Achieving Secure and Cooperative Wireless Networks with Trust Modeling and Game Theory

LI, Xiaoqi

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

Aug. 2009

Achieving Secure and Cooperative Wireless Networks with Trust Modeling and Game Theory

submitted by

LI, Xiaoqi

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong

Abstract

A mobile ad hoc network (MANET) is a kind of wireless network without centralized administration or fixed network infrastructure. Because of the nature of self-organization and the limitation of individual resources, MANET always confront security and selfishness issues.

In this thesis, we introduce an idea about "trust modeling" and propose a trusted routing protocol based on the trust relationships among nodes. We derive our trust model from subjective logic. In our trust model, trust is represented by an *opinion*, which contains three elements, belief, disbelief and uncertainty, expressing the probabilities that a node can be trusted or distrusted, and the probability of uncertainty about the trustworthiness of a node, respectively. Trust combination algorithms and trust mapping functions are provided in this model, where the former can aggregate different opinions together to get a new recommendation opinion, and the latters offer the trust mapping between the evidence space and the opinion space.

Based on this trust model, we design our trusted routing protocols for MANET called TAODV on top of Ad Hoc On-demand Distance Vector (AODV) routing protocol. We extend the routing table and the routing messages of

ADOV with trust information which can be updated directly through monitoring in the neighborhood. Besides, we also present a trust recommendation protocol. When performing trusted routing discovery, unlike those cryptographic schemes that perform signature generation or verification at every routing packet, we just combine the recommended opinions together and make a routing judgment based on each element of the new opinion. In this way the computation overhead can be largely reduced, and the trustworthiness of the routing procedure can be guaranteed as well. Through simulation we can see that the bad nodes are clearly separated from the good nodes, and we do not introduce much overhead as other cryptographic schemes do.

Security issues and selfishness issues of wireless networks can also be formulated with game theory. In this thesis, we model the security and selfishness issues of wireless networks as three different games, either in non-cooperative form or in cooperative form.

First, we formulate the security issues of wireless networks as a non-cooperative game. The interactions between the attacker and the regular node are modeled into two signaling game trees according to the type of the node. Through the theoretical analysis we are able to obtain a bound for the value of payoff assignment, which can guide the design of payment schemes and incentive routing protocols.

Second, we model the security issues of wireless networks with a cooperative game, called coalitional game. We design two value functions, security characteristic function and throughput characteristic function, from the aspects of achieving maximum security and maximum throughput for a coalition, respectively. We also present the coalition formation algorithm and the integration of the algorithm with existing routing protocols. Theoretically we analyze the existence of the stable core status of the game, and the convergence speed of the stable status. From the simulation results we can observe that the malicious nodes are all isolated outside any coalitions eventually.

Third, we study the selfishness issues of wireless networks with another coalitional game. In our work, different from others, we propose an incentive routing and forwarding scheme that combines the payment mechanism and the reputation system together, and analyze it with a coalitional game. The reputation system we employ is a heat diffusion model on a weighted reputation graph. We further present a new value function for coalitions taking into account the amount of payment and cost of a node in a coalition. We theoretically prove that this coalitional game has a core status. The simulation results show that the cumulative utilities of cooperative nodes are increased steadily and the selfish nodes cannot get more utilities by behaving selfishly than cooperatively.

基于信任模型和博弈論的安全型及合作型無線網絡

李曉琦

摘 要

移動自組網絡（MANET）是一種沒有集中式管理或固定網絡基礎結構的無線網絡。由于本質上的自組性和個體節點資源的局限性，這種網絡總是存在着安全性和自私性的問題。

在本論文中，我們引入了“信任模型”的思想，提出了一個基于節點之間信任關系的可信路由協議。我們的信任模型源自主觀邏輯。在此模型中，信任值由“觀點”來表示。觀點是一個三元組， 包含“信任”、“不信任”和“不確定”三個元素，分別代表一個節點可被信任的概率、不可被信任的概率，以及對節點可信度不確定的概率。此模型提供了信任組合算法和信任映射函數，前者可將不同的觀點組合得到一個新的推薦觀點值，後者可使信任值在證據空間和觀點空間之间相互映射。

基于信任模型，我們爲MANET在自組距離向量路由協議（AODV）之上設計了一種可信路由協議，TAODV。我們擴展了AODV的路由表和路由報文，在其中添加了信任信息。這些信任信息可以直接通過鄰居節點間的監測來更新。另外，我們還提出了一個信任推薦協議。當進行可信路由發現時，不像其它基于密碼學的方案那樣，對每一個路由報文進行數字簽名的生成或檢驗，我們的方案只是把推薦來的觀點組合起來，然後根據新觀點中每一個元素的值來進行路由判斷，从而大幅縮減計算開銷，同時保證路由過程的可信度。通過模擬實驗，我們可以看到壞的節點被清楚的從好的節點中分離，並且我們的方案並沒有引入过多計算開銷。

無線網絡的安全性和自私性問題也可以用博弈論來建模。在本論文中，我們將無線網絡的這兩個問題建模成三個不同的博弈，分別采用非合作型博弈的形式或合作型博弈的形式。

首先，我們把安全性問題建模成一個非合作型博弈。攻擊者和正常節點之間的交互根據節點類型的不同被表達成兩個信號博弈樹。通過理論分析我們最終得到了一個收益分配的邊界值，這個值可用作設計支付機制和激勵型路由協議的參考。

其次，我們把安全性問題建模成一個合作型博弈，稱爲聯盟博弈。我們設計

了兩種特征函數，安全性特征函數和吞吐量特征函數，分別代表一個聯盟可以達到得最高的安全性和最大的吞吐量。我們也提出了聯盟形成的算法，並將此算法與現有的路由協議集成起來。我們從理論上分析了此博弈的穩態“核”的存在性，並研究了穩態的收斂速度。從模擬實驗結果我們觀察到惡意節點最終全部被隔離在所有聯盟之外。

再次，我們把自私性問題也建模成一個合作型博弈。我們提出一個激勵型的路由和轉發方案，與別不同的是，這個方案集成了支付機制和聲望系統，並且用一個聯盟博弈對此方案進行分析。我們采用的聲望系統是一個在加權聲望圖上進行的熱傳導模型。我們也爲聯盟定義了一個新的特征函數，此函數考慮了聯盟中節點的收入和開銷的情況。从理論上，我们證明了這個博弈有穩定的“核”狀態。模擬實驗結果顯示合作型的節點累積的收益在穩定增加，而自私型節點不能通過自私行爲得到比合作行爲更多的收益。

Acknowledgements

There are many persons I would like to thank. First and foremost, I want to thank my supervisor, Prof. Michael R. Lyu. I gain too much from his guidance in both the attitude in doing research and the detailed technique things in conducting my research work. I would like to express my sincere gratitude and appreciation to his supervision, encouragement, and support at all levels. I also thank Prof. Jiangchuan Liu, my supervisor in the second year of my Ph.D study, from whom I have learned practical research experiences and received many valuable suggestions, especially in the design of trusted routing protocol. I am grateful for the outstanding research environment fostered by our department, and also for so many related work done by our clerical staffs.

I also thank my thesis committee members, Prof. K. S. Leung and Prof. K. H. Lee, for their worthful advises to my theis work. Their professional dedication and responsible manners have motivated me to persist on my research journey. I extend my thanks to Prof. Jianwei Huang for the conservations and discussions on the issues of coalitional game formulation. These communications have profoundly shaped the ideas in this thesis.

I would like to thank my colleagues and my friends. I thank Wujie Zheng, for his great and selfless help on the modeling and simulation of my thesis work. Thank Haixuan Yang and Hao Ma, who have given me worthwhile suggestions on mathematical expressions and algorithm design. I also want to thank my groupmates Kaizhu Huang, Yangfan Zhou, Xinyu Chen, Zibin Zheng, Hongbo Deng, Jianke Zhu, Junjie Xiong, Xin Xin, Chao Zhou, and Zenglin Xu. Their broad knowledge, encouragement, love of life, sense of humor made me feel stimulant, relaxed and happy when working in the office. I extend my gratitude to all my previous officemates and the friends from Fairyland BBS.

Finally, I want to thank my parents and my brother. Without their deep love and constant support, this thesis could not have been completed.

To My Parents

獻給我的父母

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Mobile Ad Hoc Network (MANET)

A mobile ad hoc network (MANET) [14, 64] is a kind of wireless network without centralized administration or fixed network infrastructure, where nodes are free to move and communicate with each other over bandwidth-constrained wireless links, and perform routing discovery and routing maintenance in a self-organized and cooperative way. MANET can be applied to situations where an infrastructure is unavailable or deploying one is not cost effective. Such situations include disaster recovery, military fields communications, or some other crisis management services. A simple demonstration is shown in Fig. 1.1. It can also be widely applied in our daily life situations. For example, some business environment where a meeting or collaborative computing assignment is required to be conducted outside the office environment, or in some public vehicles where people want to play online games with other passengers who carry mobile phones with build-in WiFi or Bluetooth modules, etc.

The topology of MANET may change uncertainly and rapidly due to high mobility of the independent mobile nodes. Because of network decentralization, each node in MANET would act as a "router" to discover a routing path or to forward the data packets. Unlike wired networks, the functional design

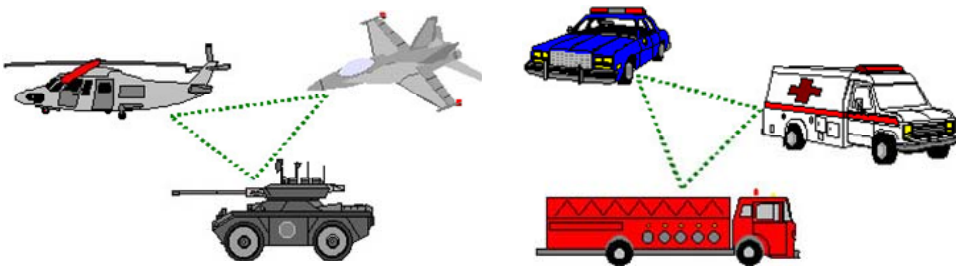


Figure 1.1: Applications of Mobile Ad Hoc Networks.

of MANET must take into account many factors such as wireless link quality, power limitation, multiuser interference and so on.

1.2 Security Issues

Self-organization, decentralization and openness are main advantages of MANET, but these characteristics also introduce insecurity. Nodes scatter in different positions moving in all directions randomly every now and then. They can join or leave the network with high flexibility because of no centralized authority. However, in such mobile situation, nodes are lacking of sufficient information about each other, which increases the risk of being compromised from either outside or inside. Malicious nodes can also join the network freely and perform all kinds of attacks to the network to eavesdrop information, interrupt normal communications, or even make the whole network denial-of-service.

1.2.1 Attacks to Mobile Ad Hoc Networks

The attacks may destroy one or several security attributes in the following: *Confidentiality*, *Authenticity*, *Integrity*, *Availability*, *Non-repudiation* and *Access control*. The detailed descriptions are as follows [41]:

Confidentiality Stored or transmitted information is accessible only by authorized parties.

Attack Method	Motivation/Result	Influence to Security Services
Eavesdropping	Obtain contents of messages	Loss of Confidentiality
Masquerading	Impersonate good nodes /Routing Redirection /Routing table poisoning /Routing Loop, etc.	Loss of Authenticity
Modification	Make a node denial of service /Obtain keys, etc.	Loss of Integrity
Tunneling	Attract traffic /Routing Redirection	Loss of Confidentiality and Availability
Flooding	Denial of Service	Loss of Availability
Dropping	Destroy normal routing progress	Loss of Non-reputation and Availability
Replaying/Delaying	Destroy normal routing progress /Destroy normal data transmission	Loss of Access Control and Integrity

Figure 1.2: Attacks to Mobile Ad Hoc Networks.

Authenticity Identity of the origin of the message is correctly identified.

Integrity Only authorized parties can modify stored or transmitted information and system assets.

Availability Network resources are available to authorized parties.

Non-repudiation Sender or receiver cannot deny the transmission.

Access Control Information resources are controlled.

We classify the attacks to MANET in terms of attack methods. Attackers may employ one or more attack methods to achieve their different goals. These attack methods are *eavesdropping*, *masquerading*, *modification*, *tunnelling*, *flooding*, and *package-oriented attack method*, which are described in detail in the following. Figure 1.2 summarizes those methods.

Eavesdropping MANET is a kind of wireless network. The property of wireless connections results in the possibility of being eavesdropped. Attackers can analyze the payloads of the packets and obtain the content

information. This attack method will destroy the confidentiality of information. Usually, encryption is needed to prevent against eavesdropping.

Masquerading It means that attackers will forge some artificial packages in order to impersonate good nodes. This attack will badly affect the authenticity of information. Almost all the traditional routing protocols for MANET do not perform routing information verification and they trust each routing request or reply by default. We take AODV routing protocol for example. In AODV [68] there are three main routing messages: Routing Request, Routing Reply and Routing Error, which will be described in detail in Chapter 3. For routing request messages, attackers can impersonate a source node by forging a routing request message with his address as the originator address. Because the destination sequence number of a node can be set in AODV, the originator of a routing request message can put a much bigger destination sequence number than the real one so as to improve the living chance of this routing request. For routing reply messages in AODV, furthermore, attackers can then forge a reply message to a node to claim a faked shorter path. The attacker can then form routing loops by sending faked shorter path reply messages to several nodes in the neighborhood of a node N_i one by one. Finally by forging routing error messages, the attacker can lie to others and convince them that node N_i is unreachable.

Modification This attack often happens when a node forwards routing messages. A malicious node will intercept messages and alter their contents before passing them on to the intended recipient. "Man-In-Middle" [49] attack belongs to this category. Integrity of information is tampered by this kind of attack. Let's also take AODV routing protocol [66] for example. When forwarding a routing request message, a malicious node can reduce the *hop count* field in the message to increase the chances of being

in the route path. The malicious node can also increase the destination sequence number of the incoming message in order to update the other intermediate nodes' routing table.

Tunnelling Tunnelling attack needs the cooperation of two or more malicious nodes. One of them may encapsulate the routing messages and exchange the encapsulated messages with the other malicious node through the normal route path of the network. Then the two nodes will decapsulate the messages and forward them out. Thus, they can establish a "virtual", "direct" path between them. By the tunnelling attack, attackers can confuse the good nodes that there is a shorter path going though the malicious nodes. This attack will influence the availability of certain network resources. It is difficult to prevent and detect the tunnelling attack. Nearly none of the existing secure routing protocol can solve this difficult problem.

Flooding Attackers may launch a great lot of messages in a short time to a node, a channel or some other network resources to make them too "crowded" to accept any more requests. This is a kind of denial-of-service attack.

Packet Oriented Attacks This category includes many attack methods baring similar properties. That is, these attacks only focus on the quantity or transmitting time of packages, including both routing and data packages. Dropping, replaying, and delaying packets all belong to this category. Dropping violates the non-reputation property in the secure services. In AODV routing protocol, a malicious node will not forward certain routing requests, routing replies and even data messages. This kind of attacks usually cannot be correctly detected because transmission errors also have the same effect [87]. Replaying means storing intercepted messages and sending them again later. Delaying is just sending messages

in a later time. These attacks can take effect without prior knowledge of the authentication policy or decrypting the messages.

From the above descriptions, we see that the attacks can be multifarious and the routing protocol is prone to be compromised. Therefore, security mechanisms must be designed to protect the individual node and the whole network.

1.2.2 Previous Security Solutions

Many security schemes from different aspects of MANET have been proposed in order to protect the routing information or data packets during communications, such as secure routing protocols [30, 31, 70, 85, 87] and secure key management solutions [11, 32, 42, 51, 92]. Due to resource scarcity (battery power, memory, and processing power) of nodes, securing MANET is quite different from traditional schemes that generally involve management and safe keeping of a small number of private and public keys [3]. The security mechanism for MANET, on one hand, must require low computation complexity and a small number of appended messages to save the node energy. On the other hand, it should also be competitive and effective in preventing misbehaviors or identifying misbehaving nodes from normal ones.

However, most of these schemes assume that there are trusted third parties or centralized servers who are responsible for issuing digital certificates and keys or monitoring the behaviors of other nodes. Centralized servers or trusted parties make the network more controllable but they destroy the self-organizing nature of MANET and reduce the network scalability. Even some schemes distribute the servers into many nodes, there are still bottlenecks due to centralization. If the scheme distributes the functions of servers into each node of the network, it will introduce significant performance overhead. What's more, by requiring nodes to generate and verify digital signatures all the time,

these solutions often bring huge computation overhead. Therefore, we need a self-organized light-weight security scheme for mobile ad hoc networks.

1.2.3 Trust Model and Trusted Routing Protocol

In our work to be described in the thesis, we will focus on designing a secure routing mechanism for MANET in a self-organized way instead of using centralized servers. Our solution is introducing the idea of "trust" to solve this problem. For example, one node N_1 can judge whether the other node N_2 could be trusted according to the trust value that N_1 obtains about N_2, thus N_1 can decide whether to go on communicating with N_2 or require N_2 to prove itself by some other ways. N_1 can also obtain a more credible trust value of N_2 by exchanging values with other nodes and calculating the new trust value of N_2 using some combination algorithms. In this way, we can achieve real self-organized trust relationships among all the nodes. Moreover, some authentication measures, such as digital signature, can be performed in a more flexible way based on the trust value so the system overhead can be greatly reduced.

Trust models have found security applications in e-commerce, peer-to-peer networks, and some other distributed systems [1, 7, 39, 76, 82]. In recent years, some research studies are conducted to apply trust models into the security solutions of MANET [19, 29]. However, there are no much concrete and applicable designs proposed for the security of routing protocols of MANET.

The trust model we employ is derived from subjective logic [35, 36, 37, 38], which qualitatively defines the representation, calculation, and combination of trust. Trust is represented by a 3-element triad called *opinion*. The three elements in an opinion are *belief*, *disbelief* and *uncertainty*, which means the probabilities that a node can be trusted or distrusted, and the probability of uncertainty about the trustworthiness of a node, respectively. The opinions are

obtained either from positive or negative evidences accumulated in the interactive communications, or from the combination of different recommendation opinions.

Based on this trust model, we design our secure routing protocol for MANET accroding to Ad hoc On-demand Distance Vector (AODV) routing protocol [67]. The new protocol, called TAODV (Trusted AODV), has several salient features: (1) Nodes perform trusted routing behaviors mainly according to the trust relationships among them; (2) A node which performs malicious behaviors will eventually be detected and denied to the whole network; and (3) System performance is improved by avoiding requesting and verifying certificates at every routing step. The idea of the trust model can also be applied to other routing protocols of MANETs, such as DSR [33], DSDV [65] and so on.

1.3 Selfishness Issues

The nature of wireless ad hoc networks is to let nodes cooperative together thus improve the connectivity and throughput of the whole network or execute some specific functions inside the network. However, nodes in this kind of network may belong to different individuals or authorities and have their own interests, and because of the limit of individual power and bandwidth, they are inherently reluctant to forward packets for others. These behaviors are called *selfishness*. Consequently, it is necessary to design incentive mechanisms to encourage cooperations among the nodes and solve the non-forwarding problem so as to increase the throughput of the networks.

To attack the selfishness problem, we investigate incentive routing and forwarding schemes in our research. Incentive routing schemes for enforcing selfish agents to cooperate in wireless networks have been studied for years. Most approaches fall into one of two main categories: approaches rewarding cooperative nodes and those punishing non-cooperative nodes. The first category

of solution applies monetary incentives to reward cooperative nodes, either virtually or practically. Payment schemes need to be designed and are usually analyzed by game theoretic methods. In these schemes, the intermediate nodes declare their costs for forwarding packages. Then the routing protocol selects the lowest cost path (LCP) based on the declared costs. Afterwards the payments are rewarded to nodes on and sometimes off the LCP with the amount no less than their declared costs. However, a problem arises when nodes may purposely declare a higher cost to take advantage of the payment algorithms. So currently more research is focused on how to avoid cheating and achieve effective and also economic payments.

The following schemes belong to this category. In Ad Hoc-VCG [4], payments are paid to nodes who forward data packets for others, consisting the actual costs incurred by forwarding data and the extra premiums. The implemented reactive routing protocol is a variation of the well-known VCG mechanism. It achieves the design objectives of truthfulness and cost-efficiency in a game-theoretic sense. But Ad Hoc-VCG is not budget-balanced. Another work [10] introduces a virtual currency called *nuglets*. The source of the packet must load it with enough nuglets to pay for the trip to the destination. Cooperation is enforced in this scheme because nodes must forward packets for others in order to build up enough nuglets to get their own packets forwarded. NUGLETS [10] is budget-balanced. Somewhat similar in scope to nuglets is SPRITE [89], which employs a Credit Clearance Service (CCS) to store the credit, as opposed to the tamper-proof module used in nuglets. However, centralized services tend to defeat the purpose of ad hoc networks. [90] designs an incentive-compatible routing and forwarding protocol integrating VCG mechanism and cryptographic technique. Payments are implemented based on VCG protocol and the application of cryptographic techniques in the design of forwarding protocol enforces the routing decision. [91] designs a

collusion-resistant routing scheme for non-cooperative wireless networks. Payments are given to nodes not only on the LCP paths but also off the paths.

The second category employs reputation systems to stimulate the nodes to cooperative. Non-cooperative nodes in the network are identified based on a reputation system and circumvented in the routing process. The primary goal of reputation-based schemes is to block selfish nodes from the network. The common idea of these schemes is that each node in the network will monitor the behaviors of its neighbors. If the neighbors are observed for not executing some functions properly, their reputations will be decreased and they will be under the threat of being blocked from the network. The key challenges are how to combine the direct neighborhood reputations together and propagate them from locally to globally. In these systems, game theoretic methods can also be employed to analyze the effectiveness of the threatening mechanism.

Several mechanisms belonging to this category are summarized as follows. In CORE [59], node cooperation is stimulated by a collaborative monitoring technique and a reputation mechanism. Each node of the network monitors the behavior of its neighbors with respect to a requested function and collects observations about the execution of that function. If the observed result and the expected result coincide, the observation will take a positive value; otherwise it will take a negative value. CONFIDANT [9] differs from CORE only in that it sends reputation values to other nodes in the network, which exposes the scheme to malicious spreading of false reputation values. Liu and Issarny [48] employ a Bayesian approach to design an incentive compatible reputation system to facilitate the trustworthiness evaluation of nodes. Some also use subjective logic to calculate uncertain trust so as to desgin secure routing protocols [46] or incentive reputation mechanisms [40]. A theory of semirings is also applied to evaluate trust and trust relations in [78].

Although most of the above schemes only employ one category of incentive solutions, in fact we consider that the methods of applying monetary incentives

and reputation systems are not mutually exclusive and they can be combined together to design a more flexible incentive scheme. Our design details of such a scheme will be described in Chapter 6

1.4 Game Theoretic Formulation

Recently game theory has been employed extensively to model networking problems, where different players may have different strategies for network usage. Game theory is a formal way to analyze interaction among a group of rational players who behave strategically. A game is the interactive situation, specified by a set of nodes, the possible actions of each node in the set, and a set of all possible payoffs. For recent years, many researchers have also tried to model the wireless network as a game. Security issues and selfishness issues of this kind of network are two main applications of game theory.

For the security issues, due to the variety of malicious behaviors, it is more difficult to apply game theory to security problems than the selfishness issues. Malicious behaviors or attack actions may be manifested with all kinds of forms, which brings the challenges to restrict them into a safe range. However, the malicious nodes still demonstrate certain behavior pattens that usually take several steps to fulfill one attack. They must be rational enough to perform harmful actions and at the same time hide themselves from being detected or denied by the network, in which case no more harmful actions are to be performed.

These security issues can be modeled as non-cooperative games played between one attacker and one target, between one attacker and the whole network, or between two or more attackers and the rest of the network. [63] is such a non-cooperative game formulation for intrusion detection system in mobile ad hoc networks. It views the interaction between the attacker and the individual node as a two-player multi-stage dynamic game with incomplete

information. Some schemes model the data forwarding in the wireless networks as a strategic game and, under certain formal assumptions, derive a forwarding rate for nodes that form a Nash equilibrium. Generous Tit-for-Tat [75] is an example of this approach. More generally, Urpi et al. [81] present several results characterizing enforceable policies in a setting where each node's utility function includes both bandwidth and energy terms.

They can also be modeled as cooperative games like [2]. In [2], the authors define a cooperative game between sensor nodes and concentrate on three fundamental factors: *cooperation*, *reputation* and *quality of security*. The more a node cooperates, the better its reputation is, which decreases when misbehavior is detected. When security of the network is compromised, the percentage of exposed traffic measures the quality of security of sensor nodes. By incorporating these three factors, sensor nodes are clustered where payoff is the largest possible individual gain for each sensor according to a defined utility metric.

Paper [60] proposes two methods to evaluate the effectiveness of the CORE mechanism based on a cooperative game approach and a non-cooperative game approach. The results obtained using the first approach define a lower bound on the number of legitimate nodes in an ad hoc network when the CORE mechanism is adopted, while the second approach describes the asymptotical behavior of a selfish node that is controlled by CORE. Paper [53] shares the experiences applying game theory to system design. It said that it is difficult to apply game theory straightforwardly to the system design.

In our work, we formulate the security issues of wireless networks through both non-cooperative games and cooperative games.

- For the non-cooperative game formulation, we regard the interactions between an attacker and a regular node as a non-cooperative dynamic

repeated game with incomplete information. Two different game structures are given, each of which depends on the different types of the attacker or the regular. We show that both of them achieve perfect Bayesian Nash equilibrium, thus leading to a defense strategy for the regular node.

- For the cooperative game formulation, we propose a novel coalitional game model [45]. We define a new throughput characteristic function, on the basis of which nodes are enforced to cooperate and form coalitions. The physical meaning of the throughput characteristic function is the maximal throughput and the most reliable traffic that a coalition can achieve.

For the selfishness issues, several incentive mechanisms [4, 9, 10, 59, 89, 90] based on game theory have been proposed. These schemes usually model the selfish behaviors as non-cooperative games. However, in wireless networks nodes cannot perform routing and forwarding behaviors individually. They must cooperate together to complete one task, so it is natural to think about modelling the wireless network behaviors as a cooperative game. In our work, we model the routing and forwarding procedures as a cooperative coalitional game with transferable payoffs, and propose an incentive routing and forwarding scheme that combines the idea of payment mechanism and reputation system together [47]. The reputation system we employ is a heat diffusion model which deals with the reputation combination and propagation issues. We also analyze that the game has a non-empty *core*, which is a stable status in cooperative game just like the Nash equilibrium in a non-cooperative game.

1.5 Contributions

The main contributions of our work are as follows:

Propose a Trust Model Based Routing Protocol for Secure Ad Hoc Networks

- We introduce the idea of trust model into the design of secure routing protocols for mobile ad hoc network.
- We derive our trust model based on subjective logic which can fully represent the properties of the trust relationships in mobile ad hoc network.
- We design a trusted routing protocol (TAODV) based on our trust model, which is both security and cost effective.

Design a Coalitional Game Model for Security Issues in Wireless Networks

- We define a new security and throughput characteristic function, on the basis of which nodes are enforced to cooperate and form coalitions. The physical meaning of the throughput characteristic function is the maximal throughput and the most reliable traffic that a coalition can achieve.
- The payoff share is given by Shapley Value after proving the feasibility of this method.
- Then a set of game rules is presented to establish a threatening mechanism to all players.
- We then describe the coalition formation procedure and the integration of this game theory model with available wireless routing protocols.
- Finally, theoretical analysis is conducted to illustrate the convergence situation and justify the correctness of the formulation.

Develop an Incentive Routing Scheme for Selfishness Issues in Wireless Networks With a Coaltional Game Model Based on Heat Diffusion

- First, we design an incentive routing and forwarding scheme that integrates reputation information into a payment mechanism, which can increase the throughput as well as the security of the network.
- Second, we introduce a heat diffusion model to combine the direct and indirect reputations together and propagate them from locally to globally.
- Third, unlike others, we model this incentive scheme using a coalitional game method. A characteristic value function of the coalition is designed and we prove that this game has a core solution.

1.6 Organization

The rest of this thesis is organized as follows:

- Chapter 2

  In this chapter, we first give a survey on the trust concept and the different trust models. Then we derive our trust model based on subjective logic. Trust relationships, trust representation, trust mapping methods between evidence and opinion spaces, and trust combination algorithms are presented in detail in our trust model. Then we propose another trust evaluation model with a Baysian approach.

- Chapter 3

  The complete trusted routing protocol for wireless networks is presented in this chapter. We first describe some existing original and secure routing protocols, then propose a trusted routing protocol based on Ad hoc

On-demand Distance Vector (AODV) routing protocol. Detailed trusted routing discovery procedure including trust recommendation, trust judging, trust updating are provided. After that, we also perform trust evaluations with an enhanced subjective logic.

- Chapter 4

  This chapter formulates the security issues of wireless networks with a non-cooperative game based on signaling game. Two attacker-regular interaction game trees are given, from which we find out the threshold of a factor for the design of secure routing protocol.

- Chapter 5

  In this chapter, we formulate the security issues of wirelss networks with coalitioanl game models. A security value function and a throughput value function are given as the foundation of the coalitional game. We then analyze the game theoretically about the existence of the stable state and the speed of convergence to the stable state.

- Chapter 6

  We first give the background of our heat diffusion model, then propose our incentive routing and forwarding scheme for the selfishness issues of wireless networks. Finally we analyze the scheme using another coalitional game model, and present the evaluation results.

- Chapter 7

  Finally we summarize all of our work and discuss about the future research direction.

Chapter 2

Trust Models for Mobile Ad Hoc Networks

2.1 A Survey on Trust

2.1.1 Definition of Trust

All kinds of transactions, interactions, and communications in human life are based on one fundamental aspect: *trust*. People often take trust into account when they do everything, even if they are not aware of it. For example, an employer may give a job to a stranger after a short-time interview and that stranger may come to work on time on the second day. They both take risk and must have basic trust between each other. So do the computer networks nowadays. Ad hoc networks may contain many peer nodes. Each node is a stranger to another. These nodes also need trust before they exchange information. Before we answer "How do they trust each other?" Let's first look at the question: "What is trust?"

There are different definitions about trust in different fields and aspects, such as social psychology, sociology, and philosophy [43, 55]. In the Oxford English Dictionary, the word *trust* is defined as follows [43]:

n. 'confidence, strong belief in the goodness, strength, reliability of something or somebody', 'responsibility'.

v. have trust in - 'believe in the honesty and reliability of someone or something', 'have confidence in', 'earnestly hope'.

Trust in Psychology

In the category of psychology, one popular definition about trust is given by Morton Deutsch in [16], which is:

1. *If an individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial (Va+) or to an event perceived to be harmful (Va-)*;

2. *He perceives that the occurrence of Va+ or Va- is contingent on the behavior of another person; and*

3. *He perceives that strength of Va- to be greater than that strength of Va+.*

   If he chooses to take an ambiguous path with such properties, I shall say he makes a trusting choice; if he chooses not to take the path, he makes a distrustful choice.

Deutsch defined trust as a kind of subjective behavior. Whether an individual will take the path or not is from his own point of view. And he also needs another person to update his perceptions. Different individuals will have different viewpoints on the same thing. So "the estimate of costs ($Va-$) and benefits ($Va+$) will be different" [55]. Then cooperation is needed to judge whether an ambiguous path is beneficial or harmful. But "one should spend hours analyzing the costs and benefits of each situation in order to derive the maximum benefit from it. However, time is valuable too, and clearly the sensible approach to this problem of processing limits is to develop a scheme in

which extensive intellectual work is only done under certain circumstances" [24, 26, 55].

This definition was updated by Deutsch later in his book *The Resolution of Conflict* [17] in 1972. He "expands the definition further, and presents clarifications, eventually arriving at the definition of trust as confidence, which is confidence that one will find what is desired from another, rather than what is feared" [55]. This definition describes such a process to make a trusting choice: First, one may feel both desired and feared to the ambiguous path; then because of the existence of fear, one must take a risk before he has confidence towards the beneficial outcome. Trust eventually becomes the confidence on the beneficial path ($Va+$).

Trust in Sociology

Niklas Luhmann's Definition Niklas Luhmann's approach to trust is sociological. His main idea was that "trust is a means for reducing the complexity of society" [55]. With more and more relations and interactions in human life the complexity of our everyday world is increasing faster and faster. Luhmann suggested that "in condition of increasing social complexity man can and must develop more effective ways of reducing complexity" [50]. "What this means is that every time we face a complex or even a simple decision-making situation, we have to make some assumptions taking into account the particular situation and the particular environment and then make some trusting choice" [43].

Bernard Barber's Definition Barber is a socialist and his work is also inherently sociological which was published in *Logic and Limits of Trust* [5]. He viewed trust as an aspect of all social relationships and presents some fundamental meanings of trust as follows:

1. *Expectation of the persistence and fulfillment of the natural and moral social orders.*

2. *Expectation of "technically competent role performance" from those we interact with in social relationships and systems.*

3. *Expectation that partners in interaction will "carry out their fiduciary obligations and responsibilities, that is, their duties in certain situations to place others' interests before their own."*

For point 1, it means that "in a general sense, trust is an expectation that the natural, physical and biological order will continue to hold true." For point 2, it refers that, for example, "we trust our doctors to perform operations well, or we trust those we elect to govern the country in a sensible and efficient manner." For point 3, "that is, for those members of society who have moral obligation and responsibilities, we expect that this will be done" [55].

This idea emphasizes an inherent social order based on trust. People in this society cannot know everybody very much, thus they must make some assumptions that another entity will not use his power against them and they must trust each other [55].

Trust in Terms of Mathematics

Diego Gambetta gave the definition about trust in terms of mathematics in the article *Can We Trust Trust?* [23] in the collection of *Trust: Making and Breaking Cooperative Relations* [24]. He defined trust as a probability, whose value is in the range of 0 to 1 [55]. His definition is [23]:

> *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 and in a context in which it affects his own action.*
>
> *When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action*

> *that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him.*
>
> *Correspondingly when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so.*

This definition gives us a different viewpoint of trust. The introduction of probability provides us a mathematical model to measure trust. Trust now "becomes more concrete than abstract compared to other definitions presented earlier" [43]. "The use of values does allow us to talk succinctly and precisely about specific circumstances in trusting behavior. In addition, it allows the straightforward implementation of formalism" [55]. Diego said that "trust is better seen as a threshold point, located on a probabilistic distribution of more general expectations, which can take a number of values suspended between complete distrust (0) and complete trust (1), and which is centered around a mid-point (0.50) of uncertainty" [23].

This definition recognizes that trust is relevant only when there is a possibility of distrust, betrayal, exit or defection [43]. That is, when someone is trusted there is a chance that the action he performs may be non-beneficial to us [43].

With this definition we will have a theoretical basis to establish our own trust model.

2.1.2 Properties of Trust Relationship

Trust relationship inherits a list of characteristics and different forms from different points of view. In this section we will describe these basic properties and forms.

Relativity

Trust relationship is not absolute. That is, two entities will keep a trust relationship only in a certain category or class. One "trusts a trustee with respect to its ability to perform a specific action or provide a specific service within a context" [28]. For example, we trust our dentists when we have toothaches, but we would better not trust them when we sprain our ankles. Also because of the large amount of trust information, we can only give certain trust to some specific information. So the trust relationship contains relativity.

Because of the relativity of trust relationship, many trust models use trust categories to represent which aspect of trust they are referring to [1].

Pervasiveness

Trust relationship can be a one-to-one relation between two entities. It can also be one-to-many and many-to-one relations such as the relationship between one employer and his many employees. It can also be many-to-many such as the mutual trust between members of a group or a committee. "In general, the entities involved in a trust relationship will be distributed and may have no direct knowledge of each other so there is a need for mechanisms to support the establishment of trust relationships between distributed entities" [27].

If one takes the view that a set is a collection of one or more entities, then the trust relation can be generalized as the relation between two sets: the trustor set and the trustee set. Thus, the trust relationship can also be viewed as a binary relation, since it occurs between trustors and trustees [27].

Asymmetry

In general, trust relationship is not symmetric. One can trust another but not vice versa. That is, the trustworthiness in the reverse direction need not exist at the same time. Thus, the trust relationship can be viewed as a one-way or

unidirectional relationship.

If mutual trust exists between the same entities, some trust models such as [1] often represent them as two separate trust relationships. This allows each of these relationships to be manipulated independently.

Transitivity

The literature [69] has mentioned that trust relationships should not be transitive as many suggestions said; however, some trust scenarios do exhibit transitivity. The concept of trust delegation is a prime example of the application of trust transitivity. For example, when Alice delegates her trust decisions to Bob, she authorizes Bob to make trust decisions on her behalf. Thus, if Bob trusts an unknown entity (say Tim), Alice will trust Tim to some extend. According to Christianson and Harbison in [13], the concept of transitivity should be avoided, as it can result in entity B adding trust assertions to an entity A's trust base without A's explicit consent leading to unintentional transitivity.

In [28], the authors agreed that transitivity of trust may have unexpected and adverse results but it may be necessary in some situations. So they viewed transitivity as inherent relationship and should be considered in the analysis of trust systems in order to determine which undesired side effects should be prevented.

In the trust model of [1], Alfarez defined the trust relationship as a conditionally transitive relationship.

Measurability

One's belief is a measurable concept. To offer this capability, a trust level is often associated with a trust relationship [57]. The trust level is a measure of one's belief in another entity and thus by definition, it is a measure of one's belief in the honesty, competence and dependability of this entity. It is not a measure, however, of the *actual* competence, honesty, security or dependability

of a trustee. Some entities may be trusted more than others with respect to performing an action. It is not fixed whether the trust level should be discrete or continuous. If discrete values are used, then a qualitative label such as high, medium or low may be sufficient. Some systems support arithmetic operations on trust recommendations so numeric quantification is more appropriate. It is also possible to provide a mapping from qualitative to numeric labels.

Uncertainty

In many situations, trust is uncertain. There exists a grey zone between trust and distrust, that is, one may be ignorant or uncertain about an entity's trustworthiness if he is lack of context, experience or complete information. Some certainty mechanisms specify trust values only according to known facts and desired behaviors, but ignore the existence of uncertainty.

Jϕsang's opinion model, based on *subjective logic*, may be a suitable technique for solving this problem [36, 37, 38]. An opinion is a representation of the belief and is modelled as a triplet, consisting of b (a measure of one's belief), d (a measure of one's disbelief) and u (a measure of ignorance), such that $b+d+u = 1$. It is assumed that b, d and u are continuous and between 0 and 1 (inclusive). This model's strength lies in the ability to reason about the opinions (on a mathematically sound basis) and its consensus, recommendation and ordering operators [35].

2.1.3 Different Forms of Trust Relationship

Although the previous section summarizes the different definitions of trust, other people have treated trust differently. Sometimes they have defined trust from a different perspective and sometimes they have linked trust with something else such as cooperation and commodity. This section highlights some of the extensions of trust and how trust is related to some other things like

cooperation, commodity, etc [43].

Trust and Cooperation

Gambetta [23] related trust with cooperation in the sense that cooperation has demands on the level of trust. If trust is only unilateral, then cooperation cannot succeed. Similarly, if there is complete distrust between the involved agents, then there cannot be any cooperation between them. A higher level of trust generally leads to a higher likelihood of cooperation. It can be argued that blind trust can make cooperation work since there is no possibility of distrust; however, the important thing to note in case of blind trust is that there can be an incentive to deception.

Trust and Recommendation

Recommendation plays a significant role in trust systems. In any decent-sized society it is impossible for everyone to know and to trust everyone else. In a situation where we do not know whether to trust someone or not, we tend to ask a third person, who we know and trust. Based on the third person's recommendation we make our own decisions. Normally we consider how much we can trust the third person and how much the third person trusts our concerned target. If the third person does not know the target then he may get a recommendation from another person who knows the target and so on. Generally, the longer the recommendation chain becomes, the more difficult it is to make the trust decision, and the lower trust information we can get. There is no magical formula here; it is simply the way we perceive trust.

Trust and Commodity

Dasgupta in [15] gave another view of trust. He believed that although trust does not have any units in which it can be measured, one can still measure its

value and its worthwhileness. It is similar to commodities such as knowledge or information. Dasgupta's view of trust resembles some of the definitions described earlier. Furthermore, trust, in some sense, can be a way of dealing with the freedom of others [55]. Later in the article he concluded that trust is based on reputation and that reputation has ultimately to be acquired through behaviours over time in well-understood circumstances.

2.1.4 Trust Models

Several trust models have been published. In this section, we survey them one by one.

Trust Model Using Direct and Recommendation Trust

In [7], the authors presented a method for the valuation of trust. They indicated that the semantic of direct trust values is different from that of recommendation trust values.

This trust model was derived originally from the work of Yahalom, Klein and Beth in [82]. When doing authentication in open networks, an entity often requires other entities' recommendations. These entities can be viewed as Authentication Servers (AS). To prevent contradicting or malicious recommendations from different authentication servers, it is necessary to provide a means of estimating the trustworthiness of AS. The trust model proposed in [82] is to solve this trust estimationproblem. It introduces a formal way to represent trust relationships using trust values, and shows how to derive and combine trust values from existing ones.

There are two types of trust in this model: direct trust and recommendation trust. Direct trust means that an entity can trust another entity directly using all existing experiences it obtains about that entity. Recommendation trust expresses "the belief in the capability of an entity to decide whether another

entity is reliable in the given trust class and in its honesty when recommending third entities" [7].

Direct Trust Direct trust is defined as follows:

$$P \;\; trusts_x^{seq} \;\; Q \;\; value \;\; V \tag{2.1}$$

A direct trust relationship exists if all experiences with Q regarding trust class x, which P knows about, are positive experiences. Seq is the sequence of entities that mediate the experiences (recommendation path) excluding P and Q. V is the value of a trust relationship, which is an estimation of the probability that Q behaves well when being trusted. This is based on the number of positive experiences with Q.

Let p be the number of positive experiences with Q which P knows about with regard to the trust class x. Then the value v_z of these experiences is computed as follows:

$$v_z(p) = 1 - \alpha^p \tag{2.2}$$

This value is the probability that Q has a reliability of more than α, founded on the information P possesses about Q. The reliability is the probability that Q turns out to be reliable when being entrusted with a single task. α should be chosen reasonably high to ensure sufficiently safe estimations.

Recommendation Trust The authors of [7] defines recommendation trust like this:

$$P \;\; trusts.rec_x^{seq} \;\; Q \;\; when.path \;\; S_p \;\; when.target \;\; S_t \;\; value \;\; V \tag{2.3}$$

A recommendation trust relationship exists if P is willing to accept reports from Q about experiences with third parties with respect to trust class x. This trust is restricted to experiences with entities in S_t (the target constraint set) mediated by entities in S_p (the path constraint set). V is the value of the

trust relationship. It represents the portion of offered experiences that P is willing to accept from Q and is based on the experiences of P with the entities recommended by Q.

If p and n represent positive and negative experiences respectively with the recommended entities, the recommendation trust value Vr is computed according to the following equation:

$$V_r(p,n) = \begin{cases} 1-\alpha^{p-n} & , \quad p > n \\ 0 & , \quad else \end{cases} \tag{2.4}$$

This value can be regarded as a degree of similarity between P and Q, taking into account that different entities may have different experiences with a third party.

A recommending entity may not behave well all the time, so it is sufficient to state certain dissimilarity and to lower the trust value. This is modelled by the following properties in Eq. (2.5):

- $v_r(p,n) = 0 \quad \text{for} \quad p = 0.$
- $v_r(p,n)$ approaches 1 with growing p and fixed n.
- $v_r(p,n)$ approaches 0 with growing n and fixed p. (2.5)

If the negative experiences outnumber the positive experiences, the value becomes zero and the entity is excluded from the recommendation constraint set.

Deriving Trust Relationships The authors presented an example showing how new trust is established when a recommendation is performed. With the help of some defined rules, a new trust relationship can be derived from a given set of initial relationships. Figure 2.1 depicts the derivation of trust relationships.

Consider the trust relationship shown in the left-hand side of Fig. 2.1, where V_1 and V_3 represent recommendation trust and V_2 represents direct trust.

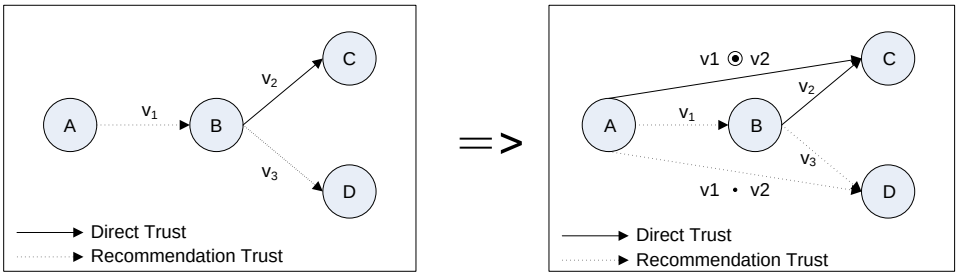


Figure 2.1: Derivation of trust relationships

Based on these existing trust relationships, new trust relationships between A and C as well as A and D can be derived. These derived trusts can be represented by Eq. (2.6).

Derived direct trust between A and C which is denoted by $V_1 \odot V_2$ is:

$$V_1 \odot V_2 = 1 - (1 - V_2)^{V_1} = 1 - (1 - (1 - \alpha^p))^{V_1} = 1 - \alpha^{V_1 \cdot p}, \tag{2.6}$$

where p is the number of positive experiences B has about C.

Derived recommendation trust between A and D which is denoted by $V_1 \bullet V_3$ is:

$$V_1 \bullet V_3 = \text{simply multiplication between } V_1 \text{ and } V_3 \tag{2.7}$$

This multiplication shows that the value of the derived recommendation trust decreases as the recommendation path grows.

The rules of inference used in the above example are defined also in [7]:

RULE1: New Direct Trust

$$\begin{aligned} & P \;\; trusts.rec_x^{seq_1} \;\; Q \;\; when.path \;\; S_p \;\; when.target \;\; S_t \;\; value \;\; V_1 \\ \wedge \;\; & Q \;\; trusts.rec_x^{seq_2} \;\; R \;\; value \;\; V_2 \\ \wedge \;\; & R \in S_t \\ \wedge \;\; & \forall X : (X \in seq_2 \Rightarrow (X \in S_p \wedge X \notin P \circ seq_1)) \\ \Rightarrow \;\; & P \;\; trusts_x^{seq_1 \circ Q \circ seq_2} \;\; R \; value \;\; (V_1 \odot V_2) \end{aligned} \tag{2.8}$$

RULE2: New Recommendation Trust

$$
\begin{aligned}
& P \;\; trusts.rec_x^{seq_1} \;\; Q \;\; when.path \;\; S_{p_1} \;\; when.target \;\; S_{t_1} \;\; value \;\; V_1 \\
\wedge \;\; & Q \;\; trusts.rec_x^{seq_2} \;\; R \;\; when.path \;\; S_{p_2} \;\; when.target \;\; S_{t_2} \;\; value \;\; V_2 \\
\wedge \;\; & \forall X : (X \in seq_2 \Rightarrow (X \in S_{p_1} \wedge X \notin P \circ seq_1)) \\
\Rightarrow \;\; & P \;\; trusts_x^{seq_1 \circ Q \circ seq_2} \;\; R \\
& when.path \;\; (S_{p_1} \cap S_{p_2}) \; when.target (S_{t_1} \cap S_{t_2}) \; value \;\; (V_1 \bullet V_2))
\end{aligned} \tag{2.9}
$$

Trust derivation algorithms are required to track down all entities which can be trusted by an entity P with respect to a trust class x. One of the trust derivation algorithms is proposed in [82], which tries all the recommendation trust expressions to derive as many new trust expressions as possible, then removes from or insert in this set the considered recommendation trust, until the set is empty. The complexity of this algorithm is exponential. Another distributed algorithm presented in [83] is employed especially for tree-like network structures. The complexity is reduced to logarithmic.

Combination of Trust Values Sometimes there are several recommendation paths so the trust relationships of the same trust class between two entities are often not unique. The trust values can then be used as collective information to compute a combined value [7].

How to combine recommendation trust is shown as follows. Given n values of recommendation trust relationships between the same entities and with respect to the same trust class $V_i (i = 1 \ldots n), V_i \neq 0$, their combination V_{com} can be computed according to Eq. (2.10):

$$V_{com} = \frac{1}{n} \sum_{i=1}^{n} V_i \tag{2.10}$$

When there are several direct trust relationships between two entities with respect to the same trust class, the combination of these trust values can be obtained by Eq. (2.11):

$$
\begin{aligned}
V_{com} &= 1 - \prod_{i=1}^{m} n_i \sqrt{\prod_{j=1}^{n_i} (1 - V_{i,j})} \\
&= 1 - \prod_{i=1}^{m} n_i \sqrt{\prod_{j=1}^{n_i} \alpha^{\bar{V}_{i,j} \cdot p_i}} \\
&= 1 - \prod_{i=1}^{m} \alpha^{\frac{1}{n_i}(\sum_{j=1}^{n_i} \bar{V}_{i,j}) \cdot p_i} \\
&= 1 - \alpha^{\sum_{i=1}^{m} \frac{1}{n_i}(\sum_{j=1}^{n_i} \bar{V}_{i,j}) \cdot p_i}
\end{aligned} \tag{2.11}
$$

Summary The trust model in [7] divides trust relationships into two types: direct trust and recommendation trust. It introduces trust values to substantiate the trust, then derives new trust values from existing ones. Different trust values can be combined together to evaluate the trustworthy of an entity.

The idea of recommendation trust was adopted by many other trust model applications which we will describe later. The recommendation trust in this trust model often comes along a path. This will be effective when the one-hop trust value has been known. But how to get the one-hop trust value is also a major problem that this trust model does not deal with.

A Distributed Trust Model with Recommendation Protocol

A distributed trust model was proposed by Alfarez et al. in [1]. Different from the trust model described above which focuses on the derivation and combination of trust values, this trust model proposes a recommendation protocol to facilitate the propagation of trust information. It extends and generalizes some current approaches to security and trust management, based upon four goals [1]:

1. To adopt a decentralized approach to trust management.

2. To generalize the notion of trust.

Table 2.1: Direct Trust Value Semantics

Value	Meaning	Description
-1	Distrust	Completely untrustworthy.
0	Ignorance	Cannot make trust-related judgement about entity.
1	Minimal	Lowest possible trust.
2	Average	Mean trustworthiness. Most entities have this trust level.
3	Good	More trustworthy than most entities.
4	Complete	Completely trust this entity.

Table 2.2: Recommendation Trust Value Semantics

Value	Meaning	Description
-1	Distrust	Completely untrustworthy.
0	Ignorance	Cannot make trust-related judgement about entity.
1	‘Closeness’ of recommender’s judgement to own judgement about trustworthiness	
2		
3		
4		

3. To reduce ambiguity by using explicit trust statements.

4. To facilitate the exchange of trust-related information via a recommendation protocol.

Trust Model Description In this trust model, trust relationship is also divided into two types: direct trust relationship and recommendation trust relationship. The model also uses trust value to represent the different levels of trustworthiness of an entity. But it adopts discrete trust levels instead of continuous values with no meaningful accuracy. The direct and recommendation trust values and their descriptions are given in Table 2.1 and Table 2.2 [1].

Figure 2.2: Example: Can Alice trust Eric the mechanic?

Recommendation Protocol There are three types of messages employed in this model: RRQ (Recommendation Request Message), Recommendation Message, and Refresh Message which is used for refreshing or revoking a recommendation. The protocol flow process in this recommendation protocol is very similar to the routing discovery process of AODV (Ad hoc On-demand Distance Vector) Routing Protocol which is used for mobile ad hoc networks. The protocol flow can be described using an example from [1] as depicted in Fig. 2.2.

To describe Fig. 2.2, let us assume that Alice (the requestor) is requesting a recommendation from Bob (the recommender) about Eric (the target). Alice is interested in Eric's reputation for servicing cars, especially VW Golfs, one of which Alice drives (trust category ="CarService"). The protocol run is as follows.

1. $Alice \rightarrow Bob : Alice, rrqA01, Eric, [Car - Service], T, 20000101$

2. $Bob \rightarrow Cathy : Bob, rrqB01, Eric, [Car - Service], T, 20000101$

3. $Cathy \rightarrow Bob : Bob, rrqB01, [Cathy],$
$[(Eric, Car - Service, 3, 20000131)], PK_{Eric}$

4. $Bob \rightarrow Alice : Alice, rrqA01, [Cathy, Bob],$
$[(Eric, Car - Service, 3, 20000131)], PK_{Eric}$

The reputations of the entities change over time so there is a need to update the reputation information in the system. To revoke or refresh the recommendations, a recommender resents the same recommendation with trust value 0. The receiver will treat this as any other 0-value recommendation.

Changing the trust value to any other value (i.e., $(-1, 1 \cdot \cdot 4)$) will refresh the recommendation.

Computing Trust The algorithm for computing trust values in this trust model is simple. The following formula is used to compute the trust value of a target for a single recommendation path:

$$tv_p(T) = tv(R_1)/4 \times tv(R_2)/4 \times \cdots \times tv(R_n)/4 \times rtv(T), \tag{2.12}$$

where

$tv(R_i)$**:** Trust value of the recommender in the return path including the first recommender (who received the original RRQ) and the last recommender (who originated the recommendation). i is from 1 to n.

$rtv(T)$**:** The recommended trust value of target T given in the recommendation.

$tv_p(T)$**:** The trust value of target T derived from recommendation received through the return path p.

A requester may have multiple recommendations for a single target and thus the recommendations must be combined to yield a single value. To this point, the averaging method used by Thomas et al. in [7] is adopted. Averaging evens out the impact of any single recommendation. The final single trust value for target T is then computed as follows:

$$tv(T) = Average(tv_i(T), \ldots, tv_p(T)). \tag{2.13}$$

Summary The main advantage of this trust model is that it proposes a recommendation protocol to formalize the propagation of trust information. The main distinctive property of this model is that the trust value here is discrete and divided into some trust levels. However, the trust value calculation

algorithm is derived largely from intuition and lacks mathematical basis. A standard algorithm is necessary to reduce ambiguity in trust value recommendations, and to allow most requesters to be confident in what is received in recommendations, which should come close to that from a universal standard. Furthermore, there is also a need to look into monitoring and revising trust of other entities, so that the dynamic and non-monotonic properties of trust in the model can be maintained.

Trust Model Based on Dempster-Shafer Theory

This model is presented in the paper [76], which proposes a method to propagate and quantify trust using principles derived from Dempster-Shafer theory of evidence. This trust model is designed mainly for e-commerce environment. There are some other trust models designed for e-commerce and Internet security, but most of them are employed for the purpose of authenticating a public key to its owner. On the other hand, this model tries to describe the scenarios where trust exists between a vendor and a customer, with several intermediaries involved in a transaction in an e-commerce setting.

Dempster-Shafer Theory Propagation of trust is a major issue when several entities are involved in e-commerce transactions. This model uses Dempster-Shafer theory to solve the trust problem because Dempster-Shafer theory of evidence is able to represent "certainty about certainty." Dempster-Shafer theory of evidence aims to model and quantity uncertainty by degrees of belief. The mathematical model proposed by Shafer [71] was based on the notion of belief functions and Dempster's rule of combination. Ginsberg proposed a procedure for uncertain reasoning using Dempster-Shafer theory in [25], which is a straightforward application of Dempster-Shafer theory.

The most important assumption made by Ginsberg is that his model applies

to dichotomous frames only, i.e., those account for two propositions. Consequently, the *belief* in proposition A can be represented by a tuple (a, b) where a measures the extent to which one believes the proposition A and b measures the disbelief, i.e., belief in the complementary proposition $\bar{A}$.

To perform combinations, Demspter-Shafer theory gives us a rule. For example, if we denote $(a, b) + (c, d)$ as the inference obtained by combining the two tuples (a, b) and (c, d), the combination formula is in Eq. (2.14):

$$(a,b)+(c,d) = (1-\frac{\bar{a}c}{1-(ad+bc)}, 1-\frac{\bar{b}d}{1-(ad+bc)}), \quad if \quad ad+bc \neq 1. \quad (2.14)$$

Trust matrix This model engages a trust matrix instead of a single trust value to represent the trust relationship between two entities. Trust matrices are maintained by certain authorities, called Trust Authorities (TA), and updated based on the information that TAs receive from each completed transaction. In this trust model, the authors define two types of trust matrix, one is trust relationship between a customer and the trust authority, and the other is between a vender and the trust authority. Figure 2.3 shows the former relationship and Fig. 2.4 shows the latter.

Given the trust matrix between a customer and TA and the matrix between TA and a vendor, the new trust matrix between the customer and the vendor can be derived by merging these two matrices using the above Dempster-Shafer formula. The newly generated trust matrix is described in Fig. 2.5 [76].

Summary This trust model uses a trust matrix instead of a single trust value to represent the trust relationship between two entities in e-commerce transactions. Two or more trust matrices are combined into one new trust matrix using Dempster-Shafer's combination rule.

It is easier to maintain trust matrices in e-commerce environment because

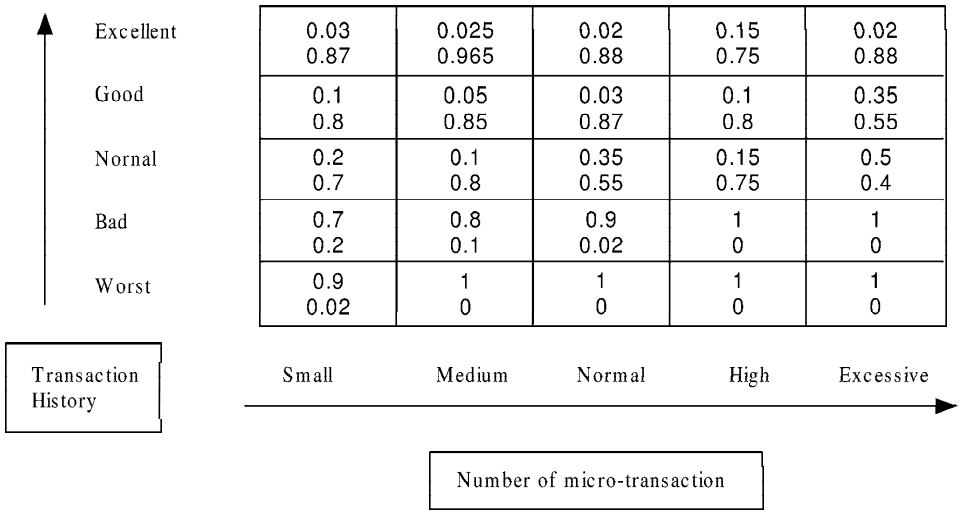


Figure 2.3: Dempster-Shafer's trust matrix between buyer and Trust Authority. Upward is belief, while downward is disbelief.

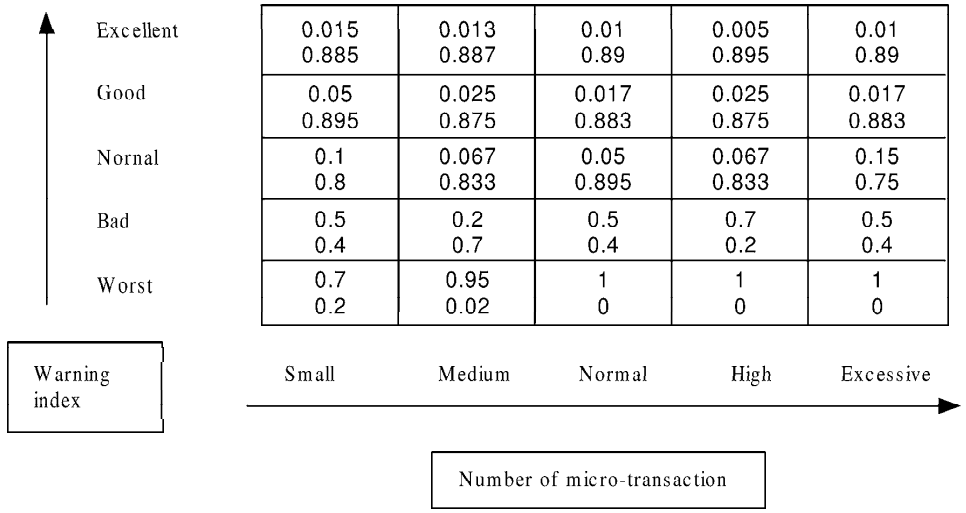


Figure 2.4: Dempster-Shafer's trust matrix between vendor and Trust Authority. Upward is belief, while downward is disbelief.

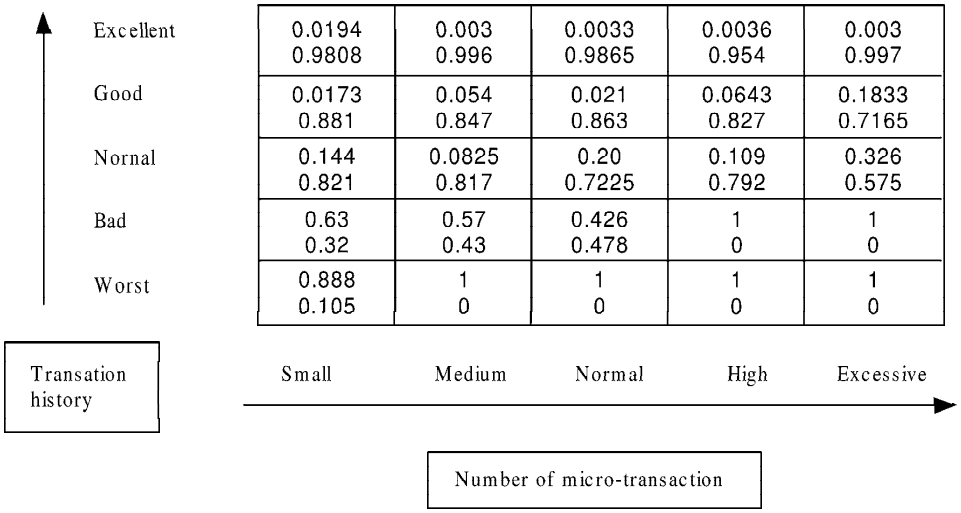


Figure 2.5: Trust matrix formed by merging the two trust matrices of Fig. 2.3 and Fig. 2.4 based on Dempster-Shafer Formula.

there are certain trust authorities which can keep the transaction history, warning index and number of micro-transactions, etc., and provide the needed information in the trust matrices in an e-commerce environment. But for ad hoc networks without centralized authorization servers, it is much complicated and not realistic to monitor and record all these information details.

Furthermore in this model the belief to a real event has been interpreted as upper and lower probability bounds, respectively, according to Dempster-Shafer theory. The two value bounds may increase the computation complexity when performing combination of trust values. However, if the trust between two entities can be viewed as probability and the combination functions can only be used to estimate probability values, there is no need to set the upper and lower bounds of one's belief.

Trust Model using Fuzzy Logic

This model is introduced in [54]. The main difference between this model and the last model is that this one applies fuzzy logic to combine the trust matrix and to verify the transactions, so as to extend trust to transaction

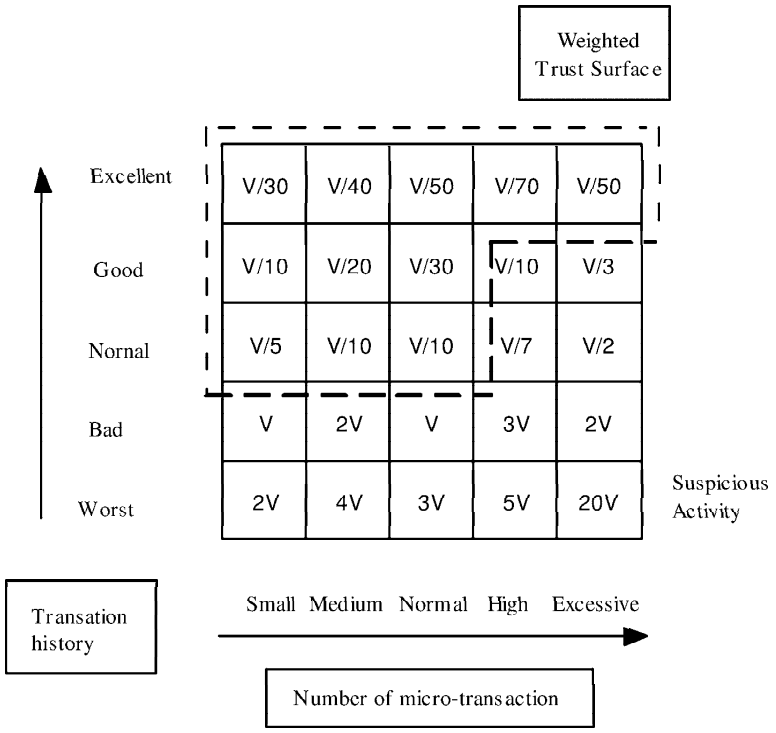


Figure 2.6: Weighted verification of transactions.

entities suitably. Furthermore, this model proposes a trust protocol which can be employed in e-commerce to protect the trust information from breach of privacy.

Trust Matrix This trust model uses Weighted Trust Surface (WTS) and Fuzzy Trust Surface (FTS) to represent the trust relationship between two entities during transactions. These two matrices are shown in the Fig. 2.6 and Fig. 2.7, respectively.

The symbol V in Fig. 2.6 and Fig. 2.7 means that the corresponding transaction should be verified. In Fig. 2.6, $V/50$ means that 1 in 50 transactions needs to be verified, and $20V$ means that the corresponding transaction may be verified more thoroughly for 20 times.

The fuzzy trust surface is then generated by replacing the numeric values by fuzzy subsets of linguistic values, shown in Fig. 2.7, which allows easy

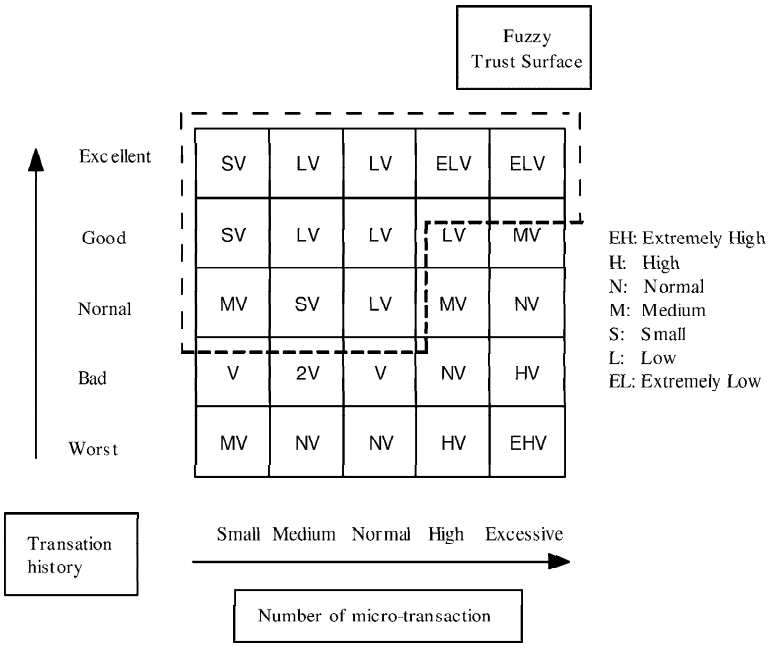


Figure 2.7: A fuzzy trust matrix.

interpretation of the matrix entities.

Fuzzy Logic Inference A set of trust matrices is obtained by engaging trust propagation techniques. Fuzzy inference can then be applied on the trust matrices to perform various actions, including verification, indemnity required, etc.

> **Definition 1 (Zadeh's Compositional Rule of Inference [12, 86])** *Let $R(x)$, $R(x,y)$ and $R(y)$ where $x \in X$, $(x,y) \in X \times Y$, and $y \in Y$ be fuzzy relations in X, $X \times Y$ and Y respectively. Let A and B denote particular fuzzy sets in X and $X \times Y$. Then the compositional rule of inference asserts that the solution of $R(x) = A$ and $R(x,y) = B$ is given by $R(y) = A \circ B$, where $A \circ B$ is the composition of A and B.* □

Application of Fuzzy Logic Inference [12] Let A and B be fuzzy sets defined over X and Y respectively. A fuzzy rule $A \rightarrow B$ is first transformed

into fuzzy relation $R_{A\rightarrow B}$ that represents a correlation between A and B and is defined as

$$\mu_R(x,y) = min(\mu_A(x), \mu_B(y)), \quad x \in X, y \in Y. \tag{2.15}$$

Given a fact A' and a rule $A \rightarrow B$, applying Zadeh's compositional rule gives

$$\begin{aligned} B' &= A' \circ R_{A\rightarrow B} \\ \mu_{B'}(y) &= max_x min(\mu_{A'}(x), \mu_R(x,y)) \\ &= min(\alpha, \mu_B(y)), \end{aligned} \tag{2.16}$$

where $\alpha = maxmin(\mu_{A'}(x), \mu_A(x))$.

Fuzzy logic inference is applied in building fuzzy expert systems to reason on trust parameters. Another example of trust systems (though not from the view point of e-commerce) using expert systems was discussed in Referee [22]. A fuzzy logic based expert system using Fuzzy CLIPS [23] (Fuzzy CLIPS is an expert system building tool) is developed to carry out the inference process.

Summary The main contributions of this trust model can be concluded as follows. Firstly, in this model, the identification and measurement of variables of trust are based on a quantifiable notion for trust. Secondly, this model engages fuzzy verification of transactions. Thirdly, the propagation of trust and the computation of a single trust matrix are performed between the customer and the vendor that govern the transaction. Another contribution is that this model proposes a suitable protocol to protect privacy of the trust information.

Trust Model using Subjective Logic

Subjective logic was proposed by Audun Jϕsang in [36, 37, 38]. Subjective logic is "a logic which operates on subjective beliefs about the world, and uses the term *opinion* to denote the representation of a subjective belief" [38].

In this model, the trust between two entities is represented by *opinion*. An opinion can be interpreted as a probability measure containing a secondary uncertainty, and can be seen as an extension of both probability calculus and binary logic.

Subjective logic is different from fuzzy logic. Fuzzy logic "operates on crisp and certain measures about linguistically vague and fuzzy propositions whereas subjective logic operates on uncertain measures about crisp propositions" [38].

The trust models introduced before mainly employ discrete values or continuous probabilities to represent trust. But the discrete values are not sufficient because they can only provide a small set of possible trust values, while the continuous values and probabilities often seem counterintuitive when applying their operators to combine trust. That is, some components such as ignorance and uncertainty which cannot be reflected by probabilities are missing when modelling trust as a probability [37].

To represent uncertain probabilities, subjective logic uses elements derived from Dempster-Shafer belief theory. Different from Dempster-Shafer's theory in which belief functions and possibility measures have been interpreted as upper and lower probability bounds, the belief functions used in subjective logic is to estimate probability values instead of setting bounds, because the probability of a real event can never be determined with certainty, and neither can upper or lower bounds be set accordingly.

Opinion Model Opinion is originally a 3-dimensional metric representing belief or trust and is extended to contain a 4^{th} redundant parameter for simple usage in combination with logical operators. The definition of opinion is as follows [38]:

> **Definition 2 (Opinion)** *Let Θ be a binary frame of discernment with two atomic states x and $\neg x$, and let m_Θ be a BMA (see Def. 3*

below) on Θ where $b(x)$, $d(x)$, $u(x)$, and $a(x)$ represent the belief, disbelief, uncertainty and relative atomicity functions on x in Θ respectively. Then the opinion about x, denoted by ω_x, is the quadruple defined by:

$$\omega_x \equiv (b(x), d(x), u(x), a(x)). \tag{2.17}$$

□

BMA is a belief mass assignment on Θ whose definition is [38]:

Definition 3 (Belief Mass Assignment) *Let Θ be a frame of discernment. If with each substate $x \in 2^{\Theta}$ a number $m_{\Theta}(x)$ is associated such that:*

$$\begin{aligned} &1. \quad m_{\Theta}(x) \geq 0 \\ &2. \quad m_{\Theta}(\phi) = 0 \\ &3. \quad \sum_{x \in 2^{\Theta}} m_{\Theta}(x) = 1 \end{aligned} \tag{2.18}$$

then $m_{\Theta}(x)$ is called a belief mass assignment on θ, or BMA for short. For each substate $x \in 2^{\Theta}$, the number $m_{\Theta}(x)$ is called the belief mass of x. □

Note $b(x)$ represents the belief function which is interpreted as an observer's total belief that a particular state is true. $d(x)$ is the disbelief function that is interpreted as the total belief that a state is not true. The uncertainty function $u(x)$ represents an observer's uncertainty regarding the truth of a given state. The sum of $b(x), d(x)$, and $u(x)$ is 1, that is:

$$b(x) + d(x) + u(x) = 1. \tag{2.19}$$

So the uncertainty function can be interpreted as something that fills the void in the absence of both belief and disbelief.

A frame of discernment with a corresponding BMA can be used to determine a probability expectation value for any given state. Uncertainty contributes to the probability expectation but will have different weight depending on the relative atomicities. The following definition is from [38].

> **Definition 4 (Probability Expectation)** *Let Θ be a frame of discernment with BMA m_Θ, then the probability expectation function corresponding with m_Θ is the function $E : 2^\Theta \mapsto [0, 1]$ defined by:*
>
> $$E(x) = \sum_y m_\Theta(y)a(x/y), \qquad y \in 2^\Theta. \tag{2.20}$$
>
> □

Combination of Opinions There are two operators for combinating opinions in this opinion model: discounting and consensus.

Discounting: Assume two entities A and B, where A has an opinion about B and B has an opinion about a proposition x. Entity A can then form an opinion about x by discounting B's opinion about x with A's opinion about B. The discounting definition is as follows [38]:

> **Definition 5 (Discounting)** *Let A and B be two agents where $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$ is A'S opinion about B's advice, and let x be a proposition where $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ is B's opinion about x expressed in an advice to A. Let $\omega_x^{AB} = (b_x^{AB}, d_x^{AB}, u_x^{AB}, a_x^{AB})$ be the opinion such that*

$$
\begin{aligned}
&1. \quad b_x^{AB} = b_B^A b_x^B \\
&2. \quad d_x^{AB} = b_B^A d_x^B \\
&3. \quad u_x^{AB} = d_B^A + u_B^A + b_B^A u_x^B \\
&4. \quad a_x^{AB} = a_x^B
\end{aligned} \tag{2.21}
$$

then ω_x^{AB} is called the discounting of ω_x^B by ω_B^A expressing A's opinion about x as a result of B's advice to A. By using the symbol '$\otimes$' to designate this operator, we define $\omega_x^{AB} \equiv \omega_B^A \otimes \omega_x^B$. □

Consensus: The consensus opinion of two opinions is an opinion that reflects both opinions in a fair and equal way. As presented in [38], the consensus is defined as follows:

Definition 6 (Consensus) *Let $\omega_x^A = (b_x^A, d_x^A, u_x^A, a_x^A)$ and $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ be opinions respectively held by agents A and B about the same proposition x. Let $\omega_x^{A,B} = (b_x^{A,B}, d_x^{A,B}, u_x^{A,B}, a_x^{A,B})$ be the opinion such that*

$$
\begin{aligned}
&1.\ b_x^{A,B} = (b_x^A u_x^B + b_x^B u_x^A)/\kappa \\
&2.\ d_x^{A,B} = (d_x^A u_x^B + d_x^B u_x^A)/\kappa \\
&3.\ u_x^{A,B} = (u_x^A u_x^B)/\kappa \\
&4.\ a_x^{A,B} = \frac{a_x^B u_x^A + a_x^A u_x^B - (a_x^A + a_x^B) u_x^A u_x^B}{u_x^A + u_x^B - 2u_x^A u_x^B}
\end{aligned} \tag{2.22}
$$

where $\kappa = u_x^A + u_x^B - 2u_x^A u_x^B$, $\kappa \neq 0$, and $a_x^{A,B} = (a_x^A + a_x^B)/2$ when $u_x^A, u_x^B = 1$. Then $\omega_x^{A,B}$ is called the consensus between ω_x^A and ω_x^B, representing an imaginary agent $[A, B]$'s opinion about x, as if it represents both A and B. By using the symbol '$\oplus$' to designate this operator, we define $\omega_x^{A,B} \equiv \omega_x^A \oplus \omega_x^B$. □

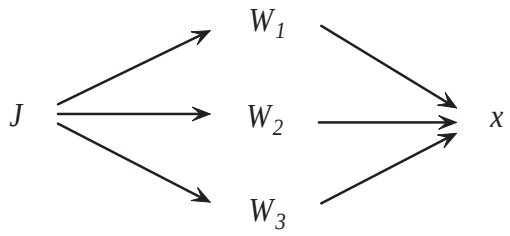


Figure 2.8: Trust in testimony from witnesses.

This model can be employed to assess the testimony from different witnesses. We take the example from [38]. There are three witnesses W_1, W_2, W_3 who are giving testimony to express their opinions about a verbal proposition x which has been made about the accused. The judge J has to determine his own opinion about x. This situation is illustrated in Fig. 2.8.

The effect of each individual testimony on the judge J can be computed using the discounting operator, so that, for example, W_1's belief in x is discounted by the judge's trust in W_1. This causes the judge to form the opinion about the truth x as a result of the testimony from W_1:

$$\omega_x^{JW_1} = \omega_{W_1}^J \otimes \omega_x^{W_1}. \tag{2.23}$$

Assuming that the opinions resulting from each witness are independent, they can finally be combined using the consensus operator to produce the judge's own opinion about x:

$$\omega_x^{J(W_1,W_2,W_3)} = (\omega_{W_1}^J \otimes \omega_x^{W_1}) \oplus (\omega_{W_2}^J \otimes \omega_x^{W_2})) \oplus (\omega_{W_3}^J \otimes \omega_x^{W_3})). \tag{2.24}$$

Summary This trust model engaging subjective logic can express the human cognitive phenomenon better than the previous trust models. It introduces the term *opinion* to represent the ignorance and uncertainty about a proposition and it is more suitable for the expression of human's subjective consciousness. The discounting and consensus operators are quite handful in the situations

that one entity needs to give its own belief about a proposition when given many different recommendations.

2.1.5 Comparison of Trust Models

The trust models discussed above have their own application fields. For the trust management in ad hoc networks, we prefer to employ the model of subjective logic because it inherits many advantages when compared with other trust models. The following is the comparison between subjective logic with the other trust models:

Subjective Logic vs. Fuzzy Logic Fuzzy Logic operates on certain measures about fuzzy propositions while subjective logic operates on uncertain measures about crisp propositions. Because of the mobility and flexibility of ad hoc networks, the nodes in the networks often do not know each other. So a node is often ignorant of the trustworthiness about another node. This kind of uncertainty measures belongs to the category of subjective logic.

Subjective Logic vs. Dempster-Shafer Theory Dempster-Shafer theory takes uncertainty and ignorance into consideration but it interprets the possibility measures as upper and lower probabilities. While in reality we usually want to just estimate probability values but not to set its bounds. The uncertainty function provided by subjective logic is a more direct way to express the uncertainty.

Opinion in Subjective Logic vs. Continuous Probability The definition about trust in [23] represents trust as a subjective probability. However, this definition misses some important components of human intuitiveness: uncertainty and ignorance. Since opinion includes belief, disbelief and also uncertainty, thus it can reflect more consciousness of human

beings. Nodes in ad hoc networks are the same that they will also have no ideas about the trustworthiness of other nodes.

Opinion in Subjective Logic vs. Discrete Trust Value Obviously the discrete trust value can only express limited information about one's belief. But the concept of dividing trust into levels or degrees may be useful when one node needs updating trust values if its opinion about another node has been changed. We will talk about this concept again when we describe the secure routing protocol which is based on our proposed trust model.

2.2 Our Trust Model Based on Subjective Logic

After performing a comprehensive survey on trust concepts and trust models, we further explore the choice of subjective logic as the basis of our trust model for ad hoc networks. In this section, we first study the characteristics of trust relationships in ad hoc networks and demonstrate that subjective logic is feasible for this application. Then we give the specific forms of our trust model.

2.2.1 Trust Relationships in Mobile Ad Hoc Networks

In ad hoc networks, a trust relationship exists between two nodes if one holds a belief about another's trustworthiness. According to the properties introduced in Section 2.1.2, the trust relationships in our proposed trust model for ad hoc networks should exhibit the following characteristics:

Relativity The trustworthiness between two nodes can be used to issue a certificate of public key and can also be employed to perform routing discovery. So trust relationships in ad hoc networks should be classified

into categories so that a node can express trust towards another about particular characteristics or functions of that node.

One-to-One Relationship In our proposed trust model, the trust relationship only exists between exactly two nodes. That means a node cannot hold one general belief about a group of nodes.

Asymmetry The trust relationship in our proposed model is non-symmetrical. That is, node A has an opinion about node B's trustworthiness while it is not necessary for B to have an opinion about A's trustworthiness. Even if B has this opinion, these two opinions do not need to be equal.

Conditional Transitivity In our trust model, the transitivity of trust relationship is conditional. For example, A, B, C are three nodes on the same routing path in an ad hoc network. A has a trust belief about B and so does B to C, then the trust belief from A to C cannot be simply passed from A to B to C. We will present a combination algorithm to combine these two beliefs into one in our trust model.

Uncertainty In ad hoc networks, nodes join and leave the network frequently. Without past experience they are uncertain about other nodes' trustworthiness. Our proposed trust model is able to expresses this property.

There are two types of trust relationships in our trust model for ad hoc networks: direct trust and recommendation trust. Direct trust can be obtained from the direct communication with other nodes in the neighborhood. It is the evaluation about other nodes' trustworthiness by the observations of itself. Recommendation trust is acquired from the combination of the recommendation opinions from other nodes.

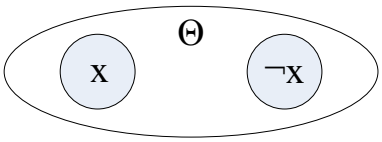


Figure 2.9: Example of a frame of discernment.

2.2.2 Trust Representation

In our trust model, we propose to represent trust on the basis of subjective logic. We also use the term *opinion* to represent the trust or belief from one node to another. Different from the original subjective logic, the frame of discernment Θ in our model only contains two states, x and $\neg x$, exactly one of which is assumed to be true at any time. Figure 2.9 illustrates such situation.

The proposition x here represents that "a node N in the network will perform normally according to the routing protocol," and $\neg x$ means the negation of x. Normally we only concern the trust relationship established on the category of the execution of routing and forwarding behaviors. Then the definition of belief mass assignment (BMA) $m_\Theta(x)$ on this frame of discernment Θ is the same as Def. 3 in Section 2.1.4. Accordingly the belief function $b(x)$, disbelief function $d(x)$, uncertainty function $u(x)$, and relative atomicity $a(x)$ are also derived from the BMA as before. Because Θ only contains exactly two contrary states, we claim that the relative atomicity $a(x)$ of each state is always equal to 1/2. Originally ω_x^M means M's opinion about the propostition x, which in our model x stands for "M's opinion about node N's behaviors on whether it will perform normally according to the routing protocol or not." To simplify the expression, we then use ω_N^M to indicate M's opinion about N's trustworthiness on its routing and forwarding behaviors. Therefore, we formally define our specific *opinion* for ad hoc networks in our model as follows.

Definition 7 (Opinion) *Let $\omega_B^A = (b_B^A, d_B^A, u_B^A)$ denote any node A's opinion about any node B's trustworthiness in ad hoc networks, where b_B^A, d_B^A, and u_B^A correspond to A's belief, disbelief, and uncertainty on B respectively. The*

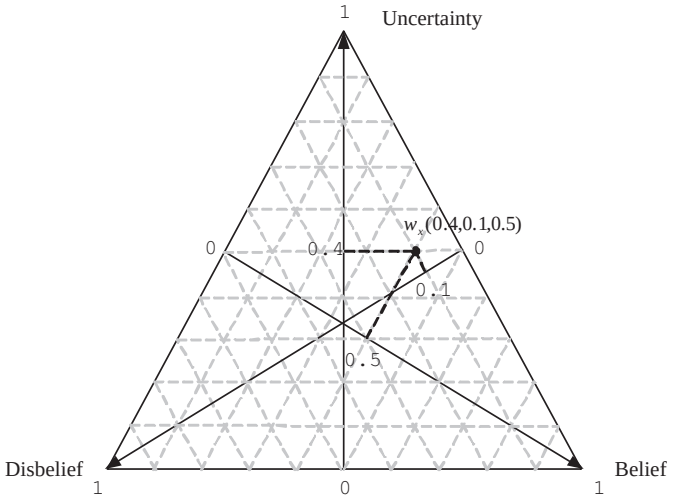


Figure 2.10: An graphical example of opinion $(0.4, 0.1, 0.5)$.

relative atomicity a_B^A *is seperated from the opinion tuple and* $a_B^A \equiv 0.5$*. Besides, the first three elements satisfy:*

$$b_B^A + d_B^A + u_B^A = 1 \tag{2.25}$$

□

In this definition, belief implies the probability of a node B can be trusted by a node A on B's behaviors of whether it will perform normally according to the routing protocol or not, and disbelief implies the probability of B not being trusted by A. Then uncertainty u_B^A fills the void in the absence of both belief and disbelief, and sum of these three elements is 1. Opinion can also be illustrated graphically using a triangle as shown in Fig. 2.10 [37].

Probability expectation can be employed to order different opinions. Based on the definition of *opinion*, we define our own probability expectation in our trust model as follows.

Definition 8 (Probability Expectation) *Let* $\omega_B^A = (b_B^A, d_B^A, u_B^A)$ *be the opinion from* A *to* B*, and* $a_B^A = 0.5$ *is the relative atomicity, then the probability expectation* $E \in [0, 1]$ *of this opinion is defined by:*

$$E(\omega_B^A) = b_B^A + a_B^A \cdot u_B^A \ . \tag{2.26}$$

□

Opinions can be ordered according to probability expectation value, but additional criteria are needed in case of equal probability expectation values to meet the specific requirements of trust model for ad hoc networks. The following definition determines the order of opinions in our trust model.

Definition 9 (Ordering of Opinions) *Let ω_B^A and ω_C^A be two opinions from A to B and to C. They can be ordered according to the following criteria by priority:*

1. *The opinion with the greatest probability expectation is the greatest opinion.*

2. *The opinion with the least uncertainty is the greatest opinion.*

□

In ad hoc networks a node's opinions about the other nodes' trustworthiness will change after they communicate with each other. So the *opinion* should be a dynamic variable. This issue was not mentioned in [38] but it is a common phenomenon in real applications. We will present a direct opinion update algorithm when we describe the design of our trusted routing protocol.

2.2.3 Trust Mapping Between Evidence and Opinion Spaces

A node in MANET will collect and record all the positive and negative evidences about other nodes' trustworthiness. With these evidences we can obtain the opinion value by applying the following mapping equation which is derived from [38].

Definition 10 (Mapping) *Let $\omega_B^A = (b_B^A, d_B^A, u_B^A)$ be node A's opinion about node B's trustworthiness in ad hoc networks, and let p and n respectively be*

the positive and negative evidences collected by node A about node B, then ω_B^A *can be expressed as a function of p and n according to:*

$$\begin{cases} b_B^A &= \frac{p}{p+n+2} \\ d_B^A &= \frac{n}{p+n+2} \\ u_B^A &= \frac{2}{p+n+2} \end{cases}, \quad where \quad u_B^A \neq 0. \tag{2.27}$$

Accordingly, provided the opinion $\omega_B^A = (b_B^A, d_B^A, u_B^A)$*, the corresponding number of positive and negative evidences p and n can be deduced as a function of* b_B^A*,* d_B^A*, and* u_B^A *as follows:*

$$\begin{cases} p &= 2b_B^A/u_B^A \\ n &= 2d_B^A/u_B^A \end{cases}, \quad where \quad u_B^A \neq 0. \tag{2.28}$$

□

2.2.4 Trust Combination

In our trust model, a node will collect all its neighbors' opinions about another node, and combine them together using combination operations. In this way, the node can make a relatively objective judgment about another node's trustworthiness even in case several nodes are lying. The followings are two combination operations nodes may adopt: *Discounting Combination* and *Consensus Combination.*

Discounting Combination

Let's consider such a situation: Node A wants to know node C's trustworthiness, and node B gives its opinion about C to A. Assuming A already has an opinion about B. Then A will combine the two opinions, A to B and B to C, to obtain a *recommendation opinion* A to C. Discounting combination is for this purpose.

Definition 11 (Discounting Combination) *Let A, B and C be three nodes where $\omega_B^A = (b_B^A, d_B^A, u_B^A)$ is A'S opinion about B's trustworthiness, and $\omega_C^B = (b_C^B, d_C^B, u_C^B)$ is B's opinion about C's trustworthiness. Let $\omega_C^{AB} = (b_C^{AB}, d_C^{AB}, u_C^{AB})$ be the opinion such that*

$$\begin{cases} b_C^{AB} &= b_A^B b_B^C \\ d_C^{AB} &= b_A^B d_B^C \\ u_C^{AB} &= d_A^B + u_A^B + b_A^B u_B^C \end{cases}, \tag{2.29}$$

ω_C^{AB} is called the discounting of ω_C^B by ω_B^A, which expresses A's opinion about C as a result of B's advice to A. By using the symbol '$\otimes$' to designate this operator, we define $\omega_C^{AB} \equiv \omega_B^A \otimes \omega_C^B$. □

The discounting combination can be employed along a recommendation path.

Consensus Combination

Different nodes may have different, or even contrary, opinions about one node. To combine these opinions together to get a relative objective evaluation about that node's trustworthiness, we may use *Consensus Combination*.

Definition 12 (Consensus Combination) *Let $\omega_C^A = (b_C^A, d_C^A, u_C^A)$ and $\omega_C^B = (b_C^B, d_C^B, u_C^B)$ be opinions respectively held by nodes A and B about node C's trustworthiness. Let $\omega_C^{A,B} = (b_C^{A,B}, d_C^{A,B}, u_C^{A,B})$ be the opinion such that*

$$\begin{cases} b_C^{A,B} &= (b_C^A u_C^B + b_C^B u_C^A)/k \\ d_C^{A,B} &= (d_C^A u_C^B + d_C^B u_C^A)/k \\ u_C^{A,B} &= (u_C^A u_C^B)/k \end{cases}, \tag{2.30}$$

where $k = u_C^A + u_C^B - 2u_C^A u_C^B$, $k \neq 0$, then $\omega_C^{A,B}$ is called the consensus between ω_C^A and ω_C^B, representing an imaginary node $[A, B]$'s opinion about C's trustworthiness, as if it represents both A and B. By using the symbol '$\oplus$' to designate this operator, we define $\omega_C^{A,B} \equiv \omega_C^A \oplus \omega_C^B$. □

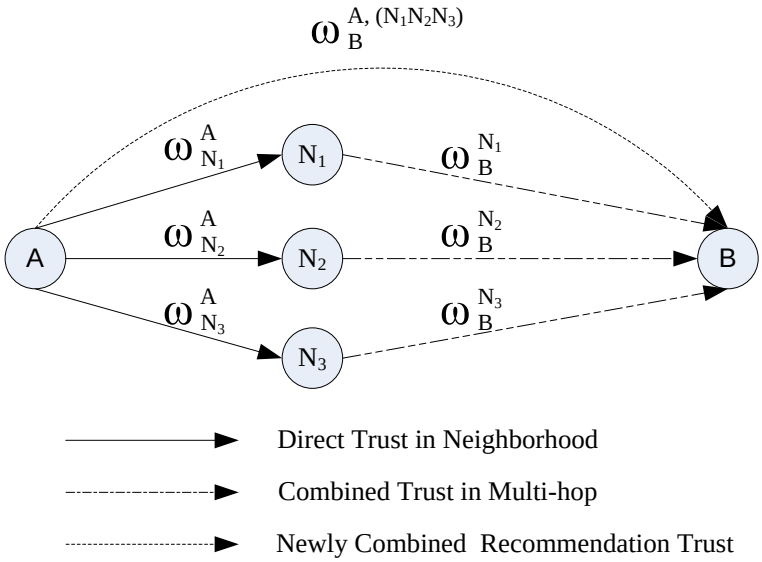


Figure 2.11: An example of trust combination.

The consensus combination can reduce the uncertainty of one's opinion. These two types of combinations will normally be employed together for a node to judge another node's trustworthiness in ad hoc network applications.

An Example of Trust Combination

Let's take an example to illustrate these two combination algorithms. Suppose that in an ad hoc network environment, A has three neighbors N_1, N_2 and N_3, and A wants to know B's trustworthiness, as shown in Fig. 2.11. Now A's opinion about B, ω_B^A should be $(0, 0, 1)$, which means total uncertainty. Assume these neighbors' opinions about B are:

$$\omega_B^{N_1} = (0.90, 0.00, 0.10)$$
$$\omega_B^{N_2} = (0.90, 0.00, 0.10)$$
$$\omega_B^{N_3} = (0.90, 0.00, 0.10)$$

Also assume A's opinions about N_1, N_2, N_3 are:

$$\omega_{N_1}^{A} = (0.90, 0.00, 0.10)$$
$$\omega_{N_2}^{A} = (0.00, 0.90, 0.10)$$
$$\omega_{N_3}^{A} = (0.10, 0.00, 0.90)$$

First we can apply discounting combination algorithm to compute the separate opinions about B, which are:

$$\omega_{B}^{A,N_1} = (0.81, 0.00, 0.19)$$
$$\omega_{B}^{A,N_2} = (0.00, 0.00, 1.00)$$
$$\omega_{B}^{A,N_3} = (0.09, 0.00, 0.91)$$

Then we can apply consensus combination algorithm to combine these new opinions again into one opinion. The result is:

$$\omega_{B}^{A,(N_1N_2N_3)} = (0.8135, 0.0000, 0.1865)$$

Therefore A will consider B as 81.35% of trustable. Moreover, the uncertainty about B is decreased from 1 to 0.1865.

2.3 Another Trust Model with a Baysian Approach and Entropy

Previously we have proposed a trust model based on subjective logic which can express *uncertainty*. In this section we will present another trust evaluation method that expresses uncertainty in another way. We give the simple model formulation and discussion, and leave the comprehensive study to the future work.

2.3.1 Entropy

Entropy is a measure of randomness, suggested by Claude E. Shannon in [72] and reprinted in [73]. Shannon defined entropy in terms of a discrete random event X, with possible states $1, \cdot\cdot, n$ as:

$$H(X) = -K \sum_{i=1}^{n} p(i) \log p(i), \tag{2.31}$$

where K is a constant corresponding to a choice of measurement units, and $p(i)$ is the probability of outcome i of event X.

That is, the entropy of the event X is the sum, over all possible outcomes i of X, of the product of the probability of outcome i and the logrithm of the probability of i (which is also called X's surprisal, and the entropy of X is the expected value of its outcomes' surprisal). We can also apply it to a general probability distribution, rather than a discrete-valued event.

$H(X)$ is not a function of X. It is a function of the probability distribution of the random variable X. The above definition of H justifies the following statement: $H(p)$ is the quantitative measure of the amount of uncertainty associated with a probability distribution p.

Entropy is one of the expressions of trust uncertainty. The greater the entropy of an event is, the greater the degree of uncertainty is.

2.3.2 A Bayesian Approach

Background

In this trust model, we propose to use a Bayesian approach for the representation and building of trust relationship as well as for subsequent decision-making depending on the trust relation. Since the true probability of a node to act maliciously, say θ, is unknown, we make an estimation of θ by inference from the data obtained by direct or indirect observations. *Bayes' Theorem* is shown

in Eq. (2.32). It is used to calculate the probability of a random variable given an observation.

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)} \quad . \tag{2.32}$$

A so-called *prior* distribution reflects the initial belief. Any up-front information can be fed into the prior to give it a head start. The prior, however, can also be chosen such that it reflects ignorance towards the initial situation. Given this prior, at each observation the information available is updated to reflect the added knowledge and to increase the precision of a belief. If the likelihood of a property is binomial, i.e., successes and failures occur independently, then a good prior density is the Beta function. The Beta function is the conjugate prior for binomial likelihood and thus the posterior density is also Beta [6]. It is defined as follows.

$$f(\theta) = Beta(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

$$\Gamma(x+1) = x\Gamma(x), \qquad \Gamma(1) = 1. \tag{2.33}$$

A binomial likelihood is assumed as $P(X) = \theta^n(1-\theta)^{1-n}$. The process of updating beliefs is as follows. First, choose a prior. To represent a non-informative prior and thus a uniform likelihood, we use $Beta(1,1)$. Then calculate the posterior distribution and update at each observation. We denote s to represent the number of successes and f for the number of failures. Then, $Beta(\alpha, \beta)' = Beta(\alpha', \beta')$ with $\alpha' = \alpha + s$ and $\beta' = \beta + f$.

The advantage of using the Beta function is that it only needs two parameters α and β that are continuously updated whenever observations are made or reported. These two parameters reflect the current belief. The higher the Beta curve is, the more the evidence samples have been taken in. If the peak in the curve is high and narrow, then we have high confidence in the belief that

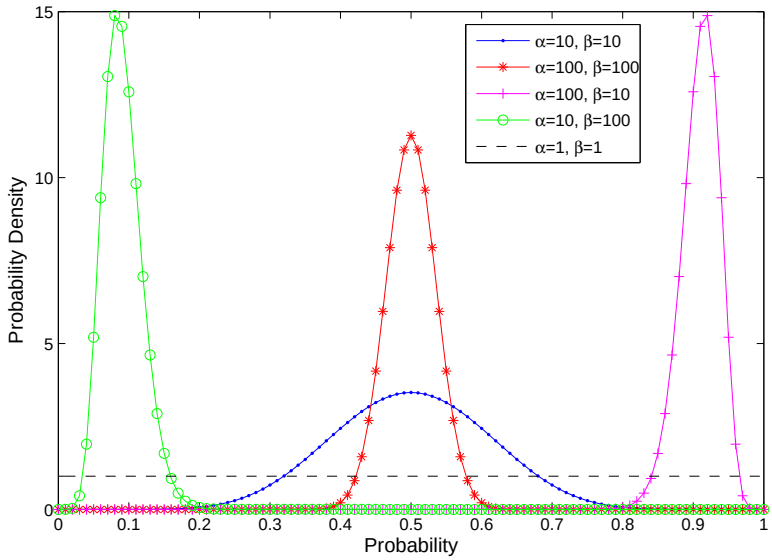


Figure 2.12: Probability density function of Beta Distribution at different α and β.

there is a certain probability around center of the observations. Figure 2.12 shows the probability density function of a beta distribution with difference parameters.

The Beta function offers moments that are simple to calculate as shown in the following.

$$E(Beta(\alpha, \beta)) = \frac{\alpha}{\alpha + \beta} \quad , \tag{2.34}$$

$$\sigma^2(Beta(\alpha, \beta)) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \quad . \tag{2.35}$$

Trust Representation

For a system with trust relationship, every node, say i, has a trust component that receives as input first-hand or second-hand behavior observations on other nodes, say j. It outputs decisions (misbehaving or not) for those j where node i forms an opinion.

We also use *opinion* ω_B^A to represent this opinion, which consists of two

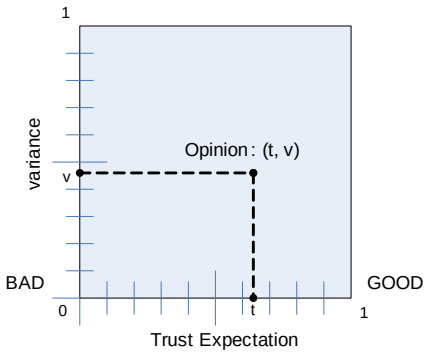


Figure 2.13: Opinion demonstration.

elements: One is the expected trust value that node A gives to node B obtained through the expected value of beta function according to A's own observations on α and β; and the other is the variance between the object trust value and the expected one. The former corresponds to the issuer's estimate of the target's trustworthiness. The variance value corresponds to the accuracy of the trust value assignment. We normalize the expected trust value and the variance in the range of $[0, 1]$. Theodorakopoulos et al. also employed two-tuples to reprensent opinion in [77], but the two elements of them are the *trust* value and the *confidence* value, respectively. We were inspired by their work and derived our own opinion form with different physical meanings. Later, we will present a graphic representation and combination method similar to their formulations. The opinion is demonstrated in Fig. 2.13.

Trust Combination

A trust model also needs to provide a trust combination method so that local direct trust values can be combined together to get the indirect trust values about the destination node through intermediate ones. Here we also denote $\otimes$ and $\oplus$ as two operators for combining opinions along a path or across multiple paths respectively (see Fig. 2.14). In [77], the authors chose two different operations $\otimes$ and $\oplus$. We will utilize their formulations and slightly modify

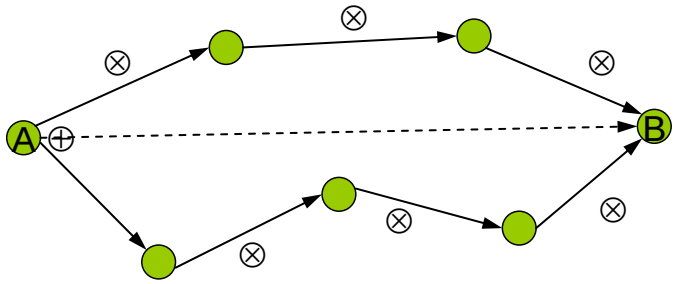


Figure 2.14: $\otimes$ Along a path and $\oplus$ accross paths.

them according to our definition, then we plan to optimize the combination values through entropy theory. The first two choices in the following are derived from [77], and the last choice is our own proposed one for future work.

Minimized Variance In this combination choice, the opinion space is $S = [0,1] \times [0,1]$. The choice for the $\otimes$ and $\oplus$ operators is as follows:

$$(t_{ik}, v_{ik}) \otimes (t_{kj}, v_{kj}) = (t_{ik}t_{kj}, v_{ik}v_{kj}), \tag{2.36}$$

$$(t_{ij}^{p1}, v_{ij}^{p1}) \oplus (t_{ij}^{p2}, v_{ij}^{p2}) = \begin{cases} (t_{ij}^{p1}, v_{ij}^{p1}), & if \quad v_{ij}^{p1} < v_{ij}^{p2} \\ (t_{ij}^{p2}, v_{ij}^{p2}), & if \quad v_{ij}^{p1} > v_{ij}^{p2} \\ (max(t_{ij}^{p1}, t_{ij}^{p2}), v_{ij}^{p1}), & if \quad v_{ij}^{p1} = v_{ij}^{p2}, \end{cases} \tag{2.37}$$

where $(t_{ij}^{p1}, v_{ij}^{p1})$ is the opinion that i has formed about j along the path $p1$.

When opinions are aggregated across multiple paths, the one with the lowest variance prevails. If two opinions have equal variance but different trust values, we pick the one with the highest trust value [77]. Fig. 2.15 captures this idea.

Parallel Resistors In this choice, the opinion space is $S = [0, \infty] \times [0,1]$. Before combination, the pair (expected trust, variance)=$(t,\ v)$ is mapped to the weight $(v/t,\ v)$. The binary operators are then applied to this weight, and the result is mapped back to a (expected trust, variance) pair. The whole

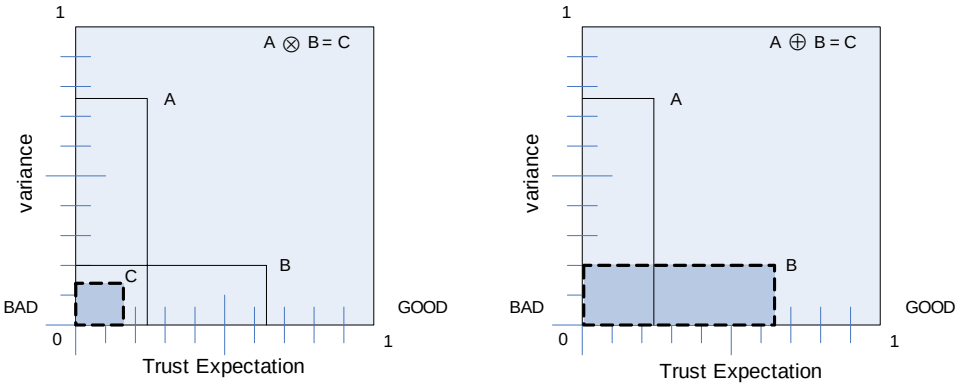


Figure 2.15: $\otimes$ and $\oplus$ operators for minimized variance combination.

process is displayed in the following equations, where arrows denote mappings, and equal signs denote actual calculations based on the operators.

$$\begin{aligned}(t_{ik}, v_{ik}) \otimes (t_{kj}, v_{kj}) &\longrightarrow (\frac{v_{ik}}{t_{ik}}, v_{ik}) \otimes (\frac{v_{kj}}{t_{kj}}, v_{kj}) \\ &= (\frac{v_{ik}v_{kj}}{t_{ik}} + \frac{v_{ik}v_{kj}}{t_{kj}}, v_{ik}v_{kj}) \\ &\longrightarrow (\frac{1}{\frac{1}{t_{ik}} + \frac{1}{t_{kj}}}, v_{ik}v_{kj}),\end{aligned} \tag{2.38}$$

$$\begin{aligned}(t_{ij}^{p1}, v_{ij}^{p1}) \oplus (t_{ij}^{p2}, v_{ij}^{p2}) &\longrightarrow (\frac{v_{ij}^{p1}}{t_{ij}^{p1}}, v_{ij}^{p1}) \oplus (\frac{v_{ij}^{p2}}{t_{ij}^{p2}}, v_{ij}^{p2}) \\ &= (\frac{v_{ij}^{p1}}{t_{ij}^{p1}} + \frac{v_{ij}^{p2}}{t_{ij}^{p2}}, v_{ij}^{p1} + v_{ij}^{p2}) \\ &\longrightarrow (\frac{v_{ij}^{p1} + v_{ij}^{p2}}{\frac{v_{ij}^{p1}}{t_{ij}^{p1}} + \frac{v_{ij}^{p2}}{t_{ij}^{p2}}}, v_{ij}^{p1} + v_{ij}^{p2}).\end{aligned} \tag{2.39}$$

As demonstrated in Fig. 2.16, when aggregating along a path, trust values are combined like parallel resistors. We can see here the effect of the mapping: Two resistors in parallel offer lower resistance than either of them in isolation. Also, a zero trust value in each opinion will result in a zero trust value in the resulting opinion, while a trust value equal to infinity will cause

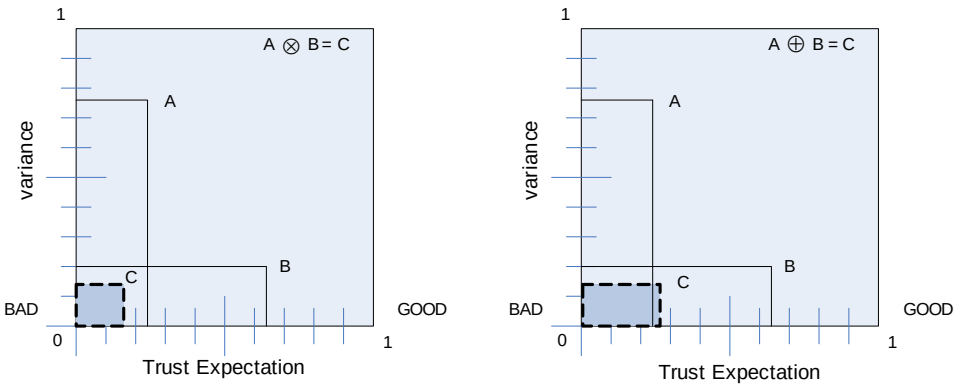


Figure 2.16: $\otimes$ and $\oplus$ operators for parallel resistors combination.

the corresponding opinion to disappear from the result. When aggregating across the paths, the total trust value is the weighted harmonic average of the components, with weights proportional to their variance values. So, the result is a value between the two component values, but closer to the one with the least variance [77].

Optimization for Trust Value with Entropy In this choice, the $\otimes$ operator has the same operation with the first choice. But when aggregating across multiple paths, we use entropy theory to achieve an optimization of trust value, so that we can get a path whose expected trust value is closest to the maximum optimization. We will leave the further study for this choice in the future work.

Demonstration of Trust Evaluation

There are usually some bad nodes in the networks. They always have the best opinion (MAX_T, MIN_V) for their neighboring bad nodes, and the worst opinion (MIN_T, MIN_V) for their neighboring good nodes. Good nodes would update their opinions for their neighbors according to some predefined rules. In the future we would like to identify those bad nodes from good ones using this trust model. For instance, suppose that one good node in the network has

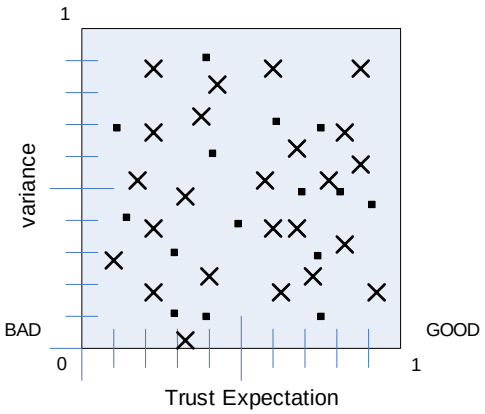


Figure 2.17: Initial opinion distribution.

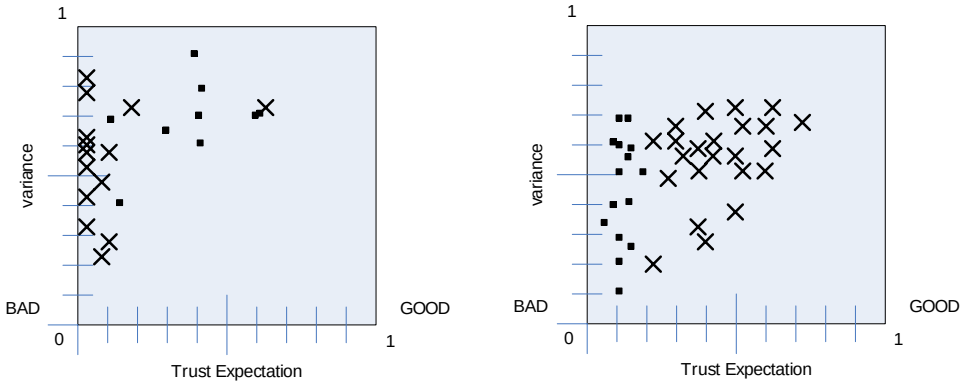


Figure 2.18: Left: Rounds m. Right: Rounds n ($m < n$). Good nodes are marked in crosses, while bad nodes are marked in squares.

an initial opinion distribution for all other nodes shown in Fig. 2.17.

We would like to obtain such a result that in the present of some percentage of bad nodes, after several rounds of trust evaluations, the good nodes and the bad nodes can be clearly seperated like those shown in Fig. 2.18.

Chapter 3

Trusted Routing Protocols for Mobile Ad Hoc Networks

3.1 Background of Routing Protocols and Key Managements

3.1.1 Non-secure Routing Protocols

Several routing protocols have been proposed for mobile ad hoc networks, such as AODV [68], DSR [34], DSDV [65] and so on. We will introduce AODV and DSR routing protocols in the following. The former is what our trusted protocol is based on.

AODV: Ad Hoc On-Demand Distance Vector Routing Protocol

Ad hoc On-demand Distance Vector (AODV) routing protocol [66, 67, 68] is one of the most popular routing protocols for Mobile Ad Hoc Networks (MANETs). On-demand is a major characteristic of AODV, which means that a node only performs routing behaviors when it wants to discover or check routing paths towards other nodes. This will greatly increase the efficiency of the routing processes. Routing discovery and routing maintenance are two basic operations in AODV protocol.

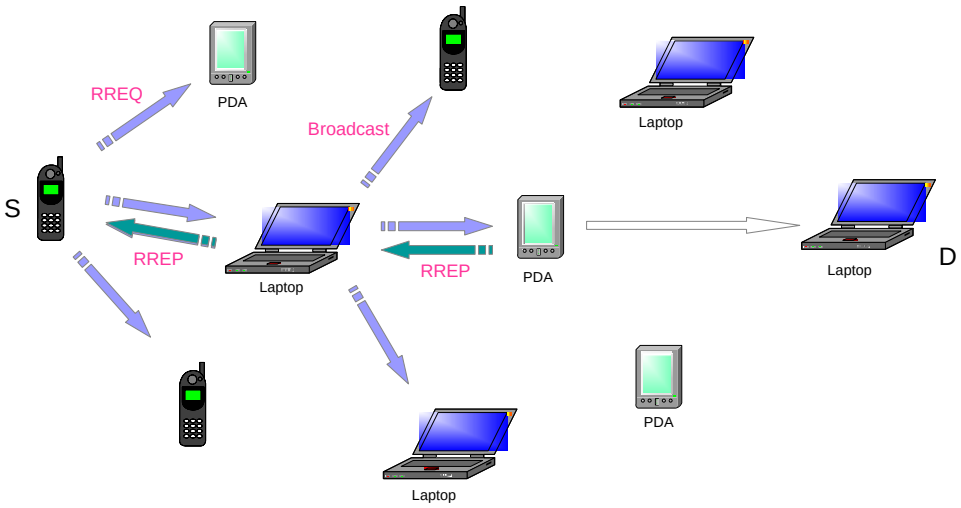


Figure 3.1: Routing discovery in AODV routing protocol.

Routing discovery happens when a node wants to communicate with a destination while it obtains no proper route entry for that destination. In this situation, this source node (originator) will broadcast an RREQ (Routing REQuest) message to all its neighbors (see Fig. 3.1). Each neighbor who receives this RREQ will check in its own routing table if it contains the route entry for that destination. If not, it will set up a reverse path towards the originator of RREQ and rebroadcast this routing request. Any node which receives this RREQ will generate an RREP (Routing REPly) message if it either has a fresh enough route to satisfy the request or is itself the destination. Then this intermediate or destination node will unicast the RREP message to the next hop toward the originator of the RREQ, as indicated by the routing entry for that originator. When a node receives an RREP message, it first updates some fields of its routing table and the routing reply message, and then forwards it to the next hop towards the originator. In this way, this RREP will ultimately reach the source node and a bidirectional routing path will be established between the source and destination. Thus, these two ends can communicate with each other through the routing path just set up.

Routing maintenance is performed through two ways. One is that a node

may positively offer connectivity information by broadcasting hello messages locally so that its neighbors can determine the connectivity by listening to the hello packets. The other way is that a node can maintain local connectivity to its next hops using some link or network layer mechanisms, such as the detection mechanism of IEEE802.11 MAC (Media Access Control) protocol.

Our secure routing protocol is based on AODV and is called TAODV (Trusted AODV), which concerns trust information when performing routing discovery and routing maintenance. We will talk about it in later sections.

DSR: Dynamic Source Routing Protocol

Dynamic Source Routing protocol (DSR) [33, 34] is also an on-demand ad hoc network routing protocol composed of two parts: Route Discovery and Route Maintenance.

In DSR, when a node has a packet to send to some destination and does not currently have a route to that destination in its Route Cache, the node initiates Route Discovery to find a route; this node is known as the *initiator* of the Route Discovery, and the destination of the packet is known as the Discovery's *target*. The initiator transmits a ROUTE REQUEST packet as a local broadcast, specifying the target and a unique identifier from the initiator. For each node receiving the ROUTE REQUEST, if it has recently seen this request identifier from the initiator, discards the REQUEST; otherwise, it appends its own node address to a list in the REQUEST and rebroadcasts the REQUEST. When the ROUTE REQUEST reaches its target node, the target sends a ROUTE REPLY back to the initiator of the REQUEST, including a copy of the accumulated list of addresses from the REQUEST. When the REPLY reaches the initiator of the REQUEST, it caches the new route in its Route Cache.

Route maintenance is the mechanism by which a node sending a packet along a specified route to some destination detects if that route has broken,

for example because two nodes in the route have moved too far apart. DSR is based on source routing: When sending a packet, the originator lists in the header of the packet the complete sequence of nodes through which the packet is to be forwarded. Each node along the route forwards the packet to the next hop indicated in the packet's header, and attempts to confirm that the packet was received by that next node; a node may confirm this by means of a link layer acknowledgment. If, after a limited number of local retransmissions of the packet, a node in the route is unable to make this confirmation, it returns a ROUTE ERROR to the original source of the packet, identifying the link from itself to the next node as broken. The sender then removes this broken link from its Route Cache; for subsequent packets to this destination, the sender may use any other route to that destination in its Cache, or it may attempt a new Route Discovery for that target if necessary.

Some secure routing protocols have been proposed based on DSR routing protocol, such as Ariadne [31]. We will introduce this secure routing protocol in the following section.

3.1.2 Secure Routing Protocols

Although the existing routing protocols are effective and efficient in routing processes, they are designed without security consideration. The following sections are several secure routing protocols proposed to improve the security of the original routing protocols.

SAODV: Secure AODV Routing Protocol

Secure AODV (SAODV) proposed by M. G. Zapata and N. Asokan in [87] is based on AODV routing protocol. Two mechanisms are used to secure the AODV messages: hash chains to secure the hop count information which is the only mutable information in the messages; and digital signatures to

authenticate the non-mutable fields of the messages. The information relative to the hash chains and the signatures is transmitted with the AODV message as an extension message.

SAODV employs hash chains to authenticate the hop count of RREQ and RREP messages in such a way that it allows every intermediate or destination node that receives the message to verify that the hop count has not been decremented by an attacker. A hash chain is formed by applying a one-way hash function repeatedly to a seed.

Digital signatures are used in SAODV to protect the integrity of the non-mutable data in RREQ and RREP messages, which means they sign everything but the Hop_Count of the AODV message and the Hash from the SAODV extension. The main problem in applying digital signatures is that an RREP message generated by an intermediate node should be able to sign it on behalf of the final destination. SAODV offers Double Signature Extension to solve this problem, which is that every time a node generates an RREQ message, it also includes the RREP flags, the prefix size and the signature of RREP.

When a node receives an RREQ, it first verifies the signature. Only if the signature is verified, will it store the route. If the RREQ has a Double Signature Extension, the node will also store the signature for the RREP and the lifetime in the route entry. An intermediate node will reply to an RREQ with an RREP only if it fulfills the AODV's requirements to do so and the node has the corresponding signature and old lifetime to put into the Signature and Old Lifetime fields of the RREP Double Signature Extension. Otherwise, it will rebroadcast the RREQ. When an RREQ is received by the destination itself, it will reply with an RREP only if it fulfills the AODV's requirements to do so. This RREP will be sent with an RREP Single Signature Extension.

When a node receives an RREP, it first verifies the signature before creating or updating a route to that host. Only if the signature is verified, will it store the route with the signature of the RREP and the lifetime.

SAODV can prevent several attacks commonly performed to AODV routing protocol. However, SAODV's signatures require a processing power that might be excessive for certain kinds of ad hoc scenarios.

Ariadne

Ariadne [31] is an on-demand secure routing protocol based on DSR which withstands node compromise and relies on symmetric cryptography. Ariadne protocol is designed in three stages. First it enables the target to verify the authenticity of the ROUTE REQUEST message by using a Message Authentication Code (MAC) with a key shared between the initiator and the target; then each intermediate node can employ three alternative techniques, the TESLA protocol, digital signatures, and standard MACs, to perform data (node list) authentication in ROUTE REQUEST and ROUTE REPLY; and finally it presents an per-hop hashing mechanism to guarantee that no node can be removed from the node list in the REQUEST.

TESLA is a broadcast authentication protocol for authenticating routing messages. It adds a MAC computed with a shared key to a message, which can provide secure authentication in point-to-point communications. TESLA achieves asymmetry from clock synchronization and delayed key disclosure. When Ariadne performs Route Discovery using TESLA broadcast authentication, it assumes that every node has a TESLA one-way key chain.

A ROUTE REQUEST in Ariadne extend original DSR Route Request to eight fields: *ROUTE REQUEST, initiator, target, id, time interval, hash chain, node list*, and *MAC list*. The *time interval* is the TESLA time interval at the pessimistic expected arrival time to the target, accounting for clock skew.

When any node A receives a ROUTE REQUEST for which it is not the target, besides checking the receive repetition, the node checks whether the *time interval* is valid in that it must not be too far in the future and the

key corresponding to it must not have been disclosed yet. If the time interval is not valid, the node discards the packet; otherwise, the node modifies the REQUEST by appending its own address, A, to the node list in the REQUEST, replaces the hash chain field with $H[A, hash\ chain]$, and appends a MAC of the entire REQUEST to the MAC list. The node uses the TESLA key K_{Ai} to compute the MAC, where i is the index for the specified time interval in the REQUEST. Finally, the node rebroadcasts the modified REQUEST, as in DSR.

When the target node receives the ROUTE REQUEST, it checks the validity of the REQUEST by determining that the keys from the time interval specified have not been disclosed yet, and that the hash chain field is correct. If the target node determines that the REQUEST is valid, it returns a ROUTE REPLY to the initiator, containing eight fields: *ROUTE REPLY, target, initiator, time interval, node list, MAC list, target MAC*, and *key list*. The *target MAC* is set to a MAC computed on the preceding fields in the REPLY with the key K_{DS}. The ROUTE REPLY is then returned to the initiator of the REQUEST along the source route.

A node forwarding a ROUTE REPLY waits until it is able to disclose its key from the time interval specified; it then appends its key from that time interval to the *key list* field in the REPLY and forwards the packet according to the source route indicated in the packet.

When the initiator receives a ROUTE REPLY, it verifies that each key in the key list is valid, that the *target MAC* is valid, and that each MAC in the *MAC list* is valid. If all of these tests succeed, the node accepts the ROUTE REPLY; otherwise, the node discards it.

Ariadne is efficient because it uses only symmetric cryptographic primitives. But it requires clock synchronization to achieve "asymmetry", which is arguably an unrealistic requirement for ad hoc networks.

3.1.3 Key Managements for Secure Ad Hoc Networks

Other than implementing security in the network layer of MANET, some key management schemes have been proposed above the network layer to provide cryptography solutions to MANETs. Traditional key management solutions commonly employ a trusted third-party or centralized servers, which violate the nature of MANETs. Recently researchers have proposed several self-organized or semi-self-organized key managements schemes. We will introduce two of them in the following.

Self-Organized Public-Key Management for MANET

The work in [11] proposes a fully self-organized public-key management scheme that does not rely on trusted authority or fixed server. Each user in this mechanism is her own authority domain and issues public key certificates to other users. Each user also keeps a local certificate repository containing a subset of certificates issued by herself and certificates selected according to an appropriate algorithm issued by other users. Key authentication is performed via a chain of certificates. When user u wants to verify the authenticity of the public key of user v, they both merge their local certificate repositories, and u tries to find an appropriate certificate chain from u to v in the merged repository.

In order to defend attacks by dishonest users, the authors in [11] extended their scheme with authentication metrics. A set of criteria are proposed for the design of the local repository construction algorithms, based on which they consider a tradeoff between the size of the local repositories of the users and the communication load/key usage.

This public-key management system is realized in a fully self-organized, yet scalable way. However, it only provides probabilistic guarantees.

Providing Robust and Ubiquitous Security Support for MANET

The work in [42] describes a solution that supports ubiquitous security services for mobile hosts, scales to network size, and is robust against break-ins. It distributes the certification authority functions through a threshold secret sharing mechanism, in which each entity holds a secret share and multiple entities in a local neighborhood jointly provide complete services. Each secret share is updated periodically to resist gradual break-ins.

The system is based on public key infrastructure and the system Certification Authority (CA) has a key pair $\{SK, PK\}$, where SK, Secret Key, is shared among the network entities and PK, Public Key, is well-known to the whole system. Each entity v_i holds a secret share P_{v_i}, and any K of such secret share holders can collectively function as the role of CA. SK is not visible by any component of the network except at the system bootstrapping phase. Each secret share P_{v_i} can be obtained during system bootstrapping phase or through a self-initialization service. A self-initialization algorithm is devised to securely deliver the secret share to an uninitialized entity by a local coalition of K secret share holders.

When an entity requests for certification service, a local coalition of K secret share holders provides to the requester a partial certificate that is signed by a value SK_i which is directly derived from the secret share P_{v_i}. Once the requester locally collects K such partial certificates, it combines them together and obtains its complete certificate that is signed by SK.

In this system, no adversary group having less than K collaborative adversaries can forge a valid certificate so that it can tolerate up to $K-1$ break-ins from each adversary group. It thus has K-out-of-N security.

3.2 Overview of Our Trusted AODV Routing Protocol (TAODV)

3.2.1 Network Model and Assumptions

In this work, we make some assumptions and establish the network model of TAODV. We also argue why we focus our security solution on routing protocol in the network layer instead of link layer.

Mobile nodes in MANETs often communicate with one another through an error-prone, bandwidth-limited, and insecure wireless channel. We do not concern the security problem introduced by the instability of physical layer or link layer. We assume that:

1. Each node in the network has the ability to recover all of its neighbors;
2. Each node in the network can broadcast some essential messages to its neighbors with high reliability;
3. Each node in the network possesses a unique ID that can be distinguished from others.

In TAODV, we also assume that the system is equipped with some monitor mechanisms or intrusion detection units either in the network layer or the application layer so that one node can observe the behaviors of its one-hop neighbors. These mechanisms have been proposed in some previous work, such as watchdog technique in [56] and intrusion detection system in [88].

Another kind of secure routing protocol which uses cryptography technologies is recommended to take effect before nodes in TAODV establish trust relationships among one another. [31] and [87] are the latest security schemes for securing MANET, which employ cryptography technologies. We assume that the keys and certificates needed by these cryptographic technologies have

been obtained through some key management procedures before a node performs routing behaviors.

In the network layer, a new node model is designed as the basis of our trust model. Some new fields are added into a node's routing table to store its opinion about other nodes' trustworthiness and to record the positive and negative evidences when it performs routing procedures with others. By embedding our trust model into the routing layer of MANET, we can save the consuming time without the trouble of maintaining the expire time, valid state, etc., which is important in the situation of high node mobility and invalidity. Also because of this reason, it is hard to design secure solutions in the transport layer, which is an end-to-end communication mechanism.

3.2.2 Framework of TAODV

There are mainly three modules in our whole TAODV system: basic AODV routing protocol, a trust model, and trusted AODV routing protocol. Based on our trust model, the TAODV routing protocol contains such procedures as trust recommendation, trust combination, trust judging, cryptographic routing behaviors, trusted routing behaviors, and trust updating. The structure and relationship among these components are shown in Fig. 3.2. The general procedure for establishing trust relationships among nodes and for performing routing discovery is described as follows.

Let us first imagine the beginning of an ad hoc network which contains a few nodes. Each node's opinion towards one another initially is $(0, 0, 1)$ which means total uncertainty. Suppose node A wants to discover a routing path to B. Because the uncertainty element in A's opinion towards others is larger than or equal to 0.5, which means that A is not sure whether it should believe or disbelieve any other nodes, A will use the cryptographic schemes as proposed in SAODV [87] or some other schemes to perform routing discovery

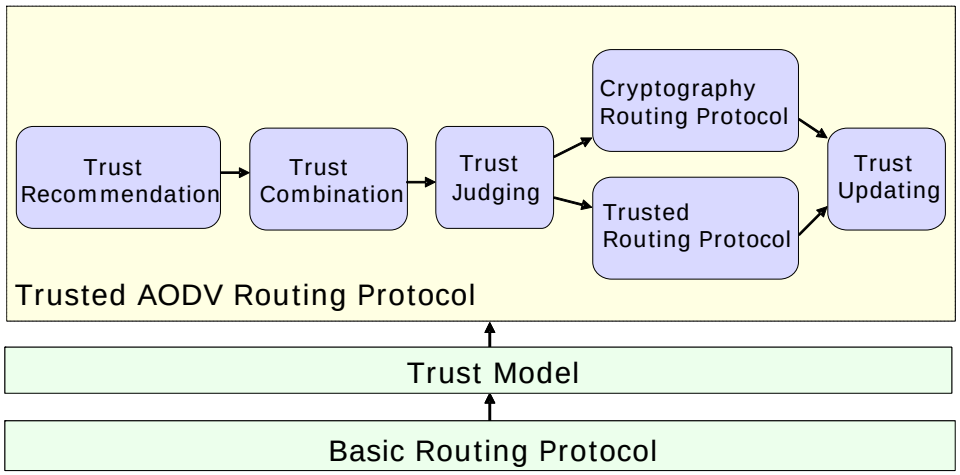


Figure 3.2: Framework of Trusted AODV (TAODV).

operations. After some successful or failed communications, A will change its opinions about other nodes gradually using the trust updating algorithm. The uncertainty elements in its opinions about other nodes will be mostly less than 0.5 after a period of time. By means of this procedure, eventually each node in the network will form more certain opinions towards other nodes eventually after the initial time period.

Once the trust relationship is established among most of the nodes in the network, these nodes can rely on our trusted routing protocol which is based our trust model to perform routing operations. Node A now will utilize the trust recommendation protocol to exchange trust information about a node, B, from its neighbors, then use the trust combination algorithm to combine all the recommendation opinions together and calculate a new option towards B. The subsequent routing discovery and maintenance operations will follow the specifications of our trusted routing protocol, which will be described in detail in Section 3.3.6. Note that the situation that one node first joins a MANET can be handled in the same way as the beginning of the whole network.

In this framework, the establishment of trust relationships among nodes and the discovery of routing paths are all performed in a self-organized way,

DestinationIP	DestinationSeq	· · ·	HopCount	· · ·	Lifetime	Positive Events	Negative Events	Opinion

Figure 3.3: Modified routing table with trust information.

which is achieved by the cooperation of different nodes to exchange information and to obtain agreements without any third-party's interventions.

3.3 Trusted Routing Operations in TAODV

3.3.1 Routing Table Extensions

We add three new fields into each node's original routing table: *positive events*, *negative events* and *opinion*. *Positive events* are the successful communication times between two nodes. Similarly *negative events* are the failed communication ones. *Opinion* means this node's belief towards another node's trustworthiness as defined before. The value of opinion can be calculated according to the mapping functions defined in Eq. (2.27) of Section 2.2.3. These three fields are the main factors when performing trusted routing. One node's routing table can be illustrated by Fig. 3.3, where some fields are omitted for highlighting the main parts. A node which has interactive behaviors with other nodes will have entries for those nodes in this node's routing table with the routing state set properly.

3.3.2 Routing Message Extensions

We extend the original AODV routing messages by appending some trust information fields. Two main types of extended messages are TRREQ (Trusted Routing REQuest) and TRREP (Trusted Routing REPly). Figure 3.4 and Fig. 3.5 show the formats of these messages.

In trusted routing discovery procedures, every routing request and reply carries trust information, including opinions towards originator node S and

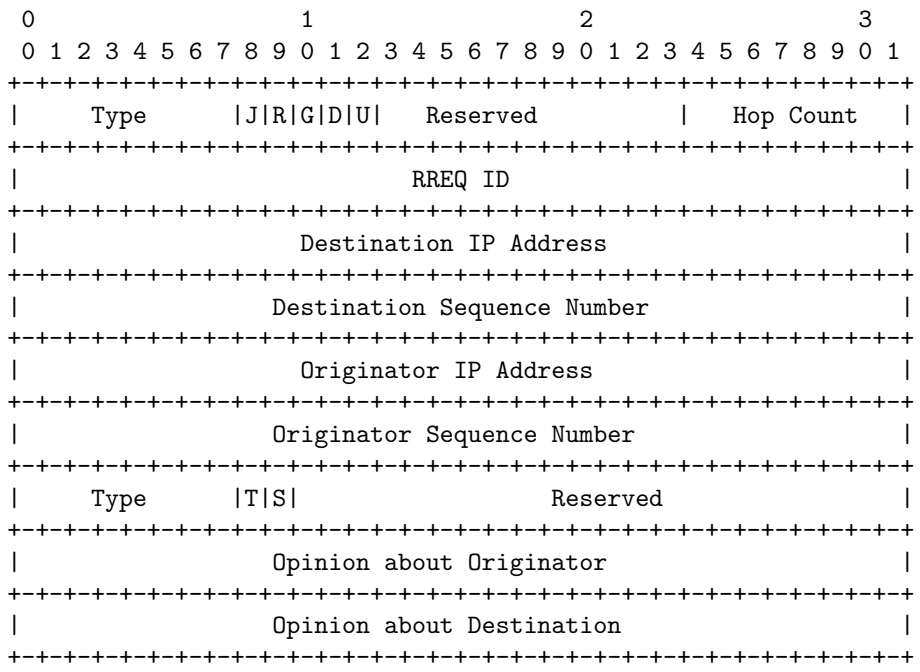


Figure 3.4: Trusted Routing Request (TRREQ) message format.

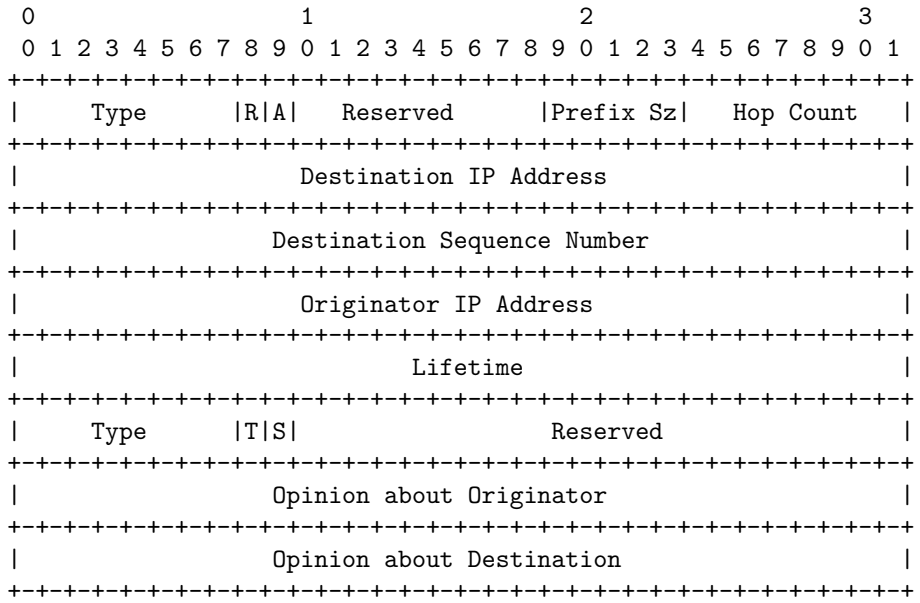


Figure 3.5: Trusted Routing Reply (TRREP) message format.

destination node D, which will be employed to calculate the credibility of S and D. When a node is required to provide its certificate information, it will fill the fields of trust information with its own signature, as proposed by some traditional security solutions for MANETs.

3.3.3 Trust Judging Rules

Before describing the process of trusted routing discovery and maintenance in detail, we define some trust judging rules here and in Table 3.1.

1. In node A's opinion towards node B's trustworthiness, if the first component *belief* of opinion ω_B^A is larger than 0.5, A will trust B and continue to perform routing related to B.

2. In node A's opinion towards node B's trustworthiness, if the second component *disbelief* of opinion ω_B^A is larger than 0.5, A will not trust B and will refuse to perform routing related to B. Accordingly the route entry for B in A's routing table will be disabled and deleted after an expire time.

3. In node A's opinion towards node B's trustworthiness, if the third component *uncertainty* of opinion ω_B^A is larger than 0.5, A will request B's digital signature whenever A has interaction (or relationship) with B.

4. In node A's opinion towards node B's trustworthiness, if the three components of opinion ω_B^A are all smaller than or equal to 0.5, A will request B's digital signature whenever A has interaction (or relationship) with B.

5. If node B has no route entry in node A's routing table, A's opinion about B is initialized as $(0, 0, 1)$.

Table 3.1: Trust Judging Rules

belief	disbelief	uncertainty	Actions
		> 0.5	Request and verify digital signature
	> 0.5		Distrust a node for an expire time
> 0.5			Trust a node and continue routing
≤ 0.5	≤ 0.5	≤ 0.5	Request and verify digital signature

3.3.4 Trust Updating Policies

Opinions among nodes change dynamically with the increase of successful or failed communication times. When and how to update trust opinions among nodes will follow some policies, which are derived as follows:

1. Each time a positive event occurs from node A to node B, B's number of successful events in A's routing table will be increased by 1.

2. Each time a negative event occurs from node A to node B, B's number of failed events in A's routing table will be increased by 1.

3. Each time when the field of the successful or failed events changes, the corresponding value of opinion will be recalculated using Eq. (2.27) from the evidence space to the opinion space.

4. Each time when the new opinion has been obtained through combination, the corresponding number of successful or failed events will be mapped back using Eq. (2.28) from the opinion space to the evidence space.

5. The positive events includes successful data or routing packets forwarding, keeping message integrity, passing cryptographic verification, and so on.

6. The negative events includes not forwarding, message forging, long delay, not passing cryptographic verification, long delay time, link instability, and so on.

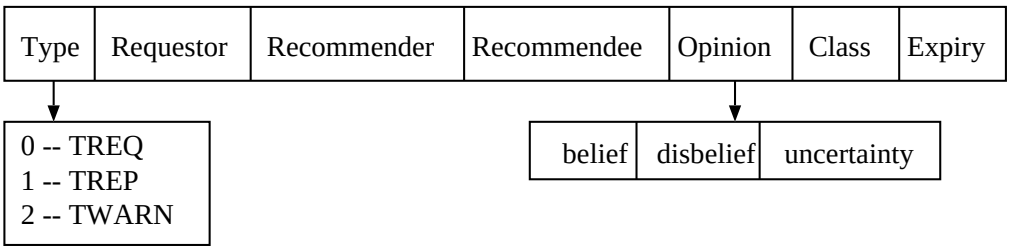


Figure 3.6: Message structure of trust recommendation protocol.

7. If node B's route entry has been deleted from node A's routing table because of expiry, or there is no B's routing entry from the beginning, the opinion ω_B^A will be set to $(0, 0, 1)$.

3.3.5 Trust Recommendation Protocol

Existing trust models seldom concern the exchange of trust information. However, it is necessary to design an information exchange mechanism when applying the trust models to network applications. In our trust recommendation protocol, there are three types of messages: Trust Request Message (TREQ), Trust Reply Message (TREP), and Trust Warning Message (TWARN). Nodes who issue TREQ messages are called *Requestor*. Those who reply TREP messages are called *Recommender*. The recommendation target nodes are called *Recommendee*. Any node may be a *Requestor*, a *Recommender*, or a *Recommendee*. These three types of messages share a common message structure, which is shown in Fig. 3.6. The exact formats of TREQ and TREP are shown in Fig. 3.7 and 3.8.

When a node A wants to know another node B's latest trustworthiness, it will broadcast a TREQ message to its neighbors. This TREQ message follows the above format and leaves the fields of *Recommender*, *Opinion* and *Expiry* empty. The *Type* field is set to 0, and the *Recommendee* field is filled with the IP address of node B. If one of A's neighbors C receives the TREQ message, C will reply with a TREP message. The *Type* field of this TREP is set to 1

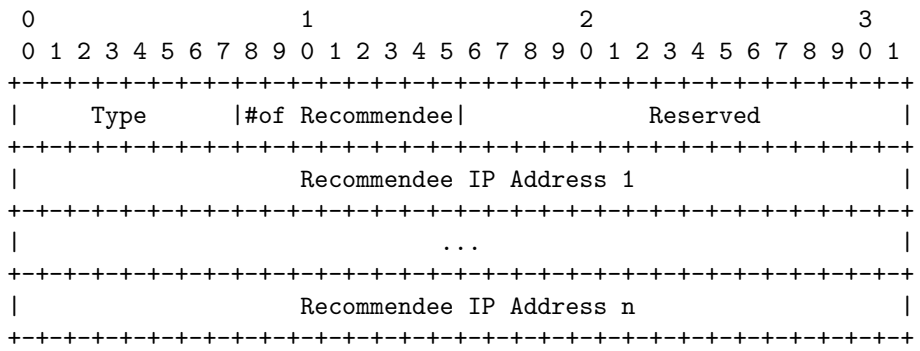


Figure 3.7: Trust Request (TREQ) message format.

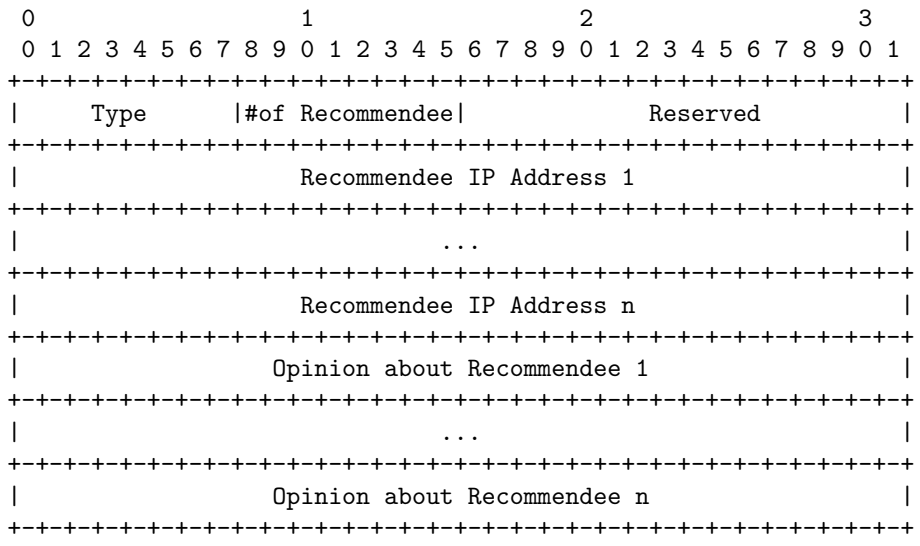


Figure 3.8: Trust Reply (TREP) message format.

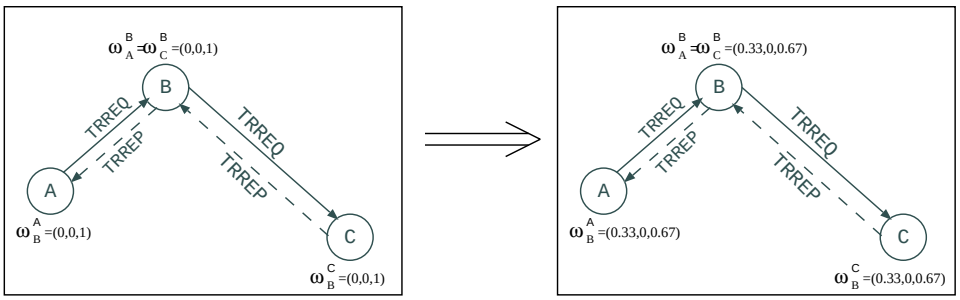


Figure 3.9: Initialization for TAODV.

and the *Opinion* field is filled with the opinion values from C to B.

When node B cannot pass the cryptography verification of node A, its opinion from the viewpoint of A will be set to $(0, 1, 0)$, which means total disbelief. B will broadcast a TWARN message with the *Type* set to 2 to its neighbors. Every node which receives this message will verify B's trustworthiness then perform corresponding opinion combination and updating. Note that, in this recommendation protocol, a node can request or reply several opinion values of different nodes simultaneously in one TREQ or TREP message. In this way, we can efficiently exchange trust information without introducing much packets overhead.

3.3.6 Trusted Routing Discovery

In this section we take AODV for example to illustrate how to perform trusted routing discovery based on our trust operations.

Scenario I: Beginning of a TAODV MANET

Let us first consider a simple MANET which only contains three nodes: A, B, and C. The topology of this minimal MANET is shown in Fig. 3.9.

In this figure, node A has only one neighbor B, node B has two neighbors A and C, and node C also has one neighbor B. Node A and B are not in neighborhood. At the beginning, there is no entry in each node's routing

table, and as described in Section 3.2.2, the initial value of each node's opinion towards one another is $(0, 0, 1)$.

Now suppose node A wants to discover a routing path to node C. The processes of node A, B, and C are listed below.

1. A broadcasts a TRREQ requesting routing path to C, then begins waiting for a TRREP from its neighbor B.

2. After B receives the TRREQ from A, then:

   (a) B checks the route to C and opinion ω_A^B and ω_C^B. Because it is the very beginning of this MANET, there should be no route to C and $\omega_A^B = \omega_C^B = (0, 0, 1)$.

   (b) B authenticates A because $u_A^B > 0.5$. B requests A's certificate and verifies it. If A passes, the number of successful events is increased by 1, and the new opinion $\omega_A^B = (0.33, 0, 0.67)$. B will then authenticate C following the previous steps. If A cannot pass, B will broadcast a TWARN message, so that the successful events will be cleared and the failed events will get a penalty number. The choice of this number will be discussed later. B will not re-broadcast the TRREQ, but deny A for an expiry time.

   (c) If C has also been authenticated, B's routing table will be updated and B will re-broadcast the TRREQ. Opinion ω_A^B becomes $(0.33, 0, 0.67)$. If C cannot pass the authentication, the operations are the same as above.

3. C receives the re-broadcasted TRREQ from B. It will also check opinion ω_B^C and B's authenticity. If B passes, C will generate a TRREP back to B, calculate ω_B^C, and update its routing table. If not, C will drop the TRREQ and perform the same operations as above.

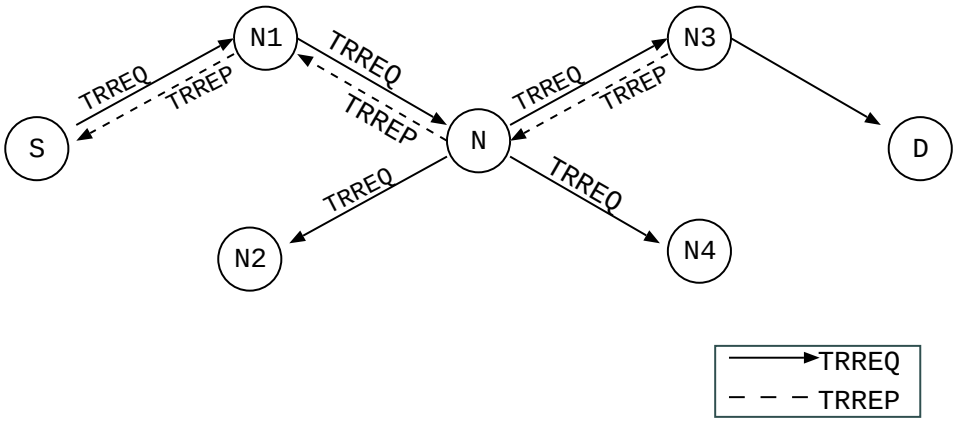


Figure 3.10: An example for trusted routing discovery.

Scenario II: A TAODV MANET After a Period of Running Time

In this case, a stable MANET has run for a period of time and the trust relationships have been established among almost all the nodes. Consequently, we can give a general description of trusted routing discovery process.

In the beginning of a MANET, because almost all the nodes are uncertain about other nodes' trustworthiness and authenticity, they have to authenticate with each other when performing routing behaviors. With the opinions being updated from time to time, the third component *uncertainty* of opinion will be decreased and the trust relationships among nodes are formed. The nodes in the MANET will thus employ the combination of different opinions to authenticate one another.

We describe the trust authentication algorithm and formulate the general procedure when performing trusted routing discovery based on the example in Fig. 3.10, shown as follows. In Fig. 3.10, the routing path from the originator S to the target node D is totally undiscovered. Node S will generate a TRREQ message and wait for the TRREP to discover a routing path to D. Node N is an intermediate node along this path, and nodes $N1$ to $N4$ are its four neighbors. We will describe what operations node N performs when it receives TRREQ/TRREP messages or TREQ/TREP/TWARN messages.

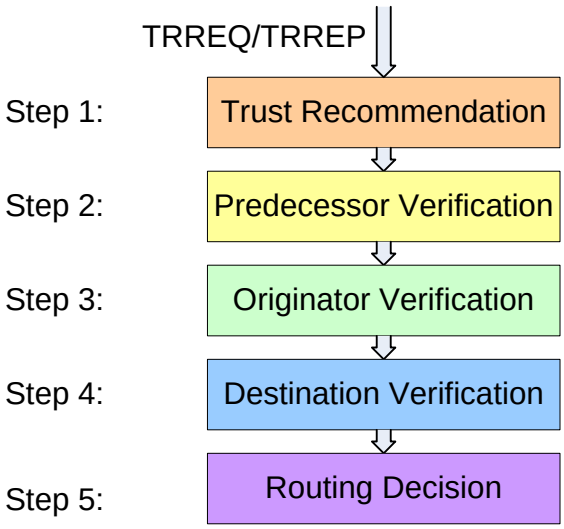


Figure 3.11: Trust routing steps at current node.

On Receiving TRREQ/TRREP When N receives a re-broadcasted TRREQ message from $N1$ or a TRREP message from $N3$, it will mainly take five steps of trust routing operations: *Trust recommendation*, *Predecessor verification*, *Originator verification*, *Destination verification*, and *Routing decision*, as shown in Fig. 3.11.

Suppose it is a TRREQ message from $N1$, so the predecessor of this message is $N1$. N will first broadcast a trust recommendation request message TREQ to ask for its neighbors' opinions about $N1$. Each neighbor receiving the TREQ, other than $N1$, will either drop the message if the disbelief value in its opinion about N is larger than 0.5, or reply to N a TREP message with its opinion about $N1$. N then collects these neighbors' recommendations towards $N1$ and combines them together following the combination equations in Section 2.2.4. The trust recommendation relationship is shown in Fig. 3.12, where the arrows denote opinion directions. N originally has opinions about $N1$, $N2$, $N3$, and $N4$: ω_{N1}^{N}, ω_{N2}^{N}, ω_{N3}^{N} and ω_{N4}^{N}. The opinions it receives from its neighbors are: ω_{N1}^{N2}, ω_{N1}^{N3}, and ω_{N1}^{N4}. The latest opinion from N to $N1$ can be calculated as follows.

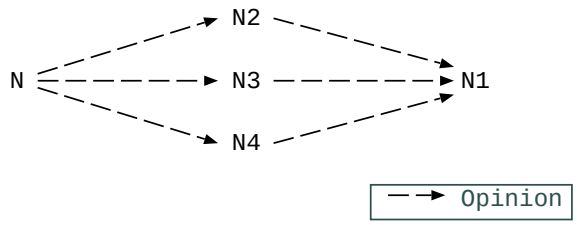


Figure 3.12: An example for trust recommendation.

First, $N1$ calculates opinions along the path using the discounting operator:

$$\begin{aligned} \omega_{N1}^{NN2} &= \omega_{N2}^{N} \otimes \omega_{N1}^{N2} \\ \omega_{N1}^{NN3} &= \omega_{N3}^{N} \otimes \omega_{N1}^{N3} \\ \omega_{N1}^{NN4} &= \omega_{N4}^{N} \otimes \omega_{N1}^{N4} \ . \end{aligned}$$

Then, the new opinion ω_{N1}^{N} can be combined across the paths using the consensus operator:

$$\begin{aligned} \omega_{N1}^{N} &= \omega_{N1}^{N(N2,N3,N4)} \\ &= \omega_{N1}^{NN2} \oplus \omega_{N1}^{NN3} \oplus \omega_{N1}^{NN4} \\ &= (\omega_{N2}^{N} \otimes \omega_{N1}^{N2}) \oplus (\omega_{N3}^{N} \otimes \omega_{N1}^{N3}) \oplus (\omega_{N4}^{N} \otimes \omega_{N1}^{N4}) \ . \end{aligned}$$

The combined opinions ω_{S}^{N} and ω_{D}^{N} are calculated without the trust recommendations to save the traffic and computation load. Because the TRREQ message carries the opinions of ω_{S}^{N1} and ω_{D}^{N1}, the lastest ω_{N1}^{N}, ω_{S}^{N} and ω_{D}^{N} can be obtained directly by discounting combination.

Everytime a node combines opinions together and gets a latest one, the corresponding number of successful or failed events should be re-calculated according to the mapping-back function in Eq. (2.28) proposed in Section 2.2.3. After trust recommendation, combination, and evidence updating, N will start to judge the opinions and verify the trustworthiness of the predecessor $N1$, the originator S, and the destination D one by one. The whole procedure is also

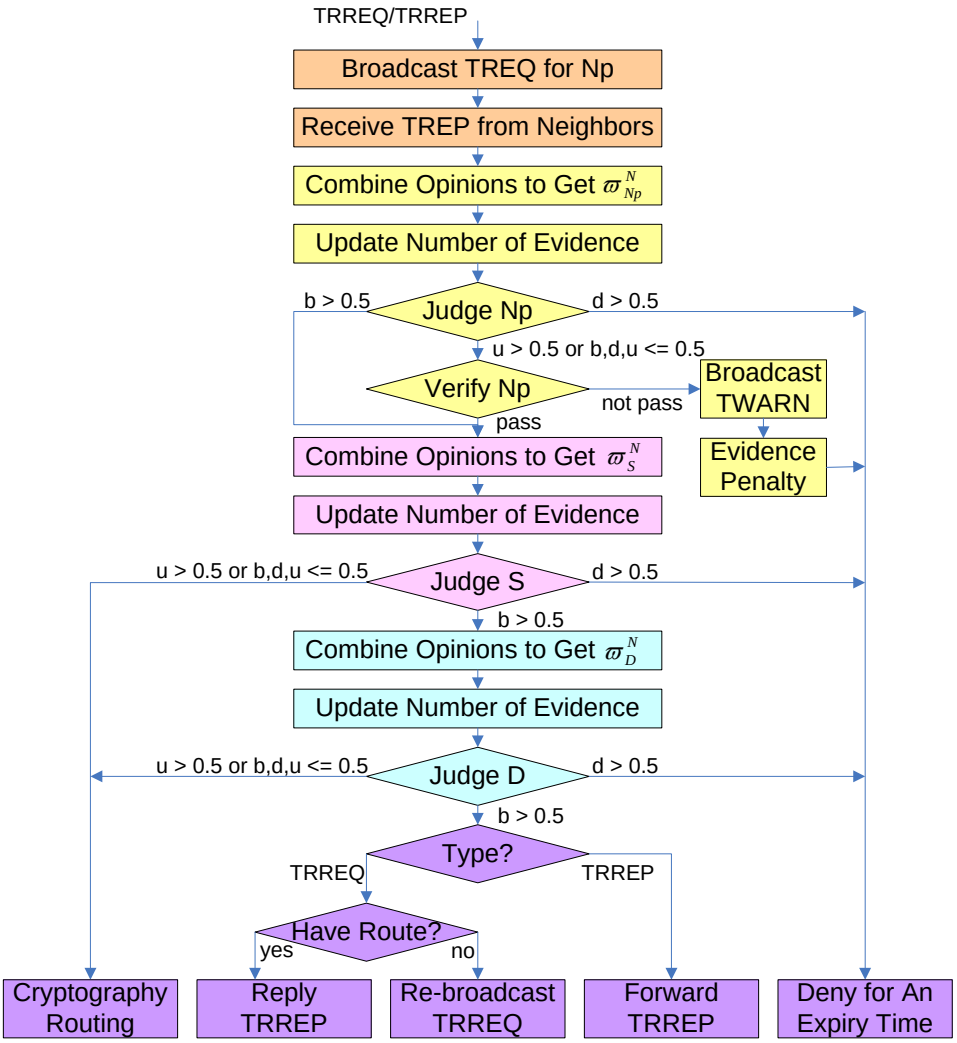


Figure 3.13: Trusted routing procedure at node N.

illustrated in Fig. 3.13. The symbol Np stands for the predecessor node, which is $N1$ or $N3$ in this example.

When judging the opinions, N follows the trust judging rules proposed in Section 3.3.3. If the uncertainty value in the new opinion of ω_{N1}^{N} is larger than 0.5, N will ask $N1$ to authenticate itself by providing digital signature or certificate. If $N1$ fails to do so, N will broadcast a TWARN message to its neighbors. At the same time, the number of evidence of $N1$ in N's routing table will get a penalty that the number of positive events is set to 0 and the number of negative events is set to a penalty number c. The penalty number c

can be decided according to the expiry time of a routing entry and the average throughput of a node in the network. The criterion is to let the node take at least an expiry time to gain enough positive events so that it can be trusted again.

For the opinions of S and D, usually the uncertainty value would be less than or equal to 0.5 because this is a scenario that the network has run for a period of time. In case of the uncertainty is larger than 0.5, cryptography routing module will take effect like the beginning of the network.

When $N1$, S, D all pass the trust judging and authentication period, N will follow the original AODV routing protocol to re-broadcast the TRREQ, forward the TRREP, or reply a TRREP. The whole procedure can also be illustrated in Algorithm 1.

On Receiving TREQ/TREP/TWARN The procedure for receiving trust recommendation messages TREQ/TREP/TWARN is simpler. When N receives a TREQ packet from M asking its opinion about L, it will first check its opinion about M. If the disbelief value in this opinion is larger than 0.5, N will simply drop the packet; otherwise, it will reply to M with a TREP packet filled with ω_L^N. When N receives a TREP or TWARN packet from M telling M's opinion about L, it will execute combination functions to get a new opinion ω_L^N. The above ideas are illustrated in Algorithm 2.

3.3.7 Trusted Routing Maintenance

The procedure of trusted routing maintenance is very similar to that of trusted routing discovery. Nodes will also use trust information to judge other nodes' trustworthiness. We omit the detailed algorithms here.

Algorithm 1: On Receiving TRREQ/TRREP Message At Node N

```
Input: Current Node N, Originator S, Destination D, Predecessor Np,
       A TRREQ or TRREP Message to N
Output: Routing Operations of N
begin
    Receiving TRREQ/TRREP;
    Recommend Broadcast ( TREQ, Np );
    Recommend Receive ( TREP, Np );
    foreach x in Np, S, D do
        ω_x^N = Trust Combine ( N → x );
        Trust Mapping ( Opinion → Evidence );
        switch ω_N1^N do
            case b > 0.5 continue;
            case d > 0.5 Deny x for an expiry time;
            case u > 0.5 or b, d, u <= 0.5
                if Authenticate (x) == true then continue;
                else
                    Recommend Broadcast ( TWARN, x );
                    Evidence Penalty ( x );
    switch Message Type do
        case TRREQ
            if N Have Route then Reply TRREP;
            else Re-broadcast TRREQ;
        case TRREP Forward TRREP;
    Packets Transmission;
    Trust Update ( Evidence → Opinion );
    Waiting TRREQ/TRREP;
end
```

3.4 Theoretical Analysis

From the performance point of view, our trusted routing protocol introduces less computation overheads than other security solutions for MANETs. Our design does not need to perform cryptographic computations in every packet, which will cause huge time and performance consumption. After the trust relationships are established, the subsequent routing operations can be performed securely according to trust information instead of acquiring certificate authentication all the time. Therefore, TAODV routing protocol improves

Algorithm 2: On Receiving TREQ/TREP/TWARN Message At Node N

Input: Current Node N, Neighbor M, Target L, A TREQ, TREP, or TWARN Message to N
Output: Recommendation Operations of N
begin
  Receiving TREQ/TREP/TWARN;
  switch *Message Type* **do**
    case *TREQ*
      if $d > 0.5$ **then** Drop TREQ;
      else Reply with TREP;
    case *TREP or TWARN* Combine opinions;
end

the performance of security solutions. Unlike some previous security schemes [31, 87], whose basis of routing operations is "blind distrust", TAODV does not decrease the efficiency of routing discovery and maintenance. In detail, we analyze the computation overhead of TAODV from two aspects. One is the cost of each trust combination and update operation. The other is the number of trust combination and update operations when given a certain volume of data load.

The cost of trust combination is $O(v)$, where v is the number of a node's neighbors. Each trust combination needs a constant number of multiplications, where the length of factor is 16 bit. Hence the overall cost of each trust combination requires $O(16^2 v)$ bit operations. For security solutions employing digital signature authentication, we use the RSA signature scheme for example to measure the computation cost of signature generation and verification. In general when using a $2k$-bit RSA signature, the generation of signature requires $O(k^3)$ bit operations and the verification requires $O(k^2)$ bit operations, where k is recommended at least to be 1024 bits for most security applications [58]. We can conclude from this aspect that TAODV achieves better computation performance compared to the pure signature authentication solutions.

On the other hand, we compare the times of performing digital authentication and trust updating when given a certain traffic volume. The digital authentication scheme usually needs to generate or verify signature for every routing message. While in TAODV protocol, with the help of expiry time of trust values, the trust updating times can be significantly reduced. Let us assume that the total number of routing packets propagated in the whole network is n, the average packet transmission interval is t, and the average expiry time of a trust value is e. Obviously the number of times in performing digital authentication is a constant value n because the generation or verification is required for each packet. The number of times in performing trust updating can be obtained by Eq. (3.1) in the following. The policy for updating trust used in this equation is that we combine periodical update and on-demand update together. When nodes in the MANET all have high mobility, the routing messages are sent in a high-frequency way. If the average packet sending interval t is smaller than the average expiry time, we update trust values periodically. When the nodes in the MANET stay in more stable positions, the average packet sending interval t is long. If the average packet interval value t is larger than the expiry time, we update the trust in an on-demand way.

$$U = \begin{cases} \lfloor \frac{nt}{e} \rfloor & , \quad t < e \\ n & , \quad t \geq e \end{cases} \tag{3.1}$$

We now assume that the total number of routing packets are 600 and the average expiry time is 10s, then we can draw a figure according to Eq. (3.1) in Fig. 3.14. It can be concluded that when the network has a high throughput it is quite efficient in using TAODV routing protocol. Comparing to those solutions that perform signature authentication not only for routing packets but also for data packets, the computation overheads of our solution will be largely reduced because we do not perform trust updating when transmitting data packets if we have established trust routes between the source nodes and

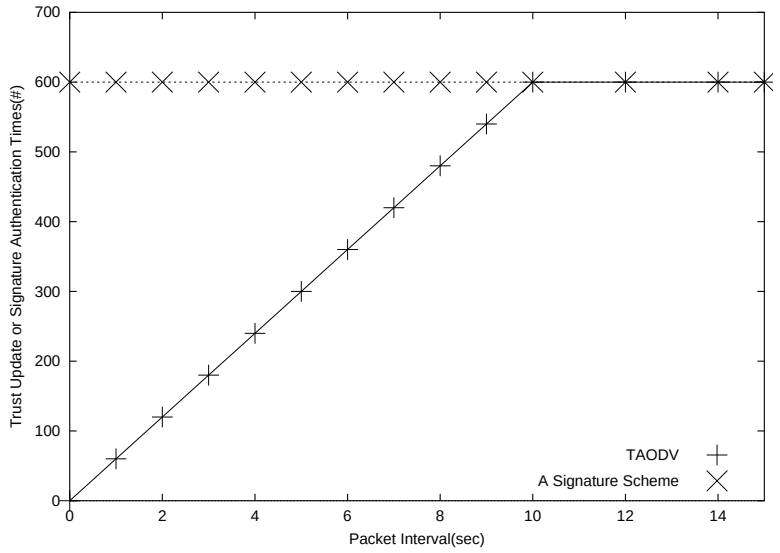


Figure 3.14: Times of trust update and signature authentication at different packet intervals.

the destinations.

From the security point of view, our design will resist the nodes' misbehaviors finally and reduce the harm to the minimum extent. When a good node is compromised and becomes a bad one, its misbehavior will be detected by its neighbors. Then with the help of the trust update algorithm, the opinions from the other nodes to this node will be updated shortly. Thus this node will be denied access to the network. Similarly, a previously bad node can become a good one if the attacker leaves the node or the underlying links are recovered. In this situation, our design allows this node's opinion from other nodes' points of view to be updated from $(0, 1, 0)$ to $(0, 0, 1)$ after a period of expiry time.

From the flexibility point of view, TAODV gives each node the flexibility to define its own opinion threshold. The default opinion threshold is 0.5, which can be increased by a node to maintain a high security level and also can be decreased to meet demands of some other applications.

3.5 Simulation

3.5.1 Simulation Environment

We perform a set of simulations on *NS-2* [61] with an extension for wireless networks, which is developed by Monarch research group in CMU. This extended simulator has good support for simulating complete wireless network protocol model from physical and data link layer, MAC layer, routing layer to application layer. Lucent's WaveLAN [18] [80] is the radio model with 2Mbps bit-rate and 250 meters radio range. The MAC layer is implemented according to IEEE 802.11 Distributed Coordination Function (DCF).

To evaluate the performance of TAODV without attackers, the basic movement and traffic models in our simulation are as follows. 50 nodes scatter in a $1500m * 300m$ field following a *random waypoint* model [8] with a velocity uniformly distributed between 0 and 20 m/s. In this moving pattern, each of the node moves from a random location to a randomly chosen destination initially. On arriving, the node will stop for a *pause time* then move to the next random destination. By varying the *pause time* we achieve different network mobility. This process repeats in each simulation run of 900 seconds. To simulate communication traffic, 20 source-destination pairs are chosen and randomly distributed over the network. The traffic source sends CBR (Constant Bit-Rate) data packets with the size of 512 bytes at the rate of 4 packets per second. The parameters of the simulation environment are listed in Table 3.2.

3.5.2 Misbehaving Model

To evaluate the TAODV with internal malicious or abnormal nodes, we provide a misbehaving model for simulation. The misbehavior we focus on in our simulation is *no forwarding* behaviors which commonly occur in mobile ad hoc networks caused by internal attackers, selfish nodes or unavailable nodes.

Table 3.2: Parameters for TAODV Simulation

Number of Nodes	50
Node Velocity Range	0-20 m/s
Simulation Field	1500 m * 300 m
Source-Destination Pairs	20
Source Packet Rate	4 pkts/s
Source Data Packet Size	512 bytes
Physical Link Bandwidth	2 Mbps
Nominal Radio Range	250 m

No forwarding means that a node participates in the wireless network but does not forward any routing operation packets, such as ROUTE REQUEST and ROUTE REPLY messages, or it performs normal routing operations but silently drops certain data packets.

We vary the number of misbehaving nodes among the 50 nodes from 0 to 20, so that the max percentage of misbehaving nodes in the simulated network is 40%, which is an extremely high ratio in real network environment. These nodes are chosen randomly by TCL's [44] built-in pseudo-random number generator. They keep dropping routing packets or data packets for a period of 200 seconds.

3.5.3 Metrics

We compute the following metrics to evaluate our TAODV.

Packet Delivery Ratio The ratio of the number of data packets received at the destination to the number of those originated at the application layer by the CBR traffic sources.

Average End-to-end Delay of Data Packets End-to-end delay represents the application level latency between the source and the destination application level. In our simulation, the end-to-end delay takes into account

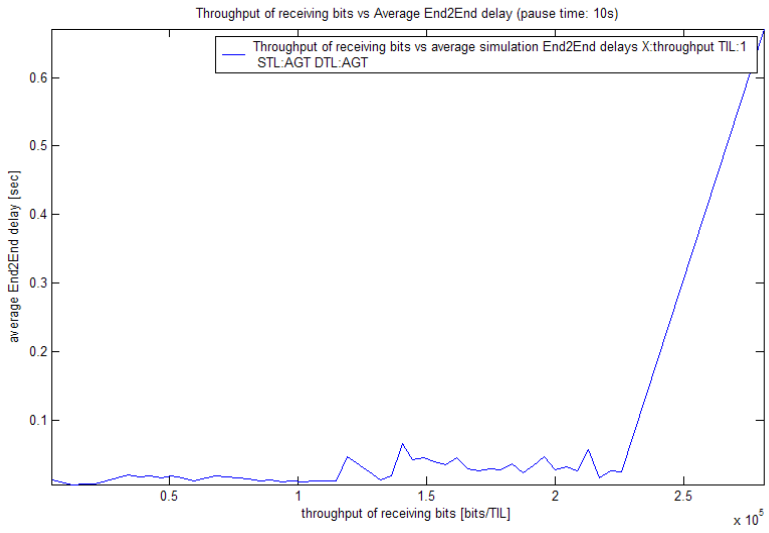


Figure 3.15: Throughput of receiving bits vs. average end-to-end delay.

not only the routing discovery latency, the queuing and buffering delays at the interface queue, the data propagation and transmission time, and the retransmission delays at the MAC layer, but also the computation delays caused by computation and verification of trust values or signatures.

Normalized Routing Load The number of routing packets needed to deliver a data packet to the destination.

False Positive Ratio The ratio of the good nodes reported to be misbehaving among all the nodes.

False Negative Ratio The ratio of the misbehaving nodes not reported among all the misbehaving nodes.

A selected result is shown in Fig. 3.15.

3.6 Trust Evaluation with Enhanced Subjective Logic

Although subjective logic provides a way to represent the concept of uncertainty and proposes mapping methods between evidence space and opinion space, the mapping function still introduces counter-intuitiveness. The value of uncertainty is only related to the numbers of positive and negative events, while human usually expect the result according to the ratio of positive and negative events. Thus the mapping function of u is not reasonable in some situations. Recall that u is mapped from evidence space to opinion space as follows:

$$u_B^A = \frac{2}{p+n+2}. \tag{3.2}$$

We can see that when the number of positive and negative events are nealy equal and both numbers large enough, the produced value of uncertainty will close to 0, which means total certainty. However, this is a counter-intuitive result because from the human subjective belief, if the positive and the negative outcomes have almost the same probability to happen, the uncertainty about this event should be more close to 1, which means total uncertainty.

Regarding to this instance, we will propose a more general and correct way to represent uncertainty and in turn benefit the evaluation of trust in an open environment, such as mobile ad hoc network.

3.6.1 Illustrating Opinion in a New Way

We also employ the definition of opinion described in Section 2.2.2. The four elements have identical meanings as in Def. 7. But we will graphically illustrate the opinion in a more compact way. As one of the three elements *belief*, *disbelief* and *uncertainty* is redundant, so instead of drawing the opinion as a triangle we represent the opinion in the rectangular coordinate. Because the sum of b,

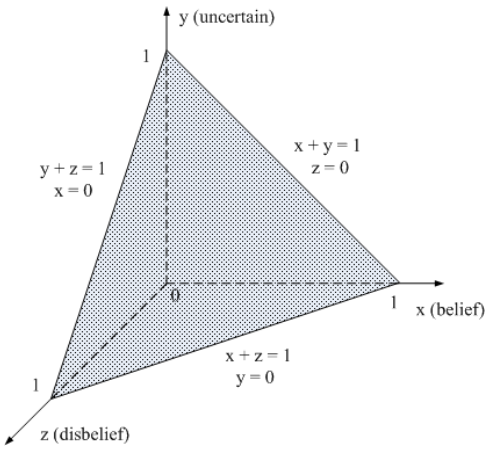


Figure 3.16: New opinion illustration in 3-dimension space.

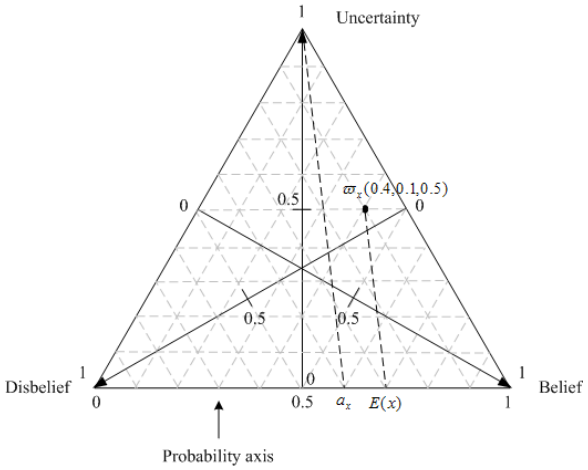


Figure 3.17: Original opinion illustration in triangular.

d and u is 1 according to Eq. (2.25), the opinion can be shown in Fig. 3.16 in a 3-dimension space.

So the new illustration of opinion in the rectangular coordinate compared to the original opinion form is shown in Fig. 3.17 and 3.18. The example opinion is $\omega_x = (0.4, 0.1, 0.5)$.

3.6.2 Re-Distribution of Opinions

For the situation that p and n are large and nearly equal, which makes the opinion around $(0.5, 0.5, 0)$, we plan to propose some opinion re-distribution

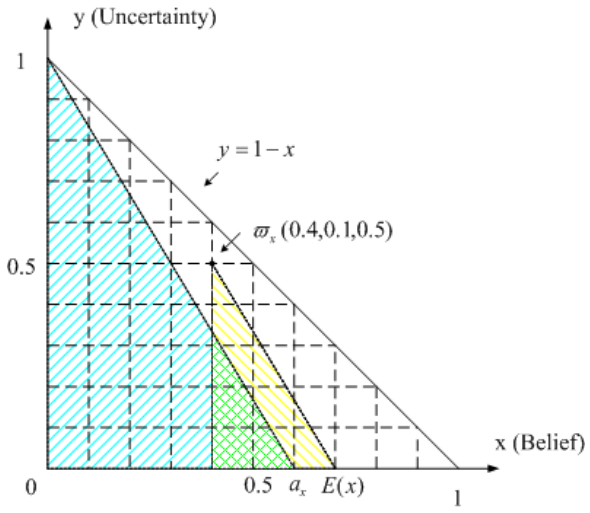


Figure 3.18: New opinion illustration in rectangular coordinate.

methods which can map the 0-uncertainty to other values, thus throwing the counter-intuitive opinions out of that range. The following equations are our three possible re-distribution solutions.

$$\begin{cases} b' = b, \quad u' = 1 - b - \varepsilon, & \text{if } b > d \\ b' = \varepsilon, \quad u' = u, & \text{if } b < d \end{cases}, \tag{3.3}$$

where ε is the allowable value of uncertainty.

$$u' = u^{|b-d|} \tag{3.4}$$

$$u' = u^{\log(b/d)} \tag{3.5}$$

After re-calculating u first, we adjust b and d according to the ratio of original b and d, and at the same time let them meet the requirement of $b + d + u = 1$. With the re-calculated opinions, we can demonstrate the new opinion distribulation with the following figures. There are totally 200 opinion points in each figure. Figure 3.19 is the original opinion distribution with many opinions around $(0.5, 0.5, 0)$, and Fig. 3.20, 3.21, and 3.22 illustrate the re-distributed opinions based on Eq. (3.3), (3.4) and (3.5) respectively.

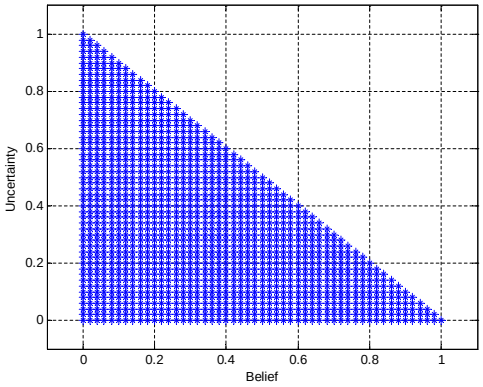


Figure 3.19: Original opinion distribution.

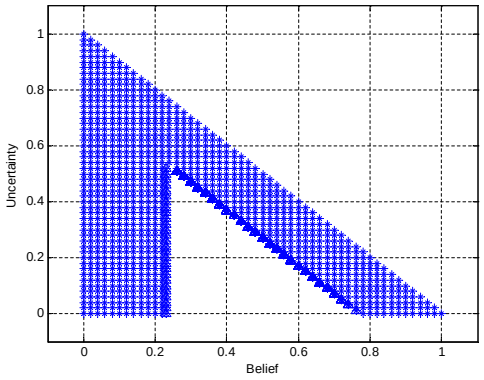


Figure 3.20: 1st opinion redistribution.

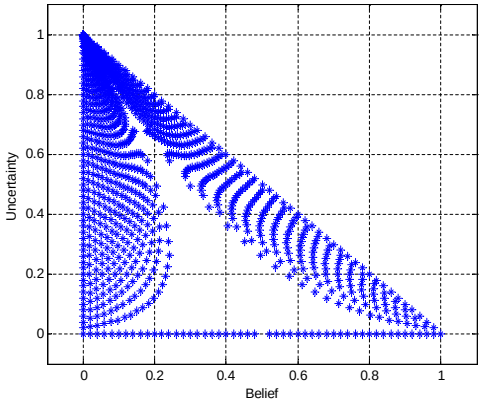


Figure 3.21: 2nd opinion redistribution.

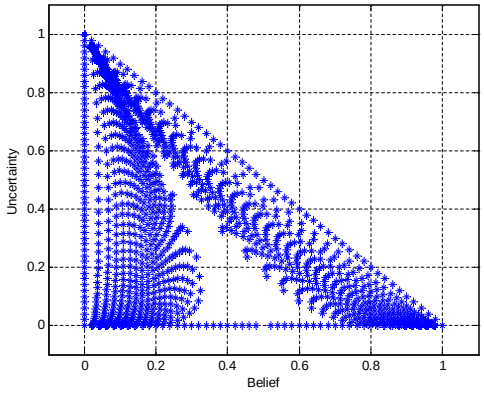


Figure 3.22: 3rd opinion redistribution.

Observing these figures we can intuitively get that Fig. 3.22 pushes the opinions more evenly and more consistently with the original opinion distribution. Therefore, we will employ Eq. (3.5) in the simulation to justify its feasibility and validity.

3.6.3 Simulation

We put 100 nodes randomly in a 100×100 square. Each node has 8 neighbors in average. When the network is "born", nodes are statistically assigned to be bad nodes or good nodes. We define a percentage of bad nodes m, e.g. $m = 30\%$. Nodes in neighborhood know if their neighbors are good or bad.

We select a good node as a delegate to evaluate the global indirect trust.

Initially bad nodes have best opinion for their neighboring bad nodes, e.g. $(0.9, 0.05, 0.05, 1-m)$. Bad nodes have worst opinion for their neighboring good nodes, e.g. $(0.05, 0.9, 0.05, 1-m)$. Good nodes adjust their direct opinions to their neighbors according to Beta distribution around low belief and high uncertainty.

The initial opinions from a delegated good node to all other nodes has high uncertainty. We want to make the uncertainty lower and lower, which means that the good node will have more and more definite opinions about other nodes trustworthiness.

At each simulation round, three things happen:

1. Each node performs an interaction with its neighbors. For bad node's neighbors, negative events will increase by a count, and for good node's neighbors, positive events will increase by a count.

2. According to the new evidence events, update the opinions in each direct neighborhood using the mapping function. Push the opinions using the re-distribution function.

3. Combine all the opinions from the selected good node to all other nodes through different paths using the discounting and consensus algorithm.

Selected results can be found in Fig. 3.23 to Fig. 3.27.

We can observe from the results that the re-distributed opinions convergence better than the original subjective logic opinions after 30 rounds.

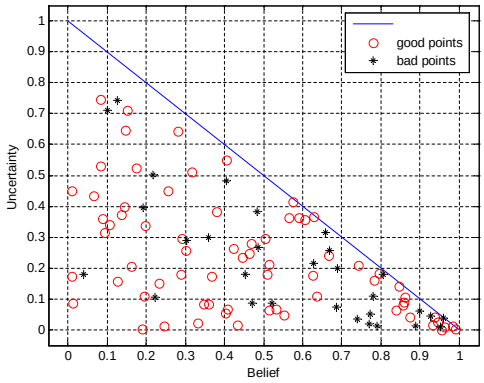


Figure 3.23: Initial opinion distribution.

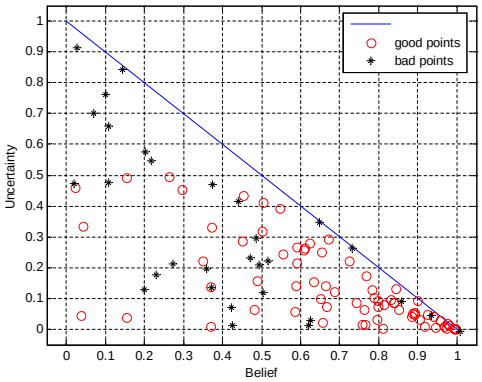


Figure 3.24: Subjective logic opinion distribution after 30 rounds.

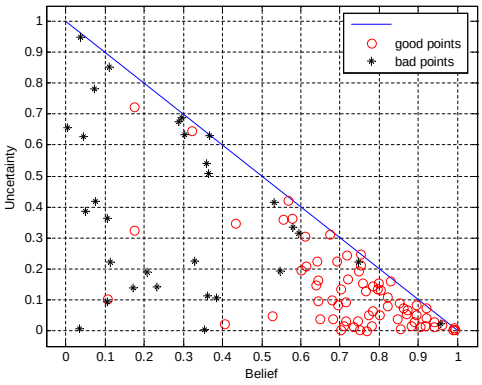


Figure 3.25: Enhanced subjective logic opinion distribution after 30 rounds.

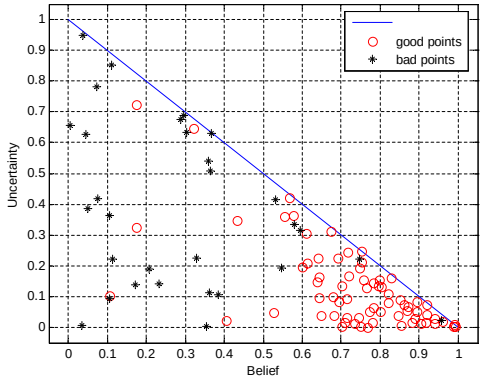


Figure 3.26: Subjective logic opinion distribution after 30+1 rounds.

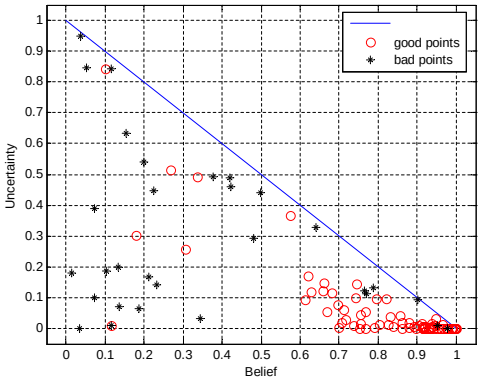


Figure 3.27: Enhanced subjective logic opinion distribution after 30+1 rounds.

Chapter 4

Non-cooperative Game Model for Security Issues

Previously we employ trust modeling approach to achieve a trusted wireless network. From this chapter on, we will apply game theory to address both security and selfishness issues of such networks. This chapter will first present a non-cooperative game theoretic model to analyze interactions between an attacker node and a regular node in mobile ad hoc networks. We view the interaction between an attacker and a regular as a two-player dynamic non-cooperative game with incomplete information. Two attacker-regular game trees are given according to different types of the stranger. From the game trees we can find out a threshold on the payoff assignment for the design of the secure routing protocol.

4.1 Background of Game Theory

Game theory is a branch of applied mathematics that is used in the social sciences (most notably economics), biology, engineering, political science, international relations, computer science, and philosophy. Game theory attempts to mathematically capture behavior in strategic situations, in which an individual's success in making choices depends on the choices of others [79]. It

provides us with tools to study situations of conflict and cooperation. Such situations exist when two or more decision makers who have different objectives act on the same system or share the same set of resources. Therefore, game theory is concerned with finding the best actions for individual decision makers in such situations and achieving stable outcomes [63].

"Game theory has been traditionally divided into *cooperative game theory* and *non-cooperative game theory*. The two branches of game theory differ in how they formalize interdependence among the players. In non-cooperative game theory, a game is a detailed model of all the moves available to the players. In contrast, cooperative game theory abstracts away from this level of detail and describes only the outcomes that result when the players come together in different combinations. In our work, we will formulate the security and selfishness issues of wireless networks in both ways."[63]

4.1.1 Non-cooperative Game Theory [63]

Non-cooperative game theory studies situations in which a number of players are involved in an interactive process whose outcome is determined by the node's individual decisions and, in turn, affects the well-being of each node in a possibly different way.

Non-cooperative games can be classified into a few categories based on several criteria. They can be classified as *static* or *dynamic* based on whether the moves made by the players are simultaneous or not. In a static game, players make their strategy choices simultaneously, without the knowledge of what the other players are choosing. Static games are generally represented diagrammatically using a game table that is called the *normal form* or *strategic form* of a game. In contrast, in a dynamic game, there is a strict order of play. Players take turns to make their moves, and they know the moves played by players who have gone before them. *Game trees* are used to depict dynamic

games. This methodology is generally referred to as the extensive form of a game. A game tree illustrates all of the possible actions that can be taken by all of the players. It also indicates all of the possible outcomes at each step of the game.

Non-cooperative games can also be classified as *complete information games* or *incomplete information games*, based on whether the players have complete or incomplete information about their adversaries in the game. Here information denotes the payoff-relevant characteristics of the adversaries. In a complete information game, each player has complete knowledge about his/her adversary's characteristics, strategy spaces, payoff functions, and so on. For further details on game theory, the reader is directed to [21, 62].

4.1.2 Basic Signaling Game [63]

A basic signaling game, in its simplest form has two players – Player 1 who is the sender and Player 2 who is the receiver. For the sake of convenience we treat Player 1 as masculine and Player 2 as feminine. Nature draws the type of the sender from a type set Θ, whose typical element is θ. The type information is private to each sender. Player 1 observes information about his type θ and chooses an action a_1 from his action space A_1 . Player 2, whose type is known to everyone, observes a_1 and chooses an action a_2 from her action space A_2. Player 2 has prior beliefs, before the start of the game, about Player 1's type. In other words, before observing the sender's message, the receiver believes that the probability that the sender is some type $\theta \in \Theta$ is $p(\theta)$. The action spaces of mixed actions are A_1 and A_2 with elements α_1 and α_2 respectively.

Player i's payoff is denoted by $u_i(\alpha_1, \alpha_2, \theta)$. Player 1's strategy is a probability distribution $\sigma_1(\cdot|\theta)$ over actions a_1 for each type θ. A strategy for Player 2 is a probability distribution $\sigma_2(\cdot|\alpha_1)$ over actions a_2 for each action a_1.

After both the players have taken their actions, the payoffs are awarded

according to the message sent by the sender, the action taken by the receiver in response and the type θ of the sender chosen by Nature.

4.2 Game Formulation of Attacker-Regular Interactions

In an ad hoc network environment, there may be all kinds of nodes with various types. Attackers perform malicious behaviors in diverse ways, and regular nodes will response with different actions. The relations between attackers and regular nodes have several forms: one attacker and one regular, more attackers and one regular (e.g. collusion attack), one attacker and more regulars (e.g. denial of service (DoS) attack), and more attackers and more regulars (e.g. distributed denial of service (DDoS) attack). We want to model these relations with the assistance of game theory.

In this section, we mainly model the interaction between one attacker and one regular, which can be fitted into the frame of non-cooperative dynamic repeated game with incomplete information. Because of the inherent characteristic of MANETs, there is no centralized superintendence to monitor all the behaviors of nodes. Therefore nodes may communicate with each other on the basis of their reputations. Formulating the reputations is also one of our goals, which can guide us in the design of new routing protocols on the basis of trust concepts.

To simplify our analysis and make our model non-trivial, we make the following assumptions.

1. Nodes in this network environment will last at least a certain lifetime;

2. Each node in his/her lifetime has only one identity;

3. Each node has the liberty and the ability to equip some security mechanisms, such as cryptography technology, host-based intrusion detection system, watchdog, trust exchanging algorithm and so on.

For the convenience of discussion, we will regard the attacker as masculine and the regular as feminine in the rest of this section.

4.2.1 Formulation Considerations

Even the interactions are between only one attacker and one regular, many possibilities of belonged types and undertaken actions can be exhibited. In the initial ad hoc network, nodes are strangers to each other. A node can be either an attacker or a regular. When he is an attacker, he may perform malicious behaviors all the time or just occasionally. Even though he is a regular node, he may "accidentally" behave abnormally. An attacker can use different attack methods with different goals. These attackers may be noticeable or even totally hidden. A regular node may have armed herself with some protection mechanisms or have no previous experience to execute such actions, in which case she is more prone to attacks. In case that malicious behaviors are detected by the target node, the node will make diverse responses to protect herself or beat the intruder. Therefore, we can see that the selection of different types of nodes and different actions they may take will lead to different structure of formulations. In the following, we will give two possible model structures and discuss the Nash equilibrium of them respectively.

In the field of game theory, players take actions depending on the expected payoffs assessed by them previously or timely. In the environment of ad hoc network, there are a lot of factors that affect the payoffs. Building a suitable utility function is a tough but important step in the game modelling of nodes interactions. The most commonly used factors may include consumed energy, connection bandwidth, and so on.

Our goal is to establish an expressive, realistic, non-trivial model of interactions between attacker(s) and target(s). We therefore try to solve the model and give a possible and reasonable Nash equilibrium solution. Besides, we also want to obtain value bound of some design factors which can be employed to design a corresponding application consistent with the strategies and beliefs in the established equilibrium.

4.2.2 Belief From a Regular to a Stranger

In our game model, the regular node is not sure about whether a stranger is an attacker or not. Thus the stranger has two types: {*Malicious, Regular*}. The known regular node has a prior belief to the stranger's type. She thinks that the probability of that stranger being malicious is ε. The regular will make communications with the stranger. From the behaviors observed she will try to make the best response according to the updated belief about the stranger's type. The strategies for the regular and the stranger are mixed. If the stranger is malicious, his action space is {*Attack, Normal*}. The probability that he performs attacks is s. If the stranger is regular, she will always behave normally. For the target node, she may perform two actions to the stranger: {*Doubt, Trust*}. The probability of her doubts is t. When she doubts, she may ask for her neighbors' help to get the trustworthiness of the stranger, or request the stranger to identify himself, or take some other measures. The structure of this game and the payoff formulation are shown in Fig. 4.1.

The policies for payoff formulation are listed below:

1. If the stranger is regular, the target will get payoff a if she trusts the stranger, where $a > 1$.

2. If the stranger is malicious and he attacks successfully, he will cause harm a to the target.

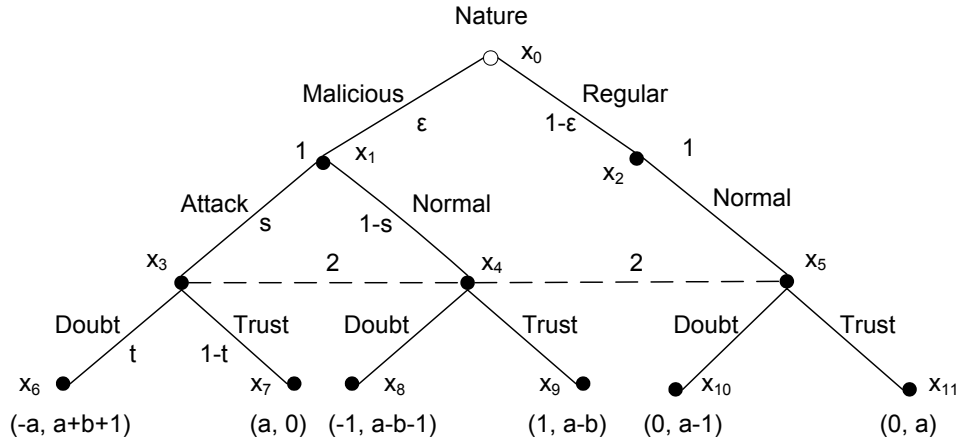


Figure 4.1: Attacker-Regular Model 1: The stranger has two types.

3. If the target doubts the stranger, it will cost 1 to her.

4. If the doubt is deserved, the target will get b amount of feedback, where $0 < b < 1$.

5. If the trust is not worthy, the target will lose b amount of payoff.

6. If the stranger is malicious but he pretends to be normal, in the current round, it will cost the target more to doubt him than to trust him, but the doubt will induce the stranger to get payoff of -1. In the long run game, the target may threat the stranger by doubting more frequently.

Next, we will try to maximize the payoffs and find the Nash equilibrium solution of this model which reflects the best responses of each player when given certain type and action. In the following Player 1 denotes the stranger and Player 2 denotes the target.

We claim that this model has no Nash equilibrium solution on pure strategy. To illustrate this, let us consider two reasonable pure strategies for example. One is (Stranger Attack, Target Doubt). If Player 1 is malicious and attacks, the best response of Player 2 is to doubt. But if Player 2 doubts, the best response of Player 1 is to behave normal. So in this case, the (Stranger Attack,

Target Doubt) is not the Nash equilibrium solution. The other strategy is (Stranger Normal, Target Trust). If Player 1 behaves normal, the best response of Player 2 is to trust (doubt is costly). But if Player 2 trusts, the best response of Player 1 is to attack. So this pure strategy is not Nash equilibrium solution neither. Other pure strategies can also be verified in a similar way.

Therefore, this model's solution, if exists, must be on mixed strategies. We claim that the model has sequential Nash equilibrium on mixed strategy, that is, the actions that the players take is a probability distribution on the action spaces. We denote the strategy profile as $\sigma = (\sigma_1, \sigma_2)$. When σ is given, $p_\sigma(x)$ denotes the probability that node x is reached, shown in Fig. 4.1. The information set h is a set containing more than one node, e.g. $h = x3, x4, x5$. Belief $\mu(x)$ specifies the probability that the player assigns to x when a certain information set h is reached.

The probability distribution on information set h is shown in Eq. (4.1).

$$\begin{cases} \mu(x_3) = & \varepsilon s \\ \mu(x_4) = & \varepsilon(1-s) \\ \mu(x_5) = & 1-\varepsilon \end{cases} . \tag{4.1}$$

So the expected payoff of Player 2 can be calculated as:

$$u_2(\sigma) = (3b-a)\varepsilon st + (a-b)\varepsilon s + (a\varepsilon - b\varepsilon - 1)t + a(1-\varepsilon). \tag{4.2}$$

We can get differential coefficient on s which is:

$$\frac{\partial u_2}{\partial s} = (3b-a)\varepsilon t + (a-b)\varepsilon. \tag{4.3}$$

Analyzing Eq. (4.3), the following conclusions can be drawn:

- When $t > \dfrac{a-b}{a-3b}$, Eq. (4.3)>0. That is, if s is increased, the payoff of Player 2 will increase.
- When $t < \dfrac{a-b}{a-3b}$, Eq. (4.3)<0. That is, if s is decreased, the payoff of Player 2 will increase.

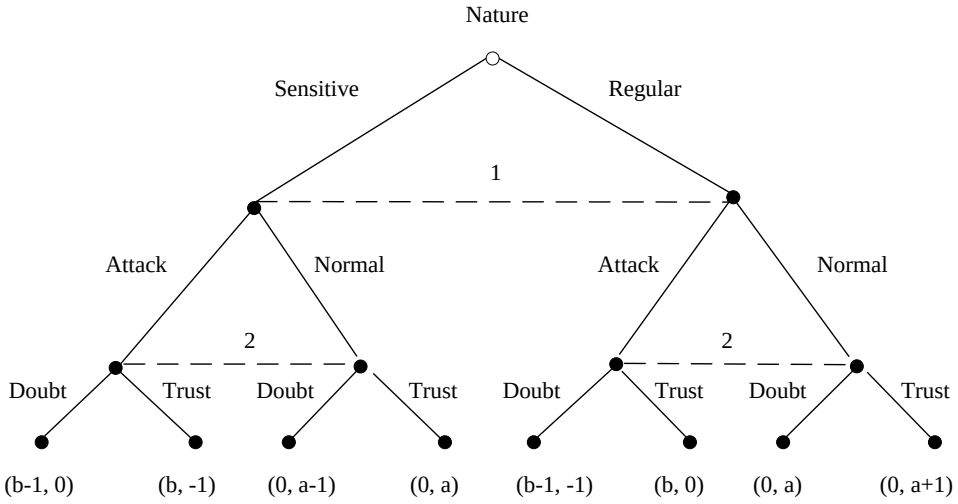


Figure 4.2: Attacker-Regular Model 2: The regular has two types.

From the above solution, we get a threshold value that can be applied to the design of corresponding secure routing protocol. In our previous secure routing protocol, if nodes opinion about another node exceeds a threshold, it will exchange opinions with its neighbors to get a more objective trustworthiness value.

4.2.3 Belief From an Attacker to a Regular

In this game model, the type of the attacker is decided while the target node has two types: {*Sensitive, Regular*}. Sensitive nodes possibly have armed with some protection mechanisms which make them easier to detect malicious behaviors or to find out the more correct trustworthiness of the attacker. An attacker wants to find out the type of a regular node and then perform corresponding attacks to her. In order to prevent attacks, the regular node may want to pretend to be a sensitive one to threat the attacker. The game tree is drawn in Fig. 4.2.

We can perform the similar analysis as above and find out the Nash equilibrium solution.

4.3 Summary

In this chapter, we employ the concept of multi-stage dynamic non-cooperative game with incomplete information to model the interactions between an attacker and a regular in the environment of wireless ad hoc networks. Two attacker-regular game trees are given according to different types of the stranger. From the game trees we find out a threshold of the payoff assignment for the design of the secure routing protocol.

Chapter 5

Coalitional Game Model for Security Issues

In the last chapter, we formulate the security issues of wireless networks as a non-cooperative game. In this chapter, we will develop a cooperative game model for the same issues. The game we employed is called a *coalitional game*. The key point of formulating a coalitional game is to define a proper payoff characteristic (value) function for any coaltion. In this chapter, we design two different characteristic functions from two different aspects: *security characteristic function* and *throughput characteristic function*.

The design of security characteristic function employs three metrics: *support rate*, *cooperation probability*, and *effective overlapping distance*. We also present a coalition formation algorithm for the nodes to follow, so that they can join coalitions and get a higher level of security. Through simulations we show that malicious nodes are identified from the good ones eventually. And we also analyze the model using a formal game theory method and prove that the nodes have incentive to form coalitions and the game will reach a stable core state under the security characteristic function.

The throughput characteristic function is defined based on the maximal throughput and the most reliable traffic that a coalition can achieve. The fair payoff share inside the coalition is given by Shapley Value after proving the

feasibility of this method. Then a set of game rules is presented to establish a threatening mechanism to all players. We then describe the coalition formation procedure and explain how to integrate this game theoretic model with available wireless routing protocols. Finally, a theoretical analysis is conducted to illustrate the convergence situation and verify the correctness of the formulation.

Please note that the model can be applied not only to mobile ad hoc networks but also to wireless sensor networks.

5.1 Security Value Function Based Coalitional Game Model

There are a great diversity of methods and goals for attackers in the environment of wireless networks. Most of the attacks are performed by tampering the routing messages or disturbing the data transmissions to achieve the goal of decreasing the total network performance or causing some nodes denial of service (DoS). Traditional attacks that intrude the terminal system and illegally acquire the super privilege still exists, but they are not the key characteristics of attacks in wireless networks. In this section, we only consider those attacks targeting routing processes and data packets.

5.1.1 Basic Idea

We model the security issue of an ad hoc network as a coalitional game with transferable payoff: $\Gamma =< N, v >$, where N is the set of nodes (players), and v is the security characteristic function that associates with every nonempty subset S of N a real number $v(S)$. The physical meaning of v is the quantified security level each coalition S can achieve. Our goal is to gracefully define the security characteristic function and show that at each time slot this

game has a stable outcome that no coalition will deviate and obtain a better security outcome for all its members. We will examine how N is partitioned into different S at a stable outcome. Nodes that cannot form any coalition are under very high suspicion of being malicious. At another time slot nodes will move around or leave the network, then the coalitions will be re-formed and a new stable state can be achieved.

5.1.2 Security Characteristic Function

In a wireless ad hoc network, nodes are distributed. It is difficult or even impossible for an individual node to acquire all the information it needs to perform a guaranteed operation. The incompleteness and imperfectness of information bring in uncertainty. Whether a node cooperates with another one is according to its own experience and previous history records it collected. In other words, individual nodes are weak in security issues. However, when nodes form coalitions together, they will jointly achieve a higher security level.

- Firstly, if nodes group together, they can testify for each other so that the coalition has more trustworthiness than individuals. In other words, nodes belonging to a coalition are more trustworthy than those who have no coalition members.

- Secondly, nodes can share observed information and transaction histories inside a coalition, which helps them make a more proper and definite decision about whether cooperating with another node or not. In such a way, the group is more robust against attacks.

- Thirdly, in our model nodes which are in closer distance have higher probability to form a coalition so that they can provide more reliable link connection. Meanwhile, this will decrease false positive alarm rate introduced by physical link instability. And data packets transmitted

along these links cost less transmission power, which fits the design of an effective routing protocol.

To correctly model a coalitional game, the key point is to assign a pay-off value to each coalition. That is, we must define a value (characteristic) function $v(S)$ for any coalition S. Based on the above reasons, our security characteristic function consists of three factors:

- support rate
- cooperation probability
- overlapping distance

Suppose there are N nodes in the network, and for any coalition $S \in 2^N$, the number of nodes in it is $|S|$. We can claim that any node in this coalition would have $|S| - 1$ nodes that testify for it. Then at time slot t, the support rate of coalition S is defined as:

$$T_t(S) = |S| - 1 \tag{5.1}$$

Suppose for each node i the history table it maintains has H_i entries. Each entry contains two elements: the other node j it concerns and the probability of cooperation p_{ij} with that node. For every node in the already formed coalition, it was admitted into this coalition with a certain probability. This admitting probability is the average opinion of all the related nodes which have history information with that node at that formation round. Nodes' admitting probabilities are different from each other. We then assign the maximal admitting probability as the cooperation probability of the whole coalition, because the larger the coalition size is, the more tolerant and robust the coalition is, and the coalition can therefore have a higher cooperation probability. The definition equation is as follows:

$$B_t(S) = \max_{j \in S} \left\{ \frac{\sum_{i \in I} p_{ij}}{|I|} \;\middle|\; I = \{i | i \in S, i \neq j, p_{ij} \neq 0\} \right\} \tag{5.2}$$

Let r_i and r_j be the transmission range radius of node i and j respectively, and d_{ij} be the distance between them. So the overlapping range between two nodes can be written as:

$$O_{ij} = r_i + r_j - d_{ij}$$

The larger overlapping value means the shorter distance between two nodes. Once the coalition is formed together it can provide more reliable links cooperatively. In other words, the link reliability of the whole coalition is increased. Therefore, we use the maximal overlapping value to represent the link reliability of the coalition

$$D_t(S) = \max_{i,j \in S} O_{ij}(t) \tag{5.3}$$

If the size of coalition S is 1, which means the node has no coalition peer, the support rate of S will be zero as indicated by Eq. (5.1), while the cooperative rate and the overlapping distance will become meaningless from Eq. (5.2) and Eq. (5.3). Therefore we will assign zero to the value of such coalitions. In summery, we have such security characteristic that when $|S| = 1$, $v(S) = 0$; otherwise, the security value function is the linear combination of the above metrics. The weight of each metric can vary depending upon different ad hoc network applications. The formal definition is written as follows:

Definition 13 (Security Characteristic Function) *The security characteristic function $v_t(S)$ is the linear combination of $T_t(S)$, $B_t(S)$ and $D_t(S)$:*

$$v_t(S) = \begin{cases} 0, & |S| = 1 \\ \alpha T_t(S) + \beta B_t(S) + \gamma D_t(S), & |S| \geq 2 \end{cases}, \tag{5.4}$$

where

1) $T_t(S)$*,* $B_t(S)$ *and* $D_t(S)$ *respectively stand for support rate, cooperrative probability and overlapping distance at time session* t*.*

2) α*,* β *and* γ *are weight parameters, and* $\alpha + \beta + \gamma = 1$*.* □

5.1.3 Coalition Formation of Nodes

Every node in the game maintains a routing table as described in the original routing protocol. Each entry in the table represents a node that the current node concerns. Now we add four elements to the entry:

1. Number of nodes in the coalition that the concerned node has joined;
2. Probability that the current node would cooperate with the concerned one;
3. The overlapping transmission range between the current node and the concerned one;
4. The security value calculated from the above metrics according to our security characteristic function. This value can be viewed as the quantified security value.

The final coalition division is formed gradually in several rounds. Before taking any coalition formation step, each node performs initialization by computing the security characteristic value for each entry in its routing table. Then the formation process begins. In the first round, the node firstly looks at its routing table and picks a node with the highest security value as its first coalition option. Then it broadcast its forming options to the network. If all the values in its table are not beyond a certain threshold, it will not pick any node. If there are two nodes which match the first option of each other, they will form a coalition. Correspondingly the routing table will be updated using the new number of coalition members. And this is the round one. At the end of round one, nodes may form several coalitions in pairs.

At the following round, each node still picks one node with the highest security value. If the first option has been matched successfully the node will pick the second option. Those whose first options have not been formed into coalitions will still broadcast their first preferences. Then some new pair coalitions form. After comparing with the result of the last round, coalitions are merged into one if they have the same member.

This process will be performed iteratively until there is no node left uncoalized or no new coalition can be formed. Now checking out the coalition list, the nodes that do not belong to any coalition can be deemed as either malicious or at least being fallow in taking part in the network forwarding functions.

The algorithm is formally given in Algorithm 3.

Algorithm 3: Coalition Formation Algorithm

```
while timeslot≠ 0 do
    while ∃ nodes want to go on forming coalition do
        Each of these nodes pick up a node with the highest security
        value unchosen;
        Publish its option;
        Matching process;
        if matching successful then
            Form a new coalition;
            Merge with previous coalitions;
            Update routing table;
    Start a timer;
    while timer < time interval do
        Do normal routing and forwarding process based on coalition;
        Update routing table normally;
    Timeslot ← timeslot-1;
```

5.2 Analysis by Game Theory

Now we will theoretically prove that based on our designed security characteristic function nodes always have incentives to form coalitions, which guarantees

no normal individual is unhappy. Futhermore, we will also prove that the coalition formation process can finally reach a stable core status at each time session, which means at that status no coalition wants to deviate and obtain a better security outcome for all its members, that is, no coalition is unhappy.

To investigate the playing feasibility and existence of core of this coalitional game model, we should find out the relationship between the individual payoff before a node joins into a coalition and the payoff share it may get after being admitted into that coalition. In Section 5.1.2 we discussed the payoff function from the coalition point of view; now we will focus on the payoff imputation share from the view of individual coalition member.

From Def. 13 we can see that if the node does not join any coalition ($|S| = 1$), its individual security value is equal to zero. Now we will discuss the payoff share when a node belongs to a coalition. According to our coalition formation algorithm, each node is admitted into a coalition by the highest security value among all the nodes' opinions of that coalition. Correspondingly, the security share of each node in the coalition also consists of three metrics. We define our payoff imputation rules inside a coalition as follows:

Firstly, for every coalition member it still has $|S| - 1$ other members to testify for it. So the testification share metric is:

$$T_t^S(i) = |S| - 1 \tag{5.5}$$

Secondly, when the node is admitted into a coalition, it will get the maximal cooperation probability among all the probabilities others assign to it.

$$B_t^S(i) = \max_{j \in S} p_{ji} \tag{5.6}$$

Thirdly, the overlapping distance share metric is defined as the maximal overlapping distance between this node and any of its neighbors.

$$D_t^S(i) = \max_{j \in NB} O_{ij}(t) \tag{5.7}$$

In summary, we give the formal definition of security imputation vector in the following:

Definition 14 (Security Imputation Vector) *The security imputation vector x is the linear combination of T_t^S, B_t^S, and D_t^S. For any $i \in S$, $|S| > 1$, its security payoff share is defined as:*

$$x_t^S(i) = \frac{1}{|S|}(\alpha T_t^S(i) + \beta B_t^S(i) + \gamma D_t^S(i)), \tag{5.8}$$

where

1) $T_t^S(i)$, $B_t^S(i)$, and $D_t^S(i)$ respectively stand for payoff share of support rate, cooperrative probability and overlapping distance at time session t.

2) α, β and γ are weight parameters, and $\alpha + \beta + \gamma = 1$. □

It is obvious that the payoff share of a node belonging to a coalition is larger than zero, which means that nodes will gain more payoff when joining coalitions than remaining alone. That is how the incentive mechanism works. However, it is not enough that players in the game are willing to form coalitions. If they keep on joining one coalition after another, the formation process will never stop and it is difficult to identify malicious nodes correctly and effectively. Therefore we must study the stable status of the game, which is called the core of the coalitional game in game theory. Now we have the following claim:

Theorem 1 This coalition game in the role of security characteristic function defined in Def. 13 has a core. □

To prove the theorm, we can follow the definition of core in game theory. For any coalition division topology, only if the sum of payoff share of all the members for each coalition is larger than the value of that coalition, then we

can say that the core exists. In our situation, we can easily get the result from our previous definition. The proof is as follows:

Proof 1 From Eq. (5.4) and Eq. (5.8) we can deduct that:

$$\sum_{i \in S} x_t^S(i) \geq v_t(S), \quad for\ all\ S \in 2^N \tag{5.9}$$

This equation satisfies the concept of the core of the coalition game. Therefore we claim that this security coalition game has a core. □

5.3 Simulation

We simulate the coalition formation process among n nodes with about 10 to 20 percent of malicious nodes. The nodes are randomly distributed in an area of $10n$ by $10n$ meters. The radio range of each node is $1.44n$ meters. The simulation process consists of the following steps:

1. Randomly generate a distribution topology of n nodes;
2. Randomly pick up a percentage of nodes from 10% to 20% as malicious nodes;
3. Initialize the values of support rate, cooperative probability and overlapping distance, then calculate the security characteristic value for each history entry of each node;
4. Run the coalition formation algorithm as described in Algorithm 3 until the loop ends;
5. Find out the nodes which do not form into any coalition.

Figure 5.1 shows the final coalition formation result with one malicious node out of ten. The blue circle stands for the highly suspected node and

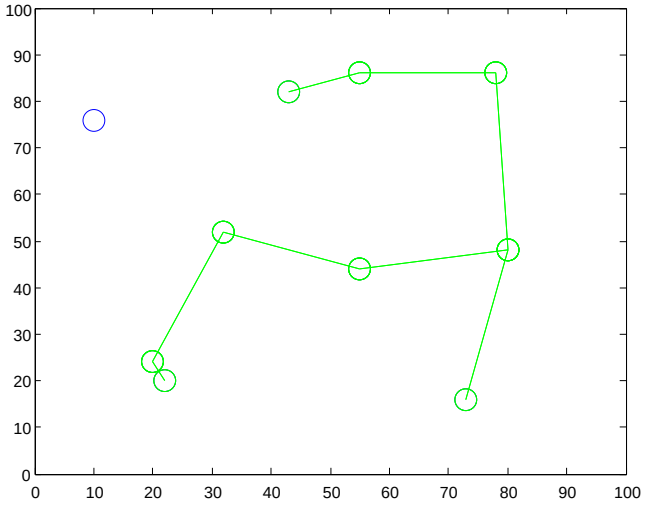


Figure 5.1: Coalition Formation Case 1: 10 nodes with 1 malicious node.

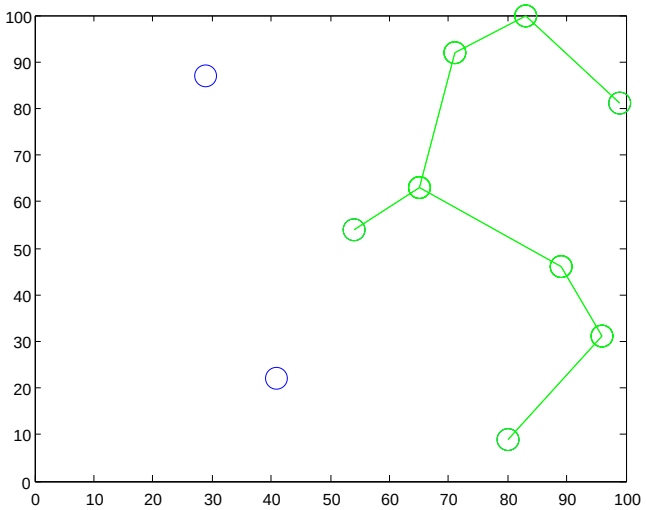


Figure 5.2: Coalition Formation Case 2: 10 nodes with 2 malicious nodes where normal nodes form into one coalition.

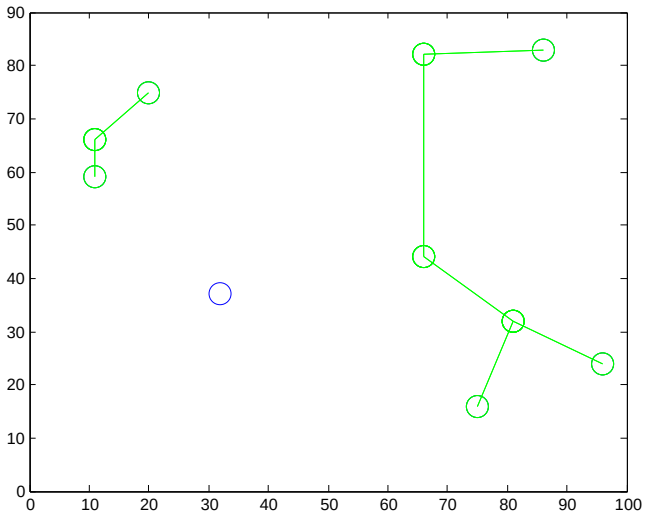


Figure 5.3: Coalition Formation Case 3: 10 nodes with 1 malicious nodes where normal nodes form into two coalitions.

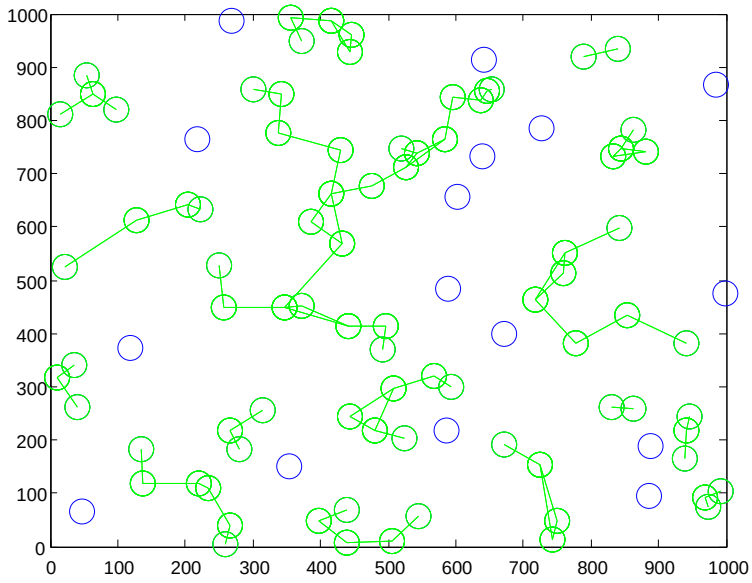


Figure 5.4: Coalition Formation Case 4: 100 nodes with 10% malicious nodes.

others are normal ones. We can see that the blue node is isolated from the grand coalition of normal nodes after the process. This grand coalition is formed from small coalitions gradually at difference processing rounds.

Figure 5.2 depicts a coalition scenario with 2 malicious nodes out of ten. In this situation, all the normal nodes join into one coalition eventually. Both of the two bad nodes are differentiated from the normal ones.

Note that at the final stable core state, the normal nodes are not necessarily formed into one coalition. According to the definition of our security characteristic function which is the foundation of our coalition formation algorithm, the nodes which have long overlapping distance or whose cooperative probability cannot achieve a certain threshold will not be united. Figure 5.3 illustrates such kind of situation that the normal nodes are divided into two coalitions.

Now we extend the simulation process to a larger scenario where there are 100 nodes and 10 of which are malicious. We can observe from Fig. 5.4 that the coalition formation algorithm is scalable to larger network capacity. Nodes in this network form into various coalitions with different sizes and situations to achieve their maximal security and reliability. In this example there are 16 nodes being marked as suspicious and excluded from the coalitions. Further investigations on sensitivity analysis and coalition effectiveness in large networks are needed, but we will leave them for future work.

5.4 Throughput Value Function Based Coalitional Game Model

In this section, we formulate the wireless network as playing a coalitional game by defining a throughput characteristic function and giving the payoff distribution method among the coalitional members. A set of game rules is

prescribed and a threatening mechanism is established, based on which we also design a coalitional formation algorithm that can be integrated into routing protocol to make it have more traffic capacity and more reliability.

5.4.1 Basic Idea

Cooperation is the inherent nature of wireless ad hoc and sensor networks. Formulating the network as a cooperative game will not destroy this nature but make full use of it. Coalitional game is one kind of cooperative game that we think will satisfy the properties of our problem.

Our coalitional game has transferable payoff and is denoted by $\Gamma =< N, v >$, where N is the set of nodes (players), and v is the throughput characteristic function that associates with every non-empty subset S of N a real number $v(S)$. The physical meaning of v is the maximal throughput and the most trustful and reliable traffic that each coalition S can achieve. It is the foundation of the coalition forming procedure and it constrains the coalition to admit or exclude a node. Our goal is to gracefully define the throughput characteristic function and also a fair payoff distribution method among coalition members. This work is done in sections 5.4.2 and 5.4.3. We will then examine how coalitions are formed under the effect of this payoff function and set game rules in sections 5.4.4 and 5.5. In such a way, nodes are enforced to take part in coalitions and those that cannot join into any coalition are under very high suspicion of being malicious.

To make our model mainly focus on the problem formulation, we give the following assumptions: 1) we assume that there is a Watchdog [56] mechanism in each node, by which it can detect whether its neighbors are forwarding data packets for it or not; 2) we also assume that a time synchronization mechanism has been implemented in the system so that we can schedule the coalition formation process synchronously.

5.4.2 Throughput Characteristic Function

We firstly give the definition of a throughput characteristic function and then explain it detailedly in the rest of this section.

Definition 15 (Throughput Characteristic Function) *The throughput characteristic value for any coalition S, $S \subseteq N$, where $|S| = 1$ and $|S| = 0$, is 0. For other coalition S, where $|S| \geq 2$, the throughput characteristic function $v(S)$ is defined as:*

$$v(S) = \frac{1}{\triangle t} \sum_{(a,b) \in SD, a,b \in S} Q_{ab} \cdot \max_{k \in P_{ab}(S)} \left\{ t(k) = \prod_{(i,j) \in k} \frac{p_{ij}}{D_{ij}^2} \right\}, \tag{5.10}$$

where

1. *$\triangle t$ is a certain time interval*
2. *$SD = \{(a,b) |$ (a,b) is a source-destination pair$\}$*
3. *Q_{ab} is the required number of data packets transmitting between pair (a,b)*
4. *$P_{ab}(S)$ is the set of routing paths inside coalition S which connect pair (a,b)*
5. *$k \in P_{ab}(S)$ is one of the path in $P_{ab}(S)$ and $k = \{(i,j) |$ i, j are the adjacent nodes on the same routing path$\}$*
6. *$t(k)$ stands for the reliability of routing path k*
7. *p_{ij} is the trustworthiness of path (i,j)*
8. *D_{ij} is the distance between node i and j*

□

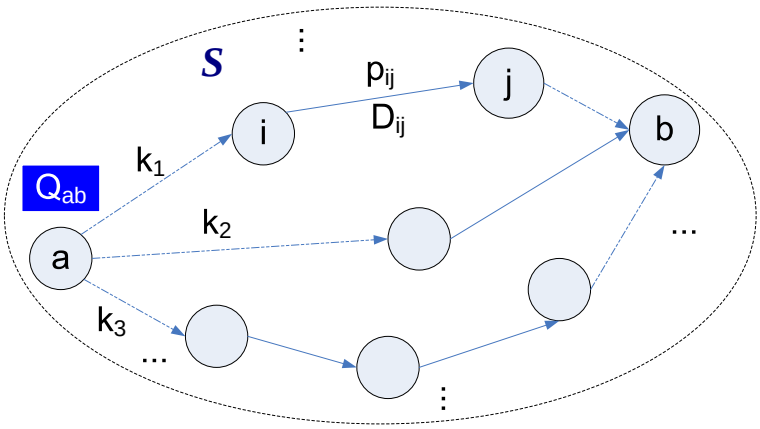


Figure 5.5: Coalition S labelled with parameters in throughput characteristic function.

Figure 5.5 shows an example coalition labelled with parameters in the throughput characteristic function. In the following paragraphs, we will explain each parameter one by one.

When a coalition is formed, it can generate a weighted directed graph $G(S)$, where vertexes are nodes inside the coalition, edges represent routing direction between nodes, and weights are the probabilities that one node wants to communicate with another. From this graph, we perform routing a discovery procedure to discover the first several possible routing paths $P(S)$ for each source-destination pair inside the coalition. The number of routing paths is related to the size of the coalition. When the coalition size increases, more possible paths can be found and more reliable routing and forwarding transmission can be obtained.

For every possible routing path k between the source-destination pair, we get a reliability evaluation $t(k)$. From the coalition point of view, the maximal value of $t(k)$ over all k means the best service that the coalition can provide to this source-destination pair. In other words, it indicates the maximal payoff that the pair can benefit from the coalition. We also use $t(i, j)$ to denote $t(k)$, where i, j are two end nodes of path k.

The probability that node i wants to communicate with node j implies the trustworthiness of the routing path from i to j. It is obtained from two ways: direct experience and indirect recommendation. The direct experience p is the fraction of observed successful transmission times by all the transmission times between i and j, shown in Eq. (5.11):

$$p = \frac{u_{succ}}{u_{all}}. \tag{5.11}$$

The indirect recommendation comes from node i's neighbors. Each neighbor of i returns probability opinions about both i and j, then i combines those probabilities of all neighbors together. Please note that we consider not only neighbors' recommendations towards j but also towards i, which represents the opinions towards the routing path from i to j. Multiplying by node i's own evaluation to its neighbors, we then get the more believable indirect probability p' of communication from i to j. The form is given in Eq. (5.12):

$$p' = \frac{\sum_{l \in NB_i} p_{il} p_{li} p_{lj}}{|NB_i|}, \tag{5.12}$$

where $|NB_i|$ is the number of neighbors of node i.

Since direct experience and indirect recommendation have different weights, which can be adjusted to fit into different applications, we then combine the probability p_{ij} in Eq. (5.13):

$$\begin{aligned} p_{ij} &= \alpha p + (1-\alpha) p' \\ &= \alpha \frac{u_{succ}}{u_{all}} + (1-\alpha) \frac{\sum_{l \in NB_i} p_{il} p_{li} p_{lj}}{|NB_i|}. \end{aligned} \tag{5.13}$$

Finally, the reliability of a routing path is determined by not only the communication probability but also the physical connection between the two nodes. Even though both nodes have good reputation, the path is still lack of reliability if they are too far away from each other. So we take another metric, distance D_{ij}, into consideration. And because the signal fading of the link is

in inverse proportion to the square of distance, so we use D_{ij}^2 to represent the connectivity of the link.

5.4.3 Payoff Allocation inside Coalition

The throughput characteristic function describes the total expected gain of a coalition from the cooperation. Since some nodes may contribute more to the coalition than others, now we consider the problem of how to fairly distribute the gains among all the nodes. In other words, what payoff can nodes reasonably expect from cooperation. Shapley value [74] is one way to distribute the total gains to players, which is applicable when the payoff function satisfies the following two conditions:

$$\begin{array}{ll} 1. & v(\phi) = 0 \\ 2. & v(S \cup T) \geq v(S) + v(T) \end{array}, \tag{5.14}$$

where S and T are disjoint subsets of N. Then the amount that player i gets is as follows:

$$x_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S)). \tag{5.15}$$

To employ this equation, we now justify that the proposed throughput characteristic function satisfies the two conditions in Eq. (5.14).

Theorem 2 Shapley Value method is applicable to the payoff allocation inside coalitions given our proposed throughput characteristic function $v(S)$. □

Proof 2 Firstly, from the definition of throughput characteristic function $v(S)$, we easily know that $v(\phi) = 0$, which satisfies the first condition of Eq. (5.14).

Secondly, on the basis of $v(S)$, we have the following equations:

$$v(S) = \frac{1}{\triangle t} \sum_{(a,b) \in SD, a,b \in S} Q_{ab} \cdot \max_{k \in P_{ab}(S)} \left\{ t(k) = \prod_{(i,j) \in k} \frac{p_{ij}}{D_{ij}^2} \right\}$$

$$v(T) = \frac{1}{\triangle t} \sum_{(a,b)\in SD, a,b\in T} Q_{ab} \cdot \max_{k\in P_{ab}(T)} \left\{ t(k) = \prod_{(i,j)\in k} \frac{p_{ij}}{D_{ij}^2} \right\}$$

$$\Longrightarrow \quad v(S \cup T) =$$

$$\frac{1}{\triangle t} \sum_{(a,b)\in SD, a,b\in S\cup T} Q_{ab} \cdot \max_{k\in P_{ab}(S\cup T)} \left\{ t(k) = \prod_{(i,j)\in k} \frac{p_{ij}}{D_{ij}^2} \right\}.$$

The larger the coalition becomes, the more number of possible routing paths can be discovered. Accordingly, the maximal reliability increases when obtained from a larger set. On the premise of certain amount of required transmission data packets and certain time interval, the expected throughput of the larger coalition will be increased. That is, $v(S \cup T) \geq v(S) + v(T)$ is satisfied. □

In summery, we can distribute the total payoff of the coalition to each players according to Shapley Value Equation.

5.4.4 Game Rules and Threatening Mechanism

There might be some misbehaving nodes in the network but they will perform bad behaviors on the premise of not compromising their own behalf. On the basis of the predefined throughput characteristic function, we can design a set of game rules so as to implementing a threatening mechanism.

The strategy space of each node is $\{join, notjoin\}$. That is, the node either joins into a coalition or doesn't join into the coalition. The game rules are:

1. A node will join into a coalition only if it can get more payoff share than it stands individually.

2. A node will deviate from the current coalition and join into another coalition only if it can get more payoff share there than that of here.

3. A coalition will refuse to admit a node if the node cannot increase the total payoff of the coalition.

4. A coalition will exclude a node if the node cannot benefit the coalition or even damage the total payoff of the coalition.

5. Nodes who are finally failed to join into any coalition will be denied from the network.

These rules form a threatening mechanism in the network. Take the selfish nodes for example, they do not forward others' routing or data packets in order to save their own communication and computation resource. But under the condition of the above game rules, they will hardly be admitted into coalitions such that their own traffic cannot be delivered to the destination because of poor reputation. This is a potential threat for them.

Before joining into or deviating from a coalition, every node will compare the possible payoff share it will obtain with the current payoff share it has obtained. Then following the above game rules, a new coalition topology will be formed.

5.5 Coalition Formation Procedure

5.5.1 Coalition Formation Algorithm

As a further refinement, we are going to design a coalition formation algorithm that satisfy the definition of $v(S)$. We introduce Gale-Shapley Deferred Acceptance Algorithm (DAA) [22] to help nodes forming coalitions. This algorithm was proposed to solve the stable marriage problem and was proven that at the end of the algorithm, no one wants to switch partners to increase his/her happiness. In this work we firstly apply this algorithm to the coalition formation of wireless networks.

The coalition formation procedure is conducted iteratively by all the nodes in the network. It is described in Algorithm 4 and 5.

Algorithm 4: Coalition Formation Algorithm

```
while timeleft≠ 0 do
    for 0 to Δt do
        Normal routing and forwarding process, gain experience;
    Update direct probability p_ij, distance D_ij;
    Compute t(i, j) for every neighbor of i, and sort them;
    foreach coalition S containing any src a or any dst b do
        Findmatch(S);
    foreach node i not in any coalition S do
        Degrade i;
    Timeleft ← timeleft - Δt;
```

Algorithm 5: Find Matching Partner Algorithm

```
Findmatch(S) {
foreach a ∈ S do
    Chose first several preferences with highest t(a, .);
    Conduct DAA algorithm to find partner a' of a;
    Add new match {a, a'} to coalition S;
    Update all members' routing table and corresponding state of S;
}
```

5.5.2 Integration with Wireless Routing Protocols

The proposed coalitional game model can be integrated with all kinds of routing protocols, such as AODV [68], DSR [34], DSDV [65] and so on, of many types of wireless network, e.g. mobile ad hoc networks and wireless sensor networks. We take AODV routing protocol for example to illustrate how to integrate the game model with routing behaviors.

Firstly, we extend the original routing table of AODV protocol by adding four fields:

1. Number of members in the coalition that the concerned entry has joined into;

2. Direct communication probability from the current node to the concerned entry;

3. Indirect communication probability from the current node to the concerned entry;

4. Distance between the current node with the concerned entry.

Secondly, besides original routing request and reply (RREQ, RREP) packet types, several new control packet types are defined, such as Matching REQuest/REPly (MREQ, MREP) and Probability REQuest/REPly (PREQ, PREP) and so on. MREQ/MREP are matching request and reply packets to exchange the matching preference list and notify the matching result. PREQ/PREP packets are used to collect neighbors' recommendation of communication probability.

Thirdly, a new dedicated timer must be set up to control the iteration of coalition formation procedure.

5.6 Theoretical Analysis

We now theoretically analyze our model from two aspects: 1) Speed of convergence and size of coalition and 2) Non-emptiness of core [62]. We will show that the coalition formation speed is fast and the size of the coalition keeps growing and even a grand coalition can be reached. We will also show that cooperation is made attractive from the individual point of view because the cost of participating in the network operation is compensated with a higher reputation value. On the other hand, when the number of cooperating nodes increases, the cost for participation is compensated by a more reliable network that in turn increases the benefit of cooperation.

5.6.1 Speed of Convergence and Size of Coalition

From the coalition formation algorithm we can see that at each round of formation, every coalition member tries to find a partner. So the coalition size is increased almost at a rate of two times. Therefore, the speed of coalition formation is fast, which means the convergence time of formation is short. And the size will keep growing until a grand coalition is reached or all misbehaving nodes are identified.

5.6.2 Non-emptiness of Core

The stable status of coalitional game is that no coalition can obtain a payoff that exceeds the sum of its members' current payoffs, which means no deviation is profitable for all of its members. The core is the set of imputation vectors which satisfies the following two conditions:

$$\begin{aligned} &1. \quad \sum_{i=1}^{n} x_i = v(N) \\ &2. \quad \sum_{i \in S} x_i \geq v(S), \forall S \in 2^N \end{aligned} \tag{5.16}$$

The first condition is to guarantee the efficiency of payoff allocation. N is called the grand coalition. The second condition ensures that no coalition is unhappy, and it is a very strong constraint. We can see that whether the core is nonempty or not is determined by the definition of characteristic function $v(S)$ and the payoff distribution method among the coalition members.

We have defined the throughput characteristic function and the payoff allocation method among coalition in previous sessions. Based on the definition, we now discuss the several situations of the core.

Suppose that we have an allocation profile $x(S) = \sum_{i \in S} x_i(S), \forall S \in 2^N$. The relation between $x(S)$ and $v(S)$ has two situations.

$x(S) < v(S)$

In this situation, the core is empty. But when $|S| = 1$, which means the node do not belong to any coalition, this node cannot form a source-destination pair and consequently no throughput can be obtained. While considering the Shapley value in Eq. (5.15), the payoff share is always larger than 0, which implies that rational nodes always have incentive to cooperate with each other.

$x(S) \geq v(S)$

If this situation can be reached, the core is nonempty. The stable outcome will last for a certain time under certain conditions. However, in the mobile ad hoc network, there are some factors that will destroy the current equilibrium and enforce the network to re-organize again. The first factor is that not all the nodes are reasonable, and the second one is the incompleteness of information due to the nodes mobility, underlying detection mechanism and so on.

If that is the case, we can still observe $x(S) - v(S)$. The difference between them means how hard the core status will be destroyed. The larger the difference, the lower the probability that coalition S will deviate. Then we can get the probability that the core would remain as follows:

$$p_{keep} = 1 - \prod_{S} [\ 1 - p_{deviate}(x(S) - v(S))\ ] \tag{5.17}$$

where $p_{deviate}(x(S) - v(S))$ can be approximated as an exponential distribution for further investigation.

5.7 Summary

We propose a coalitional game model for the security issue of wireless networks. A new method of quantify security concept is proposed by designing a security characteristic function. We theoretically prove that in the role of our security charateristic function, the game players have incentives to join coalitions then obtain higher security payoffs, and the game will reach a stable core status so that the malicious nodes can be identified correctly and effectively. We also verify our model by simulation. The results show that all the malicious nodes are differentiated after a certain number of coalition formation rounds. In the future we will improve our security value function and combine it with the routing protocol of wireless networks.

We also define a throughput characteristic function which not only describes the network performance metric but also expresses the quantification of security metric. A payoff distribution method for coalition members to fairly share the utility value is proposed. After that, a coalition formation algorithm is designed and integrated with routing protocols of wireless networks. From the theoretical analysis, we conclude that the convergence of coalition formation is quite fast and the coalition size can be very large, which means nodes are ready to form into coalitions and perform good behaviors, so that we can prevent bad behaviors and identify misbehaving nodes effectively. We also discuss the nonemptiness of stable status of coalition formation and conclude that the core in wireless networks is difficult to achieve and easy to be destroyed. But we can then still investigate the node deviation probability and get certain network properties for future applications.

Chapter 6

Coalitional Game Model for Selfishness Issues

In this chapter, we focus on the selfishness issues of wireless networks. We model the routing and forwarding procedures as a cooperative coalitional game with transferable payoff, which is not the usual non-cooperative game like others. An incentive routing and forwarding scheme is proposed, which integrates a reputation system with a monetary payment mechanism to encourage nodes cooperation in the network. The reputation system we employed, for the first time in the literature, is a heat diffusion model which provides us a way of combining the direct and indirect reputation together and propagating the reputation from locally to globally. We also analyze that the game has a non-empty *core*, which is a stable status in cooperative game just like the Nash equilibrium in a non-cooperative game. From the evaluation we can see that the cumulative utility of nodes increases when nodes stay in the core.

6.1 Background of Heat Diffusion on Weighted Directed Graph

6.1.1 Motivation

A reputation system usually needs to address two problems: 1) how to combine subjective direct reputations with indirect reputations from neighbors to make them become more objective; and 2) how to propagate the reputation from locally to globally. Previously there are different solutions to these problems such as [59] and [46]. In this work, we will employ the heat diffusion model to fulfill the requirements.

In nature, heat always flows from high temperature positions to low temperature positions via conductive media. A heat diffusion model describes this phenomenon that heat can diffuse from one point to another through an underlying manifold structure in a given time period. The higher the thermal conductivity of the medium, the easier the heat flows, which implies that the two end points have some cohesive relations. Diffusion behaviors are also affected by the underlying geometric structures. Some achievements have been made based on the heat diffusion model such as classification in machine learning field, page ranking in information retrieval [84] and marketing candidates selection in social computing [52], but to our best knowledge, there is no previous work that has been performed on the incentive routing in wireless networks.

We see that in the process of heat diffusion, each node's heat comes from all of its incoming links and diffuses out to its successors as long as it can. If we diffuse heat on a weighted directed graph, the amount of heat a node can get depends not only on the heat of its neighbors but also on the weights of the links connecting them. The higher the weight, the more thoroughly the heat can be diffused. Therefore, if we let the weight be the direct reputation value of the link, then the amount of heat will be the overall reflection of the

underlying reputation information. The course of heat diffusion through all possible links can also be deemed as a propagation of the reputations.

6.1.2 Heat Diffusion on Weighted Directed Reputation Graph

We construct a heat diffusion model on the reputation graph $G = (M, E, R)$, where $M = \{1, 2, ..., m\}$ is the node set. $E = \{(i, j)|\ i$ and j are in communication range and the transmission direction is from i to $j\}$. The heat only flows from i to j if $(i, j) \in E$. R is the reputation set $\{r_{ij}|\ r_{ij}$ is the direct reputation of edge $(i, j)\}$. We use $f_i(t)$ to describe the heat value of node i at time t, beginning from an initial distribution of heat $f_i(0)$ at time zero. $\vec{f}(t)$ denotes the vector consisting of $f_i(t)$.

The heat diffusion modelling is as follows. Suppose, at time t node i diffuses $H_D(i, t, \Delta t)$ amount of heat to its subsequent nodes. We assume that: a) the heat H_D is proportional to the time period Δt; b) H_D is proportional to the heat of node i; c) each node has the same ability to diffuse heat; and d) node i intends to distribute H_D uniformly to each of its subsequent nodes, but the actual heat it can diffuse is proportional to the corresponding reputation weight of the edge. On the basis of the above considerations, we state that node i will diffuse $\lambda p_{ik} f_i(t)\Delta t/l_i$ amount of heat to each of its subsequent node k, where l_i is the outdegree of node i and λ_j is the thermal conductivity, which is the heat diffusion coefficient representing the heat diffusion ability. In the case that the outdegree of node i is zero, we assume that this node will not diffuse heat to others. Then the total amount of heat node i will diffuse is $\sum_{k:(i,k)\in E} \lambda p_{ik} f_i(t)\Delta t/l_i$.

On the other hand, each node i receives $H_R(i, j, t, \Delta t)$ amount of heat from j during a period of Δt. We also have the following assumptions: a) H_R is proportional to the time period Δt; b) H_R is proportional to the heat of node

j; c) H_R is zero if there is no link from node j to i. Based on the above considerations, we obtain $H_R(i,j,t,\Delta t) = \lambda_j f_j(t)\Delta t$. As a result, the heat that node i receives between time t and $t+\Delta t$ will be equal to the sum of the heat flowing from all its neighbors pointing to it, which is $\sum_{j:(j,i)\in E} \lambda_j f_j(t)\Delta t$. Since the amount of heat that j diffuses to i should be equal to the amount i receives from j, we have $\lambda p_{ji} f_j(t)\Delta t/l_j = \lambda_j f_j(t)\Delta t$. So we get $\lambda_j = \lambda p_{ji}/l_j$. To sum up, the heat difference at node i between time t and $t+\Delta t$ will be the amount of heat it receives deduced by what it diffuses. The formulation is therefore:

$$f_i(t+\Delta t) - f_i(t) = \lambda \left(\sum_{j:(j,i)\in E} \frac{p_{ji}}{l_j} f_j(t) - \mu_i \sum_{k:(i,k)\in E} \frac{p_{ik}}{l_i} f_i(t) \right) \Delta t, \qquad (6.1)$$

where μ_i is a flag to identify whether node i has any outlinks. If node i does not have any outlinks, $\mu_i = 0$; otherwise, $\mu_i = 1$. To find a closed form solution to Eq. (6.1), we then express it in a matrix form:

$$\frac{\vec{f}(t+\Delta t) - \vec{f}(t)}{\Delta t} = \lambda \vec{H} \vec{f}(t), \text{where} \qquad (6.2)$$

$$H_{ij} = \begin{cases} p_{ji}/l_j, & (j,i) \in E, \\ -(\mu_i/l_i) \sum_{k:(i,k)\in E} p_{ik}, & i = j, \\ 0, & otherwise. \end{cases} \qquad (6.3)$$

In the limit $\Delta t \to 0$, Eq. (6.2) becomes

$$\frac{d}{dt}\vec{f}(t) = \lambda \vec{H} \vec{f}(t). \qquad (6.4)$$

Solving the above equation, we get

$$\vec{f}(t) = e^{\lambda t \vec{H}} \vec{f}(0), \qquad (6.5)$$

where $e^{\lambda t \vec{H}}$ can be extended as:

$$e^{\lambda t\vec{H}} = \vec{I} + \lambda t\vec{H} + \frac{\lambda^2 t^2}{2!}\vec{H}^2 + \frac{\lambda^3 t^3}{3!}\vec{H}^3 + \cdots . \tag{6.6}$$

The matrix $e^{\lambda t\vec{H}}$ is called the diffusion kernel, showing that the heat diffusion process continues infinite times from the initial heat diffusion step.

However in the large wireless network computation of $e^{\lambda t\vec{H}}$ directly is very time-consuming. To improve the scalability and performance, we employ the following discrete approximation equation to computer the heat diffusion:

$$\vec{f}(t) = \left(\vec{I} + \frac{\lambda t}{q}\vec{H}\right)^q \vec{f}(0), \tag{6.7}$$

where q is a positive integer representing the number of iterations before the diffusion converge.

6.2 Technical Descriptions

Before presenting our incentive scheme and coalitional game we first give some technical notations. Our game is based on the bi-directional weighted graph $G = (M, E, P)$ described in Section 6.1. Suppose that s is the source node and d is the destination node, then the player set of this coalitional game is $N = M \setminus \{s, d\}$. Coalition is denoted by any non-empty subset $T \subseteq N$, and the overall payoff of the coalition is denoted by $v(T) \in \mathbb{R}$. Then the game is expressed by $\Gamma =< N, v >$. Players will form into coalitions to help establishing the highest effective path between s and d with the lowest cost under the constraint that each intermediate node's heat is higher than a threshold θ. If there's a tie in the total cost, s will break the tie by choosing the path with the highest heat. The source can freely choose the value of θ to meet its requirement on reputation. The larger the value θ is, which means the source has a higher demand on reputations, the higher payments it will expend. All the paths established inside the coalition T connecting s and d compose the path set $P_{sd}(T)$.

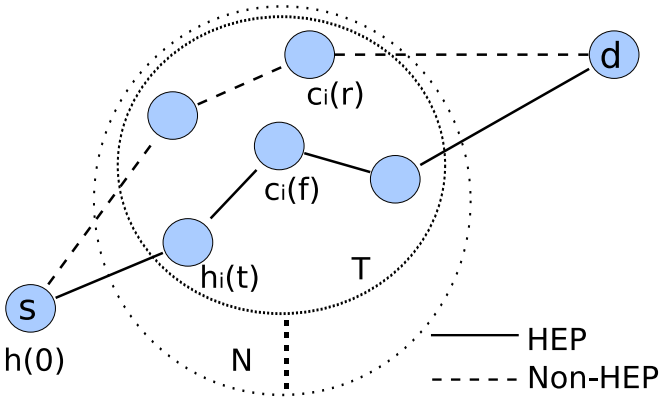


Figure 6.1: Illustration of notations of the coalitional game.

Initially, s will load a certain amount of initial heat $f(0)$ and diffuse it on the reputation graph, then at time t, each node will be diffused $f_i(t)$ amount of heat. Correspondingly each source s has an initial balance of $h(0)$, and the payment to each node $h_i(t)$ is paid by it according to $f_i(t)$. Every node evolving in the routing or forwarding procedure will cost its energy. Since the cost for sending/receiving routing and data packets are different [20], and the cost for data transmission is usually larger than that of routing packets transmission, we denote the routing and forwarding cost respectively by $c_i(r)$ and $c_i(f) \in \mathbb{R}^+$, and $c_i(r) < c_i(f)$ for all $i \in N$. Please see Fig. 6.1 for the illustration of notations.

6.3 Incentive Routing and Forwarding Scheme

The basic idea of achieving incentives is that nodes will be paid when they help others forwarding data or routing packets. Unlike other payment schemes that reward the nodes according to their claimed cost, our incentive routing and forwarding scheme pays the nodes by their reputations. The higher a node's reputation is, the higher payment it can get. The payment is given by the source node. The payment may be in the form of virtual currency like [10] or any other practical form. In our work we assume that there is such a payment

form and a payment operation daemon in the network.

In the scheme, the source node s will originate the heat diffusion process starting from itself. Then after collecting the forwarding cost of all the players, s will compute the lowest cost path under the constraint that the diffused heat of each intermediate node is higher than its assigned threshold θ. We call this path as the highest effective path (HEP). Selfish or unreliable nodes will be degraded with respect to their direct reputations by their neighbors while enthusiastic or reliable nodes will be upgraded in their reputations. The extent of increasing or decreasing a node's reputation depends on the functions it takes. Forwarding data packets will get higher reputation increments than forwarding routing packets. Correspondingly, not forwarding data packets will get heavier punishment on reputation than not forwarding routing packets. The utility a node gets in one session is the amount of payment it receives from the source node, subtracted by the cost it expends for forwarding data or routing packets. The scheme is summarized in Algorithm 6.

Algorithm 6: Incentive Routing and Forwarding Scheme

Input: Source s, destination d, reputation graph G, heat threshold θ

Output: HEP_N

foreach $i \in N$ **do**
    i claim its forwarding cost $c_i(f)$ to s;
    $f_i(0) = 0$;
$f_s(0) = f(0)$;
Execute the heat diffusion process $\vec{f}(t) = e^{\lambda t \vec{H}} \vec{f}(0)$;
s chooses the highest effective path to d with the lowest cost, subject to $f_i(t) \geq \theta$. If there is a tie, s selects the one with the highest heat;
... Data transmission process; ...
s pays $h_i(t)$ to each i according to $f_i(t)$;
s adjusts its heat threshold θ;
Updates reputation graph G;

Under the effect of the algorithm, we can see that by behaving cooperatively a node can get higher and higher reputations, thus the payment to it will also be increased, so as to the individual utility. To earn more utility, the node will then try to improve its reputation by actively forwarding for others.

Sometimes for one session a node's utility obtained for forwarding routing packets may be higher than that of forwarding data packets. But the increasing acceleration of the latter is larger than that of the former because of the different updating way of reputation. So in the long run the cumulative utility of the node in the latter will exceed that in the former. If a node declare a higher cost than its actual forwarding cost to avoid being selected in the HEP, it will suffer the same situation. The above are some intuitive thoughts behind the scheme; for precise analysis we give it in Section 6.4.

We also want to address that the underlying reputation representing and updating mechanism is not limited to the heat diffusion model. Any reputation system that can accomplish the combination, globalization and quantification of reputations can fit into our incentive scheme.

6.4 Our Coalitional Game

In this section we will analyze the proposed incentive scheme by modelling the routing and forwarding procedure as a cooperative coalitional game with transferable payoff. Furthermore, we show that the game has a non-empty core.

6.4.1 Value Function of the Coalition

The value or characteristic function is the key component of a coalitional game. For each coalition T, the value $v(T)$ is the total payoff that is available for division among the members of T. It can also be interpreted to be the most payoff

that the coalition T can guarantee independent of the behavior of the coalition $N \setminus T$ [62]. Now we will define the value function of the coalition in our game. As described in Section 6.2, s is the source node and d is the destination node. The player set is $N = M \setminus \{s, d\}$. When nodes join together into one coalition, they will establish one or more paths between s and d, each of which gives the coalition a collective payoff $w_P(T)$, where $P \subseteq T$ represents a path. The collective payoff comes from each member's contributions of their reputation-based payments h_i, then subtracted by the costs they bear for performing routing or forwarding behaviors. For those who are only involved in the routing discovery process the cost is $c_i(r)$, and for those who have been selected in a path the cost should be $c_i(f)$. So for each path $P \subseteq T$, the corresponding payoff function for the coalition is $w_P(T) = \sum_{i \in T} h_i - \sum_{i \in P} c_i(f) - \sum_{i \notin P} c_i(r)$. Among all these payoffs, we say that the characteristic worth or the value of the coalition $v(T)$ should be the maximal collective payoff it can guarantee, which is:

$$v(T) = \max_{P \subseteq T} \left(\sum_{i \in T} h_i - \sum_{i \in P} c_i(f) - \sum_{i \notin P} c_i(r) \right). \tag{6.8}$$

We call the path that has the maximal $w_P(T)$ as the highest effective path HEP_T, so alternatively we can write $v(T) = \sum_{i \in T} h_i - \sum_{i \in HEP_T} c_i(f) - \sum_{i \notin HEP_T} c_i(r)$. But if there is no such path inside the coalition, the coalition is inessential and worths nothing, and the value of it is 0. Then formally, we have the definition of $v(T)$ as follows.

Definition 16 (Value Function of A Coalition) *The value of any coalition $T \subseteq N$ is 0 when there is no path between s and d inside T. That is: $v(T) = 0$, if $P_{sd}(T) = \phi$. Otherwise, $v(T)$ is:*

$$v(T) = \sum_{i \in T} h_i - \sum_{i \in HEP_T} c_i(f) - \sum_{i \notin HEP_T} c_i(r), \quad \text{if } P_{sd}(T) \neq \phi \tag{6.9}$$

6.4.2 Non-emptiness of the Core

The key issue of a cooperative game is regarding how to divide earnings inside the coalition in some effective and fair way. The adequate allocation profile is then called a solution, which is a vector $\vec{x} \in \mathbb{R}^N$ representing the allocation to each player when a grand coalition is formed. The grand coalition means all the players form into one coalition. The core is one of the solution concepts for cooperative games. If a coalitional game's core is non-empty, it means that no coalition can obtain a payoff that exceeds the sum of its members' current payoffs, which means no deviation is profitable for all of its members [62]. Theoretically the core is the set of imputation vectors which satisfies the following three conditions:

$$\begin{aligned} &1. \quad x(i) \geq v(i) \\ &2. \quad x(T) \geq v(T), \ \ \forall T \in 2^N \\ &3. \quad x(N) = v(N), \ N \text{ is the player set} \end{aligned} \tag{6.10}$$

where $x(i)$ is the payoff share of node i in this game, $x(T) = \sum_{i \in T} x(i)$, and $x(N) = \sum_{i \in N} x(i)$.

The core of a coalitional game is possibly empty. Next we will analyze in which condition our game has a non-empty core, and what the possible core is. We derive the following theorem.

Theorem 3 Under the condition of $h_i \geq c_i(f)$ for each player i, the payoff profile x is in the core of the coalitional game where

$$x(i) = \begin{cases} h_i - c_i(f), & i \in HEP_N \\ h_i - c_i(r), & i \notin HEP_N \end{cases} \tag{6.11}$$

Proof 3 Firstly, check the first requirement of Eq. (6.10). Under the condition of $h_i \geq c_i(f)$, we have $x(i) = h_i - c_i(f) \geq 0$. When the coalition has only one member i, the value of it would be 0 if i cannot establish a path between s

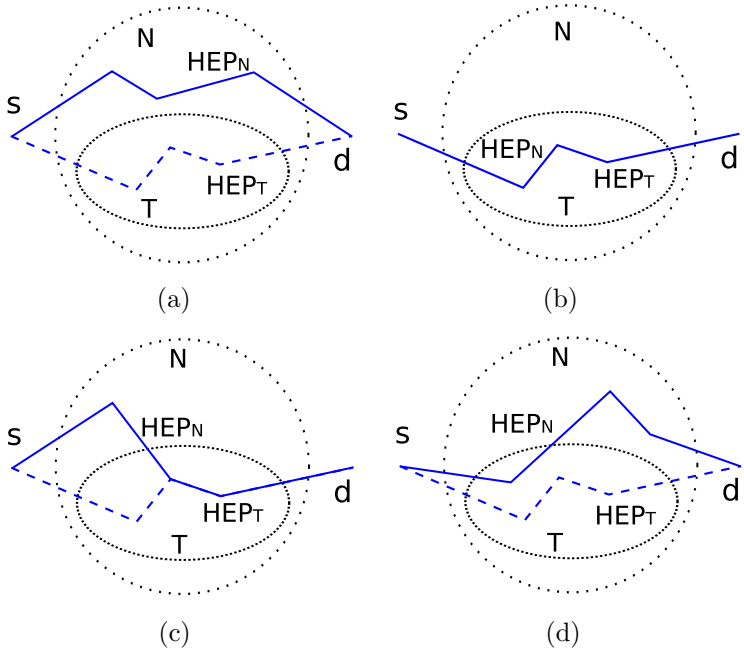


Figure 6.2: Examples of different HEP situations.

and d. That is $v(i) = 0$. Thus $x(i) \geq v(i)$ holds. If i can connect s and d, then $v(i) = h_i - c_i(f)$ as said by Def.16. In that case $x(i) = v(i)$ which also meets the first requirement.

Secondly, from Eq. (6.11) and Def. 16, we have $x(N) = \sum_{i \in N} x(i) = \sum_{i \in HEP_N}(h_i - c_i(f)) + \sum_{i \notin HEP_N}(h_i - c_i(r)) = \sum_{i \in N} h_i - \sum_{i \in HEP_N} c_i(f) - \sum_{i \notin HEP_N} c_i(r) = v(N)$. So the third requirement of Eq. (6.10) also holds.

Thirdly, to prove $\vec{x}$ satisfies the second requirement $x(T) \geq v(T)$, we will list and analyze all of the different HEP situations in the grand coalition N and an arbitrary coalition T. For those coalitions without paths inside, the values of them are 0, so we easily get $x(T) \geq v(T) = 0$. For other coalitions, there are totally four kinds of situations as illustrated in Fig. 6.2.

The proof for these four situations are similar. Because of the space limit, we only prove the most complicated situation in Fig. 6.2(d) here. In this case, when the grand coalition N is formed, the new HEP_N is different from HEP_T and part of HEP_N is inside T. For clarity we first give the following notations

for Fig. 6.2(d). We let $A = HEP_N \cap T$, $B = HEP_N \cap (N \setminus T)$, $C = HEP_T \cup A$, $D = T \setminus C$, and $E = N \setminus (HEP_N \cup HEP_T)$. According to Def. 16 and Eq. (6.11), we have:

$$v(T) = \sum_{i \in T} h_i - \sum_{i \in HEP_T} c_i(f) - \sum_{i \in A} c_i(r) - \sum_{i \in D} c_i(r)$$

$$x(T) = \sum_{i \in T} h_i - \sum_{i \in A} c_i(f) - \sum_{i \in HEP_T} c_i(r) - \sum_{i \in D} c_i(r)$$

$$x(T) - v(T) = \left[\sum_{i \in HEP_T} c_i(f) - \sum_{i \in HEP_T} c_i(r)\right] - \left[\sum_{i \in A} c_i(f) - \sum_{i \in A} c_i(r)\right] \quad (6.12)$$

HEP_N and HEP_T are two paths connecting s and d inside the grand coalition N, and HEP_N dominates HEP_T. So based on Eq. (6.8), we have $v_{HEP_N}(N) \geq v_{HEP_T}(N)$. Through deduction we get:

$$\sum_{i \in N} h_i - \sum_{i \in A} c_i(f) - \sum_{i \in B} c_i(f) - \sum_{i \in HEP_T} c_i(r) - \sum_{i \in E} c_i(r)$$

$$\geq \sum_{i \in N} h_i - \sum_{i \in HEP_T} c_i(f) - \sum_{i \in A} c_i(r) - \sum_{i \in B} c_i(r) - \sum_{i \in E} c_i(r)$$

$$\Longrightarrow \sum_{i \in HEP_T} c_i(f) - \sum_{i \in HEP_T} c_i(r) \geq \left[\sum_{i \in A} c_i(f) - \sum_{i \in A} c_i(r)\right] + \left[\sum_{i \in B} c_i(f) - \sum_{i \in B} c_i(r)\right]$$

$$\sum_{i \in HEP_T} c_i(f) - \sum_{i \in HEP_T} c_i(r) \geq \left[\sum_{i \in A} c_i(f) - \sum_{i \in A} c_i(r)\right] + \left[\sum_{i \in B} c_i(f) - \sum_{i \in B} c_i(r)\right]$$

$$> \sum_{i \in A} c_i(f) - \sum_{i \in A} c_i(r)$$

Then substitute in Eq. (6.12), we get $x(T) \geq v(T)$. So the second requirement is satisfied. In summary, under the condition of $h_i \geq c_i(f)$ for each player i, the proposed payoff profile $\vec{x}$ is in the core of this coalitional game. □

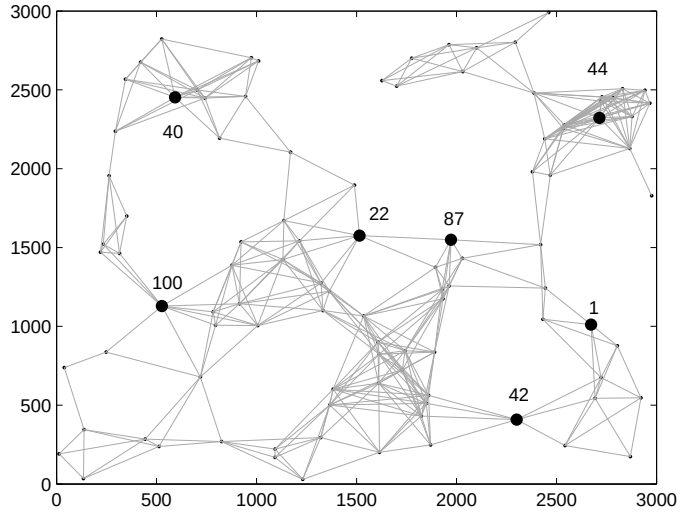
We can see that if only the payment a node gets based on its reputation is larger than the cost it needs to forward data packets, the core of this coalitional game exists. Nodes who want to get more payoff share x_i must try to improve its reputation by helping others forwarding or increasing its link reliability, so that it can get more diffusion of the heat-based payment. In this way a virtuous cycle can be created.

6.5 Evaluations

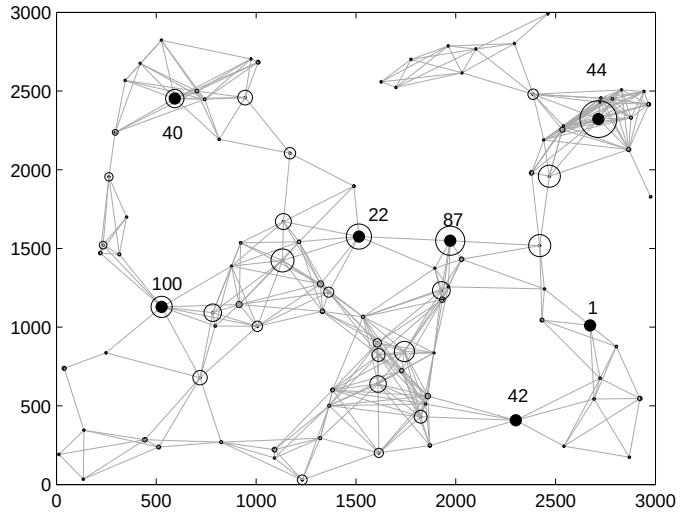
We have theoretically proved that our incentive scheme guarantees the existence of the core when modelled as the coalitional game. Now we will evaluate the scheme in two aspects through a few experiments: 1) how is the general overview of all the nodes' utility and how does the network topology affect the distribution of it; and 2) how the nodes' cumulative utilities and balances evolve over time.

We conduct the evaluation on a randomly generated wireless topology with 100 nodes scattering in an area of 3000 by 3000 meters. The radio range is set to 422.757 meters. The topology is shown in Fig. 6.3(a). There is a line connecting two nodes when they are in the communication range of each other. We label some representative nodes for further illustration. Each node has an initial balance of 100 and each directed link has a local reputation value as the weight. At each round we randomly select a source-destination pair and the source s perform the incentive routing and forwarding algorithm. We assign the parameter λ in the heat diffusion equation as 1. The evaluation runs for 1000 seconds and we observe the utility and balance of each node every second.

Our first evaluation shows the overview utility at the end of the experiment in Fig. 6.3(b). The circles around the nodes represent the cumulative utility of that node. The diameter of the circle is proportional to the amount of the utility. We observe that in general nodes in the high density area also have

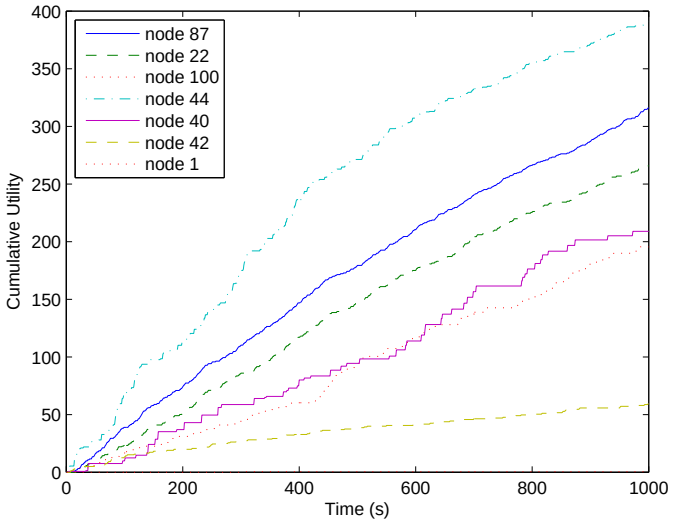


(a)

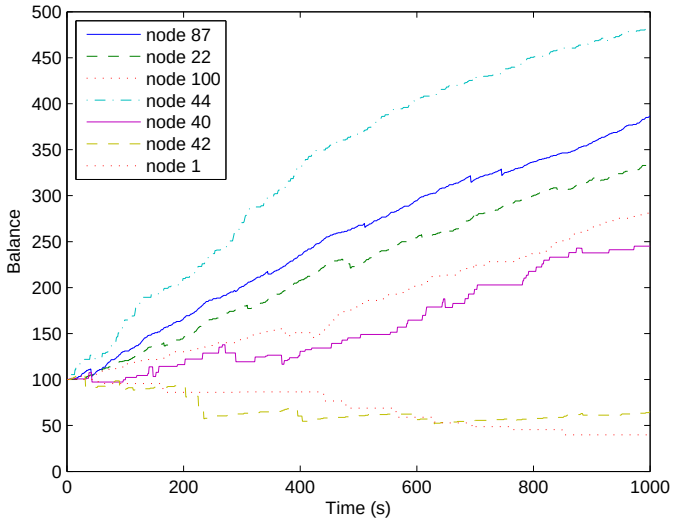


(b)

Figure 6.3: Network topology and overview of nodes' utilities.

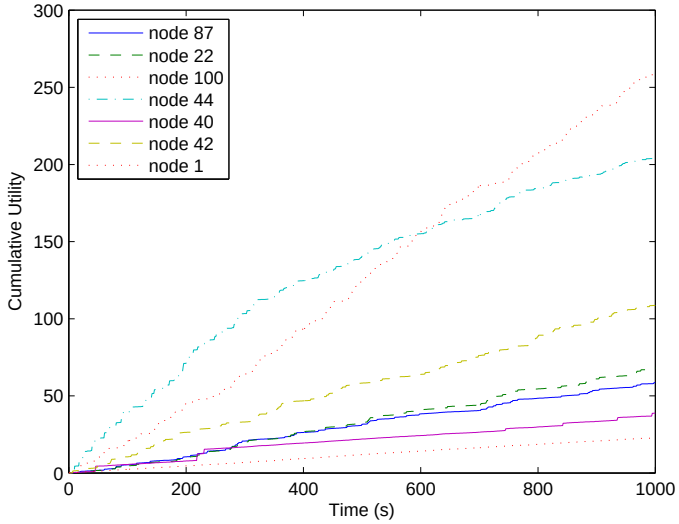


(a)

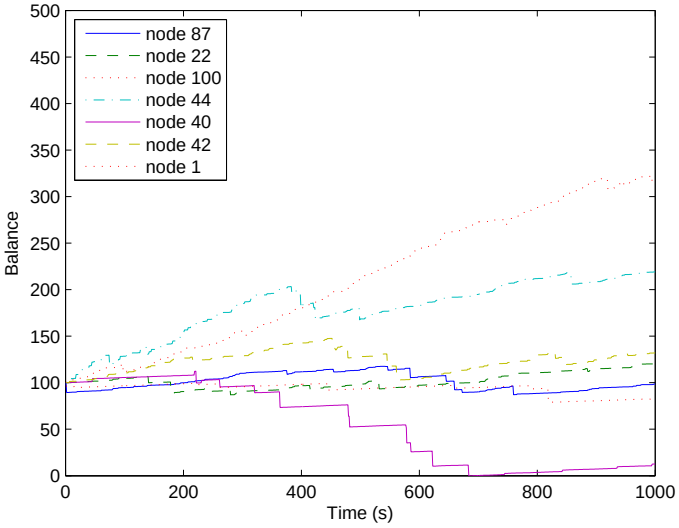


(b)

Figure 6.4: Cumulative utility and balance of nodes as a function of simulation time.



(a)



(b)

Figure 6.5: Cumulative utility and balance of nodes as a function of simulation time with taxation.

large circles around them (like node 44), and on the other hand, nodes in the sparse area usually have indistinctive circles.

Our second evaluation starts from the core of the coalitional game. Figure 6.4(a) and Fig. 6.4(b) show the cumulative utilities and balances of several typical nodes respectively over the simulation time. The balance of a node may fall below the initial balance (like node 42) because the nodes have their own data transmission requests and what they earn cannot compensate what they pay.

To solve the problem that when nodes are in the boundary region where there would be less chance for them to earn utilities since less routing and forwarding traffic is requested on them and less heat is diffused to them, we employ a taxation policy. In this policy, nodes which have earned payments will be levied tax, then the taxation will be distributed among all nodes. In this senario, new traffic source-destination pairs are randomly seleted. From Fig. 6.5(a) we can observe that with the taxation policy nodes in the boundary or low density area of the network (like node 1) can still cumulate utilities instead of earning nothing. We can also find from Fig. 6.5(b) that the balance of some nodes (like node 41) will rebound back after dropping initially so that a wave curve appears. This is because the taxation policy prevents those nodes from starvation and they can gain chances to earn utilities when there are new traffic requests.

6.6 Summary

In this chapter, we present a novel incentive routing and forwarding scheme which combines reputation system and payment mechanism together to encourage nodes to cooperate in wireless ad hoc networks. Besides, we design our reputation system based on a heat diffusion model for the first time in the literature. The heat diffusion model provides us a way of combining the direct

and indirect reputations together and propagating the reputation from locally to globally. Further, instead of using the non-cooperative game method, we model and analyze our incentive scheme using a coalitional game. We further prove that under a certain condition this game has a non-empty core. Through the evaluation we can see that the cumulative utility of nodes increases when the nodes stay in the core. In the future we will consider to apply other underlying reputation systems to our incentive scheme.

Chapter 7

Conclusion

A mobile ad hoc network (MANET) is a kind of wireless network without centralized administration or fixed network infrastructure. Self-organization, decentralization and openness are the main advantages of the mobile ad hoc network. However these characteristics also introduce insecurity. In this mobile environment, nodes are lacking of sufficient information about each other, which increases the risk of being compromised from either outside or inside. Besides, nodes in this kind of network may belong to different self-interested individuals and have limited power and bandwidth; therefore, in this thesis, they tend to be too selfish to forward packets for others.

The original routing protocol of MANET is designed without considering defensive mechanisms, thus many security schemes from different aspects have been proposed to protect the routing or data packets in the communications. However, most of these schemes assume trusted third parties or centralized servers to issue digital certificates or cryptographic keys. Moreover, generating or verifying digital signatures at every routing packets will introduce huge computation overhead. Therefore, we introduce an idea about "trust modeling" and propose a trusted routing protocol based on the trust relationships among nodes.

We derive our trust model from subjective logic. In our trust model, trust is represented by a three-element triad called *opinion*. The three elements are

belief, *disbelief* and *uncertainty*, which means the probabilities that a node can be trusted or distrusted, and the probability of uncertainty about the trustworthiness of a node, respectively. The uncertainty value fills the void in the absence of both belief and disbelief, which reasonably expresses the changefulness property of MANET. In this model, trust combination algorithms and trust mapping functions are provided. The combination algorithms can aggregate different opinions together to get a new recommendation opinion. Two operators are employed in the combination: discouting and consensus, where the former combines opinions along a path, and the latter combines opinions across multiple paths. The mapping functions offer the trust mapping between the evidence space and the opinion space. The cumulative evidences about communication status among nodes can be calculated into opinion values, and the combined opinion can also be mapped back to the evidence opinion and get an indicative number of evidences.

Based on this trust model, we design our trusted routing protocols for MANET called TAODV on top of Ad Hoc On-demand Distance Vector (AODV) routing protocol. We extend the routing table and the routing messages of ADOV with trust information which can be updated directly through monitoring in the neighborhood. The more the positive events are collected, the higher the belief value in the opinion will be. Besides, we also present a trust recommendation protocol. When performing trusted routing discovery, unlike those cryptographic schemes that perform signature generation or verification at every routing packet, we just combine the recommended opinions together and have a judgment on each element of the new opinion. Only if the uncertainty value in the opinion is higher than a threshold, will the cryptographic routing scheme take effect. When nodes have conducted more and more number of communications, the uncertainty value will become lower and lower, which means the belief or the disbelief value will dominate the trust judgment, so that the chance of performing cryptographic routing behaviors will

get lower and lower. In this way the computation overhead can be largely reduced, and the trustworthiness of the routing procedures can be guaranteed as well. Through simulation we can see that the bad nodes are clearly separated from the good nodes, and we do not introduce much overhead as other cryptographic schemes do.

Security issues and selfishness issues of wireless networks can also be formulated with game theory. Game theory is a branch of applied mathematics that employs models to study interactions (conflict and cooperation) among game players with formalized incentive structures. It has been found applications in a variety of fields in recent years. Game theory has been traditionally divided into *cooperative game theory* and *non-cooperative game theory*. In this thesis, we model the security and selfishness issues of wireless networks as three different games, either in non-cooperative form or in cooperative form.

First, we formulate the security issues of wireless networks as a non-cooperative game. The interactions between the attacker and the regular node are modeled into two game trees according to the type of the node. The game is in the form of signaling game. Through the theoretical analysis we finally obtain a bound for the value of payoff assignment, which can guide the design of payment schemes and incentive routing protocols.

Second, we model the security issues of wireless networks with a cooperative game, called *coalitional game*. The key points of coalitional game formulation are defining a characteristic function for each coalition then providing a payoff allocation solution inside the coalition, based on which nodes can form coalitions to maximize their utilities. We design two value functions, security characteristic function and throughput characteristic function, from the aspects of achieving maximum security for a coalition and achieving maximum throughput for a coalition, respectively. We also present the coalition formation algorithm and the integration of the algorithm with existing routing protocols. Theoretically we analyze the existence of the stable *core* status of

the game, and the convergence speed of the stable status. From the simulation results we can observe that the malicious nodes are all isolated outside any coalitions eventually.

Third, we study the selfishness issues of wireless networks also using a cooperative game. Previous incentive schemes for enforcing selfish nodes to cooperate in wireless networks can be classified into two categories: One category of solution is using monetary incentives; the other category is employing reputation systems to stimulate the nodes to cooperate. Most of those schemes analyze the incentive effectiveness through a non-cooperative game. In our work, different from others, we propose an incentive routing and forwarding scheme that combines the payment mechanism and the reputation system together, and analyze it with a coalitional game. The reputation system we employ is a heat diffusion model on a weighted reputation graph, which is capable of combining direct reputations and indirect reputations together, and propagating the reputation from locally to globally. We also present a new value function for coalitions taking into account the amount of payment and cost of a node in a coalition. We theoretically prove that this coalitional game has a *core* status, which means nodes in the network can be motivated to forward packets for others, and then form into one grand coalition together. The simulation results show that the cumulative utilities of cooperative nodes are increased steadily and the selfish nodes cannot get more utilities by behaving selfishly than cooperatively.

Bibliography

[1] Alfarez Abdul-Rahman and Stephen Halles. A distributed trust model. In *Proceedings of New Security Paradigms Workshop '97*, pages 48–60, 1997.

[2] Afrand Agah, Sajal K. Das, and Kalyan Basu. A game theory based approach for security in wireless sensor networks. In *Proceedings of IEEE International Conference on Performance, Computing, and Communications (IPCCC)*, pages 259–263, 2004.

[3] Afrand Agah, Sajal K. Das, and Kalyan Basu. A non-cooperative game approach for intrusion detection in sensor networks. In *Proceedings of IEEE 60th Vehicular Technology Conference (VTC-Fall)*, volume 4, pages 2902–2906, September 2004.

[4] Luzi Anderegg and Stephan Eidenbenz. Ad hoc-vcg: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *MobiCom'03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 245–259. ACM Press, 2003.

[5] Bernard Barber. *Logic and limits of Trust*. New Jersey: Rutgers University Press, 1983.

[6] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition edition, 1985.

[7] Thomas Beth, Malte Borcherding, and Birgit Klein. Valuation of trust in open networks. In *Proceedings of the European Symposium on Research in Computer Security*, pages 3–18, Brighton, UK, 1994. Springer-Verlag.

[8] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceeding of the ACM/IEEE Mobile Computing and Networking (MobiCom)*, pages 85–97, 1998. http://citeseer.nj.nec.com/broch98performance.html.

[9] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC'02)*, Lausanne, CH, June 2002. IEEE.

[10] Levente Buttyan and Jean-Pierre Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical Report DSC/2001/001, Department of Communication Systems, Swiss Federal Institute of Technology - Lausanne, 2001.

[11] Srdjan Capkun, Levente Buttyn, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2:52–64, 2002.

[12] T. Chiueh. Optimization of fuzzy logic inference architecture. *IEEE Computer*, pages 67–71, 1992.

[13] Bruce Christianson and William S. Harbison. Why isn't trust transitive? In *Security Protocols International Workshop*, pages 171–176. University of Cambridge, 1996.

[14] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations (rfc2501), January 1999. http://www.ietf.org/rfc/rfc2501.txt.

[15] Partha Dasgupta. Trust as a commodity. In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, volume 4, pages 49–72. Department of Sociology, University of Oxford, electronic edition, 2000.

[16] Morton Deutsch. Cooperation and trust: Some theoretical notes. In M. R. Jones, editor, *Nebraska symposium on motivation*, pages 275–319. University of Nebraska Press, 1962.

[17] Morton Deutsch. *The Resolution of Conflict: Constructive and Destructive Processes*. New Haven, CT: Yale University, 1972.

[18] David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. *SIGCOMM Computer Communication Review*, 26(4):243–254, 1996.

[19] Laurent Eschenauer, Virgil D. Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *Proceedings of the Security Protocols Workshop*, Cambridge, UK, April 2002. Springer-Verlag. http://citeseer.nj.nec.com/eschenauer02trust.html.

[20] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM'01: Proceedings of the 20th IEEE Conference on Computer and Communications*, volume 3, pages 1548–1557, 2001.

[21] Drew Fudenberg and Jean Tirole. *Game Theory*. The MIT Press, 2002.

[22] David Gale and Lloyd Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–14, 1962.

[23] Diego Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Department of Sociology, University of Oxford, electronic edition, 2000.

[24] Diego Gambetta, editor. *Trust: Making and Breaking Cooperative Relations*. Department of Sociology, University of Oxford, electronic edition, 2000.

[25] M. Ginsgerg. Non-monotonic reasoning using dempster's rule. In *Proc. of the AAAI-84*, pages 125–129, 1984.

[26] David Good. Individuals, interpersonal relations, and trust. In DIEGO GAMBETTA, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 3, pages 31–48. Basil Blackwell, 1988.

[27] Tyrone Grandison. Trust specification and analysis for internet applications. Technical report, Imperial College of Science, Thechnology and Medicine, Department of Computing, 2001.

[28] Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys*, 2000. http://citeseer.nj.nec.com/article/grandison00survey.html.

[29] Elizabeth Gray, Jean-Marc Seigneur, Yong Chen, and Christian Jensen. Trust propagation in small worlds. In *Proceedings of the 1st International Conference on Trust Management*, 2002. http://citeseer.nj.nec.com/575876.html.

[30] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Proceedings of 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, pages 3–13, June 2002. http://citeseer.nj.nec.com/hu02sead.html.

[31] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, Atlanta, USA, September 2002. http://citeseer.nj.nec.com/article/hu02ariadne.html.

[32] Jean-Pierre Hubaux, Levente Buttyan, and Srdan Capkun. The quest for security in mobile ad hoc networks. In *MobiHoc'01: Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[33] David B. Johnson and David A. Maltz. Dynamic source routing protocol in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, Boston, USA, 1996.

[34] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). Internet Draft, Apr 2003. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt.

[35] Audun Jøsang. Artificial reasoning with subjective logic. In *the 2nd Australian Workshop on Commonsense Reasoning*, 1997. http://www.idt.ntnu.no/ ajos/papers.html.

[36] Audun Jøsang. Prospectives for modelling trust in information security. In *Proceedings of Australasian Conference on Information Security and Privacy*, pages 2–13, 1997. http://citeseer.nj.nec.com/josang97prospectives.html.

[37] Audun Jøsang. A subjective metric of authentication. In *Proceedings of European Symposium on Research in Computer Security (ESORICS '98)*. LNCS, Springer-Verlag, 1998. http://citeseer.nj.nec.com/josang98subjective.html.

[38] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[39] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molna. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International World Wide Web Conference (WWW '03)*, Budapest, Hungary, 2003.

[40] Kevin Kane and James C. Browne. Using uncertainty in reputation methods to enforce cooperation in ad-hoc networks. In *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, pages 105–113. ACM, 2006.

[41] Etin Kaya Koc. Emerging technologies applications in communications wireless security implementation (etacom'96). 1996. http://citeseer.nj.nec.com/287432.html.

[42] Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *ICNP'01: Proceedings of IEEE International Conference on Network Protocols*, 2001.

[43] Pradip Lamsal. Understanding trust and security, 2001. http://citeseer.nj.nec.com/lamsal01understanding.html.

[44] TCL Programming Language. http://wiki.tcl.tk/.

[45] Xiaoqi Li and Michael R. Lyu. A novel coalitional game model for security issues in wireless networks. In *GLOBECOM'08: Proceedings of IEEE Global Telecommunications Conference*, December 2008.

[46] Xiaoqi Li, Michael R. Lyu, and Jiangchuan Liu. A trust model based routing protocol for secure ad hoc networks. In *Proceedings of IEEE Aerospace Conference*, volume 2, pages 1286–1295, March 2004.

[47] Xiaoqi Li, Wujie Zheng, and Michael R. Lyu. A coalitional game model for heat diffusion based incentive routing and forwarding scheme. In *Proceedings of IFIP Networking 2009*, pages 664–675, May 2009.

[48] Jinshan Liu and Valrie Issarny. An incentive compatible reputation mechanism for ubiquitous computing environments. *IJIS*, 6(5):297–311, September 2007.

[49] Gavin Lowe. An attack on the needham-schroeder public key protocol. *Journal of Information Processing Letters*, 56:131–133, 1995.

[50] Niklas Luhmann. *Trust and Power Chichester*. John Wiley and Sons, 1979.

[51] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, and Lixia Zhang. Self-securing ad hoc wireless networks. In *Proceedings of IEEE ISCC '02*, 2002.

[52] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *CIKM'08: Proceedings of the 16th International Conference on Information and Knowledge Management*, pages 233–242, October 2008.

[53] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Experiences applying game theory to system design. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS'04)*, pages 183–190. ACM Press, 2004.

[54] Daniel W Manchala. Trust metrics, models and protocols for electronic commerce transactions. In *Proceedins of the 18th International Conference on Distributed Computing Systems*, page 312, 1998.

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

[56] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of Mobile Computing and Networking (MobiCom '00))*, pages 255–265, 2000.

[57] Frank L. Mayer. A brief comparison of two different environmental guidelines for determining ̈levels of trust:. In *Sixth Annual Computer Security Applications Conference*, 1990.

[58] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, fifth edition, 1996.

[59] Pietro Michiardi and Refik Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *CMS 2002: Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

[60] Pietro Michiardi and Refik Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks. In *Proceedings of Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt'03)*, pages 3–5, March 2003.

[61] The Network Simulator NS-2. http://www.isi.edu/nsnam/ns/.

[62] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory.* The MIT Press, 1994.

[63] Animesh Patcha and Jung-Min Park. A game theoretic formulation for intrusion detection in mobile ad hoc networks. *International Journal of Network Security*, 2(2):146–152, March 2006.

[64] Charles E. Perkins, editor. *Ad Hoc Networking*. Boston: Addison-Wesley, 2001.

[65] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pages 234–244, London, UK, 1994. ACM Press. http://doi.acm.org/10.1145/190314.190336.

[66] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of MILCOM'97 Panel on Ad Hoc Networks*, November 1997. http://citeseer.nj.nec.com/article/perkins97adhoc.html.

[67] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, 1999. http://citeseer.nj.nec.com/article/perkins97adhoc.html.

[68] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector (aodv) routing. Internet Draft, February 2003. http://www.ietf.org/rfc/rfc3561.txt.

[69] Dean Povey. Trust management for e business. *E Business Security Conference. Brisbane, Australia*, 1999.

[70] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. citeseer.nj.nec.com/551839.html.

[71] G. Shafer. *A Methematical Theory of Evidence*. Princeton University Press, 1976.

[72] Claude E. Shannon. A mathematical theory of communnication. *Bell System Tech*, 27:379–423, 1948.

[73] Claude E. Shannon and W. Weaver. *A Mathematical Theory Of Communnication*. University of Illinois Press, 1963.

[74] Lloyd S. Shapley. *A Value for n-person Games*, volume 28 of *Annals of Mathematical Studies*, chapter Contributions to the Theory of Games, pages 307–317. Princeton University Press, 1953.

[75] Vikram Srinivasan, Pavan Nuggehalli, CarlaFabiana Chiasserini, and Ramesh R. Rao. Cooperation in wireless ad hoc networks. In *INFOCOM'03: Proceedings of IEEE Conference on Computer Communications*, 2003.

[76] Yun Teng, Vir V. Phoha, and Ben Choi. Design of trust metrics based on dempster-shafer theory. http://citeseer.nj.nec.com/461538.html.

[77] George Theodorakopoulos and John S. Baras. Trust evaluation in ad-hoc networks. In *WiSe'04: Proceedings of the 3th ACM Workshop on Wireless Security*, pages 1–10. ACM, 2004.

[78] George Theodorakopoulos and John S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, February 2006.

[79] Game Theory. http://en.wikipedia.org/wiki/game_theory.

[80] Bruce Tuch. Development of wavelan, an ism band wireless lan. Technical Report 4, AT&T Lucent Technical Journal, July/august 1993.

[81] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling cooperation in mobile ad hoc networks: A formal description of selfishness. In *Proceedings of WiOpt Workshop*, 2003.

[82] R. Yahalom, B. Klein, and Thomas Beth. Trust relationships in secure systems - a distributed authentication perspective. In *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy (RSP '93)*, pages 150–164, 1993.

[83] R. Yahalom, B. Klein, and Thomas Beth. Trust-based navigation in distributed systems. *Special issue of the jounal Computing Systems: Security and Integrity of Open Systems*, 1994.

[84] Haixuan Yang, Irwin King, and Michael R. Lyu. DiffusionRank: a possible penicillin for web spamming. In *SIGIR'07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 431–438, New York, NY, USA, 2007. ACM.

[85] Hao Yang, Xiaoqiao Meng, and Songwu Lu. Self-organized network-layer security in mobile ad hoc networks. In *Proceedings of ACM Workshop on Wireless Security (WiSe'02)*, Atlanta, USA, September 2002.

[86] LA Zadeh78. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[87] Manel Guerrero Zapata and N. Asokan. Securing ad hoc routing protocols. In *Proceedings of ACM Workshop on Wireless Security (WiSe '02)*, pages 1–10, Atlanta, USA, September 2002. ACM Press. http://doi.acm.org/10.1145/570681.570682.

[88] Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom'00: Proceedings of the 6th ACM International*

Conference on Mobile Computing and Networking, pages 275–283, Boston, Massachusetts, USA, 2000. http://doi.acm.org/10.1145/345910.345958.

[89] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *INFOCOM'03: Proceedings of the 22nd IEEE Conference on Computer Communications*, volume 3, pages 1987–1997, 2003.

[90] Sheng Zhong, Li (Erran) Li, Yanbin Grace Liu, and Yang (Richard) Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks - an integrated approach using game theoretical and cryptographic techniques. In *MobiCom'05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 117–131, New York, NY, USA, 2005. ACM Press.

[91] Sheng Zhong and Fan Wu. On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks. In *MobiCom'07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 278–289. ACM, 2007.

[92] Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *Journal of IEEE Networks*, 13(6):24–30, 1999.