspective if node *j* is a friend of the source node, node *i*, or node *d* (the destination) then $T_{i,j}^{encounter,unselfishness}(t+\Delta t)$ is 1. Otherwise, node *i* will hope that node *j* is altruistic by examining the protocol compliance degree of node *j*. Specifically, node *i* applies monitoring techniques to detect altruistic behaviors, e.g., whether or not node *j* follows the prescribed protocol over $[t, t+\Delta t]$. Evidence of altruism is manifested by the behavior for executing beacon, encounter history exchange, packet receipt acknowledgement, and trust evaluation protocols expected out of node *j*. $T_{i,j}^{encounter,unselfishness}(t+\Delta t)$ is then computed by the number of positive experiences over the total experiences in unselfishness-related behavior. Here we note that node *i* will not monitor if node *j* has forwarded a packet since it is impractical to monitor packet forwarding in DTNs.
- $T_{i,j}^{encounter,connectivity}(t+\Delta t)$: While there is no pre-determined connectivity pattern in DTNs, the connectivity of one node (*j*) to another node (*d*) is inherently associated with its mobility pattern and its social activities. This trust property represents the connectivity of node *j* to the destination node *d*. If the connectivity trust is high, then node *j* would be a good candidate for packet delivery to node *d*. Node *i* deduces node *j*'s connectivity with node *d* based on its encounter matrix collected over $[0, t+\Delta t]$, including the new encounter history received from node *j*. Note that node *i* should only accept a certified encounter history (as in [34, 112]) to avoid black hole attacks.
- $T_{i,j}^{encounter,energy}(t+\Delta t)$: This trust property represents the capability or competence of node *j* to do the basic routing function. Node *i* counts the ratio of the number of acknowledgement packets received from node *j* (at the MAC layer) over transmitted packets to node *j*, over $[t, t+\Delta t]$, to estimate energy status in node *j*.

On the other hand, since there is no new "indirect trust" in this case, node *i* simply updates $T_{i,j}^{indirect,X}(t+\Delta t)$ with its past experience $T_{i,j}^{indirect,X}(t)$ decayed over Δt, i.e.,

$$T_{i,j}^{indirect,\ X}(t+\Delta t) = e^{-\lambda_d \Delta t} \times T_{i,j}^{indirect,X}(t) \tag{7.4}$$

7.2.2 Trust Update Upon Node *i* Encountering Node *m* ($m \neq j$)

When node *i* encounters node *m*, $m \neq j$, node *i* uses its 1-hop neighbors (including node *m*) as recommenders to update "indirect trust" $T_{i,j}^{indirect,X}(t+\Delta t)$ in Equation (7.2). An application-level optimization parameter is the recommender trust threshold T_{rec} for the selection of recommenders. Using T_{rec} provides robustness against bad-mouthing or ballot stuffing attacks since only recommendations from more trustworthy nodes are considered. The indirect trust evaluation toward node *j* is given in Equation (7.5) below where R_i is the set containing node *i*'s 1-hop neighbors with $T_{i,k}(t) \geq T_{rec}$ and $|R_i|$ indicates the cardinality of R_i. If node *i* considers node *k* as a trustworthy recommender, i.e., $T_{i,k}(t) \geq T_{rec,}$ then node *k* is allowed to provide its recommendation to node *i* for evaluating node *j*. In this case, node *i* weighs node *k*'s recommendation, $T_{k,j}^{X}(t)$, with node *i*'s referral trust, $T_{i,k}^{X}(t)$, toward node *k*. We aggregate indirect trust recommendations using a form of normalized weighted calculation based on the referral trust value of the recommender.

$$T_{i,j}^{indirect,\ X}(t+\Delta t) = \begin{cases} e^{-\lambda_d \Delta t} \times T_{i,j}^{indirect,X}(t), & if\ |R_i| = 0 \\ \dfrac{\sum_{k \in R_i}\{T_{i,k}^{X}(t) \times T_{k,j}^{X}(t)\}}{\sum_{k \in R_i} T_{i,k}^{X}(t)}, & if\ |R_i| > 0 \end{cases} \tag{7.5}$$

On the other hand, since there is no new "direct trust" in this case, node *i* simply updates $T_{i,j}^{direct,X}(t+\Delta t)$ with its past experience $T_{i,j}^{direct,X}(t)$ decayed over Δt, i.e.,

$$T_{i,j}^{direct,\ X}(t+\Delta t) = e^{-\lambda_d \Delta t} \times T_{i,j}^{direct,X}(t) \tag{7.6}$$

7.2.3 Encounter-Based DTN Routing

When node *i* encounters node *j*, it uses $T_{i,j}(t)$ from Equation (7.1) to decide whether or not node *m* can be the next message carrier to shorten message delay or improve message delivery ratio. Another application-level optimization parameter is the minimum trust threshold T_f for the selection of the next message carrier. Node *i* will forward the message to node *j* only if $T_{i,j}(t) \geq T_f$ and $T_{i,j}(t)$ is in the top Ω percentile among all $T_{i,m}(t)$'s. This helps the chance of selecting a trustworthy next message carrier.

7.3 Performance Model

We validate our trust management designs by a novel model-based analysis methodology via extensive simulation. Specifically we develop a mathematical model based on continuous-time semi-Markov stochastic processes (for which the event time may follow any general distribution) to define a DTN consisting of a large number of mobile nodes exhibiting heterogeneous social and QoS behaviors.

We take the concept of "operational profiles" in software reliability engineering [141] as we build the mathematical model. An operational profile is what the system expects to see during its operational phase. During the testing and debugging phase, a system would be tested with its anticipated operational profile to reveal design faults. Failures are detected and design faults causing system failures are removed to improve the system reliability. The operational profile of a DTN system specifies the operational and environmental conditions. Typically this would include knowledge regarding (a) hostility such as the expected % of misbehaving nodes and if it is evolving the expected rate at which nodes become malicious or selfish or even the expected % of misbehaving nodes as a function of time; (b) mobility traces providing information of how often nodes meet and interact with each other; (c) behavior specifications defining good behavior and misbehavior during protocol execution; and (d) resource information such as how fast energy is consumed.

We develop a probability model based on Stochastic Petri Net (SPN) techniques [178] to describe a DTN, given an operational profile as input. The SPN model for a DTN node is shown in Figure 7.1 consisting of 4 places, namely, *energy*, *location*, *maliciousness* and *selfishness*. The underlying state machine is a semi-Markov model with 4-component states, i.e., (*energy*, *location*, *maliciousness*, *selfishness*), where *energy* is an integer holding the amount of energy left in the node, *location* is an integer holding the location of the node, *maliciousness* is a binary variable with 1 indicating the node is malicious and 0 otherwise, and *selfishness* is a binary variable with 1 indicating the node is socially selfish and 0 otherwise. A selfish node will forward a packet only if the source, current carrier or the destination is in its friend list. Here we note that a node's trust value actually is a real number in [0, 1]; it is calculated by a state-probability weighed sum of trust values assigned to the states of the underlying semi-Markov model of the SPN performance model, i.e., trust value = $\sum_i$(state probability of state i × trust value in state i). In some states, the trust value is binary. For example in a state in which a node is compromised, the trust value for property "healthiness" in this state is 0. Note that each node has its own SPN model. So there are as many SPN models as they are nodes in the DTN. The operational profile specifies the % of malicious nodes and the % of socially selfish nodes. Thus, some nodes will be malicious in accordance with this specification. Similarly some nodes will be selfish based on the % of selfish nodes.

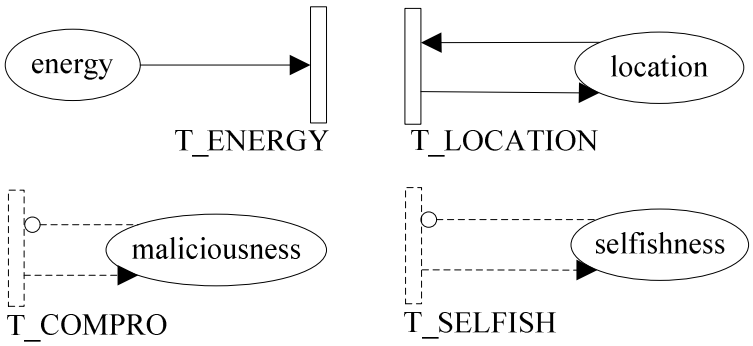


Figure 7.1: SPN Model for a Node in the DTN.

The purpose of the SPN model is to yield *ground truth* status of a node in terms of its healthiness, unselfishness, connectivity, and energy status. Then we can check *subjective trust* against *ground truth* status for validation of trust protocol designs. Below we explain how we leverage the SPN model to determine a node's ground truth status.

Location (Connectivity): The connectivity trust of node m toward node d is measured by the probability that both node m and node d are in the same location at time t. We use the *location subnet* to describe the location status of a node. Transition T_LOCATION is triggered when the node moves to a new area from its current location according to its mobility pattern. We consider both synthetic mobility models and real mobility traces. This information along with the location information of other nodes at time t provides us the probability of two nodes encountering with each other at any time t.

Energy: We use the *energy subnet* to describe the energy status of a node. Place *energy* represents the current energy level of a node. An initial energy level (E_0) of each node represented by a number of tokens is assigned according to node heterogeneity information. A token is taken out when transition T_ENERGY fires representing the energy consumed during protocol execution, packet forwarding and/or performing attacks in the case of a malicious node. The rate of transition T_ENERGY indicates the energy consumption rate which varies depending on the ground truth status of the node (i.e., malicious or selfish). The operational profile specifies the energy consumption rate of a malicious node vs. a selfish node vs. a well-behaved node.

Healthiness: A malicious node is necessarily unhealthy. So we will know the ground truth status of healthiness of the node by simply inspecting if place *maliciousness* contains a token.

Unselfishness: A socially selfish node drops packets unless the source, current carrier or the destination node is in its friend list. We will know the ground truth status of unselfishness of the node by simply inspecting if place *selfishness* contains a token.

Dynamically Changing Environment Conditions: With the goal to deal with malicious and selfish nodes in DTN routing, we consider a dynamically changing environment in which the number of misbehaving nodes (malicious or selfish) is changing over time. A node becomes malicious when it is captured and turned into a compromised node, as dictated by the per-node capture rate. The SPN output provides the probability that a node is compromised at time t. We model the capture event by a transition T_COMPRO (in dashed line) in Figure 7.1. Once the transition T_COMPRO is triggered, a token will be moved into the place *maliciousness* representing that this node is compromised. Similarly, once the transition T_SELFISH (also in dashed line) is triggered, a token will be moved into the place *selfishness* representing that this node becomes selfish. The transition rates of T_COMPRO and T_SELFISH are λ_c and λ_s, respectively. We will use the SPN model augmented with the two dashed line transitions in Section 7.6 in which we treat the subject of dynamic trust management.

Objective Trust Evaluation: The SPN model described above yields actual or ground truth status of each node. The "objective" trust of node j at time t, denoted by $T_j(t)$, is also obtained from Equation 1 except that $T_j^X(t)$ is being used instead of $T_{i,j}^X(t)$. Here $T_j^X(t)$ is simply the actual or ground truth status of node j in trust property X at time t obtainable from the SPN model for node j. The notion of "objective" trust evaluation is to validate subjective trust evaluation, that is, subjective trust evaluation is valid if the subjective trust value obtained as a result of executing our dynamic trust management protocol is accurate with respect to the objective trust value obtained from ground truth.

7.4 Numerical Results

In this section we present numerical results generated from the SPN model. Our trust evaluation results have two parts. The first part is about the convergence and accuracy of trust aggregation for individual trust properties. The second part is about maximizing application performance through trust formation (by setting the best weights to trust properties) and application-level trust optimization (by setting the best recommender trust threshold $T_{rec,}$ and message carrier trust threshold T_f). Because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, each trust property X has its own best set of (β, λ_d) under which $T_{i,j}^X(t)$ obtained from Equation (7.2) would be the most accurate, i.e., closest to actual status of node j in trust property X, or $T_j^X(t)$. Recall that a higher β value indicates that subjective trust evaluation relies more on direct observations compared with indirect recommendations provided by the recommenders and that a higher λ_d indicates a higher trust decay rate. Once we ensure the accuracy of each trust property X, we can then address the trust formation issue, i.e., identifying the best way to form the overall trust out of QoS

and social trust properties and the best way to set application-level trust parameters such that the application performance (i.e., secure routing) is maximized.

Table 7.1: System Parameter.

Param	Value	Param	Value	Param	Value	Param	Value
$m \times m$	16×16	R	250m	P_{error}	5%	P_{rand}	[0, 1]
E_0	100 *hrs*	N	20	Slope of SWIM	1.45	Pause of SWIM	≤ 4 hrs

Table 7.1 lists a set of parameters and their values (for input parameters) as prescribed by the operational profile of a DTN. We consider N = 20 nodes moving according to the SWIM mobility model [108] modeling human social behaviors in an $m \times m$=16×16 (4km×4km) operational region, with each region coving R = 250m radio radius. We use SWIM in this section for numerical results. Later in Section 7.6 we also use traces in our simulation studies. The initial energy of each node E_0 is set to 100 hours lifetime. The error probability of *direct trust* assessment of $T_{i,j}^{direct,X}(t + \Delta t)$ due to environment noise denoted by P_{error} is set to 5%. For X=healthiness, the false positive probability P_{fp} of misidentifying a healthy node as an unhealthy node is equivalent to P_{error}, i.e., 5%. For a compromised node performing random attacks with probability P_{rand} (in the range of [0, 1]) to evade detection, the false negative probability P_{fn} for missing an unhealthy node as a healthy node is $P_{error}P_{rand} + (1 - P_{error})(1 - P_{rand})$. That is, P_{fn} is P_{error} with probability P_{rand} (if attacking) and $1 - P_{error}$ with probability $1 - P_{rand}$ (if not attacking). We set $C_{i,j}^{direct,X}$ (in Equation 3) to true if the encounter duration is longer than 10 minutes as it would allow sufficient data to be collected for direct trust assessment of X; we set it to false otherwise.

In SWIM [22], a node has a home location and a number of popular places. A node makes a move to one of the population places based on a prescribed pattern. The probability of a location being selected is higher if it is closer to the node's home location or if it has a higher *popularity* (visited by more nodes). Once a node has chosen its next destination, it moves towards the destination following a straight line and with a constant speed that equals the movement distance. When reaching the destination, the node pauses at the destination location for a period of time following a bounded power law distribution with the slope set to 1.45 (as in [108]) and the upper-bound pause time set to 4 hours.

The weight of direct trust (β), trust decay parameter (λ_d), trust threshold for the recommender (T_{rec}), trust threshold for the next carrier (T_f) and weight of trust property X (w^X) are design parameters whose best settings are to be determined as output. Here we

should note that a social friendship matrix [114] and the percentages of selfish and malicious nodes, although not specified in Table 7.1, are also given as input, which we will vary in the analysis to test their effects on design parameters. Lastly, the node compromise rate (λ_c) and node selfishness rate (λ_s) for characterizing changing DTN conditions are also not specified in Table 7.1. We will consider these two parameters and treat the subject of dynamic trust management in Section 7.6.

7.4.1 Best Trust Propagation Protocol Settings to Minimize Trust Bias

Here we determine the best (β, λ_d) values that yield subjective trust evaluation closest to objective trust evaluation to minimize trust bias, given a set of parameter values as listed in Table 7.1 characterizing the operational and environmental conditions. We fix the percentage of selfish nodes to 30% and vary the percentage of malicious nodes from 0% to 45% to examine its effect. We set the recommender trust threshold (T_{rec}) to 0.6, since the trust value of a malicious node is likely to be lower than ignorance (0.5), so $T_{rec} \geq 0.6$ can effectively filter out false recommendations from malicious nodes. Since there are only two input parameters, we search the best (β, λ_d) for each trust property through exhaustive search, i.e., we compare subjective trust obtained through protocol execution under a given (β, λ_d) with objective trust. The best (β, λ_d) combination is the one that produces the lowest mean square error (MSE). This information determined at static time is recorded in a table to be used by dynamic trust management which we will discuss later in Section 7.6.

In Table 7.2, we summarize the best (β, λ_d) values for each trust property for minimizing trust bias, given the % of malicious nodes as input, for a trustor node (i.e., node i) randomly picked toward a trustee node (i.e., node j) also randomly picked. Each (β, λ_d) entry represents the best combination under which subjective trust $T_{i,j}^X(t)$ obtained as a result of executing our trust aggregation protocol for trust property X (as prescribed by Equation (7.2)) deviates the least from objective trust for property X (that is, $T_j^X(t)$). We have observed for all cases the most deviation is 3% MSE. This substantiates our claim that there exists a distinct best protocol setting in terms of (β, λ_d) for each trust property X, with X = connectivity, energy, healthiness or unselfishness. Furthermore, the best (β, λ_d) setting changes as the % of misbehaving nodes changes dynamically.

Table 7.2: Best (β, λ_d) to Minimize Trust Bias.

% of malicious nodes	Healthiness (β, $\lambda_d \times 10^4$)	Unselfishness (β, $\lambda_d \times 10^4$)	Connectivity (β, $\lambda_d \times 10^4$)	Energy (β, $\lambda_d \times 10^4$)
0%	(0.44, 0.0)	(0.41, 0.2)	(0.80, 10)	(0.39, 0.1)
5%	(0.39, 0.0)	(0.41, 0.2)	(0.80, 10)	(0.39, 0.1)
10%	(0.40, 0.0)	(0.39, 0.0)	(0.80, 10)	(0.39, 0.1)
15%	(0.39, 0.0)	(0.37, 0.0)	(0.86, 10)	(0.39, 0.1)
20%	(0.41, 0.0)	(0.33, 0.0)	(0.91, 10)	(0.39, 0.1)
25%	(0.35, 0.0)	(0.30, 0.0)	(0.91, 10)	(0.48, 0.5)
30%	(0.35, 0.0)	(0.28, 0.0)	(0.91, 10)	(0.47, 0.0)
35%	(0.35, 0.0)	(0.26, 0.0)	(0.95, 10)	(0.49, 0.5)
40%	(0.35, 0.0)	(0.21, 0.1)	(0.95, 10)	(0.49, 0.5)
45%	(0.35, 0.0)	(0.22, 0.5)	(0.95, 10)	(0.58, 0.5)

7.4.2 Best Trust Formation Protocol Settings to Maximize Application Performance

Next we turn our attention to the trust formation issue to optimize application performance. For the secure routing application, two most important performance metrics are message delivery ratio and delay. In many situations, however, excessive long delays are not acceptable to DTN applications. We define the delivery ratio as the percentage of messages that are delivered successfully within an application deadline which is the maximum delay the application can tolerate. While our protocol is generic to any deadline, we set the deadline (or a time-to-live limit) to 2 hours to reveal the tradeoff between delay and delivery ratio in this environment setting for DTN routing. Our goal is to find the best way to assign the weight w^X to X = healthiness, unselfishness, connectivity or energy to maximize the delivery ratio. Since the search space is small, we perform exhaust search to identify the best trust formation (w^X with an increment of 0.1) under which delivery ratio is maximized. This information again is recorded in a table to be used for dynamic trust management (Section 7.6). We assume that a malicious node drops all packets. A selfish node drops part of packets it receives depending on if it knows the source, current carrier or destination node socially (whether these nodes are in its friend list).

We consider two variations of secure routing protocols: single-copy forwarding (L = 1) and multi-copy forwarding (L ≥ 2), where L is the maximum number of carriers to

which a node can forward a message. Below we discuss how we identify the best setting for double-copy forwarding (L = 2). The best setting for other cases (L = 1 or L > 2) can be obtained in a similar way.

Table 7.3: Best Trust Formation to Maximize Delivery Ratio.

% of malicious nodes	$w^{healthiness}$	$w^{unselfishness}$	$w^{connectivity}$	w^{energy}
0%	0.0	0.6	0.4	0.0
5%	0.1	0.8	0.1	0.0
10%	0.3	0.3	0.3	0.1
15%	0.3	0.3	0.3	0.1
20%	0.3	0.3	0.3	0.1
25%	0.3	0.3	0.2	0.2
30%	0.3	0.3	0.3	0.1
35%	0.3	0.3	0.3	0.1
40%	0.4	0.3	0.2	0.1
45%	0.4	0.3	0.2	0.1

Table 7.3 summarizes the best trust formation for maximizing delivery ratio under double-copy forwarding, given the percentage of malicious nodes as input. We first observe there is a distinct set of optimal weight settings under which delivery ratio is maximized. Second, the optimal weight of the *healthiness* trust property increases as the % of malicious node increases. This is because in hostile environments, using a higher weight on *healthiness* helps identify malicious nodes to avoid message loss.

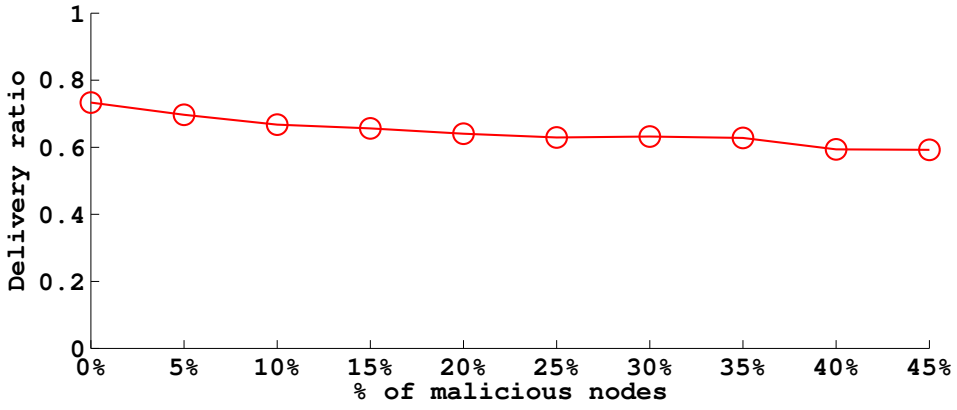


Figure 7.2: Delivery Ratio under Best Trust Formation.

Figure 7.2 correspondingly shows the maximum delivery ratio obtainable when the system operates under the best trust formation setting identified. We see that the delivery ratio remains high even as the % of malicious nodes increases to as high as 45%. This to some extent demonstrates the resiliency property of our trust-based routing protocol against malicious attacks.

7.4.3 Best Application-Level Trust Optimization Design Settings to Maximize Application Performance

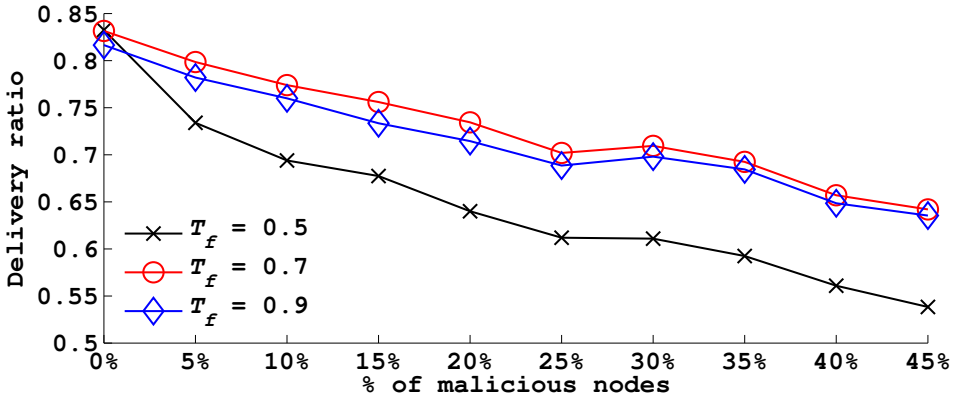


Figure 7.3: Effect of T_f on Deliver Ratio.

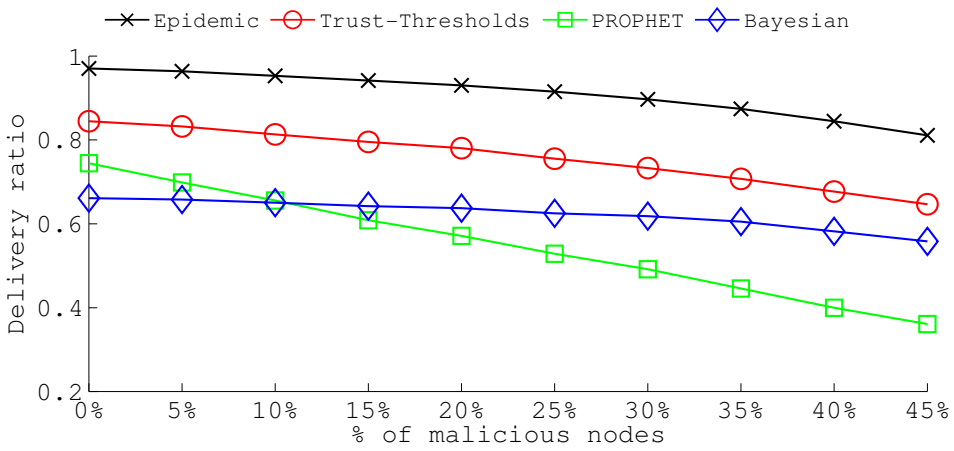
In this section, we apply the application-level trust optimization design in terms of the best message carrier trust threshold T_f to maximize delivery ratio in response to changing hostility reflected by the % of malicious nodes. Figure 7.3 shows delivery ratio vs. T_f with the percentage of malicious nodes varying in [0 - 45%]. We set the trust recommender threshold, T_{rec}, at 0.6 to isolate out its effect. We notice that there is an optimal T_f value under which delivery ratio is maximized. With the environment setting (30% selfish nodes and 0 to 45% malicious nodes), the optimal value T_f value is around 0.7. The reason is that using a higher value of T_f helps generate a higher message delivery ratio by choosing only the most trustworthy nodes as message carriers, but it also introduces a higher message delay. Therefore, T_f = 0.7 is the best setting to balance the tradeoff between message delivery ratio vs. message delay, except for the case when there are little malicious nodes for which T_f = 0.5 is the best setting.

7.4.4 Comparative Analysis

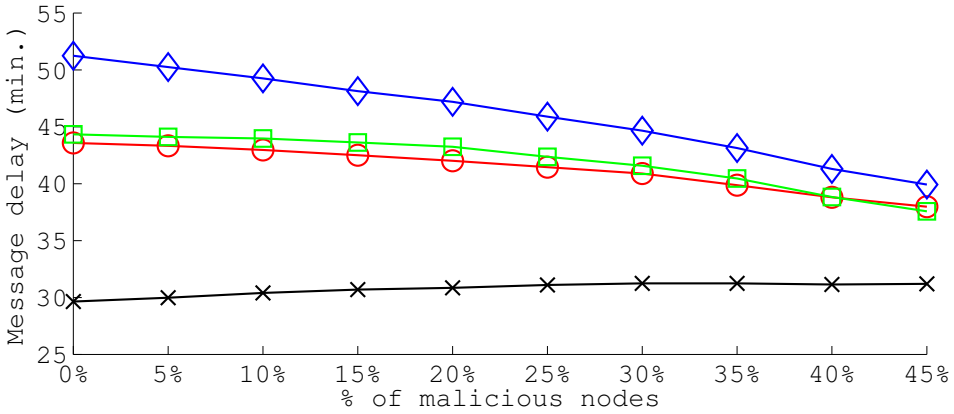
Lastly we conduct a comparative analysis, contrasting our trust-based protocol operating under the best settings identified with Bayesian trust-based routing [68, 96] and non-trust based (PROPHET [119] and epidemic [179]) protocols. PROPHET [119] uses the history of encounters and transitivity to calculate the probability that a node can de-

liver a message to a particular destination; it is considered as a benchmark "non-trust based" forwarding algorithm for DTNs in the literature. Bayesian trust-based routing on the other hand relies on the use of trust information maintained by a Bayesian based trust management system (such as a Beta reputation system [68, 96]) to make routing decisions. In a Bayesian trust management system, the trust value is assessed using the Bayes estimator, updated by both direct observations and indirect recommendations. The direct observations are directly used to update the number of positive and negative observations, whereas the recommendations are discounted by the confidence [68] or belief [96] of the trustor toward the recommender. Under Bayesian trust-based routing, a node is chosen as the message carrier only if its trust value is in the top Ω percentile and higher than the message carrier trust threshold T_f. We choose Bayesian trust-based routing because of its dominance in trust/reputation systems. We again consider double-copy forwarding (with L = 2) with nodes following the SWIM mobility model. For our trust-based secure routing protocol, we use the best settings for double-copy forwarding as identified earlier.

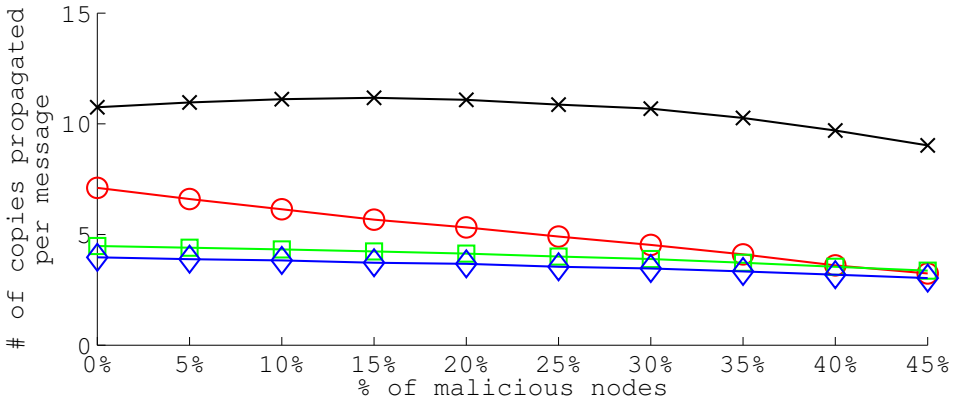
Figure 7.4 compares the message delivery ratio, delay, and overhead generated by our trust protocol against Bayesian trust-based, PROPHET, and epidemic routing protocols. The results demonstrate that our trust-based secure routing protocol designed to maximize delivery ratio can effectively trade off message overhead for a significant gain in delivery ratio. In particular, our trust-based routing protocol outperforms Bayesian trust-based routing and PROPHET in delivery ratio as it applies the best trust formation out of social and QoS trust properties. Furthermore, our trust-based routing protocol also outperforms Bayesian trust-based and PROPHET in message delay except when there is a very high % of malicious nodes (e.g., 40-45% of malicious nodes) in the system. The reason is that when there is a high % of malicious nodes, our protocol tends to use a higher weight for healthiness and consequently a lower weight for connectivity, thus causing a higher message delay. Lastly, the message overhead of our trust-based routing protocol is significantly lower than epidemic routing. We conclude that our trust-based protocol approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message overhead.



(a) Delivery Ratio.



(b) Message Delay.



(c) Message Overhead.

Figure 7.4: Performance Comparison (Analytical Results based on SWIM Mobility).

7.5 Simulation Validation

We validate analytical results through extensive simulation using ns-3 [1]. The simulated DTN environment is setup as described in Table 7.1. We simulate two mobility patterns: a synthetic mobility model (SWIM) and real mobility traces. We investigate four

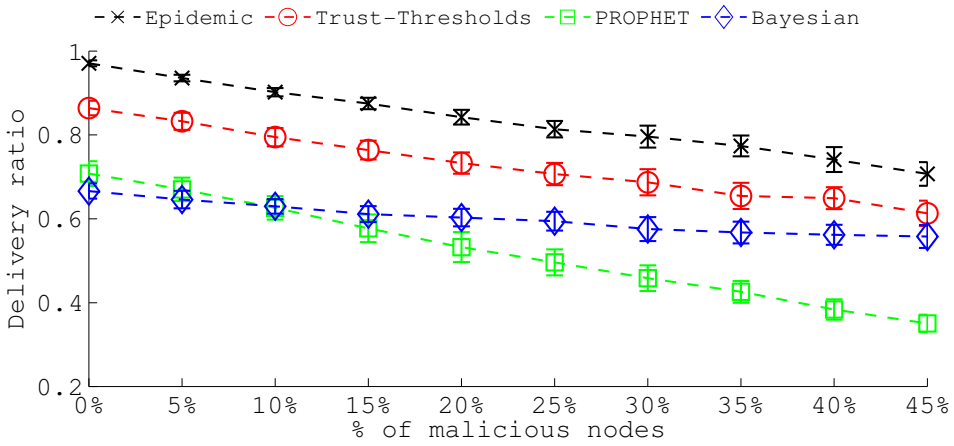
mobility traces from [165], namely *Intel*, *Cambridge*, *Infocom05* and *Infocom06*. Table 7.4 summarizes the experimental settings under which these mobility traces are obtained. During the experiment, each *internal device* records the contact/encounter event with other devices (*internal* or *external*). Due to the fact that the contact events between external devices are not recorded in the traces, we only consider internal devices in our simulation. We conduct sufficient simulation runs with disjoint random number streams and collect observations such that 5% accuracy and 95% confidence level requirements are satisfied. We mark the standard deviation from the mean by error bars in the data figures presented in this section.

Table 7.4: Experiment Setting for Mobility Traces.

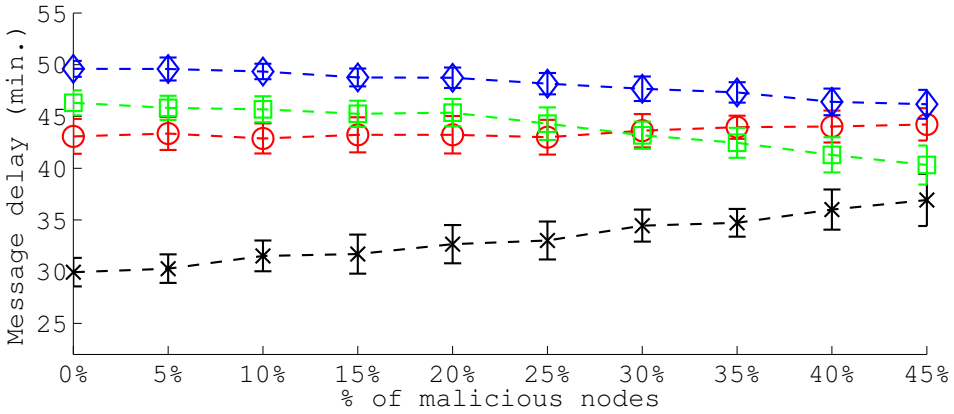
Trace	*Intel*	*Cambridge*	*Infocom05*	*Infocom06*
Participants	Researches & interns	Students & faculty	conference attendees	conference attendees
Experiment Time	4 days	5 days	3 days	4 days
Internal Devices	9 with 1 stationary	12	41	98 with 20 stationary
External Devices	119	211	233	4626

7.5.1 Simulation Results based on SWIM Mobility

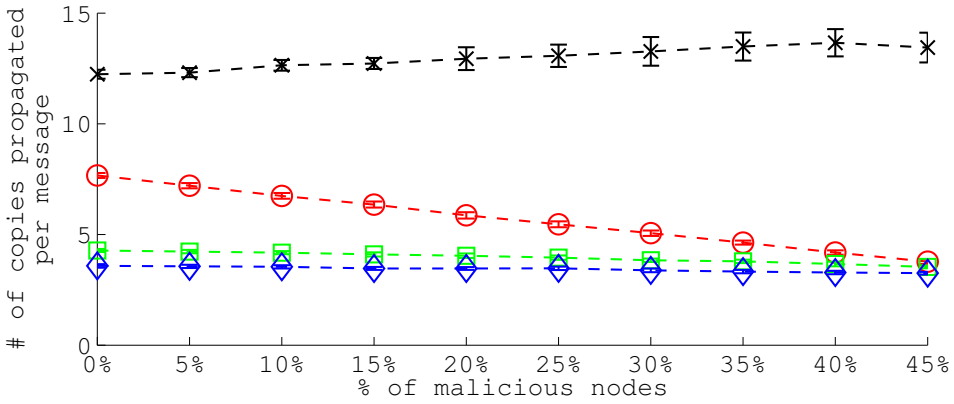
Figure 7.5 shows the simulation results in message delivery ratio, message delay, and message overhead of DTN routing under the SWIM mobility model, corresponding to the analytical results in Figure 7.4 obtained earlier. We observe that the simulation results in Figure 7.5 are virtually identical to the analytical results shown in Figure 7.4, with the MSE between the simulation results vs. the analytical results bounded by 3%.



(a) Delivery Ratio.



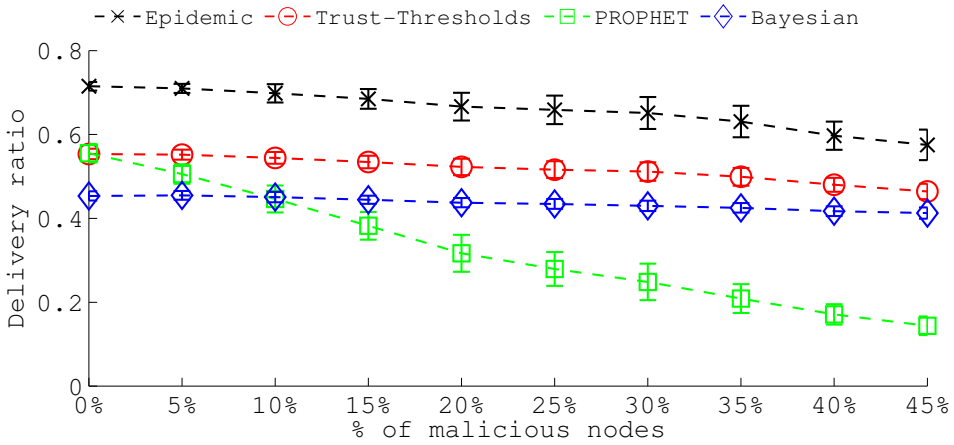
(b) Message Delay.



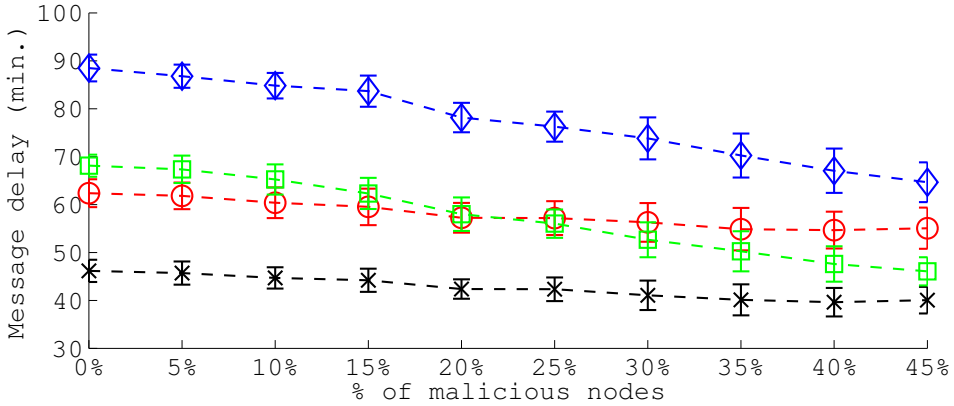
(c) Message Overhead.

Figure 7.5: Simulation Results Corresponding to Analytical Results in Figure 4 based on SWIM Mobility.

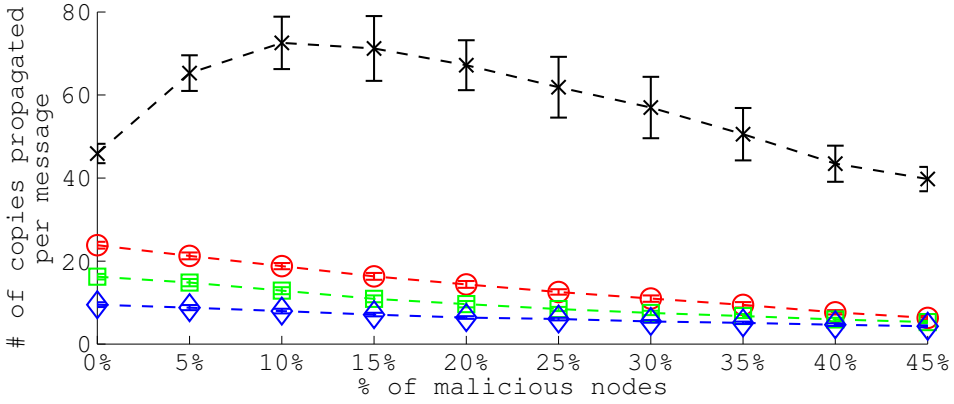
7.5.2 Simulation Results based on Mobility Traces



(a) Delivery Ratio.



(b) Message Delay.



(c) Message Overhead.

Figure 7.6: Performance Comparison of Routing Protocols based on Mobility Traces.

Figure 7.6 shows the simulation results of comparing our trust-based secure routing protocol against Bayesian trust-based routing, PROPHET, and epidemic routing protocols, based on *infocom06* mobility traces [165]. We choose *infocom06* over others since it consists of more nodes and lasts longer. The results of the other three mobility traces

exhibit the same trend and thus are not shown here. Briefly, the *infocom06* trace data contain encounter events collected by Bluetooth devices carried by conference attendees. There were a total of 98 Bluetooth devices (20 stationary nodes) used to record the encounter events over a period of four days. We select 78 mobile nodes in our simulation and use the encounter events in the traces as the time instances to perform trust updating and message forwarding (executed by each node). In each simulation run, we randomly pick a number of nodes as selfish nodes (30%) and malicious nodes (from 0% to 45%) and generate a social friendship matrix [114]. A malicious node performs attacks to disrupt the trust of the DTN, including self-promoting, ballot stuffing and bad-mouthing attacks. An altruistic node always forwards messages. A selfish node forwards a message only when it is a friend of the source, current carrier, or destination.

We first observe that Figure 7.6 obtained based on mobility traces exhibit virtually the same trends as Figure 7.5 obtained based on the SWIM mobility model. This supports our claim that our trust-based secure routing protocol can significantly outperform Bayesian trust-based routing and PROPHET in message delivery ratio regardless of the node encountering pattern. We further observe that Figure 7.6 (displaying simulation results based on traces) exhibits remarkably similar trends as Figure 7.4 (displaying analytical results based on SWIM movements) in terms of ranking routing protocols in delivery ratio, delay and overhead. As both simulation results based on traces (Figure 7.6) and SWIM movements (Figure 7.5) correlate well with analytical results (Figure 7.4), we conclude that the analytical results obtained, along with the conclusions drawn, are valid.

7.5.3 Protocol Convergence, Accuracy and Resiliency

In this section we present simulation results to demonstrate trust assessment accuracy, convergence and resiliency properties of our protocol and compare it against Bayesian trust management. We use the healthiness trust property as an example, because unlike all others it has an additional false negative probability parameter (P_{fn}) due to the possibility of a compromised node performing random attacks with probability P_{rand} to evade detection. Again we set P_{error} = 5% for direct detection error probability due to environment noises and P_{fn}=$P_{error}P_{rand} + (1 - P_{error})(1 - P_{rand})$. Also we set the % of malicious nodes to 30% so as to manifest the effect of random attacks.

Figure 7.7 shows the healthiness trust of a randomly selected healthy node (node *i*) toward a randomly selected compromised node (node *j*) by executing our dynamic trust management protocol, i.e., $T_{i,j}^{healthiness}(t)$, as a function of time *t* with the random attack probability P_{rand} of node j varying in [0, 1]. For the clarity of presentation, we do not show confidence intervals as error bars in the figure. Initially, the confidence interval at 95% confidence level is close to 0.3 due to lack of observations. As time increases and

the number of observations increases, the confidence interval decreases to less than 0.02. This is also reflected by the trust fluctuation over time in Figure 7.7. We follow the strong attack model, that is, when a malicious node performs attacks, it will provide the highest trust value, i.e., 1, for other malicious nodes for good-mouthing attacks and the lowest trust value, i.e., 0, against good nodes for bad-mouthing attacks. We first observe that $T_{i,j}^{healthiness}(t)$ eventually converges to a trust value. The warm-up time to build up trust depends on the mobility pattern and encounter frequency. Second, we observe that the trust value is close to P_{fn} after convergence. Specifically, $T_{i,j}^{healthiness}(t)$ is close to 0.95 for a malicious node exhibiting no evidence of attacks with $P_{rand} = 0$; it is close to 0.05 for a malicious node performing reckless attacks with $P_{rand} = 1$; and it is close to 0.68 for a malicious node performing attacks with $P_{rand} = 0.3$. This demonstrates that both trust convergence and accuracy properties are preserved by our protocol with the converged trust value reflecting ground truth status.

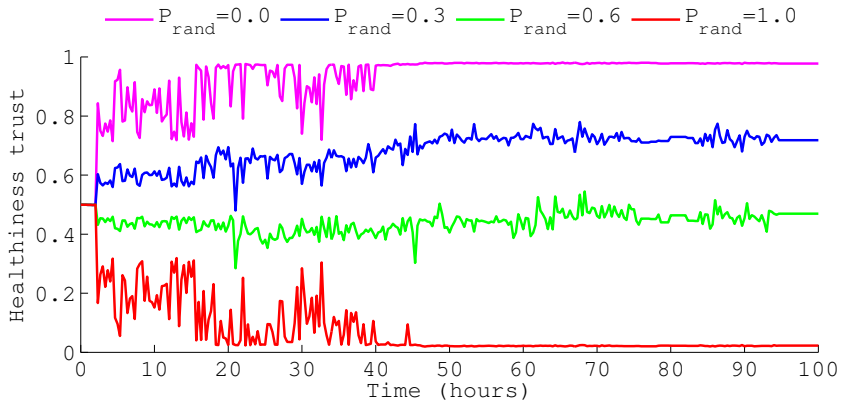
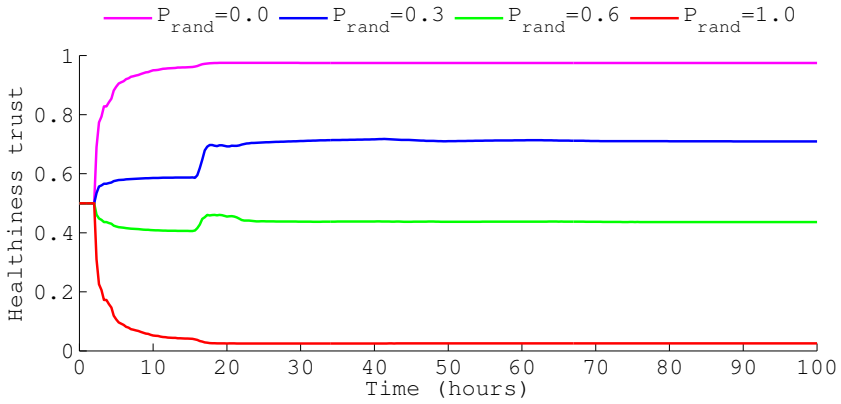


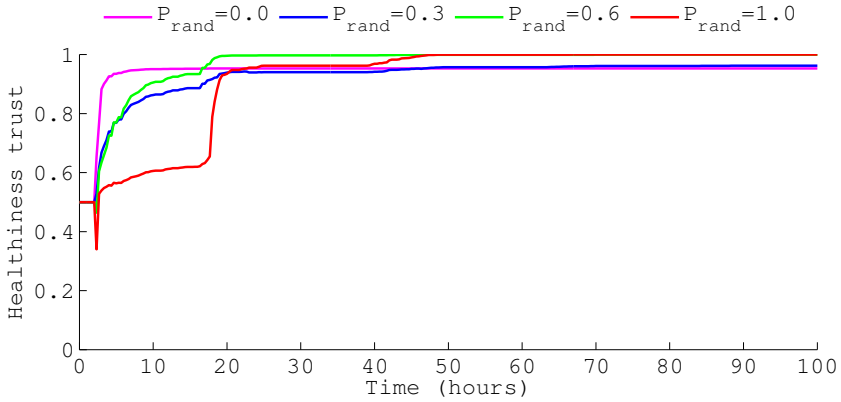
Figure 7.7: Healthiness Trust Evaluation Results of Dynamic Trust Management under Random Attacks.

Figure 7.8 shows the healthiness trust by executing Bayesian trust management, with the same trustor (node *i*) and trustee (node *j*) selected as in Figure 7.7. Similar to Figure 7.7, as time increases, the confidence interval at 95% confidence level decreases from about 0.3 to less than 0.03. In Bayesian trust management, a trustor node maintains two numbers, i.e., a positive count and a negative count, towards a trustee. The trust value is calculated as the positive count over the sum of positive and negative counts. Similarly, a recommendation in Bayesian trust also consists of a positive count and a negative count. A malicious node can provide a high positive count and a low negative count for a compromised node for good-mouthing attacks, and provide a low positive count and a high negative count against a good node for bad-mouthing attacks. For fair comparison, we consider two attack models: *weak* and *strong*. In the weak attack model, the positive count for a malicious node or the negative count against a good node is kept low, i.e., less than the number of direct observations made by the trustor node toward the trustee node, so that the effect of good-mouthing/bad-mouthing attacks is not

significant. In the strong attack model (as used for our protocol in Figure 7.7), the positive count for a malicious node or the negative count against a good node is as high as possible, i.e., greater than the number of direct observations but smaller than the accumulated observations (which is much higher than the number of direct observations) made by the trustor node toward the trustee node.



(a) Weak Attack Model



(b) Strong Attack Model

Figure 7.8: Healthiness Trust Evaluation Results of Bayesian Trust Management under Random Attacks.

Figure 7.8(a) and Figure 7.8(b) show the results under the weak and strong attack models, respectively. We observe that under the weak attack model (Figure 7.8(a)), Bayesian trust can produce very desirable trust evaluation results, since the effect of false recommendations is limited compared with direct observations. Since the attack is weak, it does not damage the trust system. On the other hand, under the strong attack model (Figure 7.8(b)), Bayesian trust management produces inaccurate trust prediction. We see that the trust value towards a compromised node will converge to a value close to 1 regardless the random attack probability. The reason is that with Bayesian trust management, a malicious node can retain a non-zero trust value because of a non-zero

false negative probability and especially by means of random attacks and then provide a high positive count to a compromised trustee node for good-mouthing attacks to break down the trust system. In contrast, our dynamic trust management protocol can effectively exclude false recommendations from malicious nodes by setting a recommender trust threshold. The results demonstrate that our dynamic trust protocol is more accurate and resilient than traditional Bayesian trust management in the presence of random attacks.

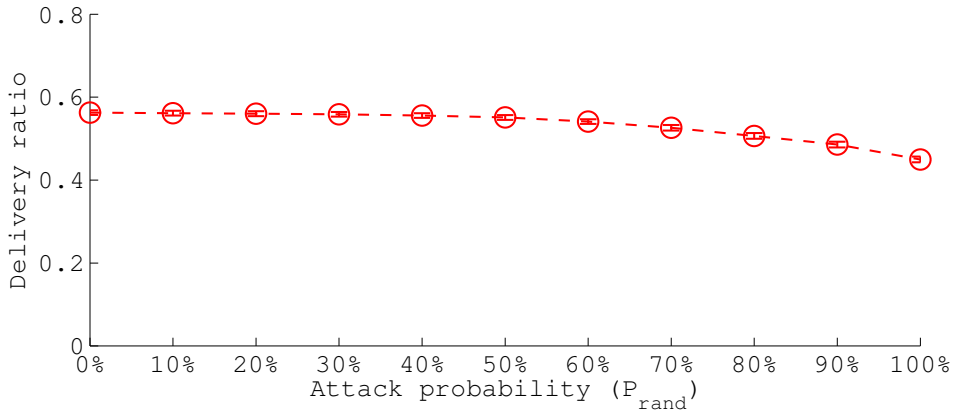


Figure 7.9: Message Delivery Ratio under Random Attacks.

Figure 7.9 shows the effect of random attacks to DTN routing performance. As expected, we see that the delivery ratio under random attacks ($P_{rand} < 1$) is higher than that under reckless attacks ($P_{rand} = 1$) since reckless attackers will always drop messages. Nevertheless, we see that the delivery ratio remains manageable as P_{rand} goes from 0 to 1. This demonstrates the resiliency property of our rust based routing protocol against random attacks by malicious nodes.

7.6 Dynamic Trust Management

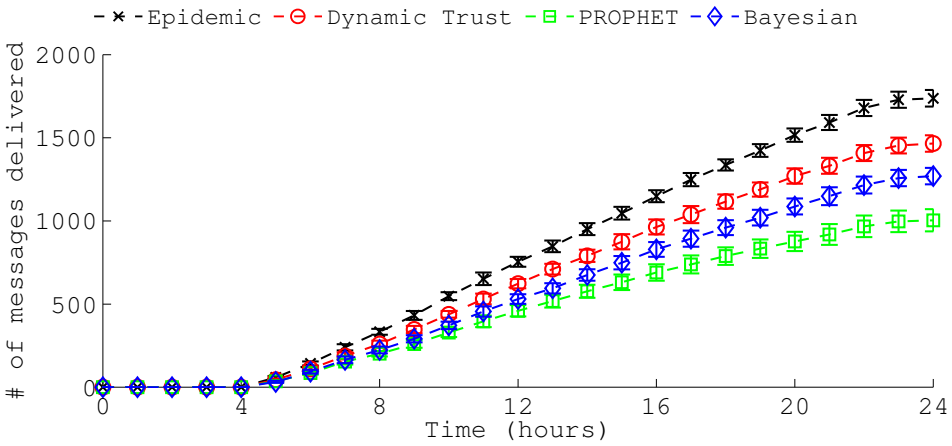
In this Section, we demonstrate the effectiveness of our dynamic trust management protocol in response to changing environment conditions. Without loss of generality, we consider hostility changes over time as modeled by the dashed line entities in the SPN model shown in Figure 7.1 with the transition rate of T_COMPRO being λ_c. Under our dynamic trust management protocol, the best protocol settings in terms of (β, λ_d), w^X, and T_f identified in Section 7.4 are applied in response to dynamically changing network conditions to minimize trust bias and to maximize DTN routing performance. Specifically, at runtime, each node senses hostility changes using its trust evaluation results (trust properties in healthiness) toward other nodes in the DTN, and then, based on the detected % of misbehaving nodes, performs a simple table lookup (e.g., into Table 7.2 and Table 7.3) to determine and apply the best protocol settings in (β, λ_d), w^X,

and T_f to minimize trust bias and to maximize DTN routing performance. As demonstrated in Figure 7.7, the healthiness trust $T_{i,j}^{healthiness}(t)$toward a compromised node will converge to P_{fn} so a node can use the fraction of "active" malicious nodes detected (i.e., those for which $T_{i,j}^{healthiness}(t)$ falls below $P_{error} + 0.5$) to perform a table lookup. Also trust convergence takes time, so a node must apply optimal protocol settings proactively.

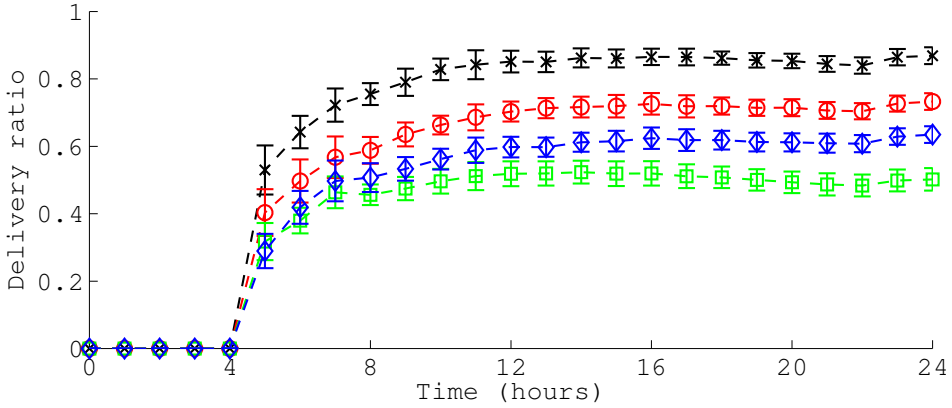
Table 7.5: Dynamic DTN Environment Setup.

Mobility	***SWIM***	***Infocom06 Trace***
Simulation Time	24 hours	100 hours
Compromise Rate (λ_c)	0.03 / hour	0.0072 / hour
# of Messages per Run	2000	2000
Warm-up Time	4 hours	10 hours
Maximum Delay	2 hours	5 hours
Routing Protocol	Dynamic trust-based routing, PROPHET, Bayesian trust-based routing, Epidemic routing	
MAC & PHY	IEEE 802.11a, Ad-Hoc	
Energy Model	3V, 17.4mA TX, 5.8mA RX, 0mA IDLE	

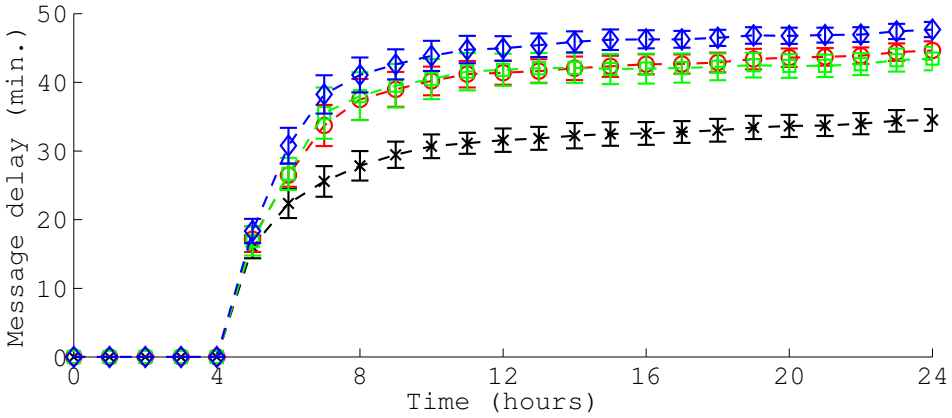
Below we perform a comparative analysis of our dynamic trust management protocol operating under best protocol settings dynamically for DTN routing against PROPHET, Bayesian trust-based routing, and epidemic routing. Similar to Section 7.5, we consider two mobility patterns: the SWIM mobility model and the *infocom06* mobility trace. Table 7.5 describes the simulation setup for each mobility pattern. Initially, there is no malicious node in the network. As time progresses, nodes become malicious with rate λ_c. The data reported is based on the average of 2000 messages. The last message is issued a few hours (the maximum delay) before the end of simulation to ensure sufficient time for message delivery.



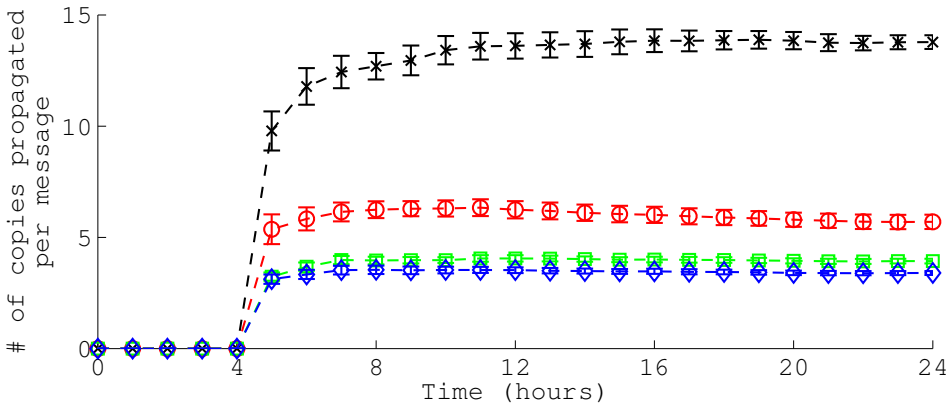
(a) Number of Delivered Messages.



(b) Delivery Ratio.



(c) Message Delay.

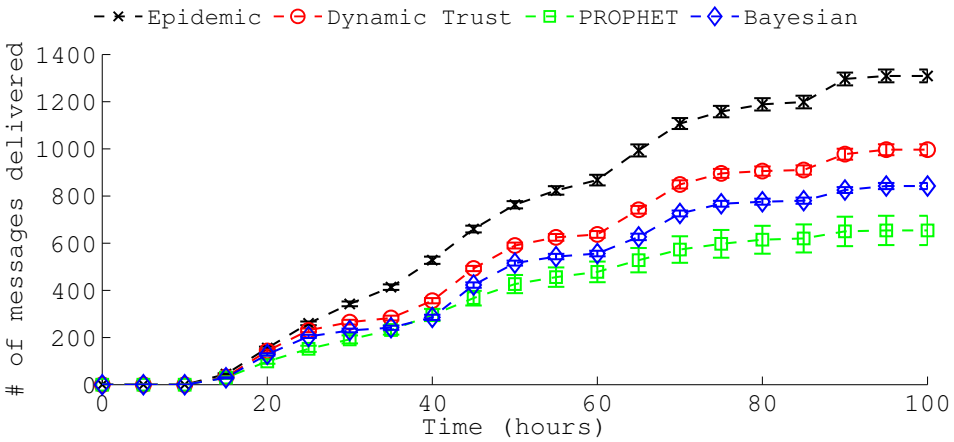


(d) Message Overhead.

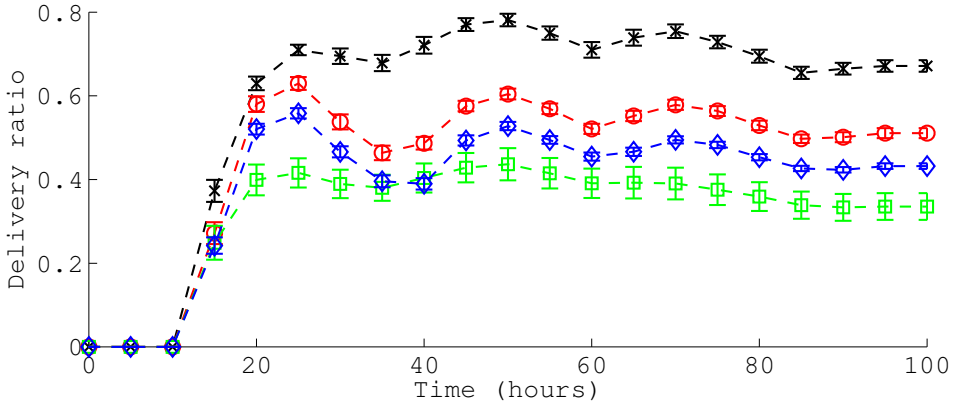
Figure 7.10: Performance Comparison of Routing Protocols based on SWIM Mobility in Dynamic DTN Environments.

Figure 7.10 shows performance comparison results based on the SWIM mobility model. The error bars mark the confidence interval at 95% confidence level. We observe that our dynamic trust-based routing protocol performs comparably to epidemic routing protocol in delivery ratio, while the other two protocols (PROPHET and Bayesian trust-based routing) have a low delivery ratio. The reason is that our trust-based routing protocol operating under the best (β, λ_d) setting can accurately identify misbehaving nodes with minimum trust bias (through the healthiness and unselfishness trust properties), thus avoiding message forwarding to misbehaving nodes. Moreover, our dynamic trust-based routing protocol operating under the best w^X, and T_f settings uses the best trust formation and application-level optimization design settings to maximize the DTN application performance in delivery ratio. We also observe that because the best protocol settings applied are geared toward maximizing the delivery ratio with a delay threshold (set to 2 hours in the experiment), it may lead to a higher message delay compared with other schemes, as only a smaller set of nodes would be selected as message carriers. However we see that when two copies (L=2) are allowed, our dynamic trust-based routing protocol approaches the ideal performance of epidemic routing in delivery ratio and message delay (Figure 7.10(b)) without incurring high message overhead (Figure 7.10(c)).

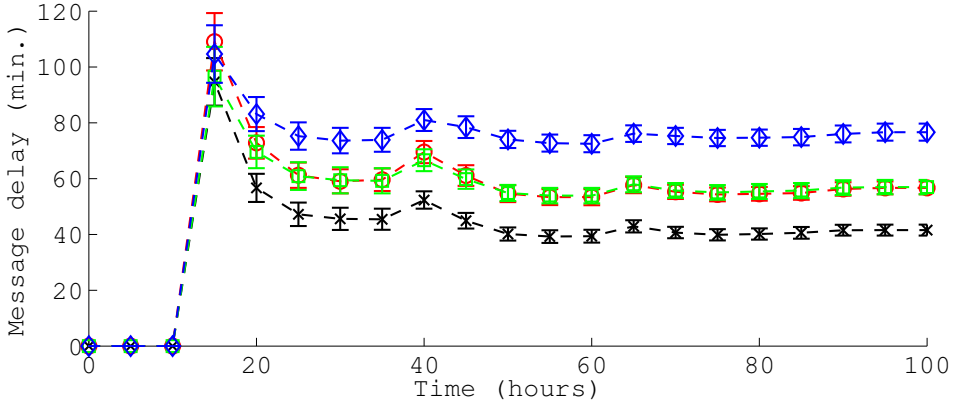
Figure 7.11 shows performance comparison results based on the *infocom06* mobility trace. We first observe that there are three peak periods in message delivery. This is caused by the three daytime periods in which people are active and most of the messages are delivered. Only a small fraction of the messages are forwarded and delivered during night. The curves in Figure 7.11 have the same trend as those in Figure 7.10, thus demonstrating the effectiveness of our dynamic trust management protocol regardless of the mobility pattern. This further validates our dynamic trust management design and its application to DTN routing in real DTN environments.



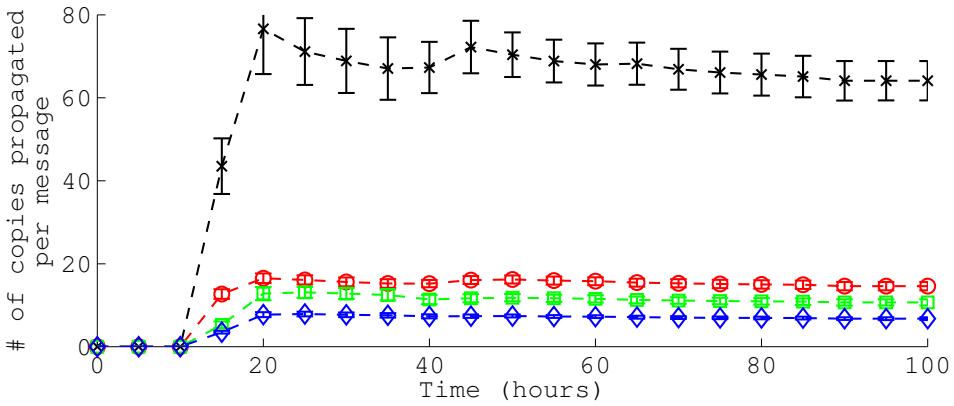
(a) Number of Delivered Messages.



(b) Delivery Ratio.



(c) Message Delay.



(d) Message Overhead.

Figure 7.11: Performance Comparison of Routing Protocols based on Mobility Traces in Dynamic DTN Environments.

7.7 Summary

In this chapter, we designed and validated a trust management protocol for DTNs and applied it to secure routing to demonstrate its utility. Our trust management protocol combines QoS trust with social trust to obtain a composite trust metric. Our design allows the best trust setting (β, λ_d) for trust aggregation to be identified so that subjective trust is closest to objective trust for each individual trust property for minimizing trust bias. Further, our design also allows the best trust formation w^x and application-level trust setting T_f to be identified to maximize application performance. We demonstrated how the results obtained at design time can facilitate dynamic trust management for DTN routing in response to dynamically changing conditions at runtime. We performed a comparative analysis of trust-based secure routing running on top of our trust management protocol with Bayesian trust-based routing and non-trust-based routing protocols (PROPHET and epidemic) in DTNs. Our results backed by simulation validation demonstrate that our trust-based secure routing protocol outperforms Bayesian trust-based routing and PROPHET. Further, it approaches the ideal performance of epidemic routing in delivery ratio and message delay without incurring high message or protocol maintenance overhead.

Chapter 8

Dynamic Trust Management for Wireless Sensor Networks and Its Applications

In this chapter, we apply design and validation principles of dynamic trust management to wireless sensor networks (WSNs) and propose a highly scalable cluster-based hierarchical trust management protocol for WSNs leveraging clustering to cope with a large number of heterogeneous sensor nodes (SNs) for scalability and reconfigurability. A wireless sensor network (WSN) is usually composed of a large number of spatially distributed autonomous SNs to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. A SN deployed in the WSN has the capability to read the sensed information and transmit or forward information to base stations or a sink node through multi-hop routing. While SNs have popularly used for various monitoring purposes such as wild animals, weather, or environments for battlefield surveillance, they also have severely restricted resources such as energy, memory, and computational power. Further, wireless environments give more design challenges due to inherently unreliable communications. A more serious issue is that nodes may be compromised and perform malicious attacks such as packet dropping or packet modifications to disrupt normal operations of a WSN wherein SNs usually perform unattended operations. A large number of SNs deployed in the WSN also require a scalable algorithm for highly reconfigurable communication operations.

To demonstrate the utility of our hierarchical trust management protocol, we apply it to trust-based geographic routing and trust-based intrusion detection. For each application, we identify the best trust composition and formation to be applied dynamically to maximize application performance. Our results indicate that trust-based geographic routing approaches the ideal performance level achievable by flooding-based routing in message delivery ratio and message delay without incurring substantial message overhead. For trust-based intrusion detection, we discover that there exists an optimal application-level trust threshold for minimizing false positives and false negatives. Fur-

thermore, trust-based intrusion detection outperforms traditional anomaly-based intrusion detection approaches in both the detection probability and the false positive probability.

8.1 System Model

We consider a cluster-based WSN consisting of multiple clusters, each with a cluster head (CH) and a number of SNs in the corresponding geographical area. CH nodes have more power and resources than SN nodes. The CH in each cluster may be selected based on an election protocol such as HEED [191] at runtime to balance energy consumption vs. CH functionality. A SN forwards its sensor reading to its CH through SNs in the same cluster and the CH then forwards the data to the base station or the destination node (or sink node) through other CHs. CHs are inherently static due to their intended functionality. SNs may have limited mobility due to natural causes (e.g., wind). When a SN moves to a neighbor cluster, it registers with the new CH to begin its new duties. We assume that a pairwise key is established between two nodes within a k-hop range at the system deployment time, so a SN moving to another cluster by natural causes can use its pairwise key to authenticate itself during the registration phase.

Leveraging this two-level of hierarchy in the WSN, our trust management protocol is conducted using *periodic* peer-to-peer trust evaluation between two SNs and between two CHs. The trust update interval Δt is a system design parameter. At the SN level, each SN is responsible to report its peer-to-peer trust evaluation results towards other SNs in the same cluster to its CH which performs CH-to-SN trust evaluation towards all SNs in its cluster. Similarly a CH is responsible to report its peer-to-peer trust evaluation results towards other CHs in the system to the base station which performs station-to-CH trust evaluation towards all CHs in the system. In Section 8.2, we will describe the protocols for performing peer-to-peer, CH-to-SN and station-to-CH trust evaluations.

We compose our trust metric by considering both *social trust* and *QoS trust* to take into account the effect of both aspects of trust on trustworthiness. Social trust in the context of wireless sensors may include intimacy, honesty, privacy, and centrality. QoS trust may include competence, cooperativeness, reliability, task completion capability, etc. We formulate our trust protocol such that it is generic and can take a combination of social trust and QoS trust metrics to form the overall trust metric. Without loss of generality, we consider *intimacy* (for measuring closeness based on interaction experiences) and *honesty* (for measuring regularity/anomaly) to measure social trust derived from social networks. We choose *energy* (for measuring competence) and *cooperativeness* (for measuring cooperativeness in protocol execution) to measure QoS trust derived from communication networks. The *intimacy* trust component reflects the relative degree of

interaction experiences between two nodes. It follows the maturity model proposed in [180] in that the more positive experiences SN *A* had with SN *B*, the more trust and confidence SN *A* will have toward SN *B*. The *honesty* trust component strongly implies whether a node is malicious or not. The assumption is that a compromised node is malicious in nature and thus dishonest. Energy is one most important metric in WSNs since SNs are extremely constrained in energy. We use energy as a QoS trust metric to measure if a SN is competent in performing its intended function. The cooperativeness trust component reflects if a SN can cooperatively execute the intended protocol.

Our trust management protocol can apply to any WSN consisting of heterogeneous SNs with vastly different initial energy levels and different degrees of maliciousness or uncooperativeness behaviors. We apply the trust management protocol to a clustered WSN in which a SN may adjust its behavior dynamically according to its own operational state and environmental conditions. A SN is more likely to become uncooperative when it has low energy or it has many cooperative neighbor nodes around. Further, a SN is more likely to become compromised when it has more compromised neighbors around. A CH consumes more energy than SNs. After a SN or CH is compromised, it may consume even more energy to perform attacks. On the other hand, an uncooperative node consumes less energy than cooperative nodes as its uncooperative behavior is reflected by stopping sensing functions and arbitrarily dropping messages.

A compromised SN can perform various attacks including forgery attacks, jamming attacks, Sybil attacks, deny of service attacks, black/sink hole attacks (absorbing and dropping packets), and slandering attacks. Depending on the system failure definition, some of these attacks if successfully performed are fatal. For example if a compromised node uses its shared secret key to perform a forgery attack and the tampered packet reaches the sink node, it can be considered as a system failure as the consequence of the sink node receiving false information may be catastrophic. Thus, the only defense of the system is to quickly detect and evict compromised nodes before a system failure occurs. In this chapter, we show that our hierarchical trust management protocol is resilient to black/sink hole attacks and slandering attacks including good-mouthing attacks (recommending a bad node as a good node), and bad-mouthing attacks (recommending a good node as a bad node) in trust-based routing applications (in Section 8.5). Also our trust management protocol can be effectively applied to implement trust-based intrusion detection (in Section 8.6) to deal with other types of attacks.

8.2 Hierarchical Trust Management Protocol

We first describe our hierarchical trust management addressing the problem of trust formation, trust aggregation and trust composition. Later we apply it to the clustered WSN described in the system model to demonstrate its effectiveness.

Our hierarchical trust management protocol maintains two levels of trust: *SN-level* trust and *CH-level* trust. Each SN evaluates other SNs in the same cluster while each CH evaluates other CHs and SNs in its cluster. The peer-to-peer trust evaluation is periodically updated based on either *direct* observations or *indirect* observations. When two nodes are neighbors within radio range, they evaluate each other based on direct observations via snooping or overhearing. Each SN sends its trust evaluation results toward other SNs in the same cluster to its CH. Each CH performs trust evaluation toward all SNs within its cluster. Similarly, each CH sends its trust evaluation results toward other CHs in the WSN to a "CH commander" which may reside on the base station if one is available, or on a CH elected if a base station is not available. The CH commander performs trust evaluation toward all CHs in the system. A possible solution to the election protocol is HEED [191].

These two levels of peer-to-peer trust evaluation process consider four different trust components described earlier: intimacy, honesty, energy, and cooperativeness. The trust value that node i evaluates towards node j at time t, $T_{ij}(t)$, is represented as a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. $T_{ij}(t)$ is computed by:

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{energy}(t) + w_4 T_{ij}^{cooperativeness}(t) \qquad (8.1)$$

where w_1, w_2, w_3, and w_4 are weights associated with these four trust components with $w_1 + w_2 + w_3 + w_4 = 1$. Deciding the best values of w_1, w_2, w_3 and w_4 to maximize application performance is a trust formation issue which we aim to explore (Section 8.5 and Section 8.6). Here we note that in the special case in which intimacy and honesty are equally important and energy and cooperativeness are also equally important, Equation (8.1) can be rewritten as $T_{ij}(t) = 0.5 w_{social} [T_{ij}^{intimacy}(t) + T_{ij}^{honesty}(t)] + 0.5 w_{QoS} [T_{ij}^{energy}(t) + T_{ij}^{cooperativeness}(t)]$ with $w_{social} + w_{QoS} = 1$.

8.2.1 Peer-to-Peer Trust Evaluation

Here we describe how peer-to-peer trust evaluation is conducted, particularly between two peer SNs or two peer CHs. When a trustor (node i) evaluates a trustee (node j) at time t, it updates $T_{ij}^{X}(t)$ where X indicates a trust component as follows:

$$T_{ij}^{X}(t) = \begin{cases} (1-\alpha)T_{ij}^{X}(t-\Delta t) + \alpha T_{ij}^{X,direct}(t), \\ \quad \text{if } i \text{ and } j \text{ are } 1-\text{hop neighbors}; \\ \underset{k \in N_i}{\text{avg}}\left\{(1-\gamma)T_{ij}^{X}(t-\Delta t) + \gamma T_{kj}^{X,recom}(t)\right\}, \\ \quad \text{otherwise.} \end{cases} \tag{8.2}$$

In Equation (8.2), if node *i* is a 1-hop neighbor of node *j*, node *i* will use its new trust based on direct observations ($T_{ij}^{X,direct}(t)$) and its old trust based on past experiences ($T_{ij}^{X}(t-\Delta t)$ where Δt is the trust update interval) toward node *j* to update $T_{ij}^{X}(t)$. A parameter α ($0 \le \alpha \le 1$) is used here to weigh these two trust values and to consider trust decay over time, i.e., the decay of the old trust value and the contribution of the new trust value. A larger α means that trust evaluation will rely more on direct observations. Here $T_{ij}^{X,direct}(t)$ indicates node *i*'s trust value toward node *j* based on direct observations accumulated over the time period $[0, t]$. Below we describe how each trust component value $T_{ij}^{X,direct}(t)$ can be obtained based on direct observations for the case node *i* and node *j* are 1-hop neighbors:

$T_{ij}^{intimacy,direct}(t)$: This measures the level of interaction experiences following the maturity model [180]. It is computed by the number of interactions between nodes *i* and *j* over the maximum number of interactions between node *i* and any neighbor node over the time period $[0, t]$.

$T_{ij}^{honesty,direct}(t)$: This refers to the belief of node *i* that node *j* is honest based on node *i*'s direct observations toward node *j*. Node *i* estimates $T_{ij}^{honesty,direct}(t)$ by keeping a count of suspicious dishonest experiences of node *j* which node *i* has observed during $[0, t]$ using a set of anomaly detection rules such as a high discrepancy in sensor readings or recommendations reported by node *j* compared to its peers has been experienced, as well as interval, retransmission, repetition, and delay rules as in [88, 170]. If the count exceeds a system-defined threshold, node *j* is considered totally dishonest at time *t*, i.e., $T_{ij}^{honesty,direct}(t)$=0. Otherwise, $T_{ij}^{honesty,direct}(t)$ is computed by 1 minus the ratio of the count to the threshold. An assumption is that a compromised node must be dishonest.

$T_{ij}^{energy,direct}(t)$: This refers to the belief of node *i* that node *j* still has adequate energy (representing competence) to perform its intended function. It may be measured by the percentage of node *j*'s remaining energy. To calculate $T_{ij}^{energy,direct}(t)$, node *i* estimates node *j*'s remaining energy by overhearing node *j*'s packet transmission activities over the time period [0, *t*], utilizing an energy consumption model as in [31, 131, 199].

$T_{ij}^{cooperativeness,direct}(t)$: This provides the degree of cooperativeness of node j as evaluated by node i based on direct observations over $[0, t]$. Node i can apply overhearing and snooping techniques to detect uncooperativeness behaviors of node j such as not faithfully performing sensing and reporting functions, data forwarding functions [168], or the prescribed trust management protocol execution. Node i may give recent interaction experiences a higher priority over old experiences in estimating $T_{ij}^{cooperativeness,direct}(t)$. An assumption is that a compromised node must be uncooperative.

On the other hand, if node i is not a 1-hop neighbor of node j, node i will use its past experience $T_{ij}^{X}(t - \Delta t)$ and recommendations from its 1-hop neighbors ($T_{kj}^{X,recom}(t)$ where k is a recommender) to update $T_{ij}^{X}(t)$. Node i will only use its 1-hop bors (N_i) as recommenders for energy conservation and scalability. If N_i is an empty set, then node i is an orphan in which case $\gamma = 0$ and node i will not be able to contribute to peer-to-peer trust management. The parameter γ is used here to weigh recommendations vs. past experiences and to consider trust decay over time as follows:

$$\gamma = \frac{\beta T_{ik}(t)}{1 + \beta T_{ik}(t)} \tag{8.3}$$

Here we introduce another parameter $\beta \geq 0$ to specify the impact of "indirect recommendations" on $T_{ij}^{X}(t)$ such that the weight assigned to indirect recommendations is normalized to $\beta T_{ik}(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust increases proportionally as either $T_{ik}(t)$ or β increases. Instead of having a fixed weight ratio $T_{ik}(t)$ to 1 for the special case in which $\beta = 1$, we allow the weight ratio to be adjusted by adjusting the value of β and test its effect on protocol resiliency against slandering attacks such as good-mouthing and bad-mouthing attacks. Here, $T_{ik}(t)$ is node i's trust toward node k as a recommender (for node i to judge if node k provides correct information). The recommendation $T_{kj}^{X,recom}(t)$ provided by node k to node i about node j depends on if node k is a good node. If node k is a good node, $T_{kj}^{X,recom}(t)$ is simply equal to $T_{kj}^{X}(t)$. If node k is a bad node, it can provide $T_{kj}^{X,recom}(t) = 0$ when node j is a good node by means of bad-mouthing attacks, and can provide $T_{kj}^{X,recom}(t) = 1$ when node j is a bad node by means of good-mouthing attacks. In our analysis we assume this worst-case attack behavior to test our protocol resiliency. The new trust value $T_{ij}^{X}(t)$ obtained from Equation (8.2) would be the average of the combined trust values of past trust information and recommendations collected at time t.

8.2.2 CH-to-SN Trust Evaluation

Each SN reports its trust evaluation toward other SNs in the same cluster to its CH. The CH then applies a generic statistical analysis method (such as Equation (8.4) below) to $T_{ij}(t)$ values received to perform CH-to-SN trust evaluation towards node *j*. Further, the CH can also leverage $T_{ij}(t)$ values received to detect if there is any outlier as an evidence of good-mouthing or bad-mouthing attacks. Based on the resulting CH-to-SN trust evaluation result toward node *j*, the CH determines whether node *j* is untrustworthy and needs to be excluded from sensor reading and routing duties. Specifically a CH, *c*, when evaluating a SN, *j*, will perform intrusion detection by comparing the system minimum trust threshold T^{th} with node *j*'s trust value, $T_{cj}(t)$, obtained by:

$$T_{cj}(t) = \underset{i \in M_c \wedge T_{ci}(t) \geq T^{th}}{\text{avg}} \{T_{ij}(t)\} \tag{8.4}$$

where M_c is the set of SNs in the cluster. CH *c* will announce *j* as compromised if $T_{cj}(t)$ is less than T^{th}; otherwise, node *j* is not compromised. Note that we only take into account the trust values received from those SNs which are considered trustworthy by the CH. That is, CH *c* will take a trust recommendation from node *i* only if $T_{ci}(t) \geq T^{th}$. Later in Section 8.6 we will illustrate a statistical analysis methodology to implement trust-based intrusion detection as an application to hierarchical trust evaluation.

8.2.3 Station-to-CH Trust Evaluation

Here we first note that the transmission power and capacity of CHs generally are higher than those of SNs. Thus, the one-hop radio range of CHs is higher than that of SNs. Also a CH after gathering and possibly aggregating sensor readings will forward the information hop-by-hop to the base station through other CHs. Thus, there are a lot of interaction experiences between two neighbor CHs in a WSN, just like two SNs in a cluster. Consequently, CH-to-CH peer evaluation will be conducted in a similar way as SN-to-SN peer evaluation, as discussed in Section 8.2.1. Each CH reports its trust evaluation toward other CHs in the WSN to the base station which is infallible with physical protection. The CH commander resided on the base station then applies the same statistical analysis method (as in Equation (8.4)) to $T_{ij}(t)$ values received from all CHs in the system to perform station-to-CH trust evaluation towards CH *j*. The base station determines whether CH *j* is considered untrustworthy and needs to be excluded from cluster head duties.

8.3 Performance Model

We develop a probability model based on stochastic Petri nets (SPN) [163] techniques to describe the behavior of each SN or CH in the WSN described in Section 8.1 given the

anticipated operational profile as input. It provides a basis for obtaining ground truth status of nodes in the system, thereby allowing us to derive *objective trust* against which subjective trust obtained as a result of executing our hierarchical trust management protocol can be checked and validated. We use SPN as our analytical tool due to its capability to represent a large number of states for complex systems where an underlying model is a semi-Markov or Markov model. Further, we develop a novel iterative hierarchical modeling technique to avoid state explosion problems and to yield efficient solutions.

Figure 8.1 shows the SPN model that describes the behavior of a SN (or a CH). We consider a heterogeneous WSN consisting of N_{SN} SNs uniformly distributed in an $M \times M$ square-shaped operational area. Each SN is attached to a CH based on its location and so the system will have N_{CH} clusters each with a CH. CHs and SNs have radio range of R and r, respectively. The trust update interval is Δt.

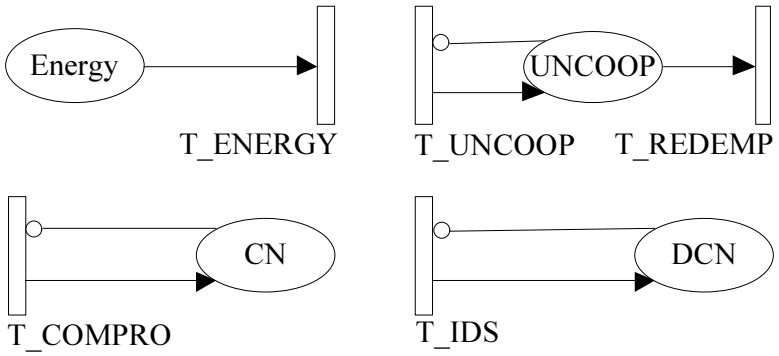


Figure 8.1: SPN Model for a Sensor Node or a Cluster Head.

Below we explain how we construct the SPN model for describing the behaviors of a single node and how we compose a performance model for the entire WSN using a number of such SPN models (one for each node in the system).

Energy: Place *Energy* indicates the remaining energy level of the node. The initial number of tokens in place *Energy* is set to E_{init}. A token will be released from place *Energy* when transition *T_ENERGY* is triggered. The rate of transition *T_ENERGY* indicates the energy consumption rate. A CH consumes more energy than a SN. The energy consumption rate is also affected by a node's state. It is lower when a node becomes uncooperative. It is higher when a node is compromised because it takes energy to perform attacks. We denote Δ_{E-SN}, Δ_{E-CH} and $\Delta_{E-compromised}$ as the energy consumption rates per Δt time for a normal SN, a normal CH, and a compromised node, respectively, which can be obtained by analyzing historical data with $\Delta_{E-SN} < \Delta_{E-CH} < \Delta_{E-compromised}$. The energy consumption rates for an uncooperative

SN and an uncooperative CH are $\rho\Delta_{E-SN}$ and $\rho\Delta_{E-CH}$ per Δt time unit, respectively, with $0 \leq \rho \leq 1$ denoting the energy saving ratio of an uncooperative node compared with a normal node.

Cooperativeness: As specified by the operational profile, a node may become uncooperative in protocol execution to save energy. An uncooperative node may stop reading data and drop packets it receives. A cooperative node may turn uncooperative in every trust evaluation interval Δt according to its remaining energy and the number of cooperative neighbors around. An uncooperative node may redeem itself as cooperative to achieve a service availability goal when it senses many uncooperative neighbor SNs around it to balance individual welfare vs. system welfare. We model these behaviors by putting a token into place *UNCOOP* when transition *T_UNCOOP* is triggered and removing the token from place *UNCOOP* when transition *T_REDEMP* is triggered. A token in place *UNCOOP* thus indicates that the node is uncooperative. A node's uncooperative probability is modeled by:

$$P_{uncoop} = \mu \frac{E_{consumed}}{E_{init}} + (1 - \mu) \frac{N_{neighbor}^{coop}}{N_{neighbor}} \tag{8.5}$$

where μ is a weight associated with the *energy* term and (1-μ) is the weight associated with the *cooperative neighborhood* term. $E_{consumed}$ is energy consumed and E_{init} is the node's initial energy level. Thus, $E_{consumed}/E_{init}$ represents the percentage of energy consumed. $N_{neighbor}^{coop}/N_{neighbor}$ is the percentage of cooperative neighbors where $N_{neighbor}^{coop}$ is the number of cooperative neighbors and $N_{neighbor}$ is the total number of neighbors. A node's uncooperative probability tends to be lower when a node has more energy and higher when the node has more cooperative neighbors as there are sufficient cooperative neighbors around to take care of sensor tasks. Thus, the rates of transitions *T_UNCOOP* and *T_REDEMP* are given by $P_{uncoop}/\Delta t$ and $(1 - P_{uncoop})/\Delta t$, respectively. All nodes are cooperative initially with no token in place *UNCOOP*. We set μ to 0.5 to give equal weighting to energy and uncooperative neighborhood terms for the example WSN described in Section 8.1.

Compromise: A node becomes compromised when transition *T_COMPRO* fires and a token is put in place *CN*. The rate to *T_COMPRO* is modeled by:

$$\lambda_c = \lambda_{c-init} \frac{N_{neighbor}^{compromised}}{N_{neighbor}^{uncomrpromised}} \tag{8.6}$$

where λ_{c-init} is the initial node compromise rate which can be obtained by first-order approximation based on historical data about the targeted network environment. $N_{neighbor}^{compromised}$ and $N_{neighbor}^{uncompromised}$ are the numbers of compromised and uncompromised nodes in the neighborhood. $N_{neighbor}^{compromised}/N_{neighbor}^{uncompromised}$ refers to the ratio of the number of compromised 1-hop neighbors to the number of not compromised 1-hop neighbors. Equation (8.6) models that a node is more likely to be compromised when there are more 1-hop compromised nodes around it due to collusive attacks. The hierarchically structured WSN has a trust-based intrusion detection system (IDS) in place (see Section 8.6). We model the IDS behavior through transition *T_IDS*. A compromised node can be caught by IDS with the rate $(1 - P_{fn})/T_{IDS}$ for transition *T_IDS* where P_{fn} is the IDS false negative probability and T_{IDS} is the IDS detection interval. When a compromised node is detected by the IDS, a token will move to place *DCN*. In addition, we model false positives generated by the IDS (i.e., diagnosing a good node as a bad node) by associating a rate of P_{fp}/T_{IDS} with transition *T_IDS* which is enabled only when the node is not compromised, that is, when there is no token in place *CN*. Note that all nodes are good, i.e., not compromised, initially. Note that trust-based intrusion detection (see Section 8.6) will be used for determining IDS P_{fn} and P_{fp}. Also since a compromised node will exhibit uncooperativeness behaviors (not following the protocol). This is modeled by moving a token to place *UNCOOP* when a token is moved into *CN*. Different from an uncooperative node, however, a compromised node will not redeem itself to become cooperative again as it is malicious in nature.

The overall performance model for describing the behaviors of a WSN consists of N_{SN} SPN subnet models one for each SN, and N_{CH} SPN subnet models one for each CH, with vastly different energy consumption, uncooperativeness/redemption and compromise rates. Below we describe how one could leverage SPN outputs to obtain subjective trust and objective trust values to validate our hierarchical trust management protocol.

8.3.1 Subjective Trust Evaluation

Recall that under our proposed trust management protocol, node *i* will subjectively assess its trust toward node *j*, $T_{ij}(t)$, based on its direct observations and indirect recommendations obtained toward node *j* according to Equation (8.1) and Equation (8.2). In particular, for the direct trust assessment part when node *j* is a 1-hop neighbor of node *i*, node *i* will apply intimacy, honesty, energy and cooperativeness detection mechanisms in the protocol design described in Section 8.2 to assess $T_{ij}^{X,direct}(t)$ based on direct observations over the time period $[0, t]$. Because the assessment is direct, assuming that the detection mechanisms are effective, $T_{ij}^{X,direct}(t)$ computed by node *i* will be

close to actual status of node *j* at time *t*, which can be obtained from the SPN model output.

Table 8.1: Status Value Assignments to Compute $T_{ij}^{X,direct}(t)$.

Item	Value	Condition (of node *j*)
$T_{ij}^{intimacy,direct}(t)$	*a*/ *c*	*If mark*(*UNCOOP*) = 1 AND *mark*(*CN*) = 0
	b/ *c*	*If mark*(*CN*) = 1
	1	Otherwise
$T_{ij}^{honesty,direct}(t)$	1	*If mark*(*DCN*) = 0
	0	Otherwise
$T_{ij}^{energy,direct}(t)$	*mark*(*Energy*)/E_{init}	none
$T_{ij}^{cooperativeness,direct}(t)$	1	*If mark*(*UNCOOP*) = 0
	0	Otherwise

In Table 8.1, we show how to compute actual status of node *j* at time *t* and thus $T_{ij}^{X,direct}(t)$ based on assigning status values to states in the underlying semi-Markov chain of the SPN model, with the state representation of node *j* being (*Energy*, *CN*, *DCN*, *UNCOOP*). Specifically, $T_{ij}^{honesty,direct}(t)$ is approximated by assigning a status value of 0 (representing complete dishonesty) to states in which node *j* is compromised detected (i.e., *DCN* is 1) and a status value of 1 (representing complete honesty) to all other states. The reason is that a compromised node must be dishonest. The dishonesty detection mechanisms employed by node *i* for direct assessment of node *j*'s dishonesty, however, are at most as good as those employed by the IDS which will announce node *j* as compromised when it identifies node *j* as compromised, i.e., when *DCN* is 1. $T_{ij}^{energy,direct}(t)$ is computed by assigning a status value of *Energy*/E_{init} to all states. $T_{ij}^{cooperativeness,direct}(t)$ is computed by assigning a status value of 1 to states in which node *j* is cooperative (i.e., *UNCOOP* is 0) and a status value of 0 to states in which node *j* is uncooperative (i.e., *UNCOOP* is 1).

To compute $T_{ij}^{intimacy,direct}(t)$, we first note that status information in intimacy is not directly available from the state representation. Based on our peer-to-peer trust evaluation protocol (Section 8.2.1), $T_{ij}^{intimacy,direct}(t)$ is computed by the number of interactions between nodes *i* and *j* over the maximum number of interactions between node *i* and any neighbor node over the time period $[0, t]$. If during the period there is no interaction between nodes *i* and *j*, then $T_{ij}^{intimacy,direct}(t) = 0$. Here we predict what $T_{ij}^{intimacy,direct}(t)$ would be when there is a normal level of interactions of data forward-

ing activities, conditioning on the status of node *j*, i.e., compromised, uncooperative or normal. We consider four types of interactions during geographic forwarding, given that node *i* is the initiating node: (1) *Requesting*: node *i* broadcasts a packet delivery request to its 1-hop neighbors; (2) *Reply*: nodes that are closer to the destination node than node *i* will reply to node *i*; (3) *Selection*: node *i* selects up to *L* nodes with the highest trust values to forward the packet; (4) *Overhearing*: node *i* overhears if the packet has been forwarded. Node *i* then keeps track of its interaction experiences with node *j* to compute $T_{ij}^{intimacy,direct}(t)$. Let the average numbers of interactions of node *i* with an uncooperative node, a compromised node and a normal node be *a*, *b* and *c*, respectively. The values of *a, b, c* are computed dynamically. Below we predict their values from node *i*'s perspective for the case in which an uncooperative node drops 50% of packets and a compromised node drops 100% of packets. On the one hand, if node *i* requests a neighbor to forward a packet then (1) the expected number of interactions between node *i* and an uncooperative node *j* is 25%×50%×3 because there will be three interactions (reply, selection, and overhearing) only if the uncooperative node is in the quadrant closest to the destination node (with 25% probability) and does not drop the packet (with 50% probability); (2) the expected number of interactions between node *i* and a compromised node *j* is 0 because a compromised node discards all requests from node *i*; and (3) the expected number of interactions between node *i* and a normal node *j* is 25%×3 because there will be three interactions only if that node is in the quadrant closest to the destination node (with 25% probability). On the other hand, if node *i* receives a request from node *j* to forward a packet, the expected number of interactions will be 25%×2 because from node *i*'s perspective there will be two interactions (reply and selection) only if node *i* is in the quadrant closest to the target node. Summarizing above, we have:

$$\begin{aligned} a &= 25\% \times 50\% \times 3 + 25\% \times 2; \\ b &= 0 + 25\% \times 2; \\ c &= 25\% \times 3 + 25\% \times 2. \end{aligned} \tag{8.7}$$

Consequently, we compute $T_{ij}^{intimacy,direct}(t)$ by assigning a status value of *a*/*c* to states in which node *j* is uncooperative (i.e., *UNCOOP* is 1), *b*/*c* to states in which node *j* is compromised (i.e., *CN* is 1), and *c*/*c* = 1 to states in which node *j* is a normal node (*UNCOOP*=0 and *CN*=0).

Here we should emphasize that in practice node *i* would just follow the protocol execution to assess $T_{ij}^{X,direct}(t)$ using detection mechanisms designed to assess trust property *X* based on local information. The computational procedure described above is to predict $T_{ij}^{X,direct}(t)$ that would have been obtained by node *i* based on the argument that a node's direct observation trust assessment would be close to ground truth. Once node

i obtains $T_{ij}^{X,direct}(t)$ for X = intimacy, honesty, energy, and cooperativeness, it will compute $T_{ij}^{X}(t)$ based on Equation (8.2) and subsequently $T_{ij}(t)$ based on Equation (8.1) for subjective trust evaluation.

8.3.2 Objective Trust Evaluation

To validate subjective trust evaluation, we compute *objective trust* based on actual status as provided by the SPN model output using exactly the same status value assignment as shown in Table 8.1 to yield ground truth status of node *j* at time *t*. The objective trust value of node *j*, $T_{j,obj}(t)$, is also a weighted linear combination of four trust component values:

$$T_{j,obj}(t) = w_1 T_{j,obj}^{intimacy}(t) + w_2 T_{j,obj}^{honesty}(t) + w_3 T_{j,obj}^{energy}(t) + w_4 T_{j,obj}^{cooperativeness}(t) \quad (8.8)$$

Note that here $T_{j,obj}^{intimacy}(t), T_{j,obj}^{honesty}(t), T_{j,obj}^{energy}(t)$ and $T_{j,obj}^{cooperativeness}(t)$ are objective trust component values, reflecting node *j*'s ground truth status at time *t*.

8.4 Trust Evaluation Results

Table 8.2: Default Parameter Values Used.

Param	Value	Param	Value	Param	Value
M	900m	R	150m	r	50m
N_{SN}	900	N_{CH}	81	Δt	80hrs
α	[0,1]	β	[0,100]	$1/\lambda_{c\text{-}init}$	[80,360]day
$\Delta_{E\text{-}SN}$	80hrs	$\Delta_{E\text{-}CH}$	160hrs	$\Delta_{E\text{-}compromised}$	240hrs
ρ	1/3	T_{IDS}	80hrs	P_{fp}, P_{fn}	[1-5]%
E_{init}	[360,480] *days* for SNs, [720,960] *days* for CHs.				

In this section, we show numerical results obtained through model-based evaluation as described in Section 8.1. The basis is the example WSN described in Section 8.1 characterized by a set of parameter values listed in Table 8.2. We consider a WSN with 900 SNs (and 81 CHs) evenly spread out in a 900m×900m operational area based on uniform distribution. The initial energy lifetime of a SN varies from 360 days to 480 days while the CHs have much higher initial energy lifetime ranging from 720 days to 960 days. The radio ranges of a SN and a CH are *r*=50m and *R*=150m, respectively. The WSN is

assumed to be deployed in a hostile environment with the node's average compromising interval in the range of 80 days to 360 days. We consider the worst case of good-mouthing attacks (providing the highest trust value of 1 for a malicious node) and bad-mouthing attacks (providing the lowest trust value of 0 against a good node). The node is a good node at time t=0 and then becomes a bad node based on its compromise rate. The false positive and negative probabilities (P_{fp} and P_{fn}) are in the range of 1% to 5% as a result of trust-based intrusion detection (see Section 8.6). Because of the anticipated long system lifetime, to save energy the trust update interval Δt is set at 80 hours. Thus, the amount of energy consumed per Δt time for a normal SN is also set to 80 hours. The amount of energy consumed per Δt time for a normal CH and a compromised node are Δ_{E-CH} = 160 hours and $\Delta_{E-compromised}$ = 240 hours, respectively. The energy saving ratio of an uncooperative node relative to a normal node, ρ, is 1/3 denoting that an uncooperative node will only consume energy at 1/3 of the speed of its cooperative counterpart.

Our trust evaluation consists of two parts. The first part is about trust composition and trust aggregation. The second part is about trust formation. Our assertion is that, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, each trust property X has its own best α and β values under which subjective assessment of $T_{ij}^{X}(t)$ from Equation (8.2) would be the most accurate against actual status of node j in trust property X. Once we are assured of the accuracy of each trust property X, we can then address the trust formation issue for each application in hand, i.e., identifying the best way to form trust out of individual QoS and social trust properties such that the application performance is maximized. We will evaluate trust formation in Section 8.5 and Section 8.6 when we apply hierarchical trust management to trust-based geographic routing and trust-based intrusion detection.

Recall that a higher α value indicates that subjective trust evaluation relies more on direct observations compared with past experiences while a higher β value indicates that subjective trust evaluation relies more on indirect recommendations provided by the recommenders compared with past experiences. Below we present CH-to-SN trust evaluation results based on peer-to-peer trust evaluation results reported by SNs in the same cluster, and compare them against objective trust evaluated based on the SN's actual status. We omit reporting station-to-CH evaluation results here as the same trends have been observed.

Figure 8.2 shows the effect of α and β on the mean square error between subjective trust obtained from Equation (8.2) and objective trust obtained from actual status for X=intimacy. The diagrams for other trust properties exhibit a similar trend. We vary α from 0 to 1 and β from 0 to 100 to cover all possible values. We see that as α increases (using a larger α indicates that subjective trust evaluation relies more on direct observations compared with past experiences), the mean square error first decreases and then

increases. Subjective trust initially approaches objective trust as more recent direct observations are used. However, there is a crossover point (e.g., $\alpha \geq 0.8$ when $\beta = 10$) after which subjective trust deviates more from objective trust because of underestimation. On the other hand, as β increases (using a larger β indicates that subjective trust evaluation relies more on indirect recommendations provided by recommenders compared with past experiences), subjective trust initially approaches objective trust, but deviates more from objective trust after a crossover point (e.g., $\beta \geq 2$ when α=0.6) is reached. This reason is that using too much indirect recommendations in subjective trust evaluation gives malicious nodes a higher change to successfully launch good-mouthing and bad-mouthing attacks. Figure 8.2 shows that using α=0.8 and β=2 yields subjective trust values very close to objective trust values in X=intimacy with the mean square error less than 0.3%.

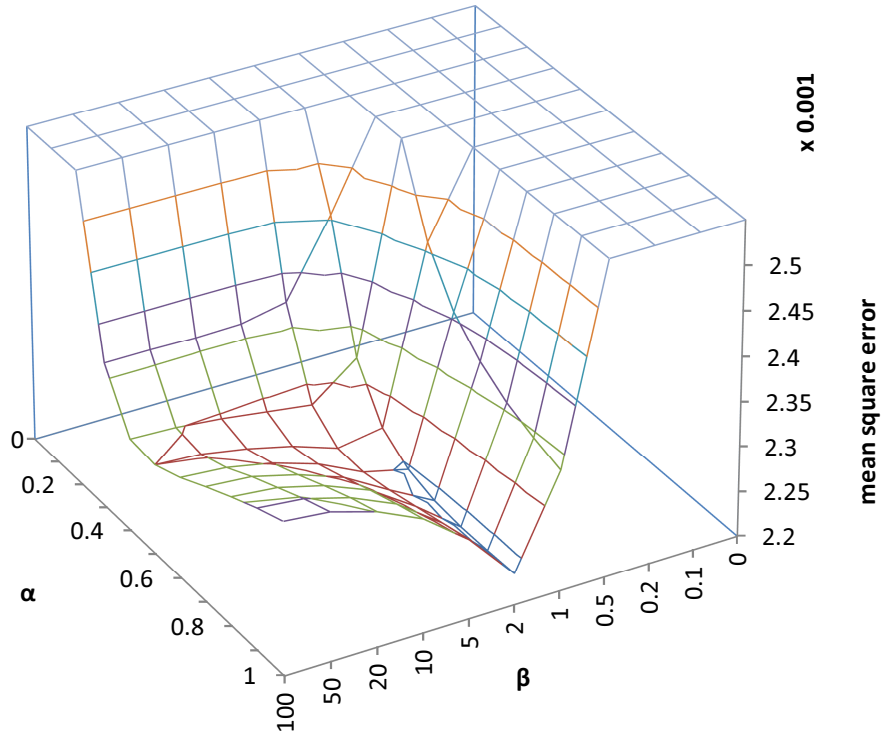


Figure 8.2: Effect of α and β on Accuracy of Trust Evaluation for X=Intimacy.

The best α and β values intrinsically depend on the nature of each trust property as well as a given set of parameter values as those listed in Table 8.2 characterizing the environmental and operational conditions. We summarize the best α and β values for each trust property in Table 8.3. The last column "MSE" shows the mean square error between subjective trust and objective trust in trust property X. Since the trust score in individual trust property X reflects the actual trust value in property X, the combined trust score given by Equation (8.1) will also reflect the actual trust value given by Equa-

tion (8.8) (i.e., with MSE ≤ 0.9% for any combination). Overall, we observe a close correlation between subjective trust evaluation and objective trust evaluation, thus supporting our claim that subjective trust obtained as a result of executing our proposed hierarchical trust management protocol approaches true objective trust.

Table 8.3: Best α and β Values for Trust Property X.

Trust Property	α	β	MSE
Intimacy	0.8	2	0.3%
Honesty	0.7	1	0.9%
Energy	0.6	1	0.1%
Cooperativeness	0.9	5	0.1%

8.5 Trust-Based Geographic Routing

In this section, we apply the proposed hierarchical trust management protocol to *trust-based geographic routing* as an application. In *geographic routing*, a node disseminates a message to a maximum of L neighbors closest to the destination node (or the sink node). In *trust-based geographic routing*, node i forwards a message to a maximum of L neighbors not only closest to the destination node but also with the highest trust values $T_{ij}(t)$. We conduct a performance analysis to compare our trust-based geographic routing protocol with baseline routing protocols, namely, flooding-based [171] and traditional geographic routing. In *flooding-based routing*, a node floods a message to all its neighbors until a copy of the packet reaches the destination node. It yields the highest message delivery ratio and the lowest message delay at the expense of the highest message overhead.

Recall that for all routing protocols, the source SN first forwards a message to its CH (through multiple hops if necessary). Then, the CH forwards the message to the sink node through other CHs. Without loss of generality, we normalize the average delay for forwarding a message between two neighbor SNs to τ. The average delay between two neighbor CHs is normalized to 2τ. We collect data for delivering 1000 messages, each with a source sensor and a sink node randomly selected. We consider two cases: L=1 and L=2 for both *trust-based geographic routing* and *geographic routing*. In the comparative analysis, we vary the degree of uncooperative or compromised nodes from 0% to 90%. Note that 30% of compromised or uncooperative nodes means that 30% of nodes are compromised or uncooperative in the system without a fixed ratio being used for these two types of nodes. We use parameter values as listed in Table 8.2 for characterizing environmental and operational conditions. We also use the optimal set of (α, β) for each

individual trust property as identified in Section 8.4 (see Table 8.3) to ensure subjective trust is close to objective trust.

8.5.1 Best Trust Formation to Maximize Application Performance

We first identify the best way to form trust out of social and QoS trust properties (i.e., identifying weights to assign to individual trust properties) so that the performance of trust-based geographical routing is maximized. Without loss of generality and for ease of disposition, we assume that the weights assigned to social trust properties, i.e., intimacy and honesty, are the same each of $0.5\times w_{social}$, and the weights assigned to QoS trust properties, i.e., energy and cooperativeness, are the same each of $0.5\times w_{QoS}$ with $w_{social} + w_{QoS} = 1$. Figure 8.3 shows the effect of w_{social} on the message delivery ratio of trust-based geographic routing with varying population percentage of compromised or uncooperative nodes. We observe that using solely either social trust (w_{social} =1) or QoS trust (w_{social} =0) yields a lower message delivery ratio, while considering both social and QoS trust properties helps generate a higher message delivery ratio. Figure 8.3 identifies that for the example WSN described in Section 8.1 characterized by a set of parameter values listed in Table 8.2, the maximum message delivery ratio performance is obtained when w_{soical}=0.4 and w_{QoS}=0.6. Hence, this weight setting represents the best trust formation in the trust-based geographical routing application.

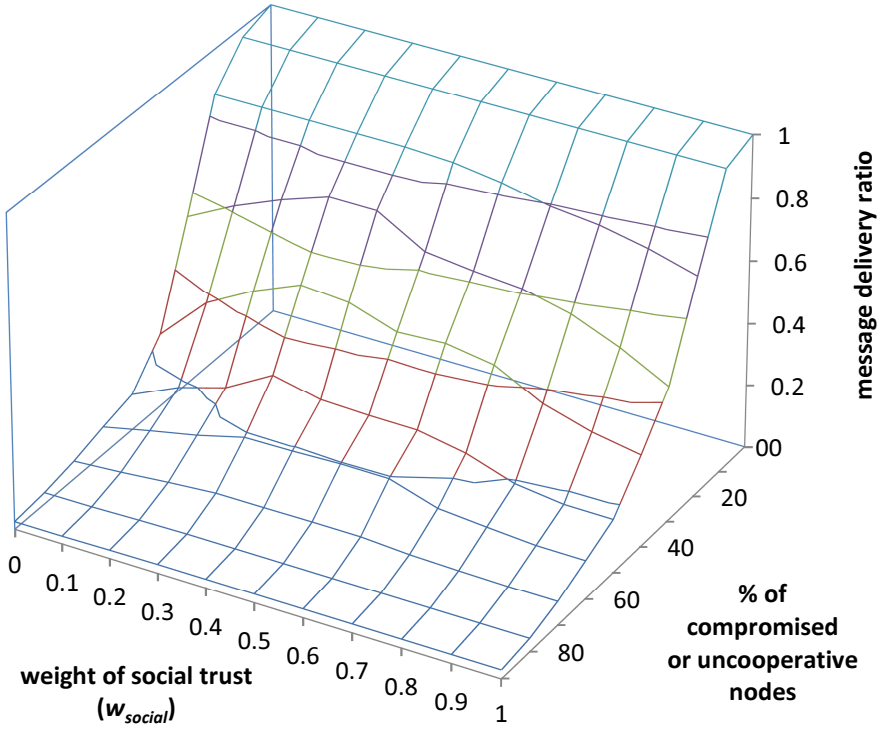


Figure 8.3: Effect of w_{social} on Message Delivery Ratio.

8.5.2 Dynamic Trust Management

Figure 8.3 illustrates the utility of dynamic trust management and application-level trust optimization for trust-based geographic routing applications, i.e., when the system senses that the hostility expressed in terms of the percentage of compromised or uncooperative nodes (the Y coordinate of Figure 8.3) is increasing, it can dynamically adjust w_{soical} (the X coordinate) to optimize application performance in message delivery ratio (the Z coordinate of Figure 8.3).

8.5.3 Performance Comparison

Figure 8.4 shows the message delivery ratio under various routing protocols. Our trust-based geographic routing protocol (L=1 or L=2) outperforms traditional geographic routing (L=1 or L=2) and approaches flooding-based routing, especially as the percentage of compromised or uncooperative nodes increases. The delivery ratio for all three routing protocols drops below 0.1 when the percentage of compromised or uncooperative nodes is higher than 80%. We observe that even the message delivery ratio of our trust-based geographic routing without redundancy (L=1) is higher than that of the geographic routing with redundancy (L=2) when the percentage of compromised or uncooperative nodes is higher than 40%. We attribute this to the ability of trust-based geographic routing being able to successfully avoid forwarding messages to untrustworthy nodes based on $T_{ij}(t)$ values obtained from our hierarchical trust management protocol.

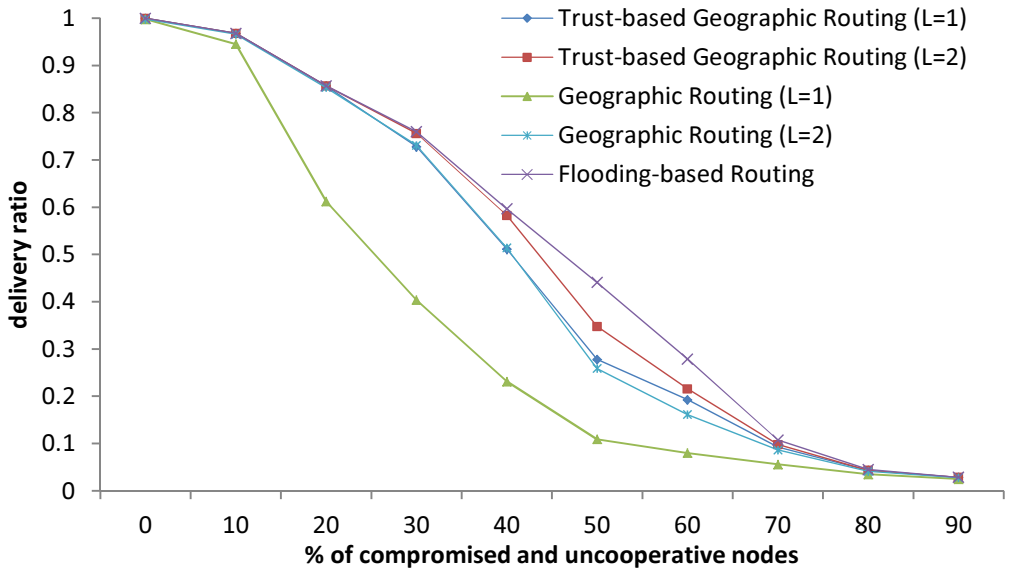


Figure 8.4: Message Delivery Ratio.

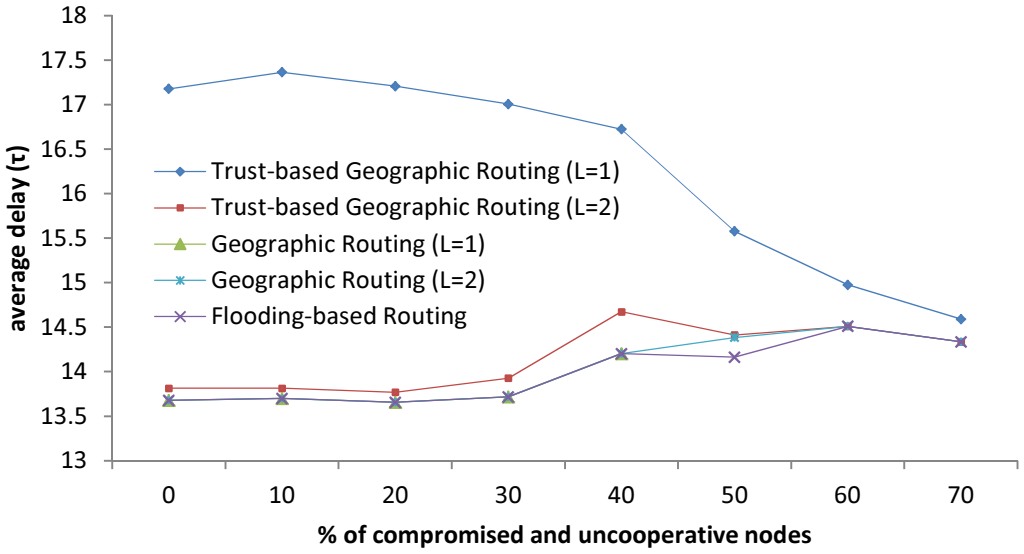


Figure 8.5: Message Delay with Source and Sink Node at a Distance Away.

Figure 8.5 shows the average delay for those messages that are successfully delivered under various routing protocols for a special case in which the source SN and the sink node are at least a distance ($700m$) away. We create this case to ensure there are sufficient intermediate nodes on any path to reach the sink node. We first observe that the message delivery delay increases as the percentage of compromised or uncooperative nodes increases due to more messages being dropped by uncooperative or malicious nodes resided on shorter routes. Flooding-based routing has the best performance since it can always find the shortest path to reach the destination sink node through flooding. Geographic routing (L=1 or L=2) has almost the same performance with flooding-based routing due to its greedy nature for selecting nodes closest to the destination sink node for message forwarding. However, geographic routing with L=1 fails to deliver any message when the percentage of compromised or uncooperative nodes is higher than 50% because there is no short route to reach the destination node over a long distance. Trust-based geographic routing with L=1 has the highest delay but with L=2 approaches the performance of flooding-based routing and geographic routing. In general traditional geographic routing performs better than trust-based geographic routing in message delay. This is expected because unlike traditional geographic routing, trust-based geographic routing tends to find forwarding nodes that are trustworthy but possibly not residing on the most direct path to the sink node. Consequently it incurs a higher delay compared with traditional geographic routing. However, we note that once we allow more message copies (e.g., L=2) to be disseminated by a node to its neighbors, trust-based geographic routing just like traditional geographic routing quickly approaches the ideal performance bound in message delay, especially as the percentage of compromised or uncooperative nodes increases.

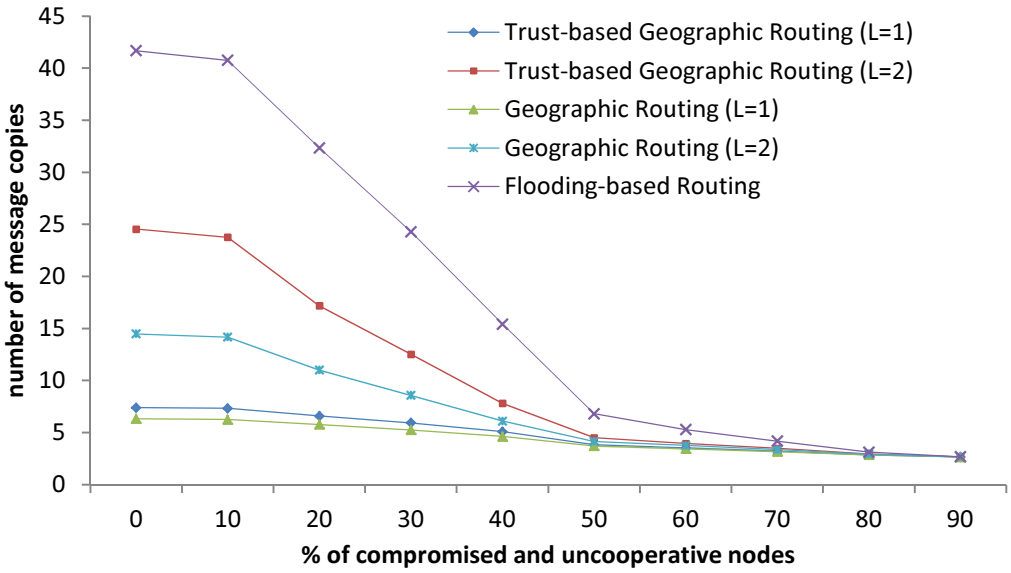


Figure 8.6: Message Overhead.

Figure 8.6 compares message overhead in terms of the number of message copies propagated before the destination sink node receives one copy. Both geographic routing and trust-based geographic routing perform significantly better than flooding-based routing. Trust-based geographic routing incurs more message overhead than traditional geographic routing because the path selected by trust-based geographic routing is often the most trustworthy path, not necessarily the shortest path. Nevertheless, we observe that the overhead increase of trust-based geographic routing over traditional geographic routing is small compared with that of flooding-based routing over traditional geographic routing. The system thus can effectively trade off message overhead for message delivery ratio and message delay. Finally, we observe that the number of message copies propagated for all three routing protocols is close to 3 when the percentage of compromised or uncooperative nodes is higher than 80%. The reason is that the message can be successfully delivered only when the source node and the sink node are close to each other. Otherwise, there is a high probability that compromised and uncooperative nodes reside on a long route will drop the message copies received.

Overall Figure 8.4 to Figure 8.6 demonstrate that our trust-based geographic routing protocol with L=2 can significantly improve the delivery ratio and message delay (close to those of flooding-based routing) in the presence of compromised or uncooperative nodes, without sacrificing too much message overhead. Here we note that the system can effectively trade off message overhead (energy consumption) for high delivery ratio and low message delay by adjusting the level of redundancy (L). As L increases the performance of our trust-based geographic routing protocol in delivery ratio and message delay will approach that of flooding-based routing.

8.6 Trust-Based Intrusion Detection

In this section we apply hierarchical trust management to trust-based intrusion detection as another application. We first describe the algorithm that can be used by a high-level node such as a CH (or a base station) to perform trust-based intrusion detection of the SNs (or CHs respectively) under its control. Then we develop a statistical method to assess trust-based IDS false positive and false negative probabilities.

Without loss of generality, in this section we illustrate how a CH performs trust-based intrusion detection on SNs in its cluster. A similar treatment applies to a base station performing trust-based intrusion detection on CHs in a WSN.

8.6.1 Algorithm for Trust-Based Intrusion Detection

Our trust-based IDS algorithm is based on selecting a system minimum trust threshold below which a node is considered compromised and needs to be excluded from sensor reading and routing duties. The underlying principle is that a compromised node will exhibit several social and QoS trust behaviors, i.e., low *intimacy* and low *honesty* (for social trust) as well as low *energy* and low *cooperativeness* (for QoS trust), thus exposing itself as a compromised node under hierarchical trust evaluation.

A CH performs CH-to-SN trust evaluation toward node j after receiving $T_{ij}(t)$ values from all SNs in the cluster. More specifically a CH, c, when evaluating a SN, j, will compute node j's trust value, $T_{cj}(t)$, by Equation (8.4). CH c will announce j as compromised if $T_{cj}(t)$ is less than T^{th}; otherwise, node j is not compromised.

8.6.2 Statistical Analysis

Consider that the trust value toward node j is a random variable following normal distribution commonly used for statistical analyses with mean value $\mu_j(t)$. Also consider that there are n sample values of $T_{ij}(t)$ submitted by n SNs considered trustworthy by the CH. With these n sample values, $X_j(t)$ is related to the sample mean, sample standard deviation and true mean following t-distribution with n - 1 degree of freedom as follows:

$$X_j(t) = \frac{\overline{T_{ij}(t)} - \mu_j(t)}{S_j(t)/\sqrt{n}} \tag{8.9}$$

where $\overline{T_{ij}(t)}$, $S_j(t)$, and $\mu_j(t)$ are the sample mean, sample standard deviation, and true mean of node j's trust value at time t, respectively. Thus, the probability that node j is diagnosed as a compromised node at time t is:

$$\Theta_j(t) = \Pr\left(\mu_j(t) < T^{th}\right) = \Pr\left(X_j(t) > \frac{\overline{T_{ij}(t)} - T^{th}}{S_j(t)/\sqrt{n}}\right) \tag{8.10}$$

The false positive of the IDS can be obtained by calculating $\Theta_j(t)$ under the condition that node *j* is not compromised. Similarly, the false negative probability can be obtained by calculating $1 - \Theta_j(t)$ under the condition that node *j* is compromised.

$$P_j^{fp}(t) = \Pr\left(X_j(t) > \frac{\overline{T_{ij}^N(t)} - T^{th}}{S_j^N(t)/\sqrt{n}}\right) \tag{8.11}$$

$$P_j^{fn}(t) = \Pr\left(X_j(t) \leq \frac{\overline{T_{ij}^C(t)} - T^{th}}{S_j^C(t)/\sqrt{n}}\right) \tag{8.12}$$

Equation (8.11) and Equation (8.12) above give the false positive probability, $P_j^{fp}(t)$, and false negative probability, $P_j^{fn}(t)$, of our proposed trust-based intrusion detection algorithm at time *t*, respectively. $\overline{T_{ij}^N(t)}$ and $S_j^N(t)$ are the mean value and standard deviation of node *j*'s trust values reported by other nodes in the same cluster, under the condition that node *j* is not compromised. $\overline{T_{ij}^C(t)}$ and $S_j^C(t)$ are the mean value and standard deviation, under the condition that node *j* is compromised. $T_{ij}^N(t)$ and $T_{ij}^C(t)$ can be easily obtained by applying the Bayes' theorem to the calculation of $T_{ij}(t)$.

$P_j^{fp}(t)$ and $P_j^{fn}(t)$ vary over time. The average false positive and false negative probabilities, denoted by P_j^{fp} and P_j^{fn}can be obtained by weighting on the probability of node *j* being compromised at time *t*, i.e.,

$$P_j^{fp} = \frac{\sum_{t=0}^{SL}\left(P_j^{fp}(t)\left(1 - P_j^C(t)\right)\right)}{\sum_{t=0}^{SL}\left(1 - P_j^C(t)\right)} \tag{8.13}$$

$$P_j^{fn} = \frac{\sum_{t=0}^{SL}\left(P_j^{fn}(t)P_j^C(t)\right)}{\sum_{t=0}^{SL} P_j^C(t)} \tag{8.14}$$

where $P_j^C(t)$ is the probability that node *j* is compromised at time *t* which can be obtained from the SPN model output, and *SL* is the anticipated WNS lifetime period over which the weighted calculation is performed.

8.6.3 Best Trust Formation to Maximize Application Performance

Here we identify the best way to form trust out of social and QoS trust properties (i.e., identifying weights to assign to individual trust properties) and to assign the minimum trust threshold, T^{th}, so that the performance of trust-based intrusion detection is maximized, i.e., both false positives and false negatives are minimized. We again consider the example WSN described in Section 8.1 characterized by a set of parameter values listed in Table 8.2 with its lifetime SL=150 days.

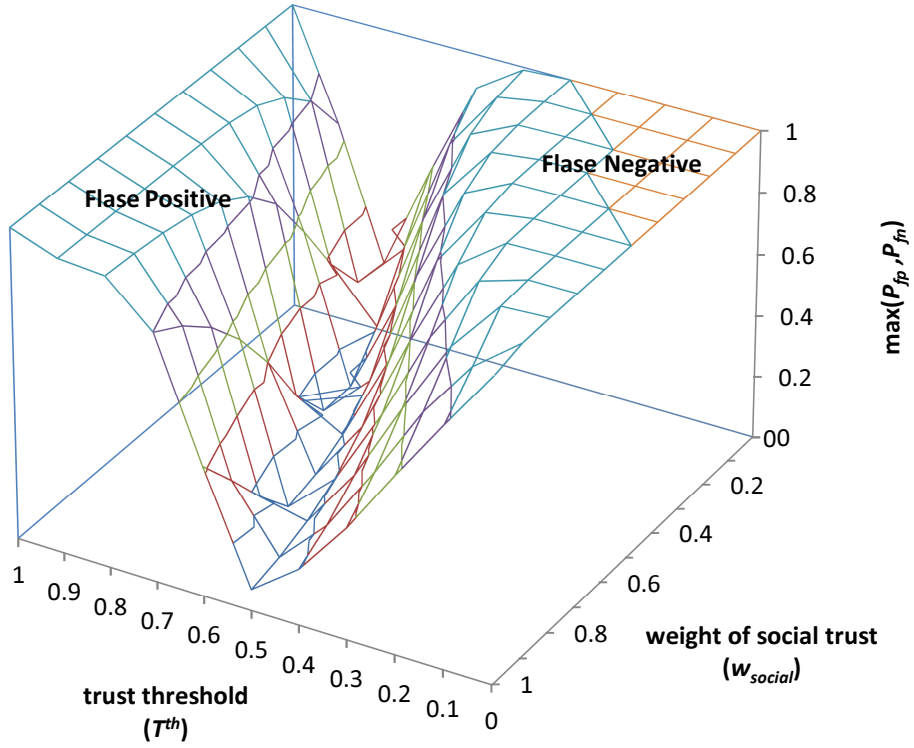


Figure 8.7: Effect of T^{th} and w_{social} on $max(P_{fp}, P_{fn})$.

Figure 8.7 shows $max(P_{fp}, P_{fn})$ vs. T^{th} and w_{social} in this system as a result of executing our trust-based intrusion detection algorithm, where P_{fp} and P_{fn} are the time-averaged false positive and false negative probabilities as calculated from Equation (8.13) and Equation (8.14), respectively, over all nodes in the system. We observe that as the minimum trust threshold T^{th} increases, the false negative probability P_{fn} decreases while the false positive probability P_{fp} increases. More importantly, there exists an optimal trust threshold $T^{th,opt}$ at which both false negative and false positive probabilities are minimized. As trust formation affects how trust is formed from social and QoS trust components, we also observe that $T^{th,opt}$ is sensitive to w_{social}. Figure 8.7 identifies that for the example WSN when $T^{th,opt} = 0.6$ and $w_{social} = 0.6$, both false positive and false negative probabilities are minimized to fall below 5%.

8.6.4 Dynamic Trust Management

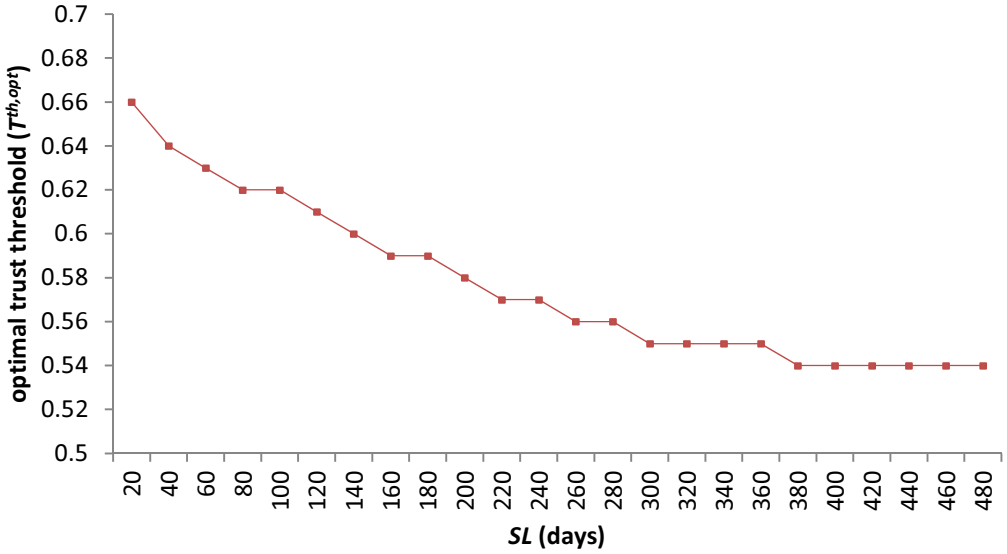


Figure 8.8: Optimal Trust Threshold vs. System Lifetime.

Figure 8.7 is for the case in which the expected system lifetime SL is 150 days of operations. Figure 8.8 shows the optimal trust threshold $T^{th,opt}$ as SL varies. Here, the value of w_{social} is fixed to 0.6 to isolate its effect. For a WSN with a prolonged operation, SL represents a time point characterized by a distinct hostility level such as the percentage of compromised and uncooperative nodes. We observe that as SL increases, the value of $T^{th,opt}$ at which the false alarm probability is minimized decreases. The reason is that a node's trust value decreases over time due to energy depletion even if the node is not compromised. The system sensing hostility change at runtime can apply the best w_{social} and $T^{th,opt}$ setting identified from static analysis to optimize application performance in false alarm probability.

8.6.5 Performance Comparison

We perform a comparative performance analysis of our trust-based intrusion detection algorithm with two anomaly detection schemes, namely, weighted summation [85] and data clustering [122]. We use the ROC (Receiver Operating Characteristic) curve [122] as the performance metric since both false negative probability (P_{fn}) and false positive probability (P_{fp}) are critical measures and ROC objectively reflects the sensitivity of detection probability (i.e., 1 - P_{fn}) as the false positive probability varies.

The first baseline anomaly detection scheme is weighted summation-based IDS [85]. In this approach, each SN has a weight associated with it and this weight changes dynamically, reflecting the trustworthiness of the SN's output relative to the average out-

put out of all SNs. We use the trust recommendation value from each SN toward a particular SN, say, SN_i, as the SN's output. The average trust recommendation value is obtained by a summation of the trust recommendation values weighted by the respective weights from all SNs except SN_i. If the trust recommendation value from a SN deviates too much from the average value, the weight value associated with that SN decreases by θ (weight penalty); otherwise the weight value remains the same. The weight value is updated dynamically until it falls below a weight threshold (w_t), in which case the corresponding SN is reported as malicious. The weight penalty (θ) and weight threshold (w_t) largely determine the false positive probability. We vary θ and w_t over the range of [0, 1] to obtain the detection probability as the false positive probability varies.

The second baseline anomaly detection scheme is fixed width data clustering-based IDS [122]. In this approach, the maximum radius of a cluster (c_w) is defined and a data point is put into a cluster if the distance between the centroid of the cluster and this data point is smaller than c_w; otherwise this data point makes a new cluster. Data points that exhibit dissimilarity with others will tend to cluster into a small cluster or standalone by themselves. These lone data points are reported as malicious. To apply fixed width data clustering-based IDS, we use trust values of SNs as collected by a CH as data points for clustering. As the maximum radius of a cluster c_w affects the false positive and negative probabilities, we vary c_w over the range of [0, 0.2] to collect the performance results.

In our trust-based intrusion detection algorithm, the false positive and negative probabilities essentially depend on the minimum trust threshold (T^{th}) and the weight of social trust (w_{social}). We vary these two parameters over the range of [0, 1] to collect the performance results.

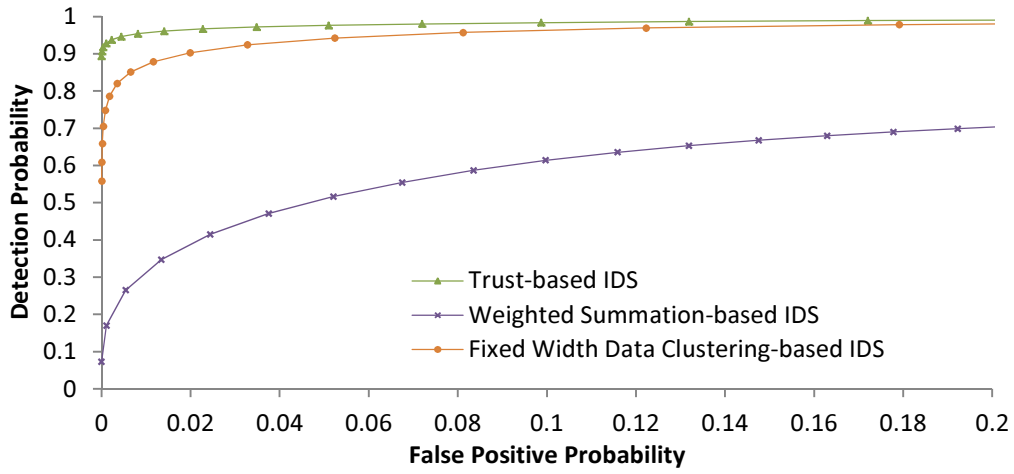


Figure 8.9: ROC Curves for IDS Performance Comparison.

In Figure 8.9 we compare the ROC curves of our trust-based IDS algorithm against those by weighted summation-based IDS and fixed width data cluster-based IDS for

SL=240 *days*. The results presented are the best results of all three IDS schemes by fine-tuning the design parameters as described above under the same network environment characterized by Table 8.2.

We observe from Figure 8.9 that as a design tradeoff, as the false positive probability increases, the detection probability increases for all IDS schemes. We observe that our trust-based IDS algorithm outperforms both weighted summation-based IDS and fixed width data clustering-based IDS, especially when the false positive probability is limited to 5% which is considered desirable in intrusion detection. The strength of our trust-based IDS algorithm is especially pronounced when the false positive probability approaches zero. This is very desirable since our trust-based IDS algorithm can still maintain a high detection probability (> 90%) when the false positive probability is close to zero at which the detection probability of anomaly detection-based IDS schemes drops sharply.

8.7 Summary

In this chapter, we proposed a hierarchical dynamic trust management protocol for cluster-based wireless sensor networks, considering two aspects of trustworthiness, namely, social trust and QoS trust. We developed a probability model utilizing stochastic Petri nets techniques to analyze the protocol performance, and validated subjective trust against objective trust obtained based on ground truth node status. We demonstrated the feasibility of dynamic hierarchical trust management and application-level trust optimization design concepts with trust-based geographic routing and trust-based IDS applications, by identifying the best way to form trust as well as use trust out of individual social and QoS trust properties at runtime to optimize application performance. The results indicated that our trust-based geographic routing protocol performs close to the ideal performance of flooding-based routing in delivery ratio and message delay without sacrificing much in message overhead compared with traditional geographic routing protocols which does not use trust. Our trust-based IDS algorithm outperforms traditional anomaly-based IDS techniques in the detection probability while maintaining sufficiently low false positives.

Chapter 9

Dynamic Trust Management for Internet of Things and Its Applications

In this chapter, we apply design and validation principles of dynamic trust management to Internet of Things (IoT) systems and propose a dynamic and scalable trust management protocol of social IoT systems to meet the scalability, compatibility, extendibility, dynamic adaptability, and resiliency requirements. An IoT system connects a large amount of tags, sensors, and mobile devices to facilitate information sharing, enabling a variety of attractive applications. Recognizing that entities in an IoT system are connected through social networks of entity owners, we consider a community of interest (CoI) based social IoT where nodes form into communities of interest. We formally prove the convergence, accuracy, and resiliency properties of our dynamic trust management protocol against trust attacks (see Appendix A). Moreover, we reveal the design tradeoff between trust convergence vs. trust fluctuation. With our dynamic trust management protocol, a social IoT application can adaptively choose the best trust parameter settings in response to changing IoT social conditions such that not only trust assessment is accurate but also the application performance is maximized. Further, given inter-CoI vs. intra-CoI social connections among entity owners as input, we identify best trust protocol settings for achieving convergence, accuracy, dynamic adaptability and resiliency properties in the presence of dynamically changing conditions and malicious nodes performing trust-related attacks. For scalability, we consider a design by which a node only keeps trust information of a subset of nodes meeting its interest and performs minimum computation to update trust (see Section 10.2).

We validate our design by extensive simulation considering both limited and ideal (unlimited) storage space. The results demonstrate that our trust management protocol using limited storage space achieves a similar performance level compared with the one under ideal storage space, and a newly join node can quickly build up trust towards other nodes with desirable accuracy and convergence behavior. The utility of dynamic

trust management is demonstrated by a trust-based service composition application in social IoT environments.

9.1 System Model

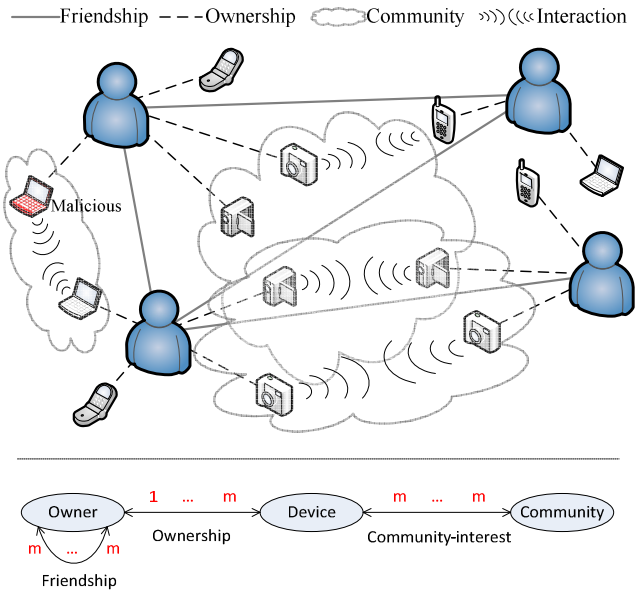


Figure 9.1: Social Relationships in a Social IoT System.

Figure 9.1 illustrates the system model for a social IoT system with socially interacting entities. We consider a social IoT environment with no centralized trusted authority. Each node is a device carried by a user. A node is able to autonomously and independently interact with other nodes, performing computation and storing information as necessary. Every device (node) has an owner and an owner could have many devices. Each owner has a list of friends, representing its social relationships. Since the ownership is a one-to-multiple relationship, when we say that two nodes (devices) are friends, we mean that their owners are friends. A device is carried or operated by its owner in certain communities of interest or working environments. Nodes belonging to a similar set of communities likely have similar interests or similar capabilities.

Two nodes belonging to the same CoI have specific social interests and strong social ties, which could be manifested by more frequent interactions. In addition, node from different communities may have different or controversial views of trust [127] towards the same trustee due to their different social interests. The multiple views of trust lead to different trust assessments even though the same behavior of the trustee is observed. This is the case especially in social IoT environments. We assume that nodes in the same CoI can achieve an agreement on trust since they share the same interests. The goal of our trust management is to make sure each node's trust evaluation converges to its

community agreement (henceforward called *CoI ground truth*). Note that although we assume the existence of the communities, a node may or may not be aware which CoI it belongs to. We consider a large IoT system in which each node has limited storage space and cannot accommodate the full set of trust values towards all other nodes. Each node can voluntarily join or leave the system.

We differentiate uncooperative nodes from malicious nodes. An uncooperative node acts for its own interest. So it may stop providing service to a service requester if it does not have a strong social tie (e.g., friendship) with the service requester. A malicious node aims to break the basic functionality of the IoT. In addition, when two nodes (devices) interact with each other, a malicious node can perform the following trust-related attacks to the other node:

- *Self-promoting attacks*: it can promote its importance (by providing good recommendations for itself) so as to be selected as the service provider, but then stop providing service or provide malfunction service.
- *Bad-mouthing attacks*: it can ruin the reputation of a well-behaved node by providing bad recommendations against the good node so as to decrease the chance of this good node being selected as a service provider.
- *Good-mouthing attacks*: it can boost the reputation of another bad node by providing good recommendations for it so as to increase the chance of this bad node being selected as a service provider.

A node's trust value is assessed based on direct observations and indirect information like recommendations. The trust of one node toward another node is updated upon encounter and interaction events. Each node will execute the trust protocol independently and will perform its direct trust assessment toward an encountered node based on specific detection mechanisms designed for assessing a trust property. Later we will discuss specific detection mechanisms employed for adaptive trust management.

9.2 Dynamic and Scalable Trust Management

The components of adaptive trust management for a social IoT system are shown in Figure 9.2. Our protocol addresses all aspects of trust management: the trust *composition* component addresses the issue of how to select multiple trust properties in IoT according to application requirements. The trust *propagation* and *aggregation* component addresses the issue of how to disseminate and combine trust information such that the trust assessment converges and is accurate. The trust *formation* addresses the issue of how to form the overall trust out of individual trust properties and how to make use of trust in order to maximize application performance. Essentially adaptive trust man-

agement is achieved by (1) selecting the best trust aggregation parameter setting to maximize trust evaluation accuracy and (2) selecting the best trust formation parameter setting to maximize application performance, in response to an evolving IoT environment.

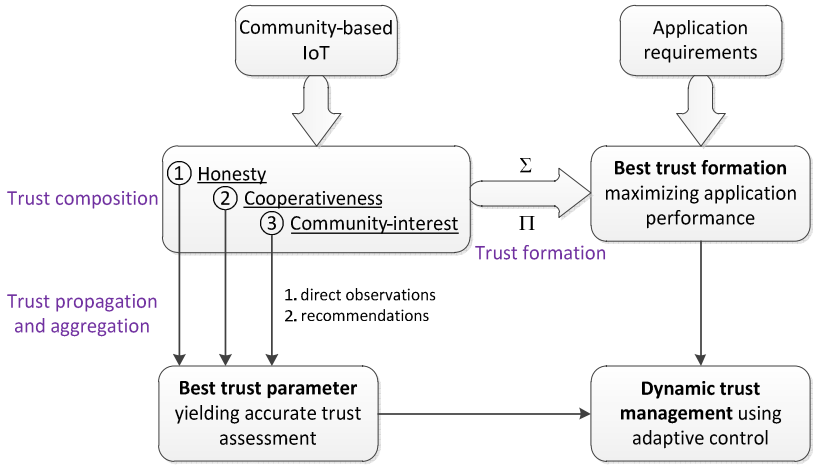


Figure 9.2: Adaptive Trust Management for a Social IoT System.

Adaptive trust management for social IoT must be distributed as a social IoT system frequently consists of free-will entities without a centralized mediator. Each node maintains its own trust assessment towards other nodes. For scalability, a node keeps its trust evaluation towards a limited set of nodes which it is most interested in. The scalable storage management is given in Section 10.2, since the design is generically applicable to dynamic trust management in other network environments. Adaptive trust management is encounter-based as well as activity-based, meaning that the trust value is updated dynamically upon an encounter event or an interaction activity. Two nodes encountering each other or involved in a direct interaction activity can directly observe each other and update their trust assessment. They also exchange their trust evaluation results toward other nodes as recommendations.

9.2.1 Trust Composition

While there is a wealth of social trust metrics available [21, 62] we choose *honesty*, *cooperativeness*, and *community-interest* as most striking social trust metrics for characterizing IoT systems, as illustrated in Figure 9.2 (2nd level):

- The *honesty* trust property represents whether or not a node is honest. In IoT, a node can be compromised, and then dishonest when providing services or trust recommendations. We select *honesty* as a trust property because a compromised node can disrupt trust management and service continuity of an IoT application.

- The *cooperativeness* trust property represents whether or not the trustee node is socially cooperative [114] with the trustor node. A node may follow the protocol execution only when interacting with its friends or nodes with strong social ties (with more common friends), but become uncooperative when interacting with other nodes. In an IoT application, a node can evaluate the cooperativeness property of other nodes based on social ties and select socially cooperative nodes in order to achieve high application performance.
- The *community-interest* trust represents whether or not the trustor and trustee nodes are in the same social communities/groups (e.g. co-location or co-work relationship [12]) or have similar capabilities (e.g., parental object relationship [12]). Two nodes with a degree of high community-interest trust have more chances and experiences in interacting with each other, and thus can result in better application performance.

Below we discuss how a node evaluates other nodes in *honesty*, *cooperativeness*, and *community-interest* trust properties by combining first-hand direct observations and second-hand recommendations.

9.2.2 Trust Propagation and Aggregation

Adaptive trust management is a continuing process which iteratively aggregates past information and new information. The new information includes both direct observations (first-hand information) and indirect recommendations (second-hand information). The trust assessment of node i towards node j at time t is denoted by $T_{ij}^{X}(t)$ where X = *honesty*, *cooperativeness*, or *community-interest*. The trust value $T_{ij}^{X}(t)$ is a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. For mobile IoT environments, node i and node j interact or encounter each other when they are within wireless radio range of each other. When node i encounters or directly interacts with another node k at time t, node i will update its trust assessment $T_{ij}^{X}(t)$ as follows:

$$T_{ij}^{X}(t) = \begin{cases} (1-\alpha)T_{ij}^{X}(t-\Delta t) + \alpha D_{ij}^{X}(t) & if\ j == k \\ (1-\gamma)T_{ij}^{X}(t-\Delta t) + \gamma R_{kj}^{X}(t) & if\ j\ != k \end{cases} \quad (9.1)$$

Here, Δt is the elapsed time since the last trust update. If node k is node j itself, node i will use its new trust assessment toward node j based on direct observation ($D_{ij}^{X}(t)$) and its old trust toward node j based on past experiences to update $T_{ij}^{X}(t)$. A parameter α ($0 \leq \alpha \leq 1$) is used here to weigh these two trust values and to consider trust decay over time, i.e., the decay of the old trust value and the contribution of the new trust val-

ue. A larger α means that trust evaluation will rely more on direct observations. On the other hand, when node k is not node j, node k will serve as a recommender to provide a trust recommendation about node j, $R_{kj}^{X}(t)$, to node i. Another parameter γ $(0 \leq \gamma \leq 1)$ is used here to weigh the recommendation trust $R_{kj}^{X}(t)$ with respect to the old trust information which decays over time.

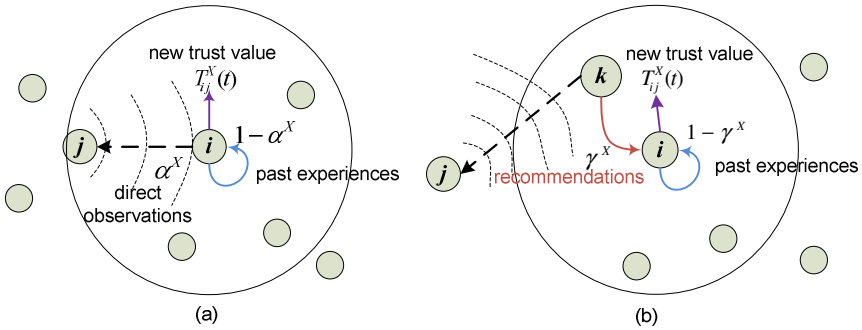


Figure 9.3: Trust Propagation and Aggregation.

Figure 9.3 illustrates the trust propagation and aggregation protocol. A key concept of our adaptive trust management protocol is that instead of having fixed weight ratios α and γ, we allow the weight ratios to be adjusted dynamically in response to changing network conditions to improve trust assessment accuracy and thus provide resiliency against slandering attacks such as good-mouthing and bad-mouthing attacks. Below we discuss in more detail how direct observation trust $D_{ij}^{X}(t)$ and indirect recommendation $R_{kj}^{X}(t)$ are obtained.

(1) Direct Observation Trust $D_{ij}^{X}(t)$

As shown in Figure 9.3(a), when node i has direct observations on node j, it will update the trust toward node j by using past information and new direct observations. For each trust property X = *honesty*, *cooperativeness*, or *community-interest*, the direct observation can be obtained as follows.

Honesty - $D_{ij}^{honesty}(t)$: This trust property refers to the belief of node i that node j is honest based on node i's direct observations toward node j. Node i estimates $D_{ij}^{honesty}(t)$ by keeping a count of suspicious dishonest experiences of node j which node i has observed during $[0, t]$ using a set of anomaly detection rules such as a high discrepancy in recommendation has been experienced, as well as interval, retransmission, repetition, and delay rules as in [88, 170]. If the count exceeds a system-defined threshold, node j is considered totally dishonest at time t, i.e., $D_{ij}^{honesty}(t) = 0$. Otherwise, $D_{ij}^{honesty}(t)$ is computed by 1 minus the ratio of the count to the threshold. Our hypothesis is that a compromised node must be dishonest. We consider non-zero false positive

probability (P_{fp}) and false negative probability (P_{fn}) for such direct detection mechanism.

Cooperativeness - $D_{ij}^{cooperativeness}(t)$: This trust property provides the degree of cooperativeness of node *j* as evaluated by node *i* based on direct observations over $[0, t]$. We use the social friendship [114] among device owners to characterize the cooperativeness. Our hypothesis is that friends are likely to be cooperative toward each other. The cooperativeness trust of node *i* towards node *j* is computed as the ratio of the number of common friends over the total number of nodes *i*'s and *j*'s friends, i.e., $\frac{|friends(i) \cap friends(j)|}{|friends(i) \cup friends(j)|}$, where $friends(i)$ denotes the set of node *i*'s friends. A node is included in its own friend list (i.e., $i \in friends(i)$) to deal with the case where two nodes are the only friends to each other. When node *i* and node *j* encounter and directly interact with each other, they can exchange their friend lists. Node *i* can validate a friend in node *j*'s list if it is their common friend. Therefore, the direct observation of cooperativeness will be close to actual status.

Community-Interest - $D_{ij}^{community-interest}(t)$: This trust property provides the degree of the common interest or similar capability of node *j* as evaluated by node *i* based on direct observations over $[0, t]$. The community-interest trust of node *i* towards node *j* is computed as the ratio of the number of common community/group interests over the total number of nodes *i*'s and *j*'s community/group interests, i.e., $\frac{|community(i) \cap community(j)|}{|community(i) \cup community(j)|}$, where $community(i)$ denotes the set of node *i*'s communities/groups. When node *i* and node *j* encounter and directly interact with each other, they can exchange their service and device profiles. Node *i* can validate whether node *j* and itself are in a particular community/group. Therefore, the direct observation of community-interest will be close to actual status.

(2) Indirect Recommendation $R_{kj}^{X}(t)$

In Equation (9.1), if node *k* is not node *j*, then node *i* will not have direct observation on node *j* and will use its past experience $T_{ij}^{X}(t - \Delta t)$ and a recommendation from node *k* (second-hand information $R_{kj}^{X}(t)$ where *k* is the recommender) to update $T_{ij}^{X}(t)$ (Figure 9.3(b)). The parameter γ is used here to weigh recommendations vs. past experiences and to consider trust decay over time as follows:

$$\gamma = \frac{\beta D_{ik}^{X}(t)}{1 + \beta D_{ik}^{X}(t)} \tag{9.2}$$

Here we introduce another parameter $\beta \geq 0$ to specify the impact of "indirect recommendation" on $T_{ij}^{X}(t)$ such that the weight assigned to indirect recommendation

$R_{kj}^{X}(t)$, is normalized to $\beta T_{ik}^{X}(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust, i.e., $R_{kj}^{X}(t)$, increases proportionally as either $D_{ik}^{X}(t)$ or β increases. Here, $D_{ik}^{X}(t)$ is node i's trust in component X toward node k as a recommender (for node i to judge if node k provides correct information). The recommendation $R_{kj}^{X}(t)$ provided by node k to node i about node j depends on the status of a node. If node k is a good node, node k (being a good node) will faithfully use its trust evaluation towards node j in component X as the recommendation, i.e., $R_{kj}^{X}(t)$ is simply equal to $D_{kj}^{X}(t)$. If node k is a bad node, node k (being a bad node) will perform bad-mouthing attacks by recommending $R_{kj}^{X}(t) = 0$ if node j is a good node, and will perform good-mouthing attacks by recommending $R_{kj}^{X}(t) = 1$ if node j is a bad node. Here we note that the protocol described by Equation (9.1) eliminates self-promoting attacks as a trustor node (node i) does not allow a trustee node (node k) to promote itself.

The underlying idea of our trust protocol is Bayesian reputation system [76] where each node calculates the trust using Bayesian estimation over historical observations. Our protocol takes an iterative approach to aggregate new direct observations with past information considering trust decay and to aggregate new recommendations with past information considering trust discounting. When two nodes have interaction and obtain the direct interaction experience, they can update the trust towards each other with minimum computation (a single iteration). In Appendix A, we analyze the convergence, accuracy, and resiliency properties of our proposed trust propagation and aggregation protocols.

9.2.3 Trust Formation

Trust formation depends on the trust requirement of the IoT application running on top of the trust management protocol. The goal of our adaptive trust management design in trust formation is to dynamically discover the best way to form trust out of identified trust components to maximize the application performance, in response to dynamically changing conditions. We discuss and illustrate adaptive trust formation design with a trust-based service composition application in Section 9.5.

9.2.4 Adaptive Control

We investigate a method based on adaptive filter theory [84] to adjust trust parameters dynamically in order to minimize trust bias. Here, we demonstrate how to dynamically calculate the best trust parameters α and β in Equations (9.1) and (9.2).

A successful trust management protocol should provide high trust toward nodes who have more positive direct feedbacks and, conversely, low trust toward those with more negative direct feedbacks. Specifically, the current trust evaluation (i.e., $T_{ij}(\alpha, \beta)$

as a function of α and β) should be close to the future direct feedbacks (i.e., $\overline{D_{ij}(\Delta t)}$, the average trust of new feedbacks within a time window Δt). Note that for notational convenience, we omit the superscript for trust property X and time variable t in $T_{ij}(\alpha, \beta)$ and $\overline{D_{ij}(\Delta t)}$. Therefore, we consider the selection of α and β as an optimization problem as follows:

$$\underset{(\alpha,\beta)}{\text{Minimum}}\, MSE = \sum_{j} \left(T_{ij}(\alpha,\beta) - \overline{D_{ij}(\Delta t)}\right)^2 \quad \text{s.t. } 0 \le \alpha \le 1, \beta \ge 0 \tag{9.3}$$

The optimization problem is to minimize the mean square error (MSE) of trust evaluations against actual feedbacks towards all target nodes. After node i obtains new direct feedbacks, it can compute the average feedback value $\overline{D_{ij}(\Delta t)}$ and can search for optimal α and β values which minimize MSE in Equation (9.3). The analytical solution can be derived if the optimization problem of Equation (9.3) is linear. Otherwise, we can use nonlinear numerical optimization methods such as downhill simplex [143], and nonlinear conjugate gradient [83] to find optimal α and β values. We will discuss the practicality issue of this method in Section 10.1.

9.3 Sensitivity Analysis on Trust Evaluation

In this section, we perform sensitivity analysis of trust parameters and network environment on trust evaluation through simulation. The simulation results are obtained from executing our adaptive trust management protocol by IoT devices in a social IoT system.

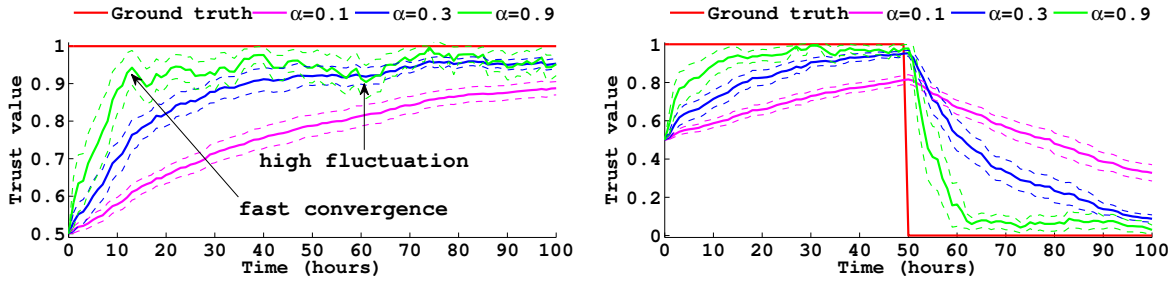
Table 9.1: Parameter Values for Sensitivity Analysis.

Parameter	Meanings	Value
N_T	Number of devices	50
N_M	Number of service providers	5
N_G	Number of user communities	10
N_H	Number of owners	20
λ	Percentage of malicious nodes	[0, 90%]
P_{fp}	Detection false positive probability	5%
P_{fn}	Detection false negative probability	5%
α	Weight on direct observation vs. past experiences in direct trust	[0, 1]
β	Weight on recommendations vs. past experiences in indirect trust	[0, 1]
T_S	Simulation period	100 hours

Table 9.1 lists the parameters used and their values. We consider a social IoT environment with $N_T = 50$ heterogeneous smart objects/devices. These devices are randomly distributed to $N_H = 20$ owners. The social cooperativeness relationship among the devices is characterized by the friendship relationship (matrix) [114] among device owners. That is, if the owners of devices i and j are friends, then there is a 1 in the ij position. Devices are used by their owners in one or more social communities or groups. A device can belong to up to $N_G = 10$ communities or groups. We consider a random waypoint mobility model where nodes move randomly and encounter or directly interact with each other when they are within the radio range. The average encountering frequency is about 0.25 per pair per hour. The total simulation time T_S is 100 hours. We consider a hostile environment where the percentage of dishonest nodes $\lambda \in [0\%, 90\%]$ is randomly selected out of all devices. A normal or good node follows the execution of the adaptive trust management protocol, while a dishonest node acts maliciously by providing false trust recommendations (good-mouthing and bad-mouthing attacks) to disrupt trust management. The initial trust value of all devices is set to ignorance with a trust level of 0.5.

9.3.1 Effect of α on Trust Evaluation

We first investigate the effect of design parameter α on trust evaluation. Recall that α is the weight associated with direct trust with respect to past experience in Equation (9.1). We vary the value of α by selecting different values (0.1, 0.3, and 0.9) and fix the value of β to 0 to isolate its effect. Here we only give the results for the *honesty* trust property evaluation. The other two trust properties follow the same trend.



(a) Static Environment (b) Dynamic Environment

Figure 9.4: Effect of α on Honesty Trust Evaluation.

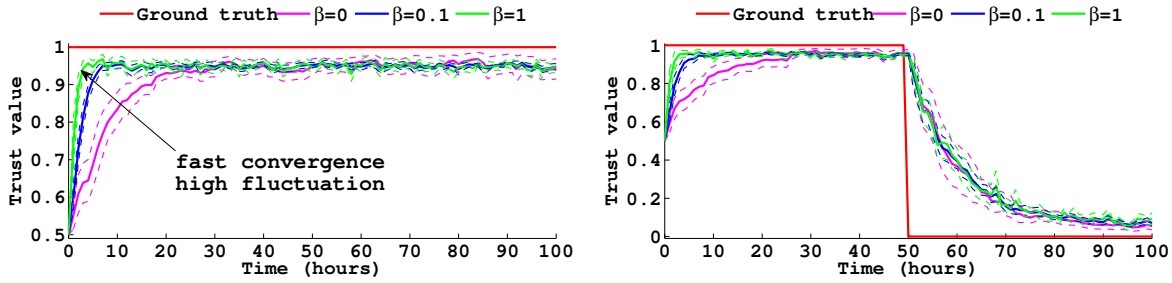
Figure 9.4(a) shows the effect of α on *honesty* trust evaluation in static environments where the ground truth status does not change as time t increases. The ground truth status is constant 1, which means that the target node is honest. Initially, the trust value is set to ignorance (0.5) due to lack of knowledge. At the early stage of trust evaluation, this may result in trust overshoot towards bad nodes and trust undershoots towards good nodes. Dash lines show the empirical confidence intervals with 90% confidence. We can see that the

trust evaluation approaches ground truth as time increases (Lemma 1). Further, we observe that as the value of α increases the trust value converges to ground truth faster (Lemma 2), but the trust fluctuation also becomes higher (Lemma 3). The reason is that new direct observation can better reflect actual node status than past trust information. So using more new direct observation (higher α) in trust evaluation can help trust converge to the actual node status quickly. However, the trust value may fluctuate more since each direct observation may deviate from the actual node status due to false positives and false negatives.

To demonstrate the performance of our adaptive trust management protocol, we consider a dynamic environment in which a node initially is a good node, and then is compromised. Figure 9.4(b) shows the results of trust evaluation for *honesty* in this setting. When a good node is compromised, there may be a temporary overshoot of trust estimates since the trust management system requires time to adaptive to dynamic environmental changes. We can see that after a status change, the trust evaluation converges towards the new ground truth status. In addition, as the value of α increases, the trust evaluation converges to the new ground truth status faster, albeit with a higher fluctuation. The results correlate well Lemmas 1-3 in Appendix A.

9.3.2 Effect of β on Trust Evaluation

Next, we investigate the effect of design parameter β on trust evaluation. Recall that β is related to γ by Equation (9.2), both representing the weight associated with indirect recommendation with respect to past experience in Equation (9.1). We fix the value of α to 0.5 to isolate its effect and vary the value of β by selecting different values (0, 0.1, and 1).



(a) Static Environment

(b) Dynamic Environment

Figure 9.5: Effect of β on Honesty Trust Evaluation.

Figure 9.5(a) shows the effect of β on *honesty* trust evaluation in static environments. Again, we can see that our trust evaluation approaches ground truth status as time increases (Lemma 1). We also observe that as β increases, the trust evaluation converges to the ground truth faster (Lemma 2), but the trust fluctuation becomes higher (Lemma 3). The reason is that using more recommendations (higher β) helps trust convergence

through effective trust propagation. However, one can see that the effect of β is insignificant compared to the effect of α as long as $\beta > 0$. The reason is that very often in a social IoT environment with a large number of nodes, the chance of the trustor encountering a recommender is higher than the chance of the trustor directly interacting with a trustee. As long as $\beta > 0$, adaptive trust management is able to effectively aggregate trust using recommendations from a large number of recommenders, thus making the effect of further increasing the value of β insignificant.

Figure 9.5(b) shows the effect of β on *honesty* trust evaluation in dynamic environments. Again, we see that after the ground truth status changes, our trust protocol quickly converges towards the new ground truth status. Initially using recommendations ($\beta > 0$) in trust evaluation helps trust convergence. However, using recommendations does not contribute much to the trust convergence speed if the ground trust status changes dynamically. The reason behind this is that an honest recommender will adversely provide obsolete and inaccurate trust recommendation, if it has not interacted with the trustee since the trustee's status changes. As the trustor will not exclude these inaccurate recommendations from good recommenders, it hinders trust convergence.

9.3.3 Adaptive Trust Management in Response to Dynamically Changing Hostility Conditions

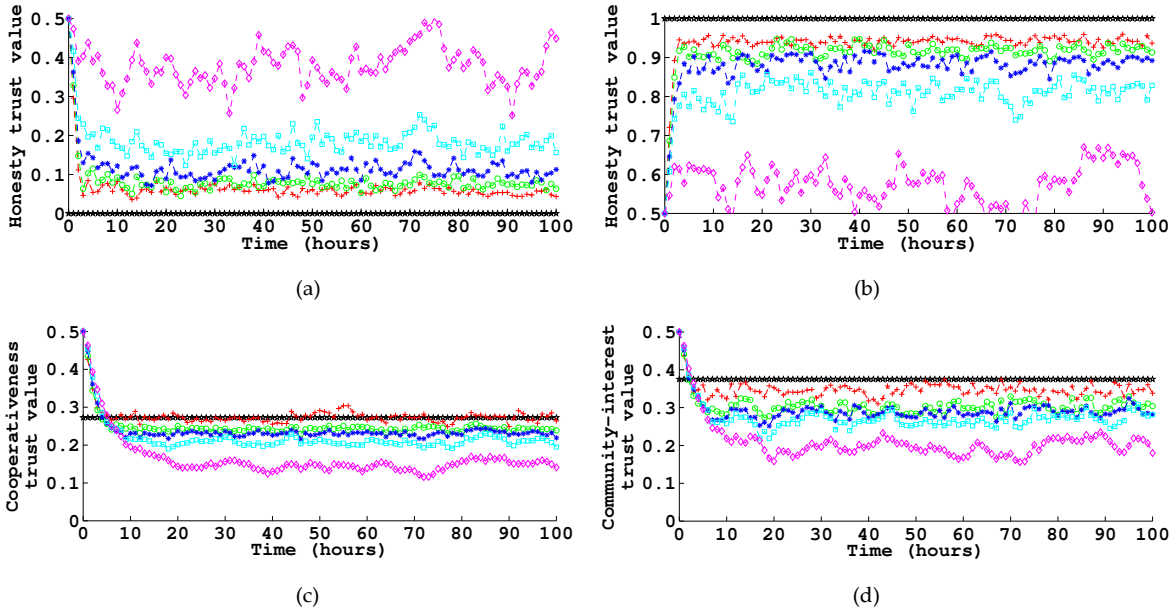


Figure 9.6: Effect of Hostility on Trust Evaluation.

From Figure 9.4 and Figure 9.5, one can see that the trust evaluation quickly converges and it is remarkably close to the ground truth status demonstrating its adaptability toward trust attacks. We further validate resiliency of our adaptive trust manage-

ment protocol toward trust attacks in IoT environments with a varying degree of hostility. We consider five different hostile environments with the percentage of malicious nodes λ being 10%, 30%, 50%, 70%, and 90%. The malicious nodes are randomly selected and perform good-mouthing and bad-mouthing attacks.

Figure 9.6 shows trust evaluation results for dishonesty (ground truth trust = 0), honesty (ground truth trust = 1), cooperativeness, and community-interest, respectively, in the 5 hostile environments. One can see that the trust evaluation quickly converges and it is remarkably close to the ground truth status when $\lambda \leq 50\%$ (under which the MAE is less than 10%), demonstrating high resiliency to trust attacks. As λ increases, the MAE of trust evaluation increases because of more false recommendations from malicious nodes. When $\lambda = 70\%$ and $\lambda = 90\%$, the MAE reaches 12% and 40%, respectively, as predicted by Lemma 4 in Appendix A. In Figure 9.6(c) and Figure 9.6(d), we observe that trust evaluation sometime overshoots, but undershoots in other cases. The reason is that we set initial trust value to 0.5 (ignorance) due to lack of knowledge. Further, as trust convergence, environment noises could lead to a temporary overshoot or undershoot of trust estimates.

Here we note that given knowledge of environment hostility (expressed in terms of the percentage of malicious nodes) possibly with help from an intrusion detection subsystem, our adaptive trust management protocol can react to changing hostility by dynamically choosing the best (α, β) values to tradeoff the trust convergence rate and trust fluctuation rate to obtain an acceptable MAE between the trust value obtained vs. ground truth.

9.4 Scalability Analysis on Trust Evaluation

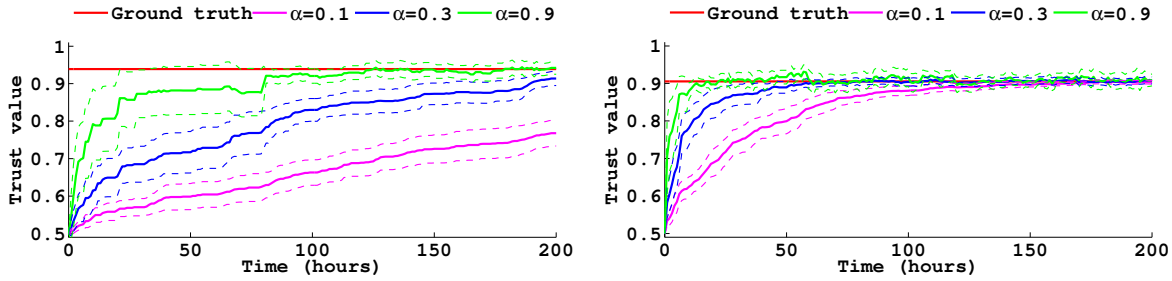
Table 9.2: Parameter Values for Scalability Analysis.

Param	Value	Param	Value	Param	Value
N_T	400	N_C	20	T	200hrs
N	40	α	[0, 1]	λ_p	~6 /pair /day
Ω	50%	β	[0, 1]	λ_c	~1 /pair /day
M	20	σ_c	0.05	σ_d	[0, 0.4]

In this section, we perform scalability analysis of our trust management protocol in a large scale IoT environment. Table 9.2 lists the default parameter values. We consider an IoT environment with N_T = 400 heterogeneous smart objects/devices. These devices are randomly distributed to N_C = 20 communities of interest. Given that trust is subjective (although it is controversial [127]) and nodes belonging to the same CoI have the

same community interests, the CoI ground truth trust status toward a trustee node is the same for two trustors in the same CoI, but may be different for two trustors in different communities. We use a standard deviation parameter σ_c (set to 0.05 in simulation) to reflect the difference. When two nodes interact with each other, their direct trust assessment based on direct observation may deviate from the CoI ground truth trust status, and the deviation is higher for more untrustworthy nodes. We use a standard deviation parameter σ_d (set in the range of [0, 0.4]) to model the difference. We randomly select P_M = 20% out of all devices as dishonest malicious nodes. A normal or good node follows the execution of our trust management protocol, while a dishonest node acts maliciously by providing false trust recommendations (Ballot stuffing, bad-mouthing, and self-promoting attacks) to disrupt trust management. The initial trust value of all devices is set to ignorance (0.5). We assume the encounter or interaction pattern follows the power-law distribution (with or without exponential cutoff) which is supported by the analysis of many real traces [103, 147, 190]. For two nodes in the same CoI, we consider that the inter-contact time follows a bounded power law distribution ([10mins, 2days]) with the slope equal to 1.4, resulting in the average interaction frequency about 6 times per day. For two nodes from two different communities, we consider that the inter-contact time follows a bounded power law distribution ([30mins, 7days]) with the slope equal to 1.2, resulting in the average interaction frequency about 1 per day. The settings here are close to those obtained from the real traces [103, 147, 190].

9.4.1 Inter-CoI vs. Intra-CoI Trust Evaluation



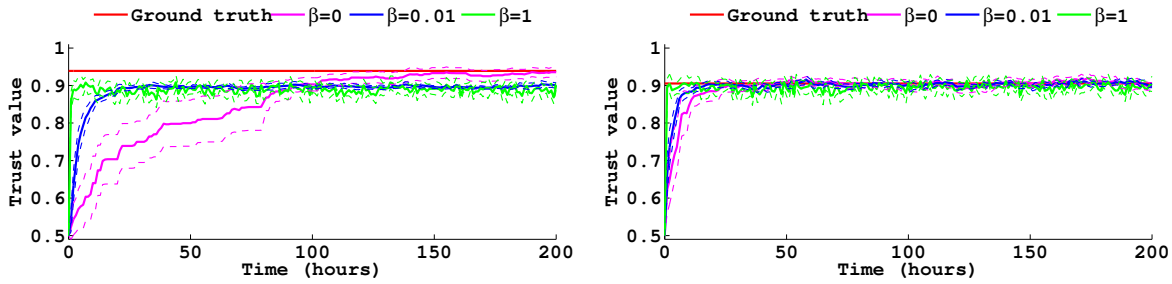
(a) Inter-CoI Trust Evaluation

(b) Intra-CoI Trust Evaluation

Figure 9.7: Effect of α on Inter-CoI and Intra-CoI Trust Evaluation.

Figure 9.7 shows the effect of trust parameter α on trust evaluation results for a trustor node and a trustee node randomly picked. We vary the value of α by choosing three values 0.1, 0.3, and 0.9, and fix the value of β to 0 to isolate its effect. The horizontal straight line on each figure indicates the actual trust value derived from CoI *ground truth*. The dash lines show the confidence interval at 95% confidence level. We observe that when the value of α increases the trust convergence time becomes shorter, but the trust evolution fluctuates more. The reason is that using more direct observations (a

higher α value) helps trust quickly converging to its CoI ground truth. However, each individual direct observation deviates from CoI ground truth status, which results in higher fluctuation. We also observe that the intra-CoI trust evaluation converges faster than the inter-CoI trust evaluation. This is expected because intra-CoI nodes interact more frequently and have more chances to directly observe each other.



(a) Inter-CoI Trust Evaluation

(b) Intra-CoI Trust Evaluation

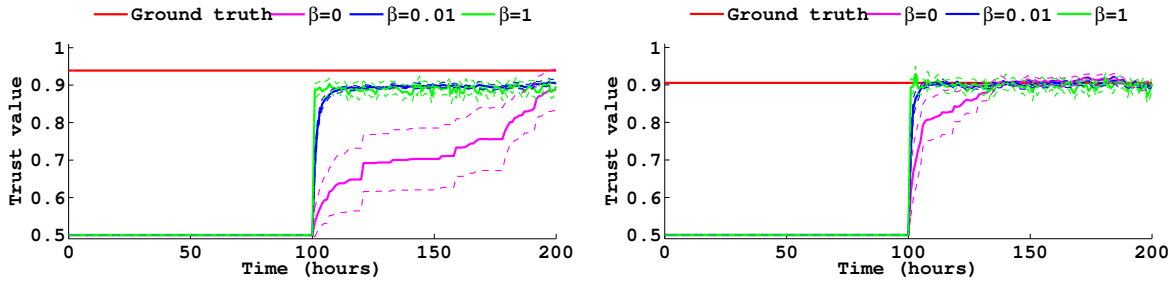
Figure 9.8: Effect of β on Inter-CoI and Intra-CoI Trust Evaluation.

Similarly, Figure 9.8 shows the effect of trust parameter β on trust evaluation results for a trustor node and a trustee node randomly picked. We vary the value of β by choosing three values 0, 0.01, and 1.0, and fix the value of α to 0.5 to isolate its effect. Again, we observe that intra-CoI trust converges faster than inter-CoI trust. In both cases, when using a higher β value, the convergence time becomes shorter, but trust fluctuation becomes higher because of the deviation of direct observation and false recommendations from attackers. Nevertheless, the mean absolute error (MAE) is less than 5% in the presence of 20% malicious nodes in the IoT system, demonstrating the resiliency of our protocol to survive from trust related attacks. Another important observation is that when using trust recommendations ($\beta > 0$), the MAE of inter-CoI trust is higher than the MAE of intra-CoI trust. The reason behind this is that trust is subjective and nodes from different communities of interest have different biased views toward CoI ground truth trust. Thus, using recommendations in trust evaluation may introduce bias if the recommender node is from a different CoI. However, the effect of trust bias can be reduced and even eliminated if the trustor and trustee are from the same CoI and interact frequently.

9.4.2 Trust Evaluation of Newly Join Nodes

Next, we consider a dynamic environment where a new node joins the IoT system. Certainly, if we do not consider any recommendations in trust evaluation, the trust evaluation of this newly join node towards others will behave the same as shown in Figure 9.7. Thus, if suffices to show the effect of β on the trust of the newly join node towards others in Figure 9.9. We can see that this newly join node very quickly builds up its trust towards both intra-CoI nodes and inter-CoI nodes in the IoT system. The reason is that

the IoT system has already reached convergence, and this newly join node can make use of such information through recommendations.



(a) Inter-CoI Trust Evaluation

(b) Intra-CoI Trust Evaluation

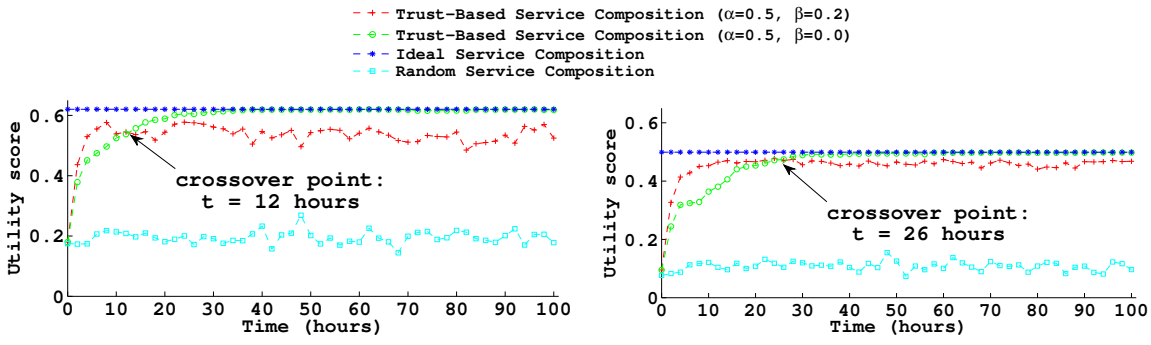
Figure 9.9: Effect of β on Trust of a Newly Join Node.

9.5 Trust-Based Service Composition

Finally, to demonstrate the effectiveness of adaptive trust management for IoT applications, we consider a trust-based service composition application in social IoT environments. The parameter settings are the same as in Table 9.1. In this application scenario, a node requests services (or information) from N_M service providers. The objective is to select the most trustworthy service providers such that the *utility* score representing the goodness of the service composition is maximized. Trust formation using the three trust components is application-specific.

We consider the trust formation design that if a selected service provider is malicious, the returning utility score is zero; otherwise, the returning utility score equals to the smaller one of the *cooperativeness* trust value and *community-interest* trust value the node has towards the service provider. In *trust-based service composition*, a node estimates the possible returning utility of each service provider based on its own knowledge and selects N_M service providers with the highest combined returning utility. The "actual" returning utility score is then computed based on actual status of the service providers selected. We compare the performance of trust-based service composition with two baseline approaches, *ideal service composition* which returns the maximum achievable utility score by selecting N_M service providers with the highest utility scores, and *random service composition* in which a node randomly selects N_M service providers without regard to trust.

Figure 9.10 compares trust-based service composition against two baseline service comparison methods in terms of the utility score. We consider two versions of trust-based service composition by selecting two different sets of design parameters: $(\alpha, \beta) = (0.5, 0.2)$ and $(\alpha, \beta) = (0.5, 0)$.



(a) λ = 10% (b) λ = 50%

Figure 9.10: Performance Comparison for Service Composition Application.

We see that as the percentage of malicious nodes increases, the utility score obtained by each protocol decreases because of fewer good service providers exist in the social IoT environment. For example, a service provider with the highest utility score when λ =10% might be compromised when λ =50%, making the combined utility score lower under *ideal service composition*. We also observe that trust-based service composition significantly outperforms random service composition and approaches the maximum achievable performance by ideal service composition. In addition, we see that there is a crossover point on the utility curves of two trust-based service composition methods. Before the crossover point, trust-based service composition under the setting of $(\alpha, \beta) = (0.5, 0.2)$ performs better, while after the crossover point, trust-based service composition under the setting of $(\alpha, \beta) = (0.5, 0)$ performs better. The reason is that while using recommendations helps trust quickly converge, it also introduces trust bias because of bad-mouthing and good-mouthing attacks. We observe that the crossover time point increases as the percentage of malicious nodes increases. Specifically, the crossover point is at t = 12 hours for λ = 10% and t = 26 hours for λ = 50%. Thus, in a dynamic IoT environment in which the hostility (in terms of the percentage of malicious nodes) changes over time, adaptive trust management is achieved by choosing the best design parameter settings (α, β) to maximize the service composition application performance.

9.6 Summary

In this chapter, we designed and analyzed a scalable, adaptive and survivable trust management protocol for a community of interest based dynamic social IoT. The trust management protocol takes dynamically changing social relationships into account. We advocate the use of three trust properties, namely, *honesty*, *cooperativeness*, and *community-interest* to evaluate social trust in social IoT environments. The protocol is distributed and each node only updates trust towards others of its interest upon encounter or interaction events. The trust assessment is updated by both direct observations and indirect

recommendations, with parameters α and β being the respective design parameters to control trust propagation for these two sources of information to improve trust assessment accuracy in response to dynamically changing conditions.

We analyzed the effect of trust parameters (α and β) on the convergence, accuracy, and resiliency properties of our adaptive trust management protocol, and validated our analysis using simulation. The results demonstrate that (1) the trust evaluation of adaptive trust management converges to the ground truth status in social IoT environments, (2) one can tradeoff trust convergence speed for low trust fluctuation, (3) adaptive trust management is resilient to misbehaving attacks, and (4) intra-CoI trust converges faster than inter-CoI under the same parameter setting. Dynamic adaptability is achieved by selecting the best trust parameter setting in response to changes to communities of interest. We also analyzed our trust protocol performance for a newly join node to the network. Making use of existing trust information in the network, a newly join node can quickly build up its trust relationship with desirable convergence and accuracy behavior. Finally, for scalability we proposed a storage management strategy to effectively utilize limited storage space in IoT devices. The results show that using the proposed method, our trust protocol with limited storage space achieves a similar performance level as that with ideal (unlimited) storage space and can perform better in the trust convergence time. We demonstrated the effectiveness of adaptive trust management by a service composition application in IoT environments. The results showed that trust-based service composition outperforms random service composition and approaches the maximum achievable performance from ground truth. We attributed this to the ability of trust-based trust service composition being able to dynamically choose the best design parameter settings in response to increasing hostility over time.

Chapter 10

Applicability and Implementation

In this chapter we address the complexity and practicality issues of the computational procedure to be executed by nodes in the system for dynamic trust management. Our design addresses the applicability and implementation issues of dynamic trust management in mobile environments where both computational power and space are limited. To tackle the computational complexity issue, we discuss two ways, namely, *design time computation* and *runtime computation*, to determine the optimal trust parameter setting. To tackle the limited storage issue, we develop a method for each node to selectively save trust results while still fulfilling application requirements. Finally, in mobile network environments such as IoT systems, nodes have pervasive connections. Low-end devices can delegate time/space-consuming tasks to designated high-end devices.

10.1 Computation of Optimal Trust Parameter Settings

10.1.1 Design Time Computation

For the *design time computation* method, the system designer needs to enumerate all possible network environment settings, pre-compute the optimal parameter values, and save these values into a table. An example is presented in Section 8.4 in which we determine the optimal α and β values for minimizing trust bias, given an environment condition as input. At run-time, each node simply performs a table lookup to retrieve the optimal parameters when the network environment changes. The advantage of this method is that it has a minimum computational requirement to mobile devices. However, it is difficult to predict all possible network conditions at design time. A mobile node has to extrapolate the optimal parameter setting if the run-time network environment does not exactly match the values stored in the table. Another disadvantage is that it requires high storage capability in mobile devices if the table is large in order to cover enough possible network states.

10.1.2 Runtime Computation

In this section, we investigate the applicability of applying heuristic search methods to determine optimal trust parameter settings at runtime. An example is presented in Section 9.2.4 in which the determination of the optimal α and β values for minimizing trust bias is formulated as an optimization problem in Equation (9.3). We consider two well-known nonlinear numerical optimization methods: *downhill simplex* [143] and *nonlinear conjugate gradient* [83]. Downhill simplex is a non-gradient algorithm. For an optimization problem with N input parameters (N-dimensional), the downhill simplex algorithm forms $N + 1$ vertices (i.e., simplex) on the N-dimensional space. A search method is performed iteratively until a convergence condition is satisfied. A new point is generated by extrapolating the objective function value at each point on the simplex and replacing one point on the simplex with this newly generated test point. Nonlinear conjugate gradient uses the gradient of objective function to find the local minimum. It works well if the objective function is approximately quadratic near the minimum.

We apply these two algorithms to solving the problem of finding optimal α and β values that minimize trust bias. The input of this optimization problem includes α and β. The objective function is to minimize the mean square error of subjective trust against objective trust. Note that since there is no analytical form for the objective function, we use numerical methods to approximate its gradient for the nonlinear conjugate gradient algorithm. We compare the results with the baseline method, brute force search.

Table 10.1: Performance Comparison of Algorithms Computing Optimal α and β.

Algorithm	Total Time (s)	Time Per Node (s)	Optimal Solution (α, β)	MSE
Brute Force Search	2886.866000	3.207629	(0.850000, 2.000000)	0.00223391503
Downhill Simplex	219.282000	0.243647	(0.837891, 2.033594)	0.00223388383
Nonlinear Conjugate Gradient	1044.933000	1.161037	(0.831848, 2.030059)	0.00223391014

Table 10.1 gives the results of computing the optimal α and β values of the intimacy trust property for 900 nodes in the network. The results of other trust properties follow the same trend. We first observe that the optimal mean square error (MSE) values generated by three algorithms are low and very close, although there are differences in optimal settings (i.e., α and β values). We also note that downhill simplex costs the least time and brute force search needs the most computation time. Note that in the brute force search method, we set the step size to 0.05. One can set a smaller step size in order to improve the results, but the computation time will also increase quadratically. The reason that downhill simplex performs better than nonlinear conjugate gradient might be that our objective function to minimize trust bias is not quadratic and has many local minimums due to environment noises. Overall, downhill simplex yield the best results and the computation time for each node is less than one second.

10.2 Storage Management

Space requirement is another challenge to trust management in mobile networks. A large-scale mobile network could have tens of thousands of nodes. However, both memory and secondary storage in most mobile devices are very limited. Storing and processing trust-related information of all nodes in the network are inefficient if not impossible. Each node has to selectively store and process trust information towards its peers while still fulfilling application requirements. In this section, we design and validate a storage management strategy to deal with this issue.

The policy of selecting which node's trust information to store may vary depending on application requirements and each node's interest. In general, nodes are more interested in others with higher trust values. However, simply saving the trust values towards the most trustworthy nodes cannot make the trust evaluation process converge and is not adaptive to dynamic environments since there is little chance to accumulate trust towards newly joining nodes. Our storage management strategy considers nodes with the highest trust values and recent interacting nodes as these nodes are most likely to share common interests.

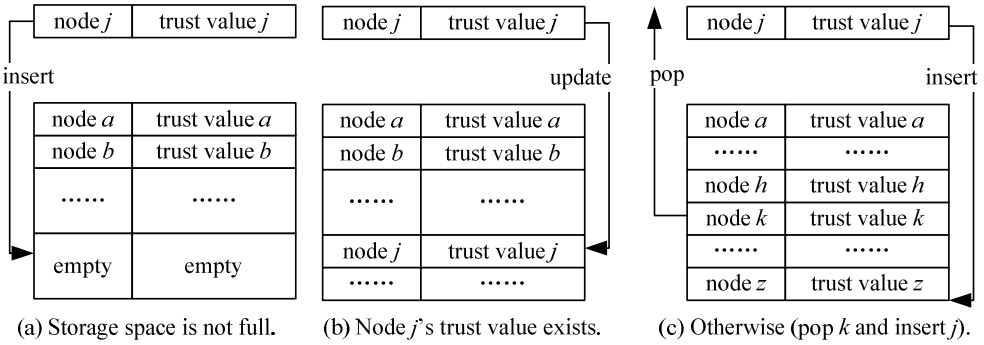


Figure 10.1: Storage Management Strategy.

Figure 10.1 illustrates how our approach works conceptualizing the storage size of each node as n (meaning that there is space to save trust values of up to n nodes). When a slot is needed, for a node's trust value to be kept it must be in the top Ω of the n trust values, or this node is one of the most recent interacting nodes. We consider $\Omega = 50\%$ here and the selection of optimal Ω value in dynamic mobile networks can be solved using the same adaptive control as for other trust parameters.

When node i obtains the trust value towards node j:

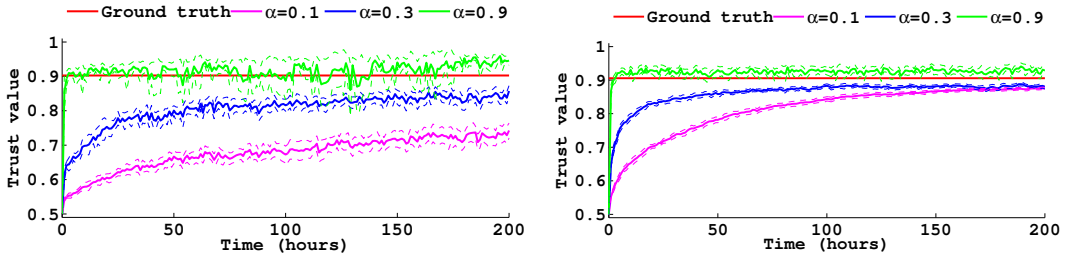
(1) If the storage space is not full or node i already has the trust information of node j in its storage space, node i will simply save the trust value towards node j.

(2) If the storage space is full and node i does not have the trust information of node j in its storage space, node i will put the trust value towards node j and pop out the trust value towards the earliest interacting node among those with trust values below the median (Ω = 50%).

The operations described above can be finished in $O(1)$ time on average by using the max-min-median heap. In addition, when two nodes interact with each other, they will only exchange the trust values kept as recommendations. This strategy can be applied to other information (e.g., direct feedbacks in user's profiles).

Validation of Scalable Storage Management in IoT Systems

We implement our design for scalable storage management in IoT systems and perform simulation validation. The network environment and parameter settings are the same as in Section 9.4 except that each node only has limited storage space to keep the trust values of up to 10% of the nodes in the system (with n = 40).



(a) Inter-CoI Trust Evaluation

(b) Intra-CoI Trust Evaluation

Figure 10.2: Effect of α on Trust (Limited Storage).

Figure 10.2 illustrates the effect of α on inter-CoI trust evaluation and intra-CoI trust evaluation for a trustee node randomly picked. We first observe a similar trend exhibited as the one in Figure 9.7 where unlimited storage is assumed, demonstrating the effectiveness of our storage management strategy. Trust fluctuation is higher especially for inter-CoI trust evaluation. The reason is that the trust value towards the trustee node may be dropped by some trustor nodes due to limited space and imperfect direct observation. We also observe that both inter-CoI trust evaluation and inter-CoI trust evaluation overestimate when α is high (e.g. α = 0.9), as a result of our storage management strategy preferring to keep nodes with high trust values. However, this overestimation is system wide, meaning that the overestimation happens on every node towards all other nodes. Thus, it does not do much harm to the usage of trust. Another byproduct of this preference is fast trust convergence. Comparing Figure 10.2 with Figure 9.7, we can see significantly improvement on trust convergence for higher α value (e.g. α = 0.9). The reason behind this is that our trust management strategy works like a filter excluding highly deviated recommendations coming from untrustworthy nodes to shield the system from false recommendation attacks.

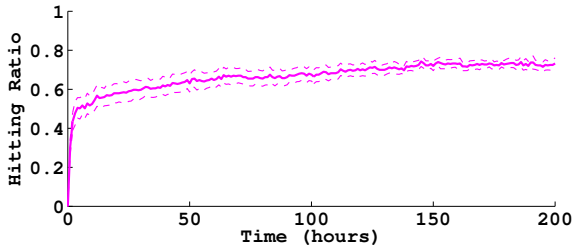


Figure 10.3: Hit Ratio.

Lastly, we demonstrate the effectiveness of the storage management strategy using the hit ratio. The top-*m* hit ratio means the percentage of the top-*m* most trustworthy nodes (with highest CoI ground truth trust) having their trust values stored in the limited *n* slots. Figure 10.3 shows the top-20 hit ratio as a function of time for a randomly selected node. We can see that initially the hit ratio is zero because there is no trust information towards others. As the trust converges, the hit ratio quickly increases and approaches 80%. This demonstrates the effectiveness and high space utilization of our storage management strategy.

10.3 Trust Delegation

In this section, we propose another solution to address the applicability issue of our dynamic trust management in large-scale mobile networks. The basic idea is that low-end devices can delegate time/space-consuming tasks to high-end devices. This is useful in user-centric network environments.

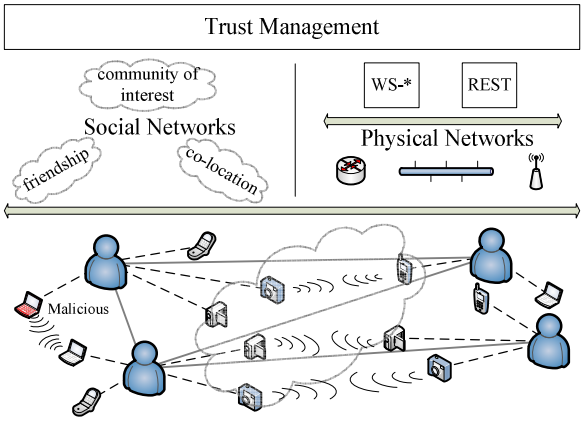


Figure 10.4: User-Centric Mobile Networks.

Figure 10.4 illustrates a user-centric mobile network where nodes are physical connected via communication networks and socially connected via users' social networks. Each user could have multiple mobile devices. At least one of these mobile devices of each user is a high-end device (i.e., a smart phone and laptop) which can be used to

store and process trust-related information and the user's profile (Figure 10.5). Other devices of the same user have the privilege to access the information. By delegating the storage and computation of social networks to a high-end device for each user, many low-end devices (i.e., sensors) are able to share and utilize the same social information in order to maximize its performance.

The designated high-end device stores each user's profile (see Figure 10.5), including trust information and other information, like, social connections, locations, direct interaction experiences, etc., which can be used for trust evaluation. In addition, this designated device is also responsible for trust computation: (1) updating trust values, (2) accepting requests from other low-end devices and providing responses, and (3) computing the optimal trust parameter setting in response to network environment changes. In contrast, each low-end device requests the trust evaluation result from this designated high-end device before interacting with other devices, and provides feedback of the direct interaction to the designated device after the interaction is completed.

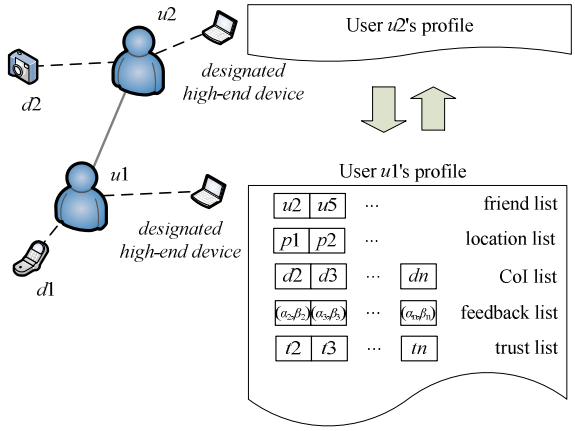


Figure 10.5: User Profile.

Trust is evaluated based on both direct feedbacks of past interaction experiences and recommendations from others. The designated high-end device receives direct feedbacks from other low-end devices of the same user, and exchanges trust recommendations with the designated devices of other users. When providing recommendations, the designated devices of two users agree on a session key and use a hash function with the session key to hide the identities of users and devices in their profiles, in order the preserve privacy to uncommon friends and devices. Then they exchange their profiles and update trust information accordingly. There are two ways to trigger trust recommendation/profile exchange: *event trigger* and *on-demand*. In the *event trigger* design, whenever two devices have an interaction, the users of the two devices will exchange their profiles in their designated devices. In the *on-demand* design, a user can send recommendation requests to its friends in order to update trust upon application requests.

Chapter 11

Conclusion

11.1 Publications

The dissertation work has resulted in three journal publications, seven conference publications, and three journal/conference submissions given below. At the end of each paper publication, we give the reference number of the publication and annotate the dissertation chapter in which part of the paper is included.

Journal Publications:

1. F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, 2012, pp. 169-183. [21] (Chapter 8.)
2. I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Integrated Social and QoS Trust-based Routing in Delay Tolerant Networks," *Wireless Personal Communications*, vol. 66, no. 2, 2012, pp. 443-459. [45] (Chapter 7.)
3. I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, accepted to appear, 2013. [46, 47] (Chapter 7.)

Conference Publications:

4. I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust Management for Encounter-Based Routing in Delay Tolerant Networks," *IEEE Global Communications Conference*, Miami, Florida, USA, Dec. 2010, pp. 1-6. [44] (Chapter 7.)
5. F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing," *26th*

ACM Symposium on Applied Computing, TaiChung, Taiwan, March 2011, pp. 1732-1738. [19] (Chapter 8.)

6. F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Trust-Based Intrusion Detection in Wireless Sensor Networks," *IEEE International Conference on Communications*, Kyoto, Japan, June 2011, pp. 1-6. [20] (Chapter 8.)
7. M. Chang, I.R. Chen, F. Bao, and J.H. Cho, "Trust-Threshold Based Routing in Delay Tolerant Networks," *IFIP WG 11.11 International Conference on Trust Management*, Copenhagen, Denmark, June 2011, pp. 265-276. [41] (Chapter 7.)
8. F. Bao and I.R. Chen, "Trust Management for the Internet of Things and Its Application to Service Composition," *IEEE WoWMoM Workshop on the Internet of Things: Smart Objects and Services*, San Francisco, CA, USA, June 2012, pp. 1-6. [18] (Chapter 9.)
9. F. Bao and I.R. Chen, "Dynamic Trust Management for the Internet of Things Applications," *International Workshop on Self-Aware Internet of Things*, San Jose, CA, USA, Sept. 2012, pp. 1-6. [17] (Chapter 9.)
10. F. Bao, I.R. Chen, and J. Guo, "Scalable, Adaptive and Survivable Trust Management for Community of Interest Based Internet of Things Systems," *11th International Symposium on Autonomous Decentralized System*, Mexico City, Mexico, March 2013. [22] (Chapter 9.)

<u>Papers Submitted:</u>

1. F. Bao and I.R. Chen, "Adaptive Trust Management for Social Internet of Things," submitted to *IEEE Transactions on Network and Service Management*, Nov. 2012. (Chapter 9.)
2. I.R. Chen, F. Bao, J. Guo, and J.H. Cho, "Integrated Social and QoS Trust Management of Mobile Groups in Ad Hoc Networks," submitted to *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Feb. 2013. (Chapter 6.)
3. F. Bao, J. Guo, and I.R. Chen, "Adaptive Trust Management of Service Composition in User-Centric Internet of Things Systems," submitted to *2013 IEEE Global Communications Conference*, March 2013. (Chapter 9.)

11.2 Summary and Future Work

Based on the design and validation principles of dynamic trust management, we have addressed trust composition by exploring both social trust and QoS trust metrics, and proposed trust aggregation and trust propagation protocols for MANETs, DTNs, WSNs

and IoT systems. We have identified the best trust formation setting for dynamic mobile network environments and applied application-level performance optimization technique for managing mobile groups in MANETs (Chapter 6), encounter-based secure routing in DTNs (Chapter 7), trust-based geographic routing and trust-based intrusion detection in WSNs (Chapter 8), and trust-based service composition in IoT systems (Chapter 9). We investigated the applicability and implementation issues of our dynamic trust management protocol, and proposed and validated designs to address computational time cost and space requirements (Chapter 10). We formally proved the convergence, accuracy, and resiliency properties of our dynamic trust management protocol for IoT systems. We validated our dynamic trust management protocols using the static-followed-by-dynamic testing strategy for MANETs, DTNs and WSNs. Further, through model-based analysis with simulation validation, we have demonstrated that our dynamic trust management protocols outperform existing trust-based protocols such as Bayesian trust and are resilient to trust related attacks.

Our research work on multi-dimensional social trust properties, including *honesty*, *selfishness*, *connectivity* and *community of interest similarity*, gains insight of the effects of these social trust properties on protocol performance, which can be further applied to trust, privacy, security, and risk management in mobile social networks. Second, our design principles in the dissertation research provide a modularized framework for trust management. This includes trust composition, trust aggregation and propagation, trust formation, and application-level optimization. Finally, the dissertation research can be applied to future generation trust-based mobile applications, such as consumer-based social mobile applications, disaster recovery, surveillance management, and trust and risk management in military applications.

There are several future research directions that can be extended further from this dissertation research:

- **Trust Delegation for User-Centric Mobile Networks**. The dynamic trust management framework proposed in this dissertation research is distributed. Compared with a centralized approach, it does not require a trusted third party or a full end-to-end connection in mobile networks, and can significantly reduce network communication cost. However, it requires each mobile node to have a certain level of computational power and storage capability, which might be difficult to satisfy in certain network environments such as WSNs and IoT systems. Trust delegation provides another alternative. Most low-end devices with little computational power and storage space can delegate the time/space-consuming tasks to designated high-end devices. This is especially useful in user-centric network environments in which each user has multiple mobile devices including at least one high-end device (e.g., a lap-

top or smartphone). Our preliminary work reported in Section 10.3 can be further extended to demonstrate its wide applicability to mobile networks.

- **Trust Management for Online Social Networks**. A future research area is to design and validate a trust management protocol for online social networks considering social communities. Today's social network platforms such as Facebook, Twitter, LinkedIn, and Google+ provide a great amount of social information. How to process this large amount of noisy social information is challenging. A variety of trust properties can be incorporated in order to improve the quality of social information in these social network platforms. For example, on the LinkedIn professional social network, people can provide recommendation for others and endorse skills or expertise of others. A multi-dimensional trust system extended from this dissertation research can be developed for online social networks to assess not only the trustworthiness of recommendations and endorsements, but also the trustworthiness of information sources. Validation of trust management protocol design can benefit from real data published by social networking companies.
- **Trust-Based Service Management**. A future research area is to explore trust-based service management applications with which we could further demonstrate the utility of our dynamic trust management protocol design. In Chapter 9, we have demonstrated the application of our dynamic trust management to service composition. This work can be extended to the condition where the composed service is constrained by workflow. In service computing, service composition can be classified as static, semi-automatic, and automatic, while service composition can be performed based on workflow and AI planning [157]. Dynamic service composition could be a complex planning problem. A future direction is to apply trust management protocols developed in the dissertation research to a template-based semi-automatic service composition application. Here, a template (based on workflow or AI) describes the data flow and logic of a composite web service. Another direction is to apply our dynamic trust management design to other components of service management, including service adverting/publishing, service subscription, location service management [50, 51, 81, 116-118], and service discovery.
- **Trust-Enhanced Secure Routing and Intrusion Detection in Wireless Mobile Networks**. In our dissertation research, we demonstrated the effectiveness of dynamic trust management when applying to secure routing and intrusion detection applications in WSNs and DTNs. One future research direction is to extend this line of research with the consideration of fuzzy failure criteria [24, 48, 49] to other wireless mobile networks such as MANETs, IoT and cyber physical systems. Another direction is to consider other applications in wireless mobile networks that can benefit from the design concept of dynamic, adaptive trust management developed in the

dissertation research. This includes, but not limited to, (a) trust-based admission control strategies as in [52, 55, 57, 58, 181, 189] used by selfish nodes to maximize their own payoffs while contributing to routing performance, (b) trust-based detection to assess trustworthiness of neighbor nodes so as to tackle the "what paths to use" problem in multipath routing decision making for intrusion tolerance [5-7], and (c) trust-enhanced intrusion detection for more complex node capture and insider attack behaviors as considered in [60, 133-135].

- **Nonlinear Trust Formation**. In our dissertation research, we considered a liner form for trust formation in Chapter 6, Chapter 7, and Chapter 8. We considered the linear form because it is simple, easy to implement, and effective. However, it might not be the optimal solution for trust formation in some cases. One future research direction is to explore other forms of trust formation, including the product form, if-then-else form, and trustee-based trust formation. In the product form, the overall trust is obtained as the product of individual trust properties. The product form of trust formation is valid if the probability of the target node taking an action relies on all trust properties, and individual trust properties are independent. The if-then-else form can be applied to applications in which certain trust properties are critical. For example, in Chapter 9, we consider *honesty* as a critical trust property. If the *honesty* trust value of a node is below the threshold, then its overall trust will be zero; otherwise, its overall trust is the minimum of the other two trust properties. In addition, we can consider trustee-based trust formation with which trust formation depends on characteristics of trustees. For example, some social trust properties may not be applicable to entities without social networking. In WSNs, *honesty* might be the most important trust property for a CH, so a minimum trust threshold in addition to a high weight is needed. It depends on how individual trust properties impact the actual node's behaviors and application performance.

Bibliography

[1] "The ns-3 Network Simulator," Nov. 2011, http://www.nsnam.org/.

[2] W. J. Adams, and N. J. Davis, "Toward a Decentralized Trust-based Access Control System for Dynamic Collaboration," *IEEE SMC Information Assurance Workshop*, West Point, NY, June 2005, pp. 317-324.

[3] E. Aivaloglou, and S. Gritzalis, "Trust–Based Data Disclosure in Sensor Networks," *IEEE International Conference on Communications*, 2009, pp. 1-6.

[4] E. Aivaloglou, and S. Gritzalis, "Hybrid Trust and Reputation Management for Sensor Networks," *Wireless Networks*, vol. 16, no. 5, July 2010, pp. 1493-1510.

[5] H. Al-Hamadi, and I. R. Chen, "Energy vs. QoS Tradeoff Analysis of Multipath Routing Protocols for Intrusion Tolerance in Heterogeneous Wireless Sensor Networks," *10th IEEE International Symposium on Parallel and Distributed Processing with Applications*, Madrid, Spain, July 2012, pp. 387-394.

[6] H. Al-Hamadi, and I. R. Chen, "Dynamic Multisource Multipath Routing for Intrusion Tolerance and Lifetime Maximization of Autonomous Wireless Sensor Networks," *IEEE 11th Symposium on Decentralized Autonomous Systems*, Mexico City, March 2013.

[7] H. Al-Hamadi, and I. R. Chen, "Redundancy Management of Multipath Routing for Intrusion Tolerance in Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Network and Service Management*, 2013.

[8] J. N. Al-Karaki, and A. E. Kamal, "Routing Rechniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, Dec. 2004, pp. 6-28.

[9] M. Aldebert, M. Ivaldi, and C. Roucolle, "Telecommunications Demand and Pricing Structure: an Economic Analysis," *Telecommunication Systems*, vol. 25, no. 1-2, Jan. 2004, pp. 89-115.

[10] D. Artz, and Y. Gil, "A Survey of Trust in Computer Science and the Semantic Web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, 2007, pp. 58-71.

[11] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks,* vol. 54, no. 15, Oct. 2010, pp. 2787-2805.

[12] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," *IEEE Communication Letters,* vol. 15, no. 11, Nov. 2011, pp. 1193-1195.

[13] E. Ayday, H. Lee, and F. Fekri, "Trust Management and Adversary Detection for Delay Tolerant Networks," *Military Communications Conference*, 2010, pp. 1788-1793.

[14] E. Ayday, H. Lee, and F. Fekri, "An Iterative Algorithm for Trust Management and Adversary Detection for Delay Tolerant Networks," *IEEE Transactions on Mobile Computing,* vol. 11, no. 9, Sept. 2012, pp. 1514-1531.

[15] A. Baier, "Trust and Antitrust," *Ethics,* vol. 96, no. 2, Jan. 1986, pp. 231-260.

[16] V. Balakrishnan, V. Varadharajan, U. K. Tupakula, and P. Lucs, "Trust and Recommendations in Mobile Ad Hoc Networks," *International Conference on Networking and Services*, Athens, Greece, June 2007, pp. 64-69.

[17] F. Bao, and I. R. Chen, "Dynamic Trust Management for the Internet of Things Applications," *International Workshop on Self-Aware Internet of Things*, San Jose, CA, USA, Sept. 2012, pp. 1-6.

[18] F. Bao, and I. R. Chen, "Trust Management for the Internet of Things and Its Application to Service Composition," *IEEE WoWMoM Workshop on the Internet of Things: Smart Objects and Services*, San Francisco, CA, USA, Juen 2012, pp. 1-6.

[19] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing," *ACM Symposium on Applied Computing*, TaiChung, Taiwan, March 2011, pp. 1732-1738.

[20] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Trust-Based Intrusion Detection in Wireless Sensor Networks," *IEEE International Conference on Communications*, Kyoto, Japan, June 2011, pp. 1-6.

[21] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management,* vol. 9, no. 2, 2012, pp. 161-183.

[22] F. Bao, I. R. Chen, and J. Guo, "Scalable, Adaptive and Survivable Trust Management for Community of Interest Based Internet of Things Systems," *11th International Symposium on Autonomous Decentralized System*, Mexico City, Mexico, March 2013.

[23] J. S. Baras, and T. Jiang, "Cooperative Games, Phase Transitions on Graphs and Distributed Trust in MANET," *IEEE Conference on Decision and Control*, Atlantis, Bahamas, Dec. 2004, pp. 93-98.

[24] F. B. Bastani, I. R. Chen, and T. Tsao, "Reliability of Systems with Fuzzy-Failure Criterion," *Annual Reliability and Maintainability Symposium*, 1994.

[25] V. Bhuse, and A. Gupta, "Anomaly Intrusion Detection in Sireless Sensor Network," *Journal of High Speed Networks,* vol. 15, no. 1, Jan. 2006, pp. 33-51.

[26] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, "Exploiting Self-Reported Social Networks for Routing in Ubiquitous Computing Environments," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication*, Avignon, France, October 2008, pp. 484-489.

[27] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management," *IEEE Symposium on Security and Privacy*, May 1996, pp. 164-173.

[28] A. Boukerche, and Y. Ren, "A Security Management Scheme using a Novel Computational Reputation Model for Wireless and Mobile Ad Hoc Networks," *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Vancouver, British Columbia, Canada, 2008, pp. 88-95.

[29] S. Braynov, and T. Sandholm, "Contracting with Uncertain Level of Trust," *Computational Intelligence,* vol. 18, no. 4, 2002, pp. 501-514.

[30] S. Buchegger, and J.-Y. L. Boudec, "Performance Analysis of the Confidant Protocol: Cooperation of Nodes - Fairness in Dynamic Ad-Hoc Networks," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2002, pp. 226-236.

[31] C. Budianu, S. Ben-David, and L. Tong, "Estimation of the Number of Operating Sensors in Large-Scale Sensor Networks with Mobile Access," *IEEE Trans. on Signal Processing,* vol. 54, no. 5, May 2006, pp. 1703-1715.

[32] N. Bui, and M. Zorzi, "Health Care Applications: A Solution Based on The Internet of Things," *the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Barcelona, Spain, Oct. 2011, pp. 1-5.

[33] E. Bulut, Z. Wang, and B. K. Szymanski, "Impact of Social Networks on Delay Tolerant Routing," *IEEE Global Telecommunications Conference*, Honolulu, HI, Nov. 2009, pp. 1-6.

[34] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for Vehicle-Based Disruption-Tolerant Networking," *IEEE Conference on Computer Communications*, Barcelona, Spain, Apr. 2006, pp. 1-11.

[35] L. Capra, and M. Musolesi, "Autonomic Trust Prediction for Pervasive Systems," *International Conference on Advanced Information Networking and Applications*, April 2006, pp. 1-5.

[36] J. Carbo, J. M. Molina, and J. Davila, "Trust Management Through Fuzzy Reputation," *International Journal of Cooperative Information Systems*, vol. 12, no. 1, 2003, pp. 135-155.

[37] C. Castelfranchi, and R. Falcone, "Trust is Much More Than Subjective Probability: Mental Components and Sources of Trust," *Hawaii International Conference on System Sciences*, Jan. 2000, pp. 1-10.

[38] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," *RFC 4838*, IETF, 2007.

[39] A. Chaintreau, P. Hui, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Impact of Human Mobility on Opportunistic Forwarding Algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, July 2007, pp. 606-620.

[40] B. J. Chang, and S. L. Kuo, "Markov Chain Trust Model for Trust Value Analysis and Key Management in Distributed Multicast MANETs," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, May 2009, pp. 1846-1863.

[41] M. Chang, I. R. Chen, F. Bao, and J. H. Cho, "Trust-Threshold Based Routing in Delay Tolerant Networks," *5th IFIP International Conference on Trust Management*, Copenhagen, Denmark, June 2011, pp. 265-276.

[42] C. Chen, and S. Helal, "A Device-Centric Approach to a Safer Internet of Things," *the 2011 International Workshop on Networking and Object Memories for the Internet of Things*, Beijing, China, Sep. 2011, pp. 1-6.

[43] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things," *Computer Science and Information Systems*, vol. 8, no. 4, Oct. 2011, pp. 1207-1228.

[44] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Trust Management for Encounter-Based Routing in Delay Tolerant Networks," *IEEE Global Communications Conference*, Miami, Florida, USA, Dec. 2010, pp. 1-6.

[45] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Integrated Social and QoS Trust-based Routing in Delay Tolerant Networks," *Wireless Personal Communications*, vol. 66, no. 2, 2012, pp. 443-459.

[46] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[47] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Supplemental Material for 'Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing'," *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[48] I. R. Chen, and F. B. Bastani, "Effect of Articial-Intelligence Planning Procedures on System Reliability," *IEEE Transactions on Reliability*, vol. 40, no. 3, 1991, pp. 364-369.

[49] I. R. Chen, F. B. Bastani, and T. W. Tsao, "On the Reliability of AI Planning Software in Real-Time Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 1, 1995, pp. 4-13.

[50] I. R. Chen, T. M. Chen, and C. Lee, "Performance Evaluation of Forwarding Strategies for Location Management in Mobile Networks," *The Computer Journal*, vol. 41, no. 4, 1998, pp. 243-253.

[51] I. R. Chen, T. M. Chen, and C. Lee, "Agent-Based Forwarding Strategies for Reducing Location Management Cost in Mobile Networks," *Mobile Networks and Applications*, vol. 6, no. 2, 2001, pp. 105-115.

[52] I. R. Chen, and T. H. Hsi, "Performance Analysis of Admission Control Algorithms Based on Reward Optimization for Real-Time Multimedia Servers," *Performance Evaluation*, vol. 33, no. 2, 1998, pp. 89-112.

[53] I. R. Chen, and D. C. Wang, "Analysis of Replicated Data with Repair Dependency," *The Computer Journal*, vol. 39, no. 9, 1996, pp. 767-779.

[54] I. R. Chen, and D. C. Wang, "Analyzing Dynamic Voting Using Petri Nets," *IEEE 15th Symposium on Reliable Distributed Systems*, 1996, pp. 44-53.

[55] I. R. Chen, O. Yilmaz, and I. Yen, "Admission Control Algorithms for Revenue Optimization with QoS Guarantees in Mobile Wireless Networks," *Wireless Personal Communications*, vol. 38, no. 3, 2006, pp. 357-376.

[56] T. Chen, F. Wu, and S. Zhong, "FITS: A Finite-Time Reputation System for Cooperation in Wireless Ad Hoc Networks," *IEEE Transactions on Computers*, vol. 60, no. 7, July 2011, pp. 1045-1056.

[57] S. T. Cheng, C. M. Chen, and I. R. Chen, "Dynamic Quota-Based Admission Control with Sub-Rating in Multimedia Servers," *Multimedia Systems*, vol. 8, no. 2, 2000, pp. 83-91.

[58] S. T. Cheng, C. M. Chen, and I. R. Chen, "Performance Evaluation of an Admission Control Algorithm: Dynamic Threshold with Negotiation," *Performance Evaluation*, vol. 52, no. 1, 2003, pp. 1-13.

[59] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks,* vol. 1, no. 1, 2003, pp. 12-64.

[60] J. H. Cho, I. R. Chen, and P. G. Feng, "Effect of Intrusion Detection on Reliability of Mission-Oriented Mobile Group Systems in Mobile Ad Hoc Networks," *IEEE Transactions on Reliability,* vol. 59, no. 1, 2010, pp. 231-241.

[61] J. H. Cho, A. Swami, and I. R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-Driven Group Communication Systems in Mobile Ad Hoc Networks," *IEEE/IFIP International Symposium on Trusted Computing and Communications*, Vancouver, BC, Canada, August 2009, pp. 641-650.

[62] J. H. Cho, A. Swami, and I. R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Communications Surveys & Tutorials,* vol. 13, no. 4, 2011, pp. 562-583.

[63] J. S. Coleman, *Foundations of Social Theory*, Cambridge, Mass.: Harvard University Press, 1990.

[64] C. Crepeau, C. R. Davis, and M. Maheswaran, "A Secure MANET Routing Protocol with Resilience against Byzantine Behaviours of Malicious or Selfish Nodes," *International Conference on Advanced Information Networking and Applications Workshops*, Niagara Falls, Ont., May 2007, pp. 19-26.

[65] E. M. Daly, and M. Haahr, "Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs," *IEEE Transactions on Mobile Computing,* vol. 8, no. 5, May 2009, pp. 606-621.

[66] E. M. Daly, and M. Haahr, "The Challenges of Disconnected Delay Tolerant MANETs," *Ad Hoc Networks,* vol. 8, no. 2, 2010, pp. 241-250.

[67] C. R. Davis, "A Localized Trust Management Scheme for Ad Hoc Networks," *International Conference on Networking*, 2004, pp. 671-675.

[68] M. K. Denko, T. Sun, and I. Woungang, "Trust Management in Ubiquitous Computing: A Bayesian Approach," *Computer Communications,* vol. 34, no. 3, 2011, pp. 398-406.

[69] M. Doering, T. Pögel, W.-B. Pöttner, and L. C. Wolf, "A New Mobility Trace for Realistic Large-Scale Simulation of Bus-based DTNs," *ACM MobiCom Workshop on Challenged Networks*, Chicago, USA, 2010, pp. 71-74.

[70] X. Du, and H.-H. Chen, "Security in Wireless Sensor Networks," *IEEE Wireless Communications,* vol. 15, no. 4, Aug. 2008, pp. 60-66.

[71] P. Ebinger, and N. Bißmeyer, "TEREC: Trust Evaluation and Reputation Exchange for Cooperative Intrusion Detection in MANETs," *Communication Networks and Services Research Conference*, May 2009, pp. 378-385.

[72] L. Eschenauer, V. D. Gligor, and J. Baras, "On Trust Establishment in Mobile Ad Hoc Networks," *International Security Protocols Workshop*, Cambridge, UK, Apr. 2002, pp. 47-66.

[73] T. E. Eugen Staab, "Tuning Evidence-Based Trust Models," *International Conference on Computational Science and Engineering*, Vancouver, Canada, August 2009, pp. 92-99.

[74] L. M. Feeney, "An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications,* vol. 16, 2001, pp. 239-249.

[75] D. Gambetta, "Can We Trust Trust?," *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, ed., Department of Sociology, University of Oxford, 2000, pp. 213-237.

[76] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *ACM Transactions on Sensor Networks,* vol. 4, no. 3, May 2008, pp. 1-37.

[77] S. C. Geyik, E. Bulut, and B. K. Szymanski, "PCFG Based Synthetic Mobility Trace Generation," *IEEE Conference on Global Communication*, Miami, FL, Dec. 2010, pp. 1-5.

[78] T. Ghosh, N. Pissinou, and K. Makki, "Towards Designing a Trust Routing Solution in Mobile Ad Hoc Net-works," *Mobile Networks and Applications,* vol. 10, no. 6, 2005, pp. 985-995.

[79] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A Survey on Facilities for Experimental Internet of Things Research," *IEEE Communications Magazine,* vol. 49, no. 11, Nov. 2011, pp. 58-67.

[80] J. Golbeck, "Computing with Trust: Definition, Properties, and Algorithms," *Securecomm Workshop Security and Privacy for Emerging Areas in Communications Networks*, Baltimore, MD, Aug. 2006, pp. 1-7.

[81] B. Gu, and I. R. Chen, "Performance Analysis of Location-Aware Mobile Service Proxies for Reducing Network Cost in Personal Communication Systems," *Mobile Networks and Applications,* vol. 10, no. 4, 2005, pp. 453-463.

[82] G. C. Hadjichristofi, W. J. Adams, and N. J. Davis, "A Framework for Key Management in a Mobile Ad Hoc Network," *International Conference on*

Information Technology: Coding and Computing, Tiejun Huang, China, April 2005, pp. 568-573.

[83] W. W. HAGER, and H. ZHANG, "A Survey of Nonlinear Conjugate Gradient Methods," *Pacific journal of Optimization*, vol. 2, no. 1, 2006, pp. 35-58.

[84] S. Haykin, *Adaptive Filter Theory*: Prentice Hall, 2002.

[85] H. Hu, Y. Chen, W.-S. Ku, Z. Su, and C.-H. J. Chen, "Weighted Trust Evaluation-Based Malicious Node Detection for Wireless Sensor Networks," *International Journal of Information and Computer Security*, vol. 3, no. 2, 2009, pp. 132-149.

[86] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole Attacks in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, 2006, pp. 370-380.

[87] E. Hyytiä, P. Lassila, and J. Virtamo, "Spatial Node Distribution of the Random Waypoint Mobility Model with Applications," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, June 2006, pp. 680-694.

[88] M. S. Islam, R. H. Khan, and D. M. Bappy, "A Hierarchical Intrusion Detection System in Wireless Sensor Networks," *Computer Science and Network Security*, vol. 10, no. 8, August 2010, pp. 21-26.

[89] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *ACM Computer Communication Review*, vol. 34, no. 4, Oct. 2004, pp. 145-158.

[90] A. J. Jara, M. A. Zamora, and A. F. G. Skarmeta, "An Internet of Things-Based Personal Device for Diabetes Therapy Management in Ambient Assisted Living (AAL)," *Personal and Ubiquitous Computing*, vol. 15, no. 4, 2011, pp. 431-440.

[91] J. J. Jaramillo, and R. Srikant, "DARWIN: Distributed and Adaptive Reputation mechanism for WIreless adhoc Networks," *ACM International Conference on Mobile Computing and Networking*, Montréal, Québec, Canada, 2007, pp. 87-97.

[92] D. B. Johnson, and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*: Kluwer Academic Publishers, 1996, pp. 153-181.

[93] D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Mobile Ad Hoc Network Working Group, IETF, 1999.

[94] K. Jones, "Trust as an Affective Attitude," *Ethics*, vol. 107, no. 1, Oct. 1996, pp. 4-25.

[95] A. Jøsang, and J. Haller, "Dirichlet Reputation Systems," *International Conference on Availability, Reliability and Security*, Vienna, April 2007, pp. 112-119.

[96] A. Jøsang, and R. Ismail, "The Beta Reputation System," *Bled Electronic Commerce Conference*, Bled, Slovenia, June 17-19 2002, pp. 1-14.

[97] A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, March 2007, pp. 618-644.

[98] A. Jøsang, and S. Pope, "Semantic Constraints for Trust Transitivity," *Asia-Pacific Conference on Conceptual Modelling*, 2005, pp. 59-68.

[99] P. Juang, H. Oki, and Y. Wang, "Energy Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," *International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, Oct. 2002, pp. 96-107.

[100] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, March 1960, pp. 35-45.

[101] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A Survey of Routing Attacks in Mobile Ad Hoc Networks," *IEEE Wireless Communications*, vol. 14, no. 5, Oct. 2007, pp. 85-91.

[102] J.-C. Kao, and R. Marculescu, "Minimizing Eavesdropping Risk by Transmission Power Control in Multihop Wireless Networks," *IEEE Transactions on Computers*, vol. 56, no. 8, Aug. 2007, pp. 1009-1023.

[103] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power Law and Exponential Decay of Intercontact Times between Mobile Devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 10, 2007, pp. 1377-1390.

[104] M. Karaliopoulos, "Assessing the Vulnerability of DTN Data Relaying Schemes to Node Selfishness," *IEEE Communications Letters*, vol. 13, no. 12, 2009, pp. 923-925.

[105] M. Khabbazian, H. Mercier, and V. K. Bhargava, "Severity Analysis and Countermeasure for the Wormhole Attack in Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, Feb. 2009, pp. 736-745.

[106] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," *USENIX/ACM Symposium on Networked System Design and Implementation*, Boston, USA, May 2005, pp. 217-230.

[107] A. Kini, and J. Choobineh, "Trust in Electronic Commerce: Definition and Theoretical Considerations," *Hawaii International Conference on System Sciences*, Kohala Coast, HI, Jan. 1998, pp. 51-61.

[108] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking," *7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Boston, MA, USA, June 2010.

[109] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," *RFC 2104*, IETF Network Working Group, 1997.

[110] S. Lakshmanan, C.-L. Tsao, R. Sivakumar, and K. Sundaresan, "Securing Wireless Data Networks against Eavesdropping Using Smart Antennas," *International Conference on Distributed Computing Systems*, 2008, pp. 19-27.

[111] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems,* vol. 4, no. 3, July 1982, pp. 382-401.

[112] F. Li, J. Wu, and A. Srinivasan, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," *IEEE Conference on Computer Communications*, 2009, pp. 2428-2436.

[113] H. Li, and M. Singhal, "Trust Management in Distributed Systems," *Computer*, vol. 40, no. 2, 2007, pp. 45 - 53.

[114] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," *IEEE Conference on Computer Communications*, San Diego, CA, March 2010, pp. 1-9.

[115] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart Community: An Internet of Things Application," *IEEE Communications Magazine,* vol. 49, no. 11, Nov. 2011, pp. 68-75.

[116] Y. Li, and I. R. Chen, "Adaptive Per-User Per-Object Cache Consistency Management for Mobile Data Access in Wireless Mesh Networks," *Journal of Parallel and Distributed Computing,* vol. 71, July 2011, pp. 1034-1046.

[117] Y. Li, and I. R. Chen, "Design and Performance Analysis of Mobility Management Schemes based on Pointer Forwarding for Wireless Mesh Networks," *IEEE Transactions on Mobile Computing,* vol. 10, no. 3, 2011, pp. 349-361.

[118] Y. Li, and I. R. Chen, "Mobility Management in Wireless Mesh Networks utilizing Location Routing and Pointer Forwarding," *IEEE Transactions on Network and Service Management,* vol. 9, no. 3, 2012, pp. 226-239.

[119] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 7, no. 3, 2003, pp. 19-20.

[120] K. Liu, N. Abu-ghazaleh, and K.-d. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," *Jounal of Parallel Distributed Computing*, vol. 67, no. 2, Feb. 2007, pp. 215-228.

[121] Z. Liu, A. W. Joy, and R. A. Thompson, "A Dynamic Trust Model for Mobile Ad Hoc Networks," *IEEE International Workshop on Future Trends of Distributed Computing Systems*, 2004, pp. 80-85.

[122] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion Detection for Routing Attacks in Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, 2006, pp. 313-332.

[123] N. Luhmann, *Trust and Power : Two Works*, Chichester, New York: Wiley, 1979.

[124] N. Luhmann, *Risk: A Sociological Theory*: Aldine Transaction, 2005.

[125] M. Maheswaran, H. C. Tang, and A. Ghunaim, "Toward a Gravity-Based Trust Model for Social Networking Systems," *International Conference on Distributed Computing Systems Workshops*, June 2007, pp. 24-31.

[126] S. P. Marsh, "Formalising Trust as a Computational Concept," Department of Computing Science and Mathematics, University of Stirling, Stirling, UK, 1994.

[127] P. Massa, and P. Avesani, "Controversial Users demand Local Trust Metrics: an Experimental Study on Epinions.com Community," *the 25th American Association for Artificial Intelligence Conference*, 2005.

[128] D. H. McKnight, and N. L. Chervany, *The Meanings of Trust*, University of Minnesota, 1996.

[129] C. McLeod, "Trust," *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed., 2011.

[130] P. Michiardi, and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *IFIP TC6/TC11 Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, 2002, pp. 107-121.

[131] R. A. F. Mini, A. A. F. Loureiroa, and B. Nath, "The Distinctive Design Characteristic of A Wireless Sensor Network: the Energy Map," *Computer Communications*, vol. 27, no. 10, June 2004, pp. 935-945.

[132] B. A. Misztal, *Trust in Modern Societies: The Search for the Bases of Social Order*: Polity Press, 1996.

[133] R. Mitchell, and I. R. Chen, "Behavior Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications," *IEEE Transactions on Smart Grids*, 2013.

[134] R. Mitchell, and I. R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, March 2013, pp. 199-210.

[135] R. Mitchell, and I. R. Chen, "On Survivability of Mobile Cyber Physical Systems with Intrusion Detection," *Wireless Personal Communications*, vol. 68, no. 4, 2013, pp. 1377-1391.

[136] M. E. G. Moe, B. E. Helvik, and S. J. Knapskog, "TSR: Trust-based Secure MANET Routing using HMMs," *ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Vancouver, British Columbia, Canada, Oct. 2008, pp. 83-90.

[137] L. Moraru, P. Leone, S. Nikoletseas, and J. D. P. Rolim, "Near Optimal Geographic Routing with Obstacle Avoidance in Wireless Sensor Networks by Fast-Converging Trust-Based Algorithms," *ACM workshop on QoS and security for wireless and mobile networks*, Chania, Crete Island, Greec, 2007, pp. 31-38.

[138] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A Survey on Jamming Attacks and Countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, 2009, pp. 42-56.

[139] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation for E-businesses," *Hawaii International Conference on System Sciences*, Jan. 2002, pp. 2431-2439

[140] J. Mundinger, and J.-Y. L. Boudec, "Analysis of a Reputation System for Mobile Ad Hoc Networks with Liars," *Performance Evaluation*, vol. 65, no. 3-4, Mar. 2008, pp. 212-226.

[141] J. D. Musa, "Operational Profiles in Software-Reliability Engineering," *IEEE Software*, vol. 10, no. 2, March 1993, pp. 14-32.

[142] D. C. Mutz, "Social Trust and E-Commerce: Experimental Evidence for the Effects of Social Trust on Individuals' Economic Behavior," *The Public Opinion Quarterly*, vol. 69, no. 3, 2005, pp. 393-416.

[143] J. A. Nelder, and R. Mead, "A Simplex Method for Function Minimization," *Computer Journal*, vol. 7, no. 4, 1965, pp. 308-313.

[144] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based Routing in DTNs," *IEEE Conference on Computer Communications*, Rio De Janeiro, Brazil, Apr. 2009, pp. 846-854.

[145] E. C. H. Ngai, and M. R. Lyu, "Trust and Clustering-Based Authentication Services in Mobile Ad Hoc Networks," *Internatinal Conference on Distributed Computing Systems Workshops*, March 2004, pp. 582-587.

[146] E. Ostrom, "A Behavioral Approach to the Rational Choice Theory of Collective Action," *The American Political Science Review,* vol. 92, no. 1, March 1998, pp. 1-22.

[147] A. Passarella, and M. Conti, "Characterising Aggregate Inter-Contact Times in Heterogeneous Opportunistic Networks," *the 10th International IFIP TC 6 Conference on Networking*, 2011, pp. 301-313.

[148] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy, "On the Efficacy of Frequency Hopping in Coping with Jamming Attacks in 802.11 Networks," *IEEE Transactions on Wireless Communications,* vol. 9, no. 10, October 2010, pp. 3258-3271.

[149] C. R. Perez-Toro, R. K. Panta, and S. Bagchi, "RDAS: Reputation-Based Resilient Data Aggregation in Sensor Network," *IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, June 2010, pp. 1-9.

[150] C. E. Perkins, E. M. Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," *RFC 3561*, IETF, 2003.

[151] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *ACM Communications,* vol. 47, no. 6, 2004, pp. 53.

[152] R. Poovendran, and L. Lazos, "A Graph Theoretic Framework for Preventing the Wormhole Attack in Wireless Ad Hoc Networks," *Wireless Networks,* vol. 13, no. 1, Jan. 2007, pp. 27-59.

[153] C. Popper, M. Strasser, and S. Capkun, "Anti-Jamming Broadcast Communication using Uncoordinated Spread Spectrum Techniques," *IEEE Journal on Selected Areas in Communications,* vol. 28, no. 5, June 2010, pp. 703-715.

[154] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly Detection in Wireless Sensor Networks," *IEEE Wireless Communications,* vol. 15, no. 4, Aug. 2008, pp. 34-40.

[155] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks," *IEEE International Conference on Communications*, Glasgow, U.K., Jun. 2007, pp. 3864-3869.

[156] F. D. Rango, and S. Marano, "Trust-based SAODV protocol with intrusion detection and incentive cooperation in MANET," *International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, Leipzig, Germany, 2009, pp. 1443-1448.

[157] J. Rao, and X. Su, "A Survey of Automated Web Service Composition Methods," *Proceedings of the First international conference on Semantic Web Services and Web Process Composition*, San Diego, CA, USA, 2004, pp. 43-54.

[158] W. Ren, "QoS-Aware and Compromise-Resilient Key Management Scheme for Heterogeneous Wireless Internet of Things," *International Journal of Network Management,* vol. 21, no. 4, July 2011, pp. 284-299.

[159] P. Resnick, and R. Zeckhauser, "Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System," *Advances in Applied Microeconomics,* vol. 11, no. 12, 2002, pp. 127-157.

[160] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer,* vol. 44, no. 9, Sep. 2011, pp. 51-58.

[161] J. B. Rotter, "A New Scale for the Measurement of Interpersonal Trust," *Journal of Personality,* vol. 35, no. 4, 1967, pp. 651-665.

[162] S. D. Roy, S. A. Singh, S. Choudhury, and N. C. Debnath, "Countering Sinkhole and Black Hole Attacks on Sensor Networks Using Dynamic Trust Management," *IEEE Symposium on Computers and Communications*, July 2008, pp. 537-542.

[163] R. A. Sahner, K. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer Systems*: Kluwer Academic Publishers, 1996.

[164] F. D. Schoorman, R. C. Mayer, and J. H. Davis, "An Ingegrative Model of Organizational Trust: Past, Present, and Future," *The Academy of Management Review,* vol. 32, no. 2, 2007, pp. 344-354.

[165] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. "CRAWDAD data set Cambridge/haggle (v. 2009-05-29)," May 2009; http://crawdad.cs.dartmouth.edu/cambridge/haggle.

[166] J. R. Searle, *The Construction of Social Reality*: Free Press, 1997.

[167] J. Sen, P. R. Chowdhury, and I. Sengupta, "A Distributed Trust Mechanism for Mobile Ad Hoc Networks," *International Symposium on Ad Hoc and Ubiquitous Computing*, Surathkal, India, Dec. 2006, pp. 62-67.

[168] R. A. Shaikh, H. Jameel, B. J. d. Auriol, H. Lee, S. Lee, and Y.-J. Song, "Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 20, no. 11, November 2009, pp. 1698-1712.

[169] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, "Incentive-Aware Routing in DTNs," *IEEE Conference on Network Protocols*, Orlando, FL, USA, Oct. 2008, pp. 238-247.

[170] A. d. Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks," *ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, Oct. 2005, pp. 16-23.

[171] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," *IEEE/ACM Trans. on Networking,* vol. 16, no. 1, Feb. 2008, pp. 77-90.

[172] W. Stallings, "PGP Web of Trust," *Byte,* vol. 20, no. 2, 1995, pp. 161-162.

[173] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," *3rd ACM Conf. on Computer and Communications Security*, New Delhi, India, Jan. 1996, pp. 31-37.

[174] Y. Sun, Z. Han, and K. J. R. Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks," *IEEE Communications Magazine,* vol. 46, no. 2, Feb. 2008, pp. 112-119.

[175] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 2, Feb. 2006, pp. 305-317.

[176] P. Sztompka, *Trust: A Sociological Theory*: Cambridge University Press, 1999.

[177] G. Theodorakopoulos, and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 2, Feb. 2006, pp. 318-328.

[178] K. S. Trivedi, "Stochastic Petri Nets Package User's Manual," Department of Electrical and Computer Engineering, Duke University, 1999.

[179] A. Vahdat, and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks,* Technical Report, Duke University, 2000.

[180] P. B. Velloso, R. P. Laufer, D. d. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model," *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT,* vol. 7, no. 3, Sept. 2010, pp. 172-185.

[181] N. Verma, and I. R. Chen, "Admission Control Algorithms Integrated with Pricing for Revenue Optimization with QoS Guarantees in Mobile Wireless

Networks," *IEEE 10th International Conference on Parallel and Distributed Systems*, Newport Beach, USA, July 2004, pp. 495-502.

[182] R. R. S. Verma, D. O'Mahony, and H. Tewari, "NTM - Progressive Trust Negotiation in Ad Hoc Networks," *IEI/IEE Symposium on Telecommunications Systems Research*, Dublin, Ireland, Nov. 2001, pp. 1-8.

[183] B. Wang, S. Soltani, and J. K. Shapiro, "Local Detection of Selfish Routing Behavior in Ad Hoc Networks," *International Symposium on Parallel Architectures, Algorithms and Networks*, Dec. 2005, pp. 392-399.

[184] F. Wang, C. Huang, J. Zhao, and C. Rong, "IDMTM: A Novel Intrusion Detection Mechanism Based on Trust Model for Ad Hoc Networks," *International Conference on Advanced Information Networking and Applications*, March 2008, pp. 978-984.

[185] W. Wang, and B. Bhargava, "Visualization of Wormholes in Sensor Networks," *ACM Workshop on Wireless Security*, Philadelphia, PA, USA, 2004, pp. 51-60.

[186] X. Wang, L. Ding, and D. Bi, "Reputation-Enabled Self-Modification for Target Sensing in Wireless Sensor Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 1, Jan. 2010, pp. 171-180.

[187] Z. Xu, Y. Jin, W. Shu, X. Liu, and J. Luo, "SReD: A Secure REputation-based Dynamic Window Scheme for disruption-tolerant networks," *IEEE Military Communications Conference*, Boston, MA, Oct. 2009, pp. 1-7.

[188] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and Localized trUst management Scheme for sensor networks security," *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Vancouver, BC, Oct. 2006, pp. 437 - 446.

[189] O. Yilmaz, and I. R. Chen, "Utilizing Call Admission Control for Pricing Optimization of Multiple Service Classes in Wireless Cellular Networks," *Computer Communications*, vol. 32, no. 2, 2009, pp. 317-323.

[190] J. Yoon, B. D. Noble, M. Liu, and M. Kim, "Building Realistic Mobility Models from Coarse-Grained Traces," *the 4th International Conference on Mobile Systems, Applications and Services*, 2006, pp. 177-190.

[191] O. Younis, and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 3, Oct.-Dec. 2004, pp. 366-379.

[192] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, June 2008, pp. 576-589.

[193] M. Yu, M. Zhou, and W. Su, "A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, Jan. 2009, pp. 449-460.

[194] Y. Yu, L. Guo, X. Wang, and C. Liu, "Routing Security Scheme Based on Reputation Evaluation in Hierarchical Ad Hoc Networks," *Computer Network,* vol. 54, no. 9, June 2010, pp. 1460–1469.

[195] P. J. Zak, and S. Knack, "Trust and Growth," *The Economic Journal,* vol. 111, no. 407, Apr. 2001, pp. 295-321.

[196] C. Zhang, X. Zhu, Y. Song, and Y. Fang, "A Formal Study of Trust-Based Routing in Wireless Ad Hoc Networks," *IEEE Conference on Computer Communications*, March 2010, pp. 1-9.

[197] J. Zhang, R. Shankaran, M. A. Orgun, V. Varadharajan, and A. Sattar, "A Trust Management Architecture for Hierarchical Wireless Sensor Networks," *IEEE Conference on Local Computer Networks*, Denver, Colorado, Oct. 2010, pp. 264-267.

[198] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet," *International Conference on Embedded Networked Sensor Systems*, Baltimore, Maryland, USA, Nov. 2004, pp. 227-238.

[199] Y. J. Zhao, R. Govindan, and D. Estrin, "Residual Energy Scan for Monitoring Sensor Networks," *IEEE Wireless Communication and Networking Conference*, Orlando, USA, March 2002, pp. 356-362.

[200] L. Zhou, and H.-C. Chao, "Multimedia Traffic Security Architecture for the Internet of Things," *IEEE Network,* vol. 25, no. 3, May-June 2011, pp. 35-40.

[201] M.-Z. Zhou, J.-S. Xu, and C. Zhu, "A Secure Data Aggregation Algorithm Based on Behavior Trust in Wireless Sensor Networks," *IEEE International Symposium on Embedded Computing*, Oct. 2008, pp. 61-66.

[202] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. S. Shen, "SMART: A Secure Multilayer Credit-Based Incentive Scheme for Delay-Tolerant Networks," *IEEE Transactions on Vehicular Technology,* vol. 58, no. 8, Oct. 2009, pp. 4628-4639.

[203] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "Robust Cooperative Trust Establishment for MANETs," *ACM Workshop on Security of Ad Hoc and Sensor Networks*, Alexandria, VA, Oct. 2006, pp. 23-34.

Appendix A

Convergence of Trust Aggregation

We consider the *instantaneous trust* (reflecting the actual instantaneous behavior) of a node j as a stochastic process $\mathcal{G}_j = \{G_j(n): n = 1,2, \dots\}$. For example, for the *honesty* trust property, a node might behave honest or dishonest, i.e., $G_j(n)$ follows Bernoulli distribution; for the *cooperativeness* trust property, $G_j(n)$ could be a random variable in the space of [0, 1]. We define *objective trust* (or ground truth trust) as the expected value of $G_j(n)$, i.e., $E\left[G_j(n)\right]$. The goal of trust management is to estimate *objective trust* using direct observations on a node's instantaneous behaviors and recommendations. The direct observations of node i towards node j (also denoted as a stochastic process, $\mathcal{D}_{ij} = \{D_{ij}(n): n = 1,2, \dots\}$) might be different from $G_j(n)$ due to noises and imperfect detection mechanisms. We assume zero-mean white noise $\varepsilon_{ij}(n)$, i.e.,

$$D_{ij}(n) = G_j(n) + \varepsilon_{ij}(n) \tag{A.1}$$

According to our trust aggregation and propagation protocol in Section 9.2, the subjective trust evaluation of node i towards node j ($\mathcal{T}_{ij} = \{T_{ij}(n): n = 1,2, \dots\}$) is updated by either direct observations $\mathcal{D}_{ij}$ or recommendations from another node k ($\mathcal{R}_{kj} = \{R_{kj}(n): n = 1,2, \dots\}$).

When node i directly encounters node j,

$$T_{ij}(n) = (1 - \alpha)T_{ij}(n-1) + \alpha D_{ij}(n) \tag{A.2}$$

Otherwise, if node i encounters another node k,

$$T_{ij}(n) = (1 - \gamma)T_{ij}(n-1) + \gamma R_{kj}(n) \tag{A.3}$$

where $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$.

We analyze the trust convergence properties of our protocol based on trust estimation error, i.e., $e_{ij}(n) = T_{ij}(n) - G_j(n)$. We define trust convergence as $\lim_{n\to\infty} E\left[e_{ij}(n)\right] = 0$, meaning that the subjective trust evaluation converges to objective trust on average. Note that our definition of trust convergence is on the "mean" trust. As the mean trust converges, the variance will be stabilized, leading to an upper bound of trust bias.

In order to make the analysis proof tractable, we make the following assumptions:

1. The zero-mean measurement error $\varepsilon_{ij}(n)$ and instantaneous trust $G_j(n)$ are stationary or non-stationary stochastic processes;
2. The zero-mean measurement error $\varepsilon_{ij}(n)$ and instantaneous trust $G_j(n)$ are independent for a given target node j;
3. The zero-mean measurement error $\varepsilon_{ij}(n)$ and instantaneous trust $G_j(n)$ are independent to the past $\varepsilon_{ij}(n')$ and $G_j(n')$ where $n' < n$;
4. The measurement errors are pair-wise independent;
5. The encounter probability is independent of measurement errors;
6. If there are malicious nodes performing trust-related attacks, we assume a random mobility model and the probability of encountering each node is the same or at least similar to each other.

A.1 Stationary Environments without Attacks

We first consider a stationary environment where $\mathcal{G}_j$ and $\mathcal{D}_{ij}$ are stationary random processes and there is no trust-related attack (i.e., $R_{kj}(n) = D_{kj}(n)$). If node i encounters another node at stage n, there are two ways to update trust:

(1) If node i encounters node j, it uses direct observations to update trust. Then, the trust estimation error of node i towards node j at stage n is:

$$e_{ij}(n) = T_{ij}(n) - G_j(n) = (1-\alpha)T_{ij}(n-1) + \alpha D_{ij}(n) - G_j(n) \tag{A.4}$$

Considering $D_{ij}(n) = G_j(n) + \varepsilon_{ij}(n)$, we have

$$e_{ij}(n) = (1-\alpha)T_{ij}(n-1) + \alpha\left(G_{ij}(n) + \varepsilon_{ij}(n)\right) - G_j(n) \tag{A.5}$$

Reformatting the equation above by subtracting $(1-\alpha)G_j(n-1)$ from the first term on the right side, then we have

$$e_{ij}(n) = (1-\alpha)\Big(T_{ij}(n-1) - G_j(n-1)\Big) + (1-\alpha)G_j(n-1) + \alpha\Big(G_j(n) + \varepsilon_{ij}(n)\Big) - G_j(n) \quad \text{(A.6)}$$

By the definition of trust estimation error, $e_{ij}(n-1) = T_{ij}(n-1) - G_j(n-1)$, therefore,

$$e_{ij}(n) = (1-\alpha)e_{ij}(n-1) + (1-\alpha)G_j(n-1) - (1-\alpha)G_j(n) + \alpha\varepsilon_{ij}(n) \quad \text{(A.7)}$$

(2) If node i encounters another node k, rather than j at stage n, it uses the recommendation from node k to update trust. Using a similar method as above, we have the measurement error of node i towards node j at stage n as follows:

$$e_{ij}(n) = (1-\gamma)e_{ij}(n-1) + (1-\gamma)G_j(n-1) - (1-\gamma)G_j(n) + \gamma\varepsilon_{kj}(n) \quad \text{(A.8)}$$

Note that in this analysis, we assume there is no trust-related attack. Therefore $R_{kj}(n) = D_{kj}(n) = G_j(n) + \varepsilon_{kj}(n)$.

We use p $(0 \le p \le 1)$ to denote the probability that the node with which node i encounters at stage n is node j. Since we assume a stationary environment and zero mean measurement noise, we have $E\big[G_j(n-1) - G_j(n)\big] = 0$, $E\big[\varepsilon_{ij}(n)\big] = 0$, and $E\big[\varepsilon_{kj}(n)\big] = 0$. Then, taking the independence assumption, we have:

$$E\big[e_{ij}(n)\big] = \big(p(1-\alpha) + (1-p)(1-E[\gamma])\big)E\big[e_{ij}(n-1)\big] = \Theta E\big[e_{ij}(n-1)\big] \quad \text{(A.9)}$$

Since $D_{ik}(n) = G_k(n) + \varepsilon_{ik}(n)$ is independent of $e_{ij}(n-1)$, $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$ is independent of $e_{ij}(n-1)$. So, as long as $0 \le \Theta < 1$, $\lim_{n\to\infty} E\big[e_{ij}(n)\big] = 0$, i.e., the trust evaluation converges. To make sure $0 \le \Theta < 1$, we need $0 < \alpha \le 1$ and $0 < E[\gamma] \le 1$. Notice that $0 < E[\gamma] = E\left[\frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}\right] \le 1$ if $\beta > 0$. Hence, we have Lemma 1 as follows:

Lemma 1: *In stationary environment, if there is no trust-related attacks, the trust evaluation in our trust management protocol converges as long as* $0 < \alpha \le 1$ and $\beta > 0$.

From Equation (A.9), we note that the trust evaluation converges exponentially when $0 \le \Theta < 1$. Then the convergence speed increases as Θ decreases. Therefore, we have Lemma 2.

Lemma 2: *In stationary environment, if there is no trust-related attacks, the trust convergence speed of our trust management protocol increases as* α *or* β *increases* ($0 < \alpha \leq 1$ and $\beta > 0$).

We measure trust fluctuation by the variance of trust measure error, i.e., $Var\left[T_{ij}(n) - E\left[G_j(n)\right]\right]$. Considering a stationary environment, we have $Var\left[T_{ij}(n) - E\left[G_j(n)\right]\right] = Var\left[T_{ij}(n)\right]$. Below we analyze the effects of trust parameters on trust fluctuation. Again, we consider two cases depending on the node with which node i encounters.

(1) If node i encounters node j, it uses direct observation to update trust. Then, we have,

$$T_{ij}(n) = (1-\alpha)T_{ij}(n-1) + \alpha\left(G_j(n) + \varepsilon_{ij}(n)\right) \tag{A.10}$$

$T_{ij}(n-1)$ is obtained based on past direct observations and measurement errors at step $n-1$, so it is independent to $G_j(n)$ and $\varepsilon_{ij}(n)$. By adopting the independence assumption, $G_j(n)$ and $\varepsilon_{ij}(n)$ are also independent. Therefore:

$$Var\left[T_{ij}(n)\right] = (1-\alpha)^2 Var\left[T_{ij}(n-1)\right] + \alpha^2\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{A.11}$$

Since $G_j(n)$ and $\varepsilon_{ij}(n)$ are stationary random processes, $Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]$ is constant. Therefore, after trust convergence ($0 < \alpha \leq 1$ and $n \to \infty$), the variance of trust evaluation will stabilize to a constant value, i.e., $Var\left[T_{ij}(n)\right] = Var\left[T_{ij}(n-1)\right]$, so,

$$Var\left[T_{ij}(n)\right] = (1-\alpha)^2 Var\left[T_{ij}(n)\right] + \alpha^2\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{A.12}$$

and,

$$Var\left[T_{ij}(n)\right] = \frac{\alpha}{2-\alpha}\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right). \tag{A.13}$$

Now, we can see that in this case, after trust convergence, the trust fluctuation increases as α increases.

(2) If node i encounters another node k, rather than j at stage n, it uses the recommendation from node k to update trust. To simplify our analysis, we consider γ as a constant. Using the same analysis, we have

$$Var[T_{ij}(n)] = \frac{\gamma}{2-\gamma}\left(Var[G_j(n)] + Var[\varepsilon_{ij}(n)]\right) \quad \text{(A.14)}$$

Since $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$ in this case, after trust convergence, the trust fluctuation increases as β increases.

Because the trust update falls into either case (1) or case (2) above, we have Lemma 3 as follows:

Lemma 3: *In stationary environment, if there is no trust-related attacks, the variance or trust fluctuation of the trust value after convergence in our trust management protocol increases as α or β increases* ($0 < \alpha \leq 1, \beta > 0$).

A.2 Stationary Environments with Attacks

When there are malicious nodes performing trust related attacks, the trust evaluation in our trust management protocol may not converge to objective trust. However, we are able to select appropriate trust parameters such that the trust evaluation can stabilize and the bias can be minimized. Suppose that the percentage of malicious nodes in the network is λ and the probability that node i encounters node j at stage n is p. Again, there are two cases:

(1) If node i encounters node j, it uses direct observation trust to update trust. Then, the measurement error of node i towards node j at stage n is:

$$e_{ij}(n) = (1-\alpha)e_{ij}(n-1) + (1-\alpha)G_j(n-1) - (1-\alpha)G_j(n) + \alpha\varepsilon_{ij}(n) \quad \text{(A.15)}$$

(2) If node i encounters another node k, rather than j at stage n, it uses the recommendation from node k to update trust. Using a similar method above, we have the measurement error of node i towards node j at stage n as follows:

$$e_{ij}(n) = (1-\gamma)e_{ij}(n-1) + (1-\gamma)G_j(n-1) - (1-\gamma)G_j(n) + \gamma\varepsilon'_{kj}(n) \quad \text{(A.16)}$$

Note that here $\varepsilon'_{kj}(n)$ might have a non-zero mean if node k is malicious and performs trust related attacks.

By adopting the random mobility assumption and combining the two cases, we have

$$E[e_{ij}(n)] = \left(p(1-\alpha) + (1-p)(1-E[\gamma])\right)E[e_{ij}(n-1)] + \lambda P_{fn}(1-p)E[\gamma]\varepsilon'_{kj}(n) \quad \text{(A.17)}$$

Here, P_{fn} is the false negative probability in detecting a malicious node. We can see that as long as $0 \leq \Theta = p(1-\alpha) + (1-p)(1-E[\gamma]) < 1$, $E\left[e_{ij}(n)\right]$ will eventually converge. Therefore, let $E\left[e_{ij}(n)\right] = E\left[e_{ij}(n-1)\right]$, we have:

$$E\left[e_{ij}(n)\right] = \frac{\lambda P_{fn}(1-p)E[\gamma]}{\left(p(1-\alpha) + (1-p)(1-E[\gamma])\right)} \varepsilon'_{kj}(n) \tag{A.18}$$

Reformatting the equation above, we have:

$$E\left[e_{ij}(n)\right] = \frac{\lambda P_{fn}(1-p)E[\gamma]}{\alpha p + (1-p)E[\gamma]} \varepsilon'_{kj}(n) < \lambda P_{fn}\varepsilon'_{kj}(n) \tag{A.19}$$

Here $E[\gamma] = E\left[\frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}\right] = \frac{\beta P_{fn}}{1+\beta P_{fn}}$. In our protocol, a trust value is in the range of [0, 1]. Therefore $E\left[e_{ij}(n)\right] < \lambda P_{fn}$.

Lemma 4: *In stationary environment, if there are malicious nodes performing trust related attacks, the trust evaluation in our trust management protocol can stabilize as long as* $0 < \alpha \leq 1$ and $\beta > 0$. *The trust bias is less that* λP_{fn} *after trust stabilizes and decreases as* α *increases or* β *decreases.*

A.3 Non-stationary Environments

In non-stationary environments, the objective trust status may change before trust converges/stabilizes since trust convergence and stabilization take time. However, from Equations (A.9) and (A.17), we can that the trust evaluation of our trust management protocol will approach the objective trust after objective trust changes. In order to quickly adapt to environment changes, we may select high α and β to shorten trust convergence time. However, increasing α and β results in high trust fluctuation (Lemma 3). The best trust parameter settings to minimize trust evaluation error depend on the application requirements and network environments.

Appendix B

Notation and Acronym

B.1 Notations

$T_{i,j}(t)$	Trust value of node *j* as evaluated by node *i* at time *t*
$T_{i,j}^{X}(t)$	Trust value of node *j* in trust component *X* as evaluated by node *i* at time *t*
$T_{i,j}^{direct,\ X}(t)$	Trust value of node *j* in trust component *X* as evaluated by node *i* at time *t* using direct observations
$T_{i,j}^{indirect,\ X}(t)$	Trust value of node *j* in trust component *X* as evaluated by node *i* at time *t* using indirect recommendations
$T_{i,j}^{1-hop,\ X}(t)$	Direct trust value of node *j* in trust component *X* as evaluated by node *i* at time *t* using direct observations in current encounter
$T_{i,j}^{encounter,\ X}(t)$	
$T_{j}^{sub,X}(t)$	Average subjective component *X* trust of node *j* at time *t*
$T_{j}^{sub}(t)$	Overall average subjective trust value of node *j* at time *t*
$T_{j}^{obj,X}(t), T_{j,obj}^{X}(t)$	Objective component *X* trust value of node *j* at time *t*
$T_{j}^{obj}(t), T_{j,obj}(t)$	Overall objective trust value of node *j* at time *t*
λ_d	Trust decay factor
Δt	Trust evaluation interval/encounter interval
$P_{mission}$	Mission success probability
$R(t)$	Mission reliability given that mission time is *t*
$R_j(t)$	Node *j*'s reliability at time *t*

M_1	Trust threshold above which a member is considered completely trustworthy for successful mission completion
M_2	Drop dead trust level below which a member is completely not trustworthy
$X_j(t)$	Instantaneous trustworthiness of node *j* at time *t*
S_{init}	Node's initial speed
R, r	Wireless ratio range
E_{init}, E_0	Node's initial energy level
λ_{com}, λ_c	Node compromising rate
E_{remain}	Node's remaining energy level
$M_{difficulty}$	Difficulty level of the given mission
S_{degree}	Degree of uncooperativeness of 1-hop neighbors
T_{gc}	Group communication interval
r_s	Reward assignment to state *s*
$P_s(t)$	State (*s*) probability at time *t*
E_T	Energy threshold below which the energy trust goes to 0
P_{fn}, P_{fn}^H	False negative probability
P_{fp}, P_{fp}^H	False positive probability
d	Time windows size to compute trust
n_r	Number of 1-hop neighbors as recommenders
L	Number of replications that a node can forward for each message
$\alpha, \beta, \beta_1, \beta_2, \gamma$	Trust aggregation parameters
w_1, w_2, w_3, w_4	Trust formation parameter (weight ratio for trust components forming overall trust)
$C_{i,j}^{direct,X}(t)$	Boolean variable indicating if the needed data for node *i* assessing component *X* trust towards node *j* is obtainalbe in current encounter interval

T_{rec}	Recommender trust threshold
T_f	Forwarding trust threshold
R_i	Set of node *i*'s 1-hop neighbors with trust value greater than recommender trust threshold
Ω	Top percentile of trust values
P_{error}	Error probability of detection mechanisms due to noise
P_{rand}	Random attacks probability
M_c	Set of sensor node in the cluster with cluster header *c*
T^{th}	System minimum trust threshold below which a node is considered as misbehaving
Δ_{E-SN}	Energy consumption rate for a normal sensor node
Δ_{E-CH}	Energy consumption rate for a normal cluster header
$\Delta_{E-compromised}$	Energy consumption rate for a compromised node
ρ	Energy saving ratio of an uncooperative node compared with a normal node
P_{uncoop}	Node's uncooperative probability
$E_{consumed}$	Node's consumed energy
$N_{neighbor}^{coop}$	Number of cooperative 1-hop neighbors
$N_{neighbor}$	Number of 1-hop neighbors
μ	Weight associated with the *energy* term vs. the *cooperative neighborhood* term determining a node's uncooperative probability
λ_{c-init}	Node's initial compromising rate
$N_{neighbor}^{compromised}$	Number of compromised 1-hop neighbors
$N_{neighbor}^{uncompromised}$	Number of uncompromised 1-hop neighbors
T_{IDS}	Intrusion detection system execution interval

B.2 Acronyms

CH	Cluster Head
CoI	Community of Interest
DDL	Device Description Language
DoS	Denial of Service
DSSS	Direct-Sequence Spread Spectrum
DTN	Delay/Disruption Tolerant Network
FHSS	Frequency Hopping Spread Spectrum
FTT	Forwarding Trust Threshold
GDH	Group Diffie-Hellman
HMAC	Keyed-Hashing for Message Authentication
HMM	Hidden Markov Model
IDS	Intrusion Detection System
IoT	Internet of Things
MAC	Media Access Control
MAE	Mean Absolut Error
MANET	Mobile Ad-Hoc Network
MSE	Mean Square Error
QoI	Quality of Information
QoS	Quality of Service
ROC	Receiver Operating Characteristic
RTT	Recommender Trust Threshold
SFD	Static-Followed-by-Dynamic
SN	Sensor Node
SPN	Stochastic Petri Net
SVM	Support Vector Machine
WSN	Wireless Sensor Network