

## ABSTRACT

Title of dissertation:       DISTRIBUTED TRUST MANAGEMENT  
                                  IN AUTONOMIC NETWORKS

Tao Jiang, Doctor of Philosophy, 2007

Dissertation directed by:   Professor John S. Baras  
                                  Department of Electrical and Computer Engineering

The management of autonomic networks has gained more and more attentions because of their wide applications and control difficulties. Autonomic networks are decentralized and self-organized. Without global knowledge on the states of autonomic networks, it is difficult to predict behaviors of such networks and thus to conduct proper network management and control. This dissertation is the starting point of my effort to theoretically understand the complex characteristics of autonomic networks. In particular, I focus on a specific application: distributed trust management.

We view trust among users as a set of relations established on the basis of trust credentials and required by specified policies. Two important components of a distributed trust management system are studied in this work: trust credential distribution and trust evaluation. In autonomic networks, trust credentials are distributed throughout the network. Given the mobility and dynamics of the networks, it is important to properly distribute trust credentials such that users are able to efficiently obtain required credentials and update existing credentials. I present a

trust credential distribution scheme based on network coding. After obtaining credentials in need, policies are required for users to evaluate trustworthiness of targets in a distributed way. In this dissertation, I model distributed trust evaluation as an estimation problem and trust evaluation policies based on local interactions are studied. I investigate the convergence of both deterministic and stochastic voting rules and prove their effectiveness with the present of misbehaving users.

Autonomic networks rely on collaboration among users. The conflict between the benefit from collaboration and the required cost for collaboration naturally leads to game-theoretic studies. I study collaboration based on cooperative games with communication constraints and give the conditions under which users are willing to collaborate. The results in this dissertation show that a well-designed trust management system is helpful to enforce collaboration. Besides collaboration, I show that trust can be used to the utility optimization problems as well. The effect of trust values is that in the routing and scheduling problems the trustworthiness of the node will be automatically considered and used. For example, packets will not be routed frequently to suspicious nodes.

DISTRIBUTED TRUST MANAGEMENT IN AUTONOMIC  
NETWORKS

by

Tao Jiang

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2007

Advisory Committee:

Professor John S. Baras, Chair/Advisor

Professor Armand M. Makowski

Professor Virgil D. Gligor

Professor Gang Qu

Professor Nick Roussopoulos

© Copyright by  
Tao Jiang  
2007

# DEDICATION

To my parents Xiancheng and Qingfang,

to my sister Rong,

and to Peng

for their invaluable love and support

## ACKNOWLEDGMENTS

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever. First and foremost I would like to thank my advisor, Dr. John S. Baras for giving me an opportunity to work on challenging and extremely interesting problems. His deep insight on amazingly broad areas of research, his endless enthusiasm on challenging problems and his timely encouragement have been the most important support for my work. I would also like to thank Dr. Armand M. Makowski, Dr. Virgil D. Gligor, Dr. Gang Qu and Dr. Nick Roussopoulos for agreeing serve on my committee and for providing feedback.

Sincerely thanks to my colleagues in the HyNet center and SEIL lab. Special thanks to Georgios Theodorakopoulos, Alvaro Cardenas, Huigang Chen, Maben Rabi, Svetlana Radosavac, Pedram Hovareshti and Punyaslok Purkayastha, with whom I always had inspiring and fruitful discussions. Also I would also like to thank Kim Edwards for her great administrative support.

This dissertation is prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. It is also supported by the U.S. Army Research Office under grant No DAAD19-01-1-0494.

# TABLE OF CONTENTS

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Autonomic Networks	1
1.2 Notions of Trust	3
1.3 Context of Trust	6
1.4 Global Trust or Local Trust	7
1.5 Trust Metrics and Uncertainty	7
1.6 Examples of Trust and Reputation Systems	8
1.6.1 Reputation management in eBay	9
1.6.2 Product review sites	10
1.6.3 Expert sites	12
1.6.4 Google's PageRank	13
1.6.5 PGP web of trust	14
1.6.6 Trust and reputation systems in P2P networks	14
1.7 Trust Management in Autonomic Networks	16
2 Trust Credential Distribution	19
2.1 Overview	19
2.2 Types of Trust Credentials	21
2.3 Network Coding	24
2.4 Network Coding Based Scheme	25
2.5 Comparison of Distribution Schemes	29
3 Trust Evaluation	32
3.1 Overview	32
3.2 An Example on Distributed Trust Evaluation	33
3.3 Global Trust Evaluation	34
3.3.1 Network Model	35
3.3.2 Direct Trust	37
3.3.3 Voting Rule	40
3.4 Deterministic Voting Rule	42
3.4.1 Convergence	43
3.4.1.1 Voting in a virtuous network	45
3.4.1.2 Voting with leaders	48
3.4.2 Network Topology	50
3.5 Stochastic Voting Rule	53
3.5.1 Stochastic Model	53
3.5.2 Estimation	54
3.5.3 Bayesian Network	55
3.5.4 Markov Random Field	61

3.5.5	Stochastic Voting . . . . .	63
3.5.5.1	Update Sequence . . . . .	63
3.5.5.2	Markov Chain Interpretation . . . . .	64
3.5.5.3	Convergence . . . . .	65
3.5.6	Binary Example . . . . .	69
3.5.6.1	Ising Model and Spin Glasses . . . . .	71
3.5.6.2	Virtuous Network . . . . .	72
3.5.6.3	Adversary Model . . . . .	77
3.5.6.4	Network Topology . . . . .	81
4	Applications of Trust . . . . .	84
4.1	Trust and Cooperation: the Game Theoretic View . . . . .	85
4.1.1	Problem Formulation . . . . .	87
4.1.2	Coalitional Games . . . . .	88
4.1.3	Cooperation in games . . . . .	90
4.1.3.1	Cooperative games with negotiation . . . . .	90
4.1.3.2	Trust mechanism . . . . .	94
4.1.4	Dynamics of Cooperation . . . . .	96
4.1.4.1	System model . . . . .	96
4.1.4.2	Game evolution . . . . .	98
4.2	Trust Aware Cross-Layer Optimization . . . . .	102
4.2.1	System Model . . . . .	105
4.2.1.1	Trust . . . . .	105
4.2.1.2	Data flow and utility function . . . . .	106
4.2.1.3	Interference model and stability . . . . .	108
4.2.2	Utility Optimization and Dual Decomposition . . . . .	109
4.2.3	Distributed Algorithm . . . . .	112
	Bibliography . . . . .	114



LIST OF TABLES

3.1 The Conditional Probability  $\Pr(d_{ij}|s_i, s_j)$  . . . . . 59

## LIST OF FIGURES

1.1	Trust management system in autonomic networks . . . . .	18
2.1	Alice and Cathy want to obtain credentials on Bob . . . . .	20
2.2	A Simple Network Coding Example . . . . .	24
2.3	Operations of our network coding based scheme . . . . .	27
2.4	Number of nodes finishing over time . . . . .	30
2.5	Network load over time . . . . .	31
3.1	Procedure of trust evaluation . . . . .	34
3.2	A trust graph . . . . .	36
3.3	The second largest eigenvalue $\lambda_2$ . . . . .	52
3.4	Convergence time . . . . .	53
3.5	Bipartite Bayesian Network for the Trust Graph 3.2 . . . . .	55
3.6	General Factor Graph for Trust Graphs . . . . .	56
3.7	Factor Graph for the Trust Graph 3.2 . . . . .	56
3.8	A Tree Trust Graph . . . . .	58
3.9	The Tree Factor Graph of Figure 3.8 . . . . .	58
3.10	Message Passing for the Tree Factor Graph . . . . .	60
3.11	Part of the Markov Chain. Suppose $S(k) = [1, -1, 1, 1]$ , then $S(k+1)$ either flips one of the element in $S(k)$ or stays at the state of $S(k)$ . . . . .	65
3.12	Transitions between $S$ and $R$ . $p_{S,R} = q_i \Pr[\bar{s}_i S]$ and $p_{R,S} = q_i \Pr[s_i R]$ . . . . .	68
3.13	$P_{correct}$ vs. $b$ with $\eta < 0$ and $\eta > 0$ . . . . .	74
3.14	$P_{correct}$ vs. $b$ with $\eta = 0$ . . . . .	74
3.15	$P_c$ vs. $b$ with link errors $p_e$ . $\eta = 0$ . . . . .	76

3.16	$P_c$ vs. the percentage of adversaries $P_m$ . The link error $p_e = 0.05$ and the degree of certainty $b = 1$ . . . . .	80
3.17	The effect of network topology. $P_{rw}$ is the percentage of shortcuts in WS small world model. The network with $P_{rw} \in [0.01, 0.1]$ has the small world property. $p_e = 0.1$ . . . . .	81
3.18	The effect of average degree with $p_e = 0.1$ . . . . .	82
4.1	System operation block-graph for a typical node . . . . .	97
4.2	Algorithm for game evolution modeling trust revocation . . . . .	100
4.3	Percentage of cooperating pairs vs. negative links . . . . .	103
4.4	Average payoffs vs. negative links . . . . .	103
4.5	An example network with 2 routes for flow from $s$ to $d$ . . . . .	107

# Chapter 1

## Introduction

### 1.1 Autonomic Networks

With the explosive growth of network techniques in the last decade, connecting to the world from any place, at any time and for any body is no longer just a dream. In the meanwhile, the fast proliferation of networked devices and applications, such as sensor networks and pervasive computing, integrates information technology into our environments. These dramatic changes create unique challenges for network management and control. Innovative solutions are required for managing network mobility and dynamics, astronomical number of data and enormous information exchanges.

The traditional centralized server-based management can no longer satisfy the requirements of next generation networks, so people started to propose new concepts of network infrastructure and management. For instance, mobile ad hoc networks (MANETs) [47] aim to provide wireless network services without relying on any infrastructure. The wireless mesh network, which has been implemented by a number of wireless network groups [2, 19], is essentially a MANET over a 802.11 wireless LAN, which can be set up with almost “zero-cost” in highly mobile environments. Another example is peer-to-peer (P2P) networks [28, 67], where a large number of data are shared among millions of network users. All the aforementioned new types

of networks share a common characteristic: they are distributed and self-organized, thus they are sometimes called *autonomic networks* [5] in the literature. In this work, our focus is on the fundamental principles and properties of these networks, rather than narrowing on a particular network prototype.

An autonomic network is one that is self-configuring, self-optimizing, self-healing and self-protecting. Such a network requires minimal administration, mostly involving policy level management. Entities<sup>1</sup> in autonomic networks all participate in network control through individual interactions. To achieve desired network management goals under such “anarchy” is not an easy job. A small misbehavior by an individual might lead to a network-wise “avalanche”. The goal of this dissertation is to understand and analyze the behavior and properties of these “anarchical” autonomic networks.

Autonomic systems have been studied in various scientific fields: in biological systems, swarms of bacteria, insects and animals yield sophisticated collective behaviors based on simple individual interactions; in physics, a group of particles interacting with their neighbors to form a magnet; even in human society, economists have modeled human individual interactions using iterated games, etc. In this dissertation, some results are inspired from those studies, such as the Ising model and spin glasses model in statistical physics, which will be investigated in details in Chapter. 3.

As a case study, we employ a specific application in autonomic networks – distributed trust management – for our study. Trust is important and critical for

---

<sup>1</sup>In this dissertation, terms node, user and entity are interchangeable.

network security. It integrates with several components of network management, such as risk management, access control and authentication. Trust management is to collect, analyze and present trust-related evidence and to make assessments and decisions regarding trust relationships between entities in a network [10].

## 1.2 Notions of Trust

In order to understand the phenomenon of trust relationships, we first need to understand the meaning of trust. Various different definitions of the term trust have been proposed.

In an informal way, trust in an entity can be explained as the belief that under certain circumstances this entity will perform in a certain way. More formally, Merriam-Websters Collegiate Dictionary states, that trust is the

“assured reliance on the character, ability, strength, or truth of someone or something.”

One definition popular in the network field is from Diego Gambetta [24] “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...”. We identify two relevant points in this definition: firstly, trust is used in order to predict an entity’s future behavior, secondly, trust is subjective. A trust relation is usually specified between two parties: a trustor, the subject that trusts a target entity, known as the trustee, i.e. the entity that is trusted. The trustor is an entity capable of thinking and making decision, whereas the trustee can be

anything from a person or physical entity, to abstract notions such as software or a cryptographic key between trustor, the entity who is trusting, and trustee, the entity who is trusted[39]. In the following, we identify various dimensions of trust relationships in addition to trustor and trustee:

**Context** Trust has very broad meaning, and there is no need to trust an entity completely if one expects it only to perform a limited task. Different forms of trust have been identified in the literature. Besides those related to security, such as access control, authentication and integrity, trust can be related to providing service, delegation, correctness and so on. Here are some examples:

- I trust my doctor to competently perform an operation.
- I trust my database manager to decide who has access to my database.
- A Trust Computing Base (TCB) is trusted to execute on a machine to support the required security policy.

**Metric** A trust metric refers to the quality of the trust relationship, which ranges from complete distrust over a neutral trust measure to full trust. The more a trustee is trusted, the higher the trust metric is supposed to be. Trust metrics may take uncertainty into consideration. Certainty of a trust metric specifies the confidence of the trustor in his or her estimation of the trustee. If this estimation is gained via only few personal experiences, the certainty is supposed to be low. While, if the estimation is gained based on certificates signed by trusted third parties, the certainty is usually high.

**Transitivity** Transitivity of trust is not always true. However most of cases, we accept transitivity in a restricted manner. Alice wants to evaluate entity X, but she can only get Bobs evaluation of X. Alice accepts this evaluation only if she believes the policy Bob used is a subset of her policy. The similar discussion can be found in [9, 21]. For instance, Alice only trust board certified doctors as her Primary Care Physicians (PCPs), and her health insurance company recommends her a couple of doctors as candidates. Alice believes the company only recommends doctors who are providers of the insurance plan and board certified doctors. So she trusts those recommended doctors, because the policy of the company is a subset of (stronger than) her policy.

**Dynamics** A trust relationship is not static, but changes dynamically on various different incidents. If for instance a patient has good experience with his doctor, the trust of the patient in his doctor will increase. In addition to own experiences, trust estimations received from others influence the own trust assessment as well. Lastly, trust relationships may also change over time when no experiences have been made. For instance, the trustworthiness of a password decreases dramatically when the validation date has passed.

**Trust, risk and reputation** Risk and reputation are two factors that contribute significantly to the trust decision. Risk refers to the probability of failure of a transaction with respect to a specific context. Reputation refers to the cumulative view of the interactions with respect to a party within a context. Both concepts are subjective and can be used in the determination of a trust decision. A very risky venture has a higher chance of being designated a decision not to trust, while a not



so risky transaction will normally lead to a positive trust decision. Entities with a positive reputation stand a higher chance of being trusted for future interactions, while entities with a negative reputation will normally lead to a distrust decision. Generally, reputation only partly affects trust. However, in this dissertation, we do not explicitly distinguish trust and reputation.

In the following three sections, we clarify some of the above dimensions in details.

### 1.3 Context of Trust

As pointed out in the previous section, applications can define various context areas in which entities can be trusted. It is important to note, that these areas are not necessarily independent from each other. Different kinds of dependencies can exist among the context areas: instance of relationships are one-level relationships for classification, is a relationships provide generalization, part of relationships enable aggregation and surely many other - potentially application specific - forms of dependencies between context areas can be imagined. For the sake of the trust model, however, we do not need to model all these relationships in detail. Instead, we model the asymmetric semantic distance between the context areas and therefore abstract from the kind of dependency. The metrics chosen for the semantic distance is a real number in the interval of  $[0, 1]$ . A distance close to 1 represents a high dependency; a distance of 0 refers to no dependency. Therefore, we organize the trust context areas as a weighted directed graph as can be seen in 1. This allows

us to spread the impact of a trust update in one area to related areas. The context areas and the semantic distances between the areas are specified by the applications to be supported with trust management. However, due to the subjectivity of trust, each user is enabled to locally modify the distances to suit his or her personal views.

## 1.4 Global Trust or Local Trust

Global trust metrics assign to a given user a unique trust score, independently of the user that is evaluating the other user's trustworthiness. On the other hand, a local trust metric (personalized trust metric) provides a personalized trust score that depends on the point of view of the evaluating user [49].

Global trust metrics are good when users have the same criterion on trustworthiness. For instance, in the context of routing, a node is trusted if it forwards others' packets, and distrusted if it drops packets. While the local trust depends on individual's preference.

## 1.5 Trust Metrics and Uncertainty

Various different representations of trust values exist in the related work. Trust values can be depicted as real numbers in certain intervals like for instance  $[-1, +1]$ , as done by Jonker and Treur [36] and Sabater [60] or probabilities in  $[0, 1]$ , as proposed among others by Jøsang and Ismail [37], Yu and Singh [71], and Kinateder and Rothermel [40]. Others propose discrete values, like the binary representation by Blaze and Feigenbaum [11] or four discrete values in PGP [73].

Not all aforementioned algorithms support the computation of an uncertainty value, which states the quality of the trust assessment represented in the trust metric. If uncertainty is mentioned [71, 37, 60], it is specified in the interval  $[0, 1]$ , whereas 0 describes complete uncertainty and 1 the total certainty.

The metric used in this dissertation is a real number in the interval of  $[-1, 1]$ . Complete distrust is represented by  $-1$  whereas 1 corresponds to full trust.

## 1.6 Examples of Trust and Reputation Systems

Trust and reputation systems are widely used to provide a basis for the choice of transaction items and partners in online service provision. The basic idea is to let participating nodes rate each other, for example after the completion of a transaction, and use the aggregated ratings about a given node to derive a trust or reputation score, which can assist other nodes in deciding whether or not to transact with that node in the future. Trust and reputation systems have proven useful in online auctioning systems such as eBay or online book stores such as Amazon, and can be viewed as a substitute for the ‘word-of-mouth’ mechanisms observed in offline personal encounters. As a result, it has recently been proposed that trust management systems be extended to autonomic systems such as Internet-based peer-to-peer systems and mobile ad-hoc networks. The need here is predominantly to provide incentives for cooperation and to protect the network from node misbehavior.

In this section, we describe the most well known applications of online trust and reputation systems<sup>2</sup>. We start with systems having centralized network archi-

---

<sup>2</sup>Some of the examples are from the work by Jøsang et. al.[38].

tructures, and then existing distributed systems are described.

### 1.6.1 Reputation management in eBay

eBay<sup>3</sup> uses the feedback profile for rating their members and establishing the members' reputation. Members rate their trading partners with a Positive, Negative or Neutral feedback and explain briefly why after completion of a transaction. The reputation is calculated by assigning 1 point for each positive comment, 0 points for each neutral comment and -1 point for each negative comment. This feedback mechanism is a centralized reputation system, where eBay collects all the ratings and computes the scores. The reputation score of each participant is the sum of all feedback points (from unique users).

The eBay reputation system is very primitive and can be quite misleading. We list the following problems for the system:

- It suffers from single point of failure because it is centralized.
- A participant with 100 positive and 10 negative ratings should intuitively appear much less reputable than a participant with 90 positive and no negatives, but on eBay both would have the same total reputation score.
- A seller may participate in many small transactions in order to build up a high positive reputation, and then defraud one or more buyers on a high-priced item.
- No special mechanism is provided to detect members that lie in their feedback.

---

<sup>3</sup><http://www.ebay.com>

Despite its drawbacks and primitive nature, the eBay reputation system seems to have a strong positive impact on eBay as a marketplace.

### 1.6.2 Product review sites

Individual reviewers of Product review sites provide information for consumers for the purpose of making better purchase decisions. The trust systems on those sites apply to products as well as to the reviewers themselves.

Epinions<sup>4</sup> is a product and shop review site. The reviews written by members consist of text and quantitative ratings from 1 to 5 stars for a set of aspects such as *Ease of Use* for products and *Customer Service* for shops. Other members can rate reviews as *Not Helpful*, *Somewhat Helpful*, *Helpful*, and *Very Helpful*, and thereby contribute to determining the rank of the review, as well as to giving the reviewer a higher status. A member can obtain the status *Advisor*, *Top Reviewer* or *Category Lead* as a function of the accumulated ratings on all his or her reviews over a period. Epinions also make use of the “Web of Trust”: members can decide to either *trust* or *block* another member. A member’s list of trusted members represents that member’s personal Web of Trust. The Web of Trust influences the automated selection of Top Reviewers and Advisors. Epinions has an incentive systems for reviewers called the *Income Share Program*, where members can earn money based on the utility of their reviews. Reviewers can potentially earn as much for helping someone make a buying decision with a positive review, as for helping someone avoid a purchase with a negative review. This is important in order not to give

---

<sup>4</sup><http://www.epinions.com>

an incentive to write biased reviews just for profit. The trust system of Epinions is highly sophisticated compared to other trust and reputation systems on the Internet. It has been shown to provide high quality reviews. Notice that the trust system of Epinions is still centralized.

Amazon<sup>5</sup> is another example of product review sites. Reviews consist of text and a rating in the range 1 to 5 stars. The average of all ratings gives a book its average rating. Users can vote on reviews as being helpful or not helpful. The numbers of helpful as well as the total number of votes are displayed with each review. As a function of the number of helpful votes each reviewer has received, as well as other parameters not publicly revealed, Amazon determines each reviewer's rank, and those reviewers who are among the 1000 highest get assigned the status of Top 1000, Top 500, Top 100, Top 50, Top 10 or #1 Reviewer. Amazon has a system of Favorite People, where each member can choose other members as favorite reviewers, and the number of other members who has a specific reviewer listed as favorite person also influences that reviewer's rank. There are many reports of attacks on the Amazon review scheme where various types of ballot stuffing has artificially elevated reviewers to top reviewer, or various types of "bad mouthing" has dethroned top reviewers. For example the Amazon #1 Reviewer usually is somebody who posts more reviews than any living person could possibly do if it would require that person to read each book, thus indicating that the combined effort of a group of people, presented as a single person's work, is needed to get to the top. Also, reviewers who have reached the Top 100 rank have reported a sudden

---

<sup>5</sup><http://www.amazon.com>

increase in negative votes which reflects that there is a ‘catfight’ taking place in order to get into the ranks of top reviewers.

### 1.6.3 Expert sites

Individuals in expert sites are willing to answer questions in their areas of expertise, and the reputation systems on those sites are there to rate the experts.

AllExperts<sup>6</sup> provides a free expert service for the public on the Internet. The reputation system on AllExperts uses the aspects: Knowledgeable, Clarity of Response, Timeliness and Politeness where ratings can be given in the interval [1; 10]. The score in each aspect is simply the numerical average of ratings received. The number of questions an expert has received is also displayed in addition to a General Prestige score that is simply the sum of all average ratings an expert has received.

Advogato<sup>7</sup> is a community of open-source programmers. Members rank each other according to how skilled they perceive each other to be, using Advogato’s trust scheme, which in essence is a centralized reputation system based on a flow model. The trust engine of Advogato computes the trust flow through a network where members constitute the nodes and the edges constitute referrals between nodes. Each member node is assigned a capacity between 800 and 1 depending on the distance from the source node that is owned by Raph Levien who is the creator of Advogato. The source node has a capacity of 800 and the further away from the source node, the smaller the capacity. Members can refer each other with the status

---

<sup>6</sup><http://www.allexperts.com>

<sup>7</sup><http://www.advogato.org>

of Apprentice (lowest), Journeyer (medium) or Master (highest). A separate flow graph is computed for each type of referral. A member will get the highest status for which there is a positive flow to his or her node.

#### 1.6.4 Google's PageRank

PageRank proposed by Page et al. [14] represents a way of ranking the best search results based on a page's reputation. Roughly speaking, PageRank ranks a page according to how many other pages are pointing at it. This can be described as a reputation system, because the collection of hyperlinks to a given page can be seen as public information that can be combined to derive a reputation score. A single hyperlink to a given web page can be seen as a positive rating of that web page. Google's search engine is based on the PageRank algorithm and the rapidly rising popularity of Google was obviously caused by the search results that the PageRank algorithm delivered.

The trust and reputation systems mentioned above are all centralized. All data for these systems are stored at a trusted centralized database, which is not ideal for autonomic networks, and lead to the usual issues of scalability and single point of failure. We present distributed systems in the rest of this section, which are the main interest of this thesis.



### 1.6.5 PGP web of trust

PGP web of trust is probably the most widely used decentralized trust system. PGP focuses on proving the identity of key holders. PGP uses user-defined thresholds to decide whether a given key is trusted or not, and different introducers can be trusted at finite set of different trust levels (completely trusted, marginally trust, marginally distrust and completely distrust). Trust in PGP is only followed through one level of indirection; i.e., if A is trying to decide the trust of B, there can be at most one person, C, in the trust path between A and B.

### 1.6.6 Trust and reputation systems in P2P networks

The peer-to-peer (P2P) paradigm allows participants to exchange resources like content, processing power, and storage capacity without the need for a centralized authority. The P2P networks that have been widely deployed to date can be observed to have several unique features. They are essentially goodwill networks of mostly unknown peers and the peer population is highly transient. The result is that these networks are open to abuse and misuse by their peers, and so the issues of trust and reputation are becoming critical to their continued success.

KaZaA<sup>8</sup> uses Integrity Rating Files for rating files and Participation Level for rating peers. Users can rate the files they share based on technical quality and accuracy of their descriptive file data. A file can be given the following ratings: excellent, average, poor, or delete file. In KaZaA a Participation Level is assigned

---

<sup>8</sup><http://www.kazaa.com>

to each user in the network based on the files that are uploaded from the user and the files the user downloads from other users. The Participation Level is defined as  $(\text{Uploads in MB}/\text{Downloads in MB}) \times 100$ .

Reputation systems can be used to motivate peers to positively contribute to the network. For instance, agents that contribute resources to the network may be rewarded with faster download speeds or reduced response latency for their requests and queries. An example of this incentive can be seen in Bittorrent<sup>9</sup>. Applying the principle of “tit-for-tat”, the client application throttles upload speeds to a peer based on the download speed it is receiving from that peer. Therefore, peers that are willing to devote more upload bandwidth are rewarded with a higher download speed. In both KaZaA and Bittorrent, malicious peers can still upload inauthentic files which will reduce the peers’ satisfaction and waste network resources.

As we have seen, the existing distributed trust and reputation systems are very primitive. They suffer from many attacks and their potentially complex behavior is not yet understood. This understanding is necessary to investigate whether distributed trust management systems might prove as useful as their counterparts for centralized systems. The focus of this dissertation is exactly on the theoretical analysis of distributed trust management.

All the above examples assume that the trust credentials are collected by a centralized server or are ready to use when needed. This is usually not the case in autonomic networks where trust credentials are produced by individual principals,

---

<sup>9</sup><http://www.bittorrent.com>

thus credential distribution is needed for users to obtain credentials they need. In Chapter 2, we study distribution of trust credentials and propose a distribution scheme which is inspired from P2P networks. Most of trust or reputation policies used in the above examples are designed in an ad hoc way. Therefore, various vulnerabilities are discovered after these policies have been used in the real life. In Chapter 3, we model distributed trust evaluation in a more general form and theoretically analyze performance of trust evaluation policies. By analyzing certain evaluation policies, we are able to derive the proper region for network design, such as requirements on the network hierarchy and parameter settings. Some of our theoretic results are even surprising, such as phase transition phenomena. Those results provide a way to properly design feasible evaluation policies and proper network structure. As we mentioned, one predominant need for trust and reputation systems in autonomic networks is to provide incentives for cooperation. For instance, Bittorrent uses reputation to encourage cooperation. In Chapter 4, we formally model the cooperation based on game theory and provide necessary conditions that can encourage cooperation among all nodes in the network.

## 1.7 Trust Management in Autonomic Networks

Before getting into the details of our solutions, we first clarify the unique properties of trust management in autonomic networks. On Jøsang et. al.'s definition ([39]), trust management is: “[t]he activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability

of potential transactions involving risk, and that also allow players and system owners to increase and correctly represent the reliability of themselves and their systems.”

The essence of trust management, on this definition, is to provide a better informational basis for analyzing the risks and benefits of the transaction. Typical examples would be the collection of relevant information about the reputation of the other party or the systematic analysis of my own experiences from previous encounters with her, and the communication of my own policies in situations of this kind. The same understanding of the point of trust management emerges from the survey of trust management provided by Ruohomaa et al. [59]. Our focus for distributed trust management is on its theoretical analysis rather than providing a strict definition, or a system model for trust management.

In traditional networks, such as Internet, sources of trust evidence are centralized control servers, such as trusted third parties (TTPs) and authentication servers (ASs). Those servers are trusted and available all the time. Most prior research efforts in this framework [66, 43, 26] assume an underlying hierarchical structure within which trust relationships between ASs are constrained.

In contrast, autonomic networks have neither fixed infrastructures, nor centralized control servers. In these networks, the sources of trust evidence are peers, i.e. the entities that form the network. We summarize the essential and unique properties of distributed trust management in autonomic networks as the followings:

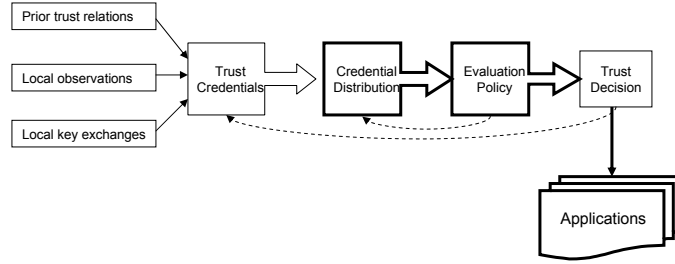


Figure 1.1: Trust management system in autonomic networks

- **uncertainty and incompleteness:** Trust evidence is provided by peers, which can be incomplete and even incorrect.
- **locality:** Trust information is exchanged locally through individual interactions.
- **distributed computation:** Trust evaluation is performed in a distributed manner.

The objective of this dissertation is to use mathematical analysis that helps us to understand and predict the emergent behaviors [27] of trust management systems in autonomic networks. Figure 1.1 describes components of a trust management system in autonomic networks and their interactions. Chapter 2 focuses on credential distribution. The evaluation policy is studied in Chapter 3. In Chapter 4, we present two applications where trust decisions that the trust management system provides can be used.

## Chapter 2

### Trust Credential Distribution

#### 2.1 Overview

As we have discussed, in autonomic networks, trust credentials are typically issued and often stored in a distributed manner. Most of existing work ([10, 9, 58, 50]) assumes that one has already gathered all the potentially relevant credentials in one place and does not consider how to gather these credentials. The assumption that all credentials are stored in one place is inconsistent with the idea of decentralized trust management. As shown in Figure 2.1, credentials on Bob are scattered in the network. Both Alice and Cathy want to evaluate the trustworthiness of Bob.

Credential distribution raises many interesting questions. Where should Alice and Cathy look for credentials? Often, they cannot look everywhere, how can they efficiently obtain requested credentials? In addition, what are the best places to store credentials, such that they can be easily located, well protected and timely updated? Trust credential distribution is the foundation of trust evaluation. It provides the input for the evaluation model. Till now, the credential management and retrieval problem that exists in distributed environments have not been well addressed. The problem of trust credential distribution shares many characteristics of distributed peer-to-peer (P2P) file sharing systems ([67, 18]). Trust credentials are files to be shared for trust management. Thus many distributed trust management systems

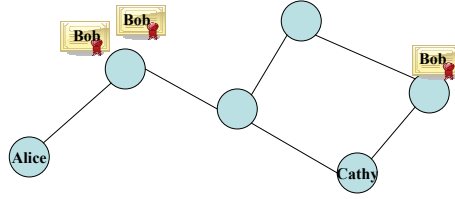


Figure 2.1: Alice and Cathy want to obtain credentials on Bob

consist of components that are inspired from P2P systems. For instance, the trust management scheme P-Grid ([1]) is based on scalable replication of tree structures, which is derived from the P2P systems. As one of the first work in the literature, Eschenauer [21] proposed a trust credential distribution scheme based on Freenet [18], where trust credentials are routed and searched using distributed hash table (DHT). More specifically, for a particular trust credential, its ID is mapped into a value using a universal defined hash function. This hash value also corresponds to a unique node ID in the network. Then the credential is routed to and stored in this node. When searching for this particular credential, the requester gets the hash value using the same hash function and set its request destination as the corresponding node. In this way, trust credentials are routed by hash-based routing instead of flooding. Request routing in Freenet avoids flooding and improves with time. Files, or trust credentials in this context, are replicated by caching at every node, which causes information to converge to where it is most needed.

Despite the similarities between trust credential distribution and P2P file sharing, there exist several unique properties of trust credential distribution. The size of trust credentials in digital documents are a lot smaller compared to the hundreds of million or billion bit files in P2P sharing. The requests in trust credential distri-

bution usually aim at multiple pieces, for instance, a request may ask for all trust credentials about the trustworthiness of node  $A$ . While in P2P one request explicitly points to one single file. Furthermore, the trust credentials update much more frequently than files in P2P networks. We propose a new trust credential distribution scheme that uses *network coding*, and we show that the new scheme handles the above unique properties and performs well in terms of efficiency and security[34].

Network coding has been used in P2P file sharing as well. Gkantsidis and Rodriguez designed a P2P file sharing system named Avalanche, in which intermediate peers produce linear combinations of file blocks as in network coding ([25]). However, again, they are focused on downloading large files from file servers, and our interest is to distribute scattered trust credentials.

In this chapter, we propose our trust credential distribution scheme that uses linear network coding to combine multiple credentials during transmissions. Our approach is inherently different with the commonly used request-response approach. We show that our scheme is extremely easy to implement and absolutely distributed, which nicely handles the dynamic nature of the problem.

## 2.2 Types of Trust Credentials

Different trust contexts require different types of credentials. Examples might be your driver's license, your social security card, or your birth certificate. Each of these has some information identifying you and some authorization stating that someone else has confirmed your identity. Some credentials, such as your passport,



are important enough confirmation of your identity that you would not want to lose them, lest someone uses them to impersonate you.

In cyberspace, users rely on digital trust documents. A digital trust document is data that functions much like a physical credential. We are focused on digital trust documents, which are called *trust documents* or documents in short. Here are some examples of digital trust documents:

- *Digital certificate*: which is issued by a certificate authority or entity and verifies that a public key is owned by a particular entity. Digital certificates are used to thwart attempts to substitute one person's key for another. A digital certificate consists of three things: a public key, certificate information ("identity" information about the user, such as name, user ID, and so on) and one or more digital signatures. Digital certificates are widely used in PGP and X.509.
- *IAFIS* (Integrated Automated Fingerprint Identification System): which uses human fingerprints as identities. It has made law enforcement officials capable to easily share information about criminals and quickly compare a suspect's fingerprint image with millions of similar imprints.

The trust documents can be generated by some trusted party as in centralized networks (commanders of the army), by friends who know each other (soldiers in the same group), or by certain monitoring schemes ([48, 72]). Creation of trust documents is not in the scope of this work.

In autonomic networks, information is usually partial and incomplete. Uncer-

tainty of trust must be introduced in the trust documents, which is usually represented by a value. This value denotes the degree of trust the issuer of the documents has on the target user. The value can be binary-valued, either trust or distrust, or multiple-valued, such as four levels of trust in PGP [73], or even continuous in an interval, say  $[-1, 1]$ .

In dynamic environments, the validity and value of trust documents change over time and space. Every user has confidence values on the documents he stores. The confidence value depends on several factors, for instance, time elapsed since the document is issued, and the communication distance taken by the document to reach the user. Normally, the confidence value is a monotonically non-increasing function of the elapsed time and communication distance. When the confidence value is below certain threshold, the corresponding document is considered to be invalid.

As an important security concept, we need consider the integrity and authenticity of trust documents as well. Digital signatures based on public-key cryptography have been studied by Maurer ([50]) and employed in PGP ([73]). However there are several difficulties with this approach: first, we have to assume that there is an immense public key infrastructure in place, which is usually not available in a self-organized network; second, there has to be a reliable way to find the identities of entities and make sure the identities are not tampered. Those problems are especially important in networks with no fixed infrastructure. One solution is to make use of side channels. For instance, in [6] and [65], users utilize a location-limited side channel, such as physical contact, to first exchange a small amount of cryptographic

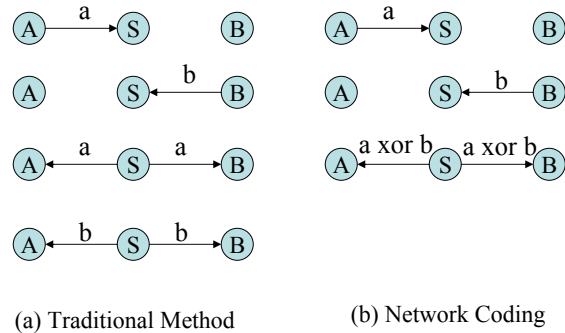


Figure 2.2: A Simple Network Coding Example

information. The information can be used to authenticate standard key exchange protocols performed over the wireless link. In the rest of the chapter, we assume authenticity and confidentiality of documents have been achieved by applying certain cryptographic primitives.

### 2.3 Network Coding

Network coding ([3]) is a recent field in information theory proposed to improve the throughput utilization of a given network topology. In communication networks today, packages are always separated with each other during transmissions. The principle behind network coding is to allow intermediate nodes to encode packets. Instead of simply forwarding data, intermediate nodes may recombine several input packets into one or several output packets. An overview of network coding and a discussion of possible Internet applications are given in [17].

To illustrate how network coding improves the propagation of information without global coordination, we consider a simple example in a wireless context with a three node topology. In Figure 2.2, nodes  $A$  and  $B$  want to exchange packets

via a wireless base station, node  $S$ . In the traditional way, nodes  $A$  and  $B$  send packet  $a$  and  $b$  to node  $S$  respectively. Then node  $S$  sends packet  $b$  and  $a$  to node  $A$  and  $B$  respectively. Because of interference in wireless environments, this procedure takes four transmissions as shown on the left side of Figure 2.2.

If network coding is used, Node  $S$  will first linearly combine packets  $a$  and  $b$ , e.g. by binary adding the two packets:  $a \oplus b$ . Instead of forwarding packets  $a$  and  $b$  separately,  $S$  broadcasts  $a \oplus b$ . After receiving the encoded packet, both  $A$  and  $B$  can recover the packet of interest, while the number of transmissions is reduced. This simple example is similar to linear network coding, where the  $\oplus$  operation is replaced by linear combinations of data over some finite field. As we will see later, linear network coding makes efficient propagation of trust documents and the associated operations are distributed.

## 2.4 Network Coding Based Scheme

In this section, we describe our proposed scheme for trust document distribution. We will outline the basic operations of this system, emphasizing some algorithmic parameters that affect its performance.

We assume a population of users that are interested in the trustworthiness of a particular user, that is the target user. The trust documents about the target user are initially stored in a number of users scattered throughout the network. These users are the issuers of the trust documents, or they have retrieved these documents from others. In the rest of the section, we only consider trust documents about one

single target user. All the operations are applied on those documents about the particular target user. Each trust document has a unique ID, which is obtained by applying a universally defined hash function to the document.

In autonomic networks, users usually cannot directly communicate with all other users; they only communicate with a small subset of users, which we call the neighborhood. We assume that the neighboring relation is symmetric, i.e. if node  $A$  is in the neighborhood of  $B$ , then, also  $B$  is in the neighborhood of  $A$ . Symmetric channels are necessary in wireless networks, since reliable transmissions require that the two nodes can exchange information to avoid collisions and interference, such as the RTS/CTS messages [30]. In our scheme, users frequently check with their neighbors for new documents. If there are new documents, these documents are forwarded.

Whenever a user wants to forward trust documents to another node, it produces a linear combination of all the documents it currently stores. The operation of the system is best described in the example of Figuer 2.3. Assume that initially user  $A$  stores several trust documents about a particular target user. User  $A$  will combine all the documents to create an *encoded document* as follows. It will pick some random coefficients  $c_i$ , then multiply each trust document  $i$  with  $c_i$ , and then add the results of the multiplications together, shown as follows

$$\text{EncodedDoc}_1 = (c_1 \cdot \text{Doc}_1) + (c_2 \cdot \text{Doc}_2) + (c_3 \cdot \text{Doc}_3). \quad (2.1)$$

All these operations take place in a finite field. If more than one user requests trust

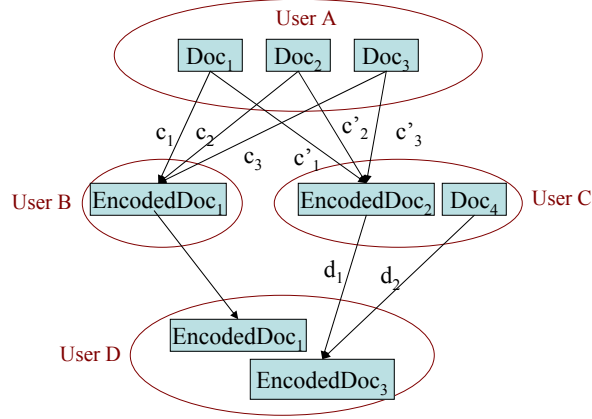


Figure 2.3: Operations of our network coding based scheme

documents from  $A$ ,  $A$  randomly picks different sets of coefficients for these users. Observe that the probability that two distinct users get the same set of coefficients depends on the size of the field. If the field is very small such “collisions” may happen and they will reduce the performance of the system. In most practical cases a field of size  $2^{16}$  should be enough.

User  $A$  will then transmit to user  $B$  the result of addition,  $\text{EncodedDoc}_1$ , the IDs of the encoded trust documents and the coefficient vector  $\mathbf{c} = (c_i)$ . User  $A$  has also another encoded document  $\text{EncodedDoc}_2$ , with its associated coefficients  $\mathbf{c}'$ . User  $A$  transmits it to user  $C$  who stores another trust document,  $\text{Doc}_4$ . User  $D$  is a neighbor of both user  $B$  and user  $C$ . He requires trust documents from both  $B$  and  $C$ . User  $B$  transmits  $\text{EncodedDoc}_1$  directly to User  $D$ , and user  $C$  picks two random coefficient  $d_1$  and  $d_2$  and transmits  $\text{EncodedDoc}_3$  to  $D$ . We have that

$$\text{EncodedDoc}_3 = (d_1 \cdot \text{EncodedDoc}_2) + (d_2 \cdot \text{Doc}_4). \quad (2.2)$$

The coefficient vectors user  $D$  now has are

$$[c_1, c_2, c_3, 0] \text{ and } d_1 \cdot [c'_1, c'_2, c'_3, 0] + d_2 \cdot [0, 0, 0, 1]$$

and the corresponding document IDs are  $\text{Doc}_1$ ,  $\text{Doc}_2$ ,  $\text{Doc}_3$  and  $\text{Doc}_4$ .

Observe that given  $n$  distinct trust documents, a user can recover them after receiving  $n$  encoded documents for which the associated coefficient vectors are *linearly independent* from each other. The reconstruction process is similar to solving linear equations.

Our network coding scheme involves only local interactions. Users need not be aware of the existence of any trust document and its location. They only contact their neighbors to check whether there are new documents or new encoded documents. This is a great advantage as compared to any request-response scheme, such as the Freenet-based scheme. Request-response schemes require routing tables. If the topology of the network changes, new nodes come in or some nodes move out, routing tables need to be updated immediately. While, our scheme adapts quickly to the topology changes, since users only contact their current neighbors. In addition, request-response schemes require that users know the document IDs before sending out requests, which needs global information exchange. Our scheme operates without knowing any document ID.

We have mentioned that there are different types of trust documents with various confidence values, and the importance of these documents varies dramatically. In other words, the trust documents in the network are *heterogeneous*. A good trust

document distribution scheme should take such heterogeneity into consideration. Credentials with high confidence values and of high importance should have high priority in transmission and they should be recovered before other documents. The simplest solution is to transmit these high priority documents separately. However, this solution essentially doubles the resources used for transmitting these documents. In our system, we mark these high priority documents, and they are always considered to be new documents. Therefore, whenever a user with high priority documents wants to forward his documents, he always combines the high priority documents with all other documents. In this way, these high priority documents have much higher chance to be recovered.

## 2.5 Comparison of Distribution Schemes

In the section, we study the performance of our trust document distribution scheme that uses network coding and compare it with the Freenet-based approach in [21]. To evaluate the performance, we calculate the time it takes for each user to obtain all trust documents for the target user and the load of the network.

To study the performance of relatively large number of users, we have implemented both our network coding based scheme and the Freenet based scheme in MATLAB. There are 100 nodes randomly placed in a  $1 \times 1$  square. If the distance between two nodes is less or equal to 0.2, they considered to be neighbors. Assume that 20 distinct documents are randomly stored in these nodes. All users want to obtain these 20 documents.



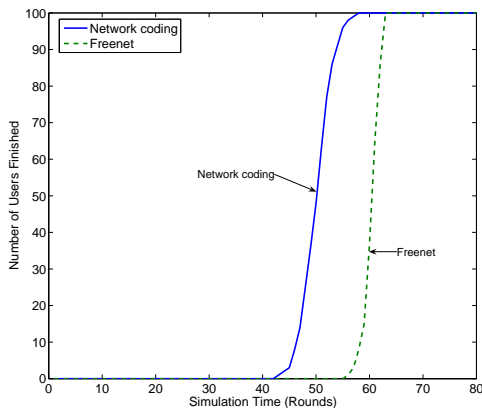


Figure 2.4: Number of nodes finishing over time

The simulation is round based. For our network coding based scheme, at the beginning of each round, each user contacts its neighbors to discover whether there are new documents or encoded documents. When new documents are obtained, users encode them according to the operations described in Section 4.1.1. Once a user can recover all the trust documents, we designate that he has finished, and the number of rounds it takes is called the *finish time*. For the Freenet based scheme, each user sends out a request for one of the documents he does not store. The routing scheme of the requests is described in [21]. Once the requested document is found and sent back, a new request is generated for another document the user does not store. The user stops sending requests once he obtains all documents. Documents are replicated by caching at every node they pass.

We start by comparing the performance of finish time. In Figure 2.4, we plot the finish time of each node, where  $x$ -axis represents the simulation time and  $y$ -axis represents the number of nodes that have finished. The performance of the network coding based scheme is better compared to the Freenet based scheme. Because by

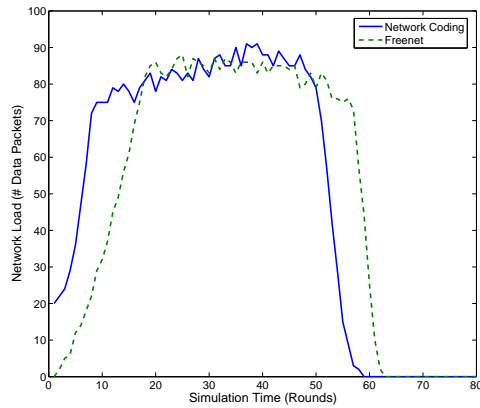


Figure 2.5: Network load over time

combining the documents with network coding, within a few rounds, documents can be spread throughout the network. While for the Freenet based scheme, just one document is transmitted per round, and the search phase at the beginning takes long time.

Figure 2.5 gives the load of the network with time. Our network coding based scheme has relatively higher network load. This is due to the controlled flooding nature of network coding.

## Chapter 3

### Trust Evaluation

#### 3.1 Overview

After distributing trust documents to proper nodes, the next and probably the most important problem is to evaluate target nodes based on these documents. The objective of trust evaluation is to estimate the trustworthiness of targets and thus help to predict their future behaviors. In this chapter, nodes which conduct the evaluation are called *the evaluating nodes*, and nodes which are the targets of the evaluation are called *the target nodes*.

In literature, two types of trust have been studied: global trust and local trust[49]. In global trust, given a target node, there is a unique trust value assigned to the node, independently of the evaluating node. On the other hand, a local trust value (aka personalized trust value) provides a personalized trust value that depends on the point of view of the evaluating node. In this chapter, we are focused on the global trust evaluation. Some of our methods can also be applied to local trust evaluation as well.

## 3.2 An Example on Distributed Trust Evaluation

Consider an example where Alice needs to identify Bob who she meets for the first time. There are several credentials that Alice can get from different sources: the passport Bob shows to Alice and the voice of Bob that Alice heard from phone conversations before they meet. Obviously, Alice has different confidence on these credentials. Usually, she has stronger confidence on the passport of Bob than Bob's voice. However, the passport may be a weaker credential, in case it is expired. Suppose Cathy also obtains some credentials on the identity of Bob, such as the pictures of Bob from the Web.

Given that we are considering autonomic networks, these credentials can not be gathered in one place and there is no centralized server to evaluate trustworthiness. However, Alice and Cathy are able to derive their *direct trust* on the identity of Bob separately based on the credentials they obtain (passport and voice of Bob for Alice and pictures of Bob for Cathy). Suppose that Alice and Cathy can exchange their direct trust information either directly or through other people by the way of *recommendations*. The problems we are interested in are how to exchange trust information and how to combine all available information such that the system can derive a proper trust value on the identity of Bob in a distributed manner. Figure 3.1 shows the procedure of trust evaluation for this example. The rest of the chapter studies policies on the derivation of direct trust and evaluated trust in details.

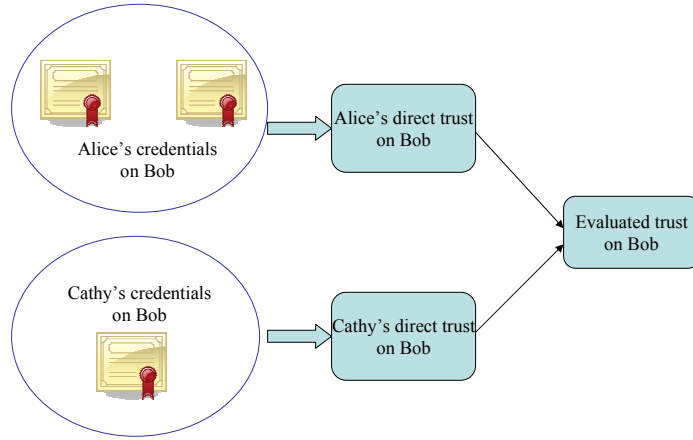


Figure 3.1: Procedure of trust evaluation

### 3.3 Global Trust Evaluation

In autonomic networks, trust documents a node obtains on the target node are usually not enough to infer the trust value of the target. However, there may be more than one nodes that obtain different trust documents about the target. The system should integrate these trust documents to achieve better estimation on the trustworthiness of targets.

As we discussed in Chapter 1, the trust states that the node behaves in a reliable way, but since there are many different things the node can or may do (offer a service, issue various types of credentials, etc.), it may be possible that the node is reliable in doing one thing but not the other. To take this into account, Maurer in [50] distinguishes different trust. A more complete attempt to classify various forms of trust has been proposed in [29, 38]. However, in the global trust evaluation, we will make the assumption that trust is a general proposition about anything the node may do, i.e. nodes are simply good or bad in certain degree.

### 3.3.1 Network Model

We model the network as a dynamic directed graph  $G(k)(V(k), E(k))$ , in which nodes are the entities/peers in the network and arcs represent trust relations. In this work, we consider discrete time and  $k$  is a specific time slot.  $V(k)$  is the set of nodes in the network at time  $k$ . If arc  $(i, j) \in E(k)$ , there are direct trust node  $i$  has on node  $j$  at time  $k$ . In the followings, we may omit the time  $k$  for easier notations when suitable.

Graph  $G$  is called the *trust graph*, which is a logical graph, as opposed to the physical graph, in which nodes are connected if they are one hop away in terms of physical transmissions. Suppose that the number of nodes in the network is  $N$ , i.e.  $|V| = N$  and nodes are labeled with indices  $V = \{1, \dots, N\}$ .

In Chapter 2, nodes have obtained trust documents needed for evaluating the trustworthiness of certain target nodes. Based on these documents, each node can derive *direct* trust values on target nodes if he has some trust documents on the target. If  $(i, j)$  is a direct arc in graph  $G$ , i.e.  $(i, j) \in E$ , node  $i$  has direct trust on  $j$ . As we have discussed, there are various types of trust documents. The evaluating node may have different confidence on these documents. Therefore, the direct trust value is no longer a binary value (trust or distrust). We take trust values continuous with range  $[-1, 1]$ , where  $-1$  stands for completely distrust,  $1$  for completely trust and  $0$  for totally neutral. Define  $d_{ij}$  as the direct trust value  $i$  has on  $j$ . Therefore, the trust graph  $G$  is a weighted graph, where  $d_{ij}$  is the weight on the arc  $(i, j)$ . If  $i$  has no direct trust relation on  $j$ , i.e.  $(i, j) \notin E$ ,  $d_{ij}$  may set to be  $0$ . Figure 3.2 is an

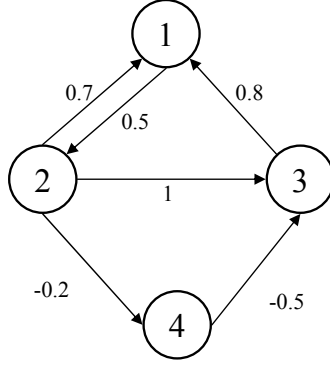


Figure 3.2: A trust graph

example of a trust graph with 4 nodes. We define the neighbor set of node  $i$  as

$$\mathcal{N}_i = \{j | (j, i) \in E\} \subseteq V \setminus \{i\},$$

which is the set of nodes that have direct trust values on  $i$ .

There are three types of trust we need distinguish: real, direct and evaluated trust.

- **Real** trust: the real state of the target nodes, denoted as  $t_i$  for node  $i$ .
- **Direct** trust: the trust values each node obtains from trust documents in hand, denoted as  $d_{ij}$  for the direct trust node  $i$  on node  $j$ . The policies used to set the direct trust values are discussed in detail in the next subsection.
- **Evaluated** trust: the value that is derived from the evaluation rule. Most of the time, the evaluated trust values are called the trust values for simplicity, denoted as  $s_i$  for node  $i$ .

The vector  $T = [t_1, \dots, t_N]$  is called the real trust vector, and  $S = [s_1, \dots, s_N]$  is the

evaluated trust vector or trust vector. Matrix  $D = \{d_{ij}\}$  is the direct trust matrix. Notice that in this dissertation we consider two types of trust metrics: deterministic trust metric and stochastic trust metric. If the deterministic trust metric is used, trust values are scalar between  $-1$  and  $1$ . If the system uses the stochastic metric, trust values are random variables with range  $[-1, 1]$ .

The purpose of the global trust evaluation is to find the best estimates of  $t_i$ 's,  $s_i$ 's, based on all the  $d_{ij}$ 's.

### 3.3.2 Direct Trust

Direct trust is derived from trust documents each node has in hand. Values of direct trust depend on the value of trust documents and their issuers. Users design their own policies on deriving direct trust. The policy for deriving direct trust is out of the scope of this dissertation. In this subsection, we just give some examples on these policies for both deterministic trust and stochastic trust.

We first consider the deterministic trust. Suppose Alice wants to identify Bob, and she has got the valid passport of Bob and recognized the voice of Bob based on their phone conversations before they meet. Alice may have very high confidence on the passport, then with the passport along she identifies Bob with trust, say  $0.9$ , where the value  $0.9$  is determined by Alice's policy. Furthermore, by adding the voice of Bob, Alice may increase the trust value to  $0.95$ .

Now we give an example on stochastic trust. Let's consider a binary case where nodes are either GOOD or BAD<sup>1</sup>, i.e.  $t_i = 1$  or  $-1$ , and  $d_{ij}$  is equal to  $1$  or  $-1$

---

<sup>1</sup>The capital letters emphasize that nodes are either completely good or completely bad. In the



as well. Suppose a GOOD node  $i$  obtains  $m$  trust documents about node  $j$ . The confidence  $i$  has on these  $m$  documents are the same. In terms of probability,  $i$  believes that the probability of the issuers of these  $m$  documents making wrong statements is  $p_e$ .

$$\Pr(\text{document states that } j \text{ is GOOD} | t_j = 1) = 1 - p_e. \quad (3.1)$$

The policy that node  $i$  uses to derive  $d_{ij}$  is the majority rule. Suppose there are  $k$  out of these  $m$  documents stating that  $j$  is GOOD, and the other  $m - k$  stating the opposite. If  $k \geq m/2$ ,  $d_{ij} = 1$ , and  $k < m/2$ ,  $d_{ij} = -1$ . Then node  $i$  can derive the conditional probability of  $d_{ij}$  given that  $i$  and  $j$  are GOOD and  $i$  obtains the  $m$  trust documents as the following

$$\begin{aligned} & \Pr(d_{ij} = 1 | t_i = t_j = 1, \text{ the } m \text{ trust documents with confidence } 1 - p_e, \text{ majority policy}) \\ = & \sum_{k=0}^{\lfloor m/2 \rfloor} \binom{m}{k} p_e^k (1 - p_e)^{m-k}. \end{aligned} \quad (3.2)$$

For the sake of simplicity, we omit the trust documents and direct trust policy in the conditional probability for the rest of the Chapter. Notice that usually direct trust policy is fixed but the trust documents may change over time, then the corresponding conditional probability changes as well. The conditional probability of  $d_{ij}$  is denoted

---

case of uncertainty ( $t_i \in [-1, 1]$ ), we say a node is good with small letters if  $t_i > 0$  and bad with small letters if  $t_i < 0$ .

as  $P(d_{ij}|t_i, t_j)$ . We have

$$\Pr(d_{ij} = -1|t_i = t_j = 1) = 1 - \Pr(d_{ij} = 1|t_i = t_j = 1). \quad (3.3)$$

Regarding the BAD nodes, if we know their policies of setting direct trust values, the conditional probability is derived accordingly. However, in most cases, we may not know their policies, then uniform distribution is chosen for the conditional probability of  $d_{ij}$  where node  $i$  is BAD. The reason is that uniform distribution achieves the maximum entropy. Therefore,

$$\Pr(d_{ij} = 1|t_i = -1, t_j = 1) = \Pr(d_{ij} = 1|t_i = -1, t_j = -1) = 1/2; \quad (3.4)$$

Another more complicate form of  $P(d_{ij}|t_i, t_j)$  is the *truncated normal distribution* where nodes in the network are good or bad in some extent, i.e.,  $t_i$  is continuous in  $[-1, 1]$ . Specifically, suppose  $X \sim N(\mu, \sigma^2)$  has a normal distribution and lies within the interval  $X \in (a, b), -\infty \leq a < b \leq \infty$ . Then the probability density function of  $X$  is

$$f(x; \mu, \sigma, a, b) = \frac{\varphi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}, \quad (3.5)$$

where  $\varphi(\cdot)$  is the probability density function of the standard normal distribution,  $\Phi(\cdot)$  is the corresponding cumulative distribution function. In the case for  $P(d_{ij}|t_i, t_j)$ ,  $d_{ij}$  is the estimate  $i$  has on the trustworthiness of  $j$ . It is reasonable to assume that  $\mu = t_i$ . The error of such estimate depends on the capability of node  $j$  which is a function of  $t_i$ . According to our definition of trust, the truncated

boundary is set as  $a = -1$  and  $b = 1$ . Therefore, we have the following definition on the conditional PDF

$$P(d_{ij}|t_i, t_j) = \begin{cases} \frac{1}{z} e^{(d_{ij}-t_j)t_i^2}, & t_i > 0 \\ \frac{1}{2}, & t_i \leq 0 \end{cases} \quad (3.6)$$

where  $z = \int_{-1}^1 e^{(x-t_j)t_i^2} dx$ . For a good node with positive real trust value ( $t_i > 0$ ), the direct trust value he has on another node  $j$  is a truncated normal distribution in the range of  $[-1, 1]$ . Before the truncation, the mean is the real trust value  $t_j$  and the variance is the inverse of  $t_i$ . The more trusted  $i$  is, the more accurate its direct trust value on  $j$ . The conditional probability is still chosen to be uniformly distributed for bad nodes.

We have described examples on the policy of direct trust. In the rest of this chapter, we assume that direct trust has been derived and given to the evaluation rule.

### 3.3.3 Voting Rule

To evaluate a particular node in the network, say node  $i$ , all the direct trust  $i$ 's neighbors has on  $i$  should be take into account. The natural approach is to aggregate all the neighbors' direct trust values. This approach is called the *voting rule*, in which votes are neighbors' direct trust values and those neighbors are called voters. However, a rule using naïve average is not a good estimation. Votes from voters with high (evaluated) trust values are more credible, so they should carry

larger weights. On the other hand, a vote from a distrusted or neutral voter (i.e. node  $j$  with  $s_j \leq 0$ ) should not be valid. Based on the above arguments, we define the voting rule as the following

$$s_i = f(d_{ji}s_j\mathbf{1}(s_j > 0)|j \in \mathcal{N}_i), \quad (3.7)$$

where  $f : \mathbb{R}^{|\mathcal{N}_i|} \rightarrow [-1, 1]$  and  $\mathbf{1}(\cdot)$  is an indicator function.

Apparently, the trust value  $s_j$  depends on trust values of  $j$ 's neighbors of  $j$ 's neighbors and their votes on  $j$ . Therefore  $s_j$  is also evaluated at the same time, and so are  $j$ 's neighbors. The whole evaluation evolve as the voting rule iterates throughout the network and Eqn. (3.7) can be written as

$$s_i(k+1) = f(d_{ji}(k)s_j(k)\mathbf{1}(s_j(k) > 0)|j \in \mathcal{N}_i). \quad (3.8)$$

Thus the trust evaluation can be considered as a dynamic process which evolves with time.

The voting rule evolves in discrete time. Even if a distributed algorithm is asynchronous and communication delays are real (i.e., not integer) variables, the events of interest may be indexed by a discrete variable. In Eqn. (3.8), each time step includes evaluation of trust values which is followed by updating the direct trust values ( $d_{ji}$ 's); so the restriction to discrete time entails no loss of generality.

Our interest is to study the evolution of the evaluated trust vector  $S$  and its values when the process converges. The motivation of trust evaluation is, of course,

to be able to detect bad nodes and trust good nodes. It is important to investigate whether  $S$  can correctly estimate the trust vector  $T$  at the steady state.

### 3.4 Deterministic Voting Rule

There are several choices for the voting rule. For instance, it can be the average, maximum or minimum of all votes. In this section, we investigate a simple rule: the weighted average of all votes [31]. The voting rule for node  $i$  is the following:

$$s_i(k+1) = \frac{1}{z_i(k)} \sum_{j \in \mathcal{N}_i(k)} d_{ji}(k) s_j(k) \mathbf{1}(s_j(k) > 0), \quad (3.9)$$

where  $z_i = \sum_{j \in \mathcal{N}_i(k)} \mathbf{1}(s_j(k) > 0)$ .

As mentioned in Sec. 3.3.3, the voting rule must satisfy that  $f : \mathbb{R}^{|\mathcal{N}_i|} \rightarrow [-1, 1]$ .

We first prove the following lemma.

**Lemma 1**  $s_i(k) \in [-1, 1], \forall 1 \leq i \leq N, k = 0, 1, \dots$  in Eqn. (3.9).

**Proof** For any  $k = 0, 1, \dots$ , we have from Eqn. (3.9)

$$\begin{aligned} |s_i(k+1)| &= \left| \frac{1}{z_i(k)} \sum_{j \in \mathcal{N}_i(k)} d_{ji}(k) s_j(k) \mathbf{1}(s_j(k) > 0) \right| \\ &\leq \frac{1}{|z_i(k)|} \sum_{j \in \mathcal{N}_i(k)} |d_{ji}(k) s_j(k) \mathbf{1}(s_j(k) > 0)| \end{aligned} \quad (3.10)$$

$$= \frac{\sum_{\substack{j \in \mathcal{N}_i(k) \\ s_j(k) > 0}} |d_{ji}(k)| |s_j(k)|}{\sum_{\substack{j \in \mathcal{N}_i(k) \\ s_j(k) > 0}} 1} \quad (3.11)$$

$$\begin{aligned} &\leq \frac{\sum_{\substack{j \in \mathcal{N}_i(k) \\ s_j(k) > 0}} |s_j(k)|}{\sum_{\substack{j \in \mathcal{N}_i(k) \\ s_j(k) > 0}} 1} \\ &\leq 1. \end{aligned} \quad (3.12)$$

Inequality (3.10) comes from the triangle inequality of summation. By removing all nonnegative  $s_j(k)$ 's, we get Eqn. (3.11). The last two inequality follows from the fact that  $|d_{ji}(k)| \leq 1$  and  $|s_j(k)| \leq 1$ . Therefore, we have  $-1 \leq s_i(k+1) \leq 1$ . ■

Thus the weighted average rule defined in Eqn. (3.9) is a legitimate voting rule. Furthermore the rule can be written in the state form. Let's define  $Z(k)$  to be the diagonal matrix whose  $i$ th diagonal element is  $z_i(k)$  and the matrix  $D(k)$  represents the values of all votes.  $d_{ji} = 0$  if node  $j$  has no direct trust relation on  $i$ . Then we have

$$S(k+1) = Z(k)^{-1}D^T(k)S^+(k). \quad (3.13)$$

$S^+(k)$  is a column vector where  $s_i^+(k) = s_i(k)\mathbb{1}(s_i(k) > 0)$ .

### 3.4.1 Convergence

In this section, we regard the trust establishment process as a dynamic system. We discuss the convergence property of the system and investigate the spreading of trust as the system reaches the steady state.

The voting rule of Eqn. (3.13) can be written as a mapping

$$S(k+1) = \mathcal{F}^{(k)}(S(k)) \quad (3.14)$$

where  $\mathcal{F}^{(k)} : [-1, 1]^N \rightarrow [-1, 1]^N$ .

We assume that after a certain time instant  $K$ , all nodes have obtained sufficient trust documents and  $d_{ij}$ 's keep as constants. Function  $\mathcal{F}^{(k)}$  does not change

over time as  $d_{ij}$ 's are constants. Set  $\mathcal{F} = \mathcal{F}^{(k)}$  when  $k > K$ . We want to prove that the iterated rule converges to the unique fixed point.

**Theorem 2** *If there exists  $i$  and  $j$  where  $< 0 < |d_{ij}(k)| < 1$ ,  $\mathcal{F}$  is a contraction mapping, that is  $|\mathcal{F}(S) - \mathcal{F}(R)| \leq l|S - R|$ , where  $l$  is a constant and  $0 < l < 1$ .*

**Proof** Let's define a square matrix  $F = Z^{-1}D^T$  and the element of  $F$  on row  $i$  column  $j$  as  $f_{ij}$ . Since for different  $S$ ,  $Z$  might be different. We use  $F_S$  (and  $Z_S$ ) to denote the matrix  $F$  (and  $Z$ ) corresponds to vector  $S$ . We first consider the case where elements of both  $S$  and  $R$  are positive. Then we have and  $F_R = F_S$ .

$$\begin{aligned} |\mathcal{F}(S) - \mathcal{F}(R)|^2 - |S - R|^2 &= (S - R)^T F^T F (S - R) - (S - R)^T (S - R) \\ &= (S - R)^T (F^T F - I)(S - R). \end{aligned} \quad (3.15)$$

Since we know that for each row of  $F$ ,

$$\sum_j |f_{ij}| < 1. \quad (3.16)$$

We have that all the diagonal elements of  $F^T F$  are strictly less than 1 and  $F^T F$  is a real symmetric matrix. Therefore,  $F^T F - I$  is a negative definite matrix. Therefore,

$$|\mathcal{F}(S) - \mathcal{F}(R)|^2 < |S - R|^2 \quad (3.17)$$

Now consider  $R$  and  $S$  with negative elements. If  $s_i < 0$ , then  $r_i < 0$  for any  $i \in \{1, \dots, N\}$ . Thus we have  $F_R = F_S$ . The proof is the same as the above. The

mapping  $\mathcal{F}$  is a contraction mapping. ■

Banach fixed point theorem states that every contraction mapping on a nonempty complete metric space has a unique fixed point, and that for any initial vector  $S(0)$  the iterated function sequence converges to the fixed point. Suppose the voting rule converges to a vector  $S$ , then we have that

$$S = \mathcal{F}(S).$$

$S$  is the fixed point of  $\mathcal{F}$ . The following inequality describes the speed of convergence:

$$|S - \mathcal{F}^k(S)| \leq \frac{l^k}{1-l} |\mathcal{F}(S(0)) - S(0)|. \quad (3.18)$$

### 3.4.1.1 Voting in a virtuous network

We start with a virtuous network with no adversary, i.e. all nodes behave rationally and all votes are “reasonable”, which means we admit the uncertainty of the voting value but no one is voting maliciously, we have  $d_{ij} \in [0, 1]$ . Furthermore, nodes start with trust value no less than 0. Otherwise if a node has trust value less than 0 initially, it is excluded by others immediately. Therefore,  $s_i(k) \geq 0$ ,  $\forall i \in V$  and  $k \geq 0$ . Our goal is to show that as  $k \rightarrow \infty$  the vector sequence  $\{S(k)\}$  converges and to find the steady-state value  $S$ .

To determine the trustworthiness of nodes, we apply a threshold rule on the steady states of trust values, which are defined as  $s_i = \lim_{k \rightarrow \infty} s_i(k)$ . Our threshold



rule is dependent on a system defined parameter  $\eta$  as follows:

$$\text{Node } i \text{ is } \begin{cases} \text{trusted,} & \text{if } s_i \geq \eta \\ \text{neutral,} & \text{if } s_i < \eta \end{cases} .$$

Define a matrix  $F = Z^{-1}D^T$ , then the state equation (3.13) is written as  $S(k) = F^k S(0)$ . First, a very simple scenario is considered where all votes are all of the value 1, which means all nodes are able to correctly verify their neighbors as trusted. Therefore  $D = A$ , where  $A$  is the adjacency matrix of graph  $G$ , and  $F$  is the normalized adjacency matrix. It's trivial to verify that  $F$  is a *stochastic* matrix<sup>2</sup>. We show in the following that the convergence behavior of  $S(k)$  depends on whether  $F$  is reducible or not, which represents the connectivity of the network.

- $G$  is connected, i.e.  $F$  is irreducible.

Since  $F$  is a stochastic matrix, the largest eigenvalue of  $F$  is 1. Let  $\pi$  be the right eigenvector corresponding to eigenvalue 1, which is an  $N$ -dimension normalized row vector<sup>3</sup>, then  $\pi F = \pi$ . We could prove by ergodicity that [61],

$$F^k = \begin{bmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{bmatrix} .$$

Thus  $\forall i \in V$ ,  $s_i \triangleq \lim_{k \rightarrow \infty} s_i(k) = \pi \times S(0) = \sum_{j=1}^N \pi_j s_j(0)$ . Therefore,

---

<sup>2</sup>A matrix is called a stochastic matrix, if the sum of the elements of each row is 1.

<sup>3</sup>For a normalized vector, the sum of all elements is equal to 1.

every node reaches the same trust value at the steady state. We can see that whether a node is trusted or not depends on the initial configuration of  $S(0)$ . Obviously, if  $\sum_{j=1}^N \pi_j s_j(0) \geq \eta$ , all nodes are trusted, and none is trusted otherwise. Therefore the initial trust value is very crucial here. It is easier to establish a complete trusted network if a large number of nodes have high trust value initially, otherwise none will get trusted.

- $G$  is not connected, i.e.  $F$  can be decomposed as  $F = \text{blockdiag}[F_1, F_2, \dots, F_L]$ , where  $F_l, l = 1, \dots, L$  are irreducible matrices of order  $N_l, \sum_{l=1}^L N_l = N$ . Thus the graph  $G$  has  $L$  components, which are disconnected with each other. Let's use  $\mathcal{C}_i, i = \{1, \dots, L\}$  to denote those components, where  $\mathcal{C}_i$  has normalized adjacency matrix  $F_i$ . Similarly, since  $F$  is irreducible, we can find the vector  $\pi_i$  which is the right eigenvector of  $F_i$  corresponding to eigenvalue 1, such that  $s_i \triangleq \lim_{k \rightarrow \infty} s_i(k) = \sum_{j \in \mathcal{C}_i} \pi_j s_j(0)$ , if  $i \in \mathcal{C}_i$ . Therefore, the trustworthiness of a node is related to the initial configuration within its connected component. It's easy to extend the results of irreducible matrices to reducible ones by applying the method above. From then on, we will assume  $G$  is connected.

More generally, the voting value  $d_{ji}$  is less or equal to 1 instead of always being 1, i.e., uncertainty is considered. Then  $F$  is a semi-stochastic matrix. It's known that for a semi-stochastic matrix  $F, F^k \rightarrow 0$  as  $k \rightarrow \infty$ , thus  $S$  also goes to 0. Therefore, nodes **cannot** be trusted at all if there is uncertainty in votes.

We have shown that using the simple voting scheme, trust can only be evaluated under certain strict conditions: all voting values are 1 and the initial configura-

tion must satisfy  $\sum_{j=1}^N \pi_j s_j(0) \geq \eta$ . A single vote with value less than 1 will result in failure of trust evaluation. The result is not obvious when we define the voting rule. This actually tells us the difficulties of designing algorithms in autonomic networks. We want to ensure trust evaluation in virtuous networks without depending on the initial states. Therefore we introduce the notion of leaders, which are agents that are always trusted with trust value 1.

### 3.4.1.2 Voting with leaders

As we just mentioned, leaders are pre-trusted agents. For instance, they can be the leader of a cluster, or agents holding a certificate signed by authorities. To simplify the discussion, we also assume all leaders only vote for nodes that they fully trust. Therefore, if a node  $i$  is trusted with  $b_i$  leaders, it will get  $b_i$  more votes with value 1. So define  $b_i$  as the number of leaders that fully trust node  $i$ . Let  $B$  be the diagonal matrix with  $i$ th diagonal element equal to  $b_i$  and  $\mathbf{1} = [1 \dots 1]'$ . Then the updating rule in (3.13) changes to

$$S(k) = (Z + B)^{-1}(DS(k-1) + B\mathbf{1}). \quad (3.19)$$

Again as we did in Sect. 3.4.1.1, first the situation with all 1-valued votes is considered, i.e.,  $D=A$ . Define  $\tilde{S}(k) = \mathbf{1} - S(k)$ . By Eqn.(3.19), we can deduce that  $\tilde{S}$  satisfies the equation

$$\tilde{S}(k) = (Z + B)^{-1}A\tilde{S}(k-1) \triangleq \tilde{F}\tilde{S}(k-1), \quad (3.20)$$

where  $\tilde{F} = (Z + B)^{-1}A$ . Thus,  $\tilde{S}(k) = \tilde{F}^k \tilde{S}(0)$ . It's easy to observe that if there is at least one node  $i$  such that  $b_i > 0$ ,  $\tilde{F}$  is a semi-stochastic matrix. We have  $\tilde{F}^k \rightarrow 0$ , as  $k \rightarrow \infty$ . Therefore  $S(k) \rightarrow 1$ . It follows that

**Corollary 3** *For a connected virture network, all nodes get fully trusted in steady state given all 1-valued votes, iff  $\exists$  at least one node that connects to one or more leaders.*

Thus adding just one leader guarantees a fully trusted network. Now consider votes with uncertainty. We have the following theorem.

**Theorem 4** *Let  $\eta$  be the threshold of trustworthiness. We have the following conditions on the number of leaders:*

1. *Define a  $N$ -dimension column vector  $V$ .  $v_i = 1$  if node  $i$  is trusted at the steady state, otherwise  $v_i = 0$ , we have that*

$$B(\eta V - \mathbf{1}) > \eta(Z - D)V.$$

2. *If all nodes are trusted at the steady state, the number of leaders for each node must satisfy*

$$B\mathbf{1} \geq \eta\mathbf{1} - \eta(Z - D)\mathbf{1}.$$

**Proof** Using a similar technique as above, let  $\tilde{S}(k) = \xi - S(k)$ . Substituting it into Eqn.(3.19), we have

$$\tilde{S}(k) = (Z + B)^{-1}D\tilde{S}(k - 1) + (Z + B)^{-1}((Z + B - D)\xi - B\mathbf{1}). \quad (3.21)$$

Let the last term on the right hand side of Eqn. (3.21) be 0. Then  $\tilde{S}(k) \rightarrow 0$  as  $k \rightarrow \infty$ , so  $T(k) \rightarrow \xi$ .

According to the decision rule, we want  $S = \lim_{k \rightarrow \infty} S(k) > \eta \mathbf{1}$ , therefore  $\xi > \eta \mathbf{1}$ . Consider the case  $\xi = \eta \mathbf{1}$ , since  $(Z + B)^{-1}((Z + B - D)\xi - B\mathbf{1}) = 0$ , we have

$$B\mathbf{1} = \frac{\eta}{1 - \eta}(Z - D)\mathbf{1}.$$

Notice that the function  $f(x) = \frac{x}{1-x}$  is strictly increasing for  $x \in [0, 1)$ . If we want  $\xi > \eta \mathbf{1}$ , then

$$B\mathbf{1} > \frac{\eta}{1 - \eta}(Z - D)\mathbf{1}.$$

The first condition can be proved similarly. ■

Similarly, for graphs that are not connected, the theorem holds for each connected component separately.

Theorem 3 proves as well as provides a network design method to establish a fully trusted network by introducing certain number of leaders. Moreover, this method only employs local interactions, and it converges to the desired result without dependence on any initial configuration.

### 3.4.2 Network Topology

Having found a simple and light-weight trust evaluation method, the next concern is the dynamics of the procedure. In particular, we investigated the time it takes to reach the steady state, in other words how fast the trust values converge.

We introduce an important theorem, the *Perron-Frobenius Theorem*[13], which states that for a stochastic matrix  $A$ ,  $A^n = \lambda_1^n v_1 u_1^T + O(n^{m_2-1} |\lambda_2|^n)$ , where  $\lambda_1$  is the largest eigenvalue with its left and right eigenvector  $u_1$  and  $v_1$  respectively<sup>4</sup>,  $\lambda_2$  is the second largest eigenvalue and  $m_2$  is the algebraic multiplicity of  $\lambda_2$ . Thus the convergence rate of  $A^n$  is of order  $n^{m_2-1} |\lambda_2|^n$ . Normalized adjacency matrices are stochastic matrices, therefore those with smaller  $\lambda_2$  converge faster.

Then the question becomes: what kind of networks or which network topology has smaller  $\lambda_2$ ? Since the well-known small-world paper by Watts and Strogatz in 1998 ([69]), research on network topology has gained voluminous attention. The small world models have two prominent properties: high clustering coefficient and small average graphical distance between any pairs. The small average distance essentially indicates that nodes can communicate with other nodes in a few hops given a very large network. In the past five years, there has been substantial research on the small-world model in various complex networks, such as Internet and biological systems. As a social concept, distributed trust networks exhibit small-world properties too. In [15], it was shown that the PGP certificate graph has the small-world property. Therefore, the study of network topology using the small-world model will help us to understand practical trust systems.

In this work, we consider one of many small-world models, the so-called  $\varphi$ -model ([69]), which is modeled by adding small number of new edges into a regular lattice. The network starts as a two-dimensional lattice with periodic boundary, and the neighbors are those one hop away. Then new edges are added by randomly

---

<sup>4</sup>Notice that  $\lambda_1 = 1$  for a stochastic matrix.

choosing two unlinked nodes. In our simulations, a network is considered with 400 nodes on the lattice. For the original lattice, each node has 4 neighbors, and the total number of edges is 800. Each simulation includes several rounds. At each round, 2 new edges are added randomly into the network. The second largest eigenvalue and the convergence time are computed for each round. Figure 3.3 shows the second largest eigenvalue  $\lambda_2$  as a function of the number of new edges added. It shows that with more edges added, the second largest eigenvalue of the graph decreases. So, generally speaking, small-world networks have smaller second largest eigenvalue than the original lattice. Figure 3.4 illustrates the substantial changes of the convergence time as new edges are added. Notice that given 8 new edges are added, which is just 1% of the total edges, the convergence time drops from 5000 rounds to 500 rounds. Thus trust is established much faster in a network with small-world property than a regular lattice. This conclusion also provides a direction for network management so as to achieve good performance.

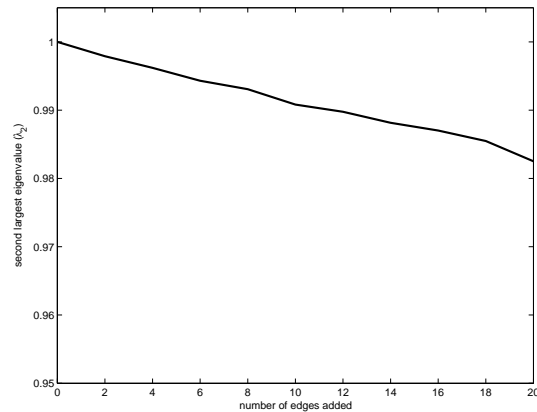


Figure 3.3: The second largest eigenvalue  $\lambda_2$

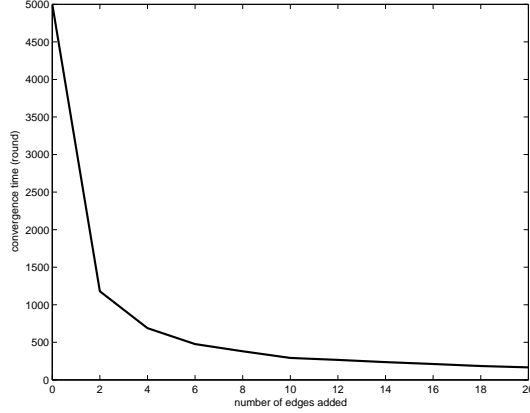


Figure 3.4: Convergence time

## 3.5 Stochastic Voting Rule

### 3.5.1 Stochastic Model

In the stochastic model, the trust evaluation problem is interpreted as an estimation problem.  $T$  is the vector to be estimated and  $S$  is the estimate. The direct trust values are the observation used for estimation. Therefore, the direct trust value  $d_{ij}$  is modeled as a random variable that conditioned on real state of  $i$  and  $j$  and the policies node  $i$  uses to derive the direct trust value.

One important issue for estimation is to correctly model the conditional probability of the direct trust value  $d_{ij}$ .  $d_{ij}$  depends on the real trust values of  $i$  and  $j$  and the policies used to derive the direct trust value. The model of the conditional probability of  $d_{ij}$  varies in different trust evaluation systems. In this work, we study the general evaluation rule and assume the conditional probability is known by the evaluating node.



Because the direct trust value  $d_{ij}$  depends only on  $t_i$  and  $t_j$  we have that

$$\Pr[d_{ij}|t_l, l \in N, d_{kl}, (k, l) \in E \text{ and } (k, l) \neq (i, j)] = \Pr[d_{ij}|t_i, t_j] \quad (3.22)$$

For four distinct nodes  $i, j, k, l$  in the network, we have

$$\Pr[d_{ij}, d_{kl}|t_i, t_j, t_k, t_l] = \Pr[d_{ij}|t_i, t_j] \cdot \Pr[d_{kl}|t_k, t_l]. \quad (3.23)$$

We further assume that the *a priori* probability on the real trust value is known, i.e.  $P(t_i)$  is known and independent with each other. The value of  $P(t_i)$  depends on the environment node  $i$  is in. For instance, a node that is physically well-protected has high probability being good. In some case,  $P(t_i)$ 's for all node  $i \in V$  are identically independent. For instance, a network with only GOOD and BAD nodes, the percentage of GOOD nodes in the network is known in advance denoted as  $p_G$ . Thus,  $P(t_i = 1) = p_G, \forall i \in V$ .

### 3.5.2 Estimation

The trust evaluation is considered as an estimation problem, where the observations are  $d_{ij}$ 's and the trust values are the estimates. We first consider an ideal case where all direct trust values (i.e., observations) are collected in one place. We have the *posterior* of  $T$  given the direct trust values as follows

$$\Pr[T|\mathcal{D}] = \frac{\Pr[\mathcal{D}|T] \Pr[T]}{\sum_T \Pr[\mathcal{D}|T] \Pr[T]}, \quad (3.24)$$

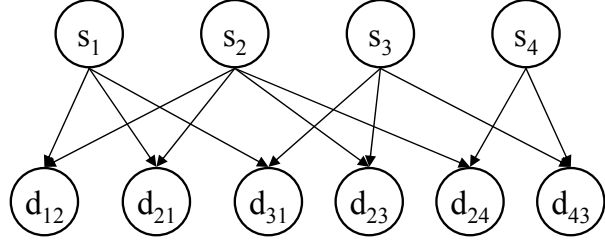


Figure 3.5: Bipartite Bayesian Network for the Trust Graph 3.2

where  $\mathcal{D} = \{d_{ij}, \forall (i, j) \in E\}$  is the set of all direct trust values and  $\Pr[T]$  is the *prior* probability of nodes' real trust values, which is assumed to be independent of each other. Therefore, the distribution of the *estimated* trust vector  $S$

$$\begin{aligned} \Pr[S] &= \Pr[T = S | \mathcal{D}] \\ &= \frac{\Pr[d_{ij}, (i, j) \in E | s_i, i \in V] \prod_{i \in V} \Pr[s_i]}{Z}, \end{aligned} \quad (3.25)$$

where  $Z = \sum_S \Pr[d_{ij}, (i, j) \in E | s_i, i \in V] \prod_{i \in V} \Pr[s_i]$  is the normalization factor.

### 3.5.3 Bayesian Network

According to the independence assumption in Eqn. (3.23), the estimation can be modeled as a bipartite Bayesian networks. Fig. 3.5 gives the corresponding Bayesian network for the trust graph Fig. 3.2.

The Bayesian network belongs to graphical models. Graphical models have been used in variety of fields, such as the Markov random field in signal processing and Viterbi decoding algorithm in coding. The algorithms for these graphic model may be viewed as instances of the *sum-product* (or *belief propagation* or *probability*

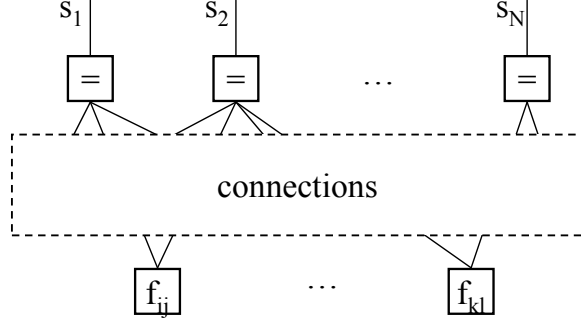


Figure 3.6: General Factor Graph for Trust Graphs

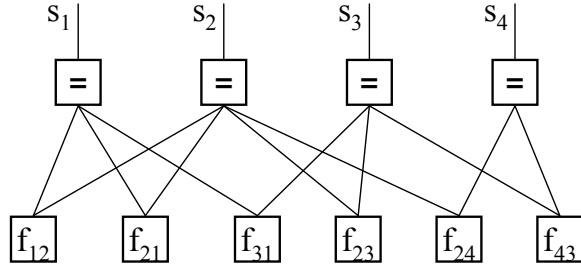


Figure 3.7: Factor Graph for the Trust Graph 3.2

*propagation* algorithm and the *max-product* (or *min-sum*) algorithm. These algorithms are operated by *message passing* in graphical models. Specific instances of such algorithms include Kalman filtering and smoothing, the forward-backward algorithm for hidden Markov models, probability propagation in Bayesian networks, and decoding algorithms for error correcting codes such as the Viterbi algorithm and the iterative decoding of LDPC codes. In recent years, researchers unified all these algorithms in terms of *factor graph*[42, 46].

The trust estimation can also be interpreted as a factor graph introduced in [23] (and there called “normal graphs”). The variables are  $s_i$ 's and the factor nodes are  $\Pr[d_{ij}|s_i, s_j]$ , denoted as  $f_{ij}(s_i, s_j)$ . The factor graph of a general trust graph is show in Fig. 3.6. Figure 3.7 shows the factor graph of Figure 3.2. For autonomic

networks, the corresponding trust graph is usually sparse: each node is connected only to a small number of other nodes.

We are interested in the marginal probability distribution  $\Pr[s_i]$  for each node  $i$ .

$$\Pr[s_i] = \sum_{\sim\{s_i\}} \Pr[S] = \sum_{\sim\{s_i\}} \prod_{(i,j) \in E} \Pr[d_{ij}|s_i, s_j] \prod_{i \in V} \Pr[s_i]/Z, \quad (3.26)$$

where  $\sim\{s_i\} = S \setminus \{s_i\}$ . Notice that in Eqn. (3.26) and the rest of this section, we assume discrete random variables. Our arguments can be easily extended to continuous random variables by replacing summation to integrals. Now define  $f_{ij}(s_i, s_j) = \Pr[d_{ij}|s_i, s_j]$ , then

$$\Pr[s_i] = \sum_{\sim\{s_i\}} \prod_{(i,j) \in E} f_{ij}(s_i, s_j) \prod_{i \in V} \Pr[s_i]/Z. \quad (3.27)$$

The computation of the marginal probability distribution belongs to the sum-product algorithm, which can be implemented using the message passing algorithm. Suppose the message send from the factor node  $f_{ij}$  to variable  $s_i$  is  $\mu_{f_{ij} \rightarrow i}$ , then the message is the function

$$\mu_{f_{ij} \rightarrow i}(s_i) \triangleq \sum_{s_j} f_{ij}(s_i, s_j) \mu_{j \rightarrow f_{ij}}(s_j), \quad (3.28)$$

where  $\mu_{j \rightarrow f_{ij}}$  (which is a function of  $s_j$ ) is the message that arrives at  $f_{ij}$  from

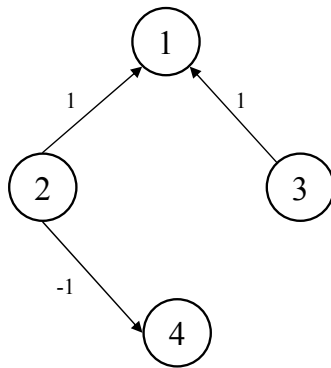


Figure 3.8: A Tree Trust Graph

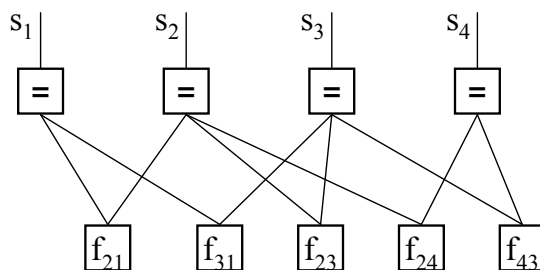


Figure 3.9: The Tree Factor Graph of Figure 3.8

variable  $s_j$ , which is defined as

$$\mu_{j \rightarrow f_{ij}}(s_j) \triangleq \prod_{k \in \hat{\mathcal{N}}_j \setminus i} \mu_{f_{jk} \rightarrow j}(s_j), \quad (3.29)$$

where  $\hat{\mathcal{N}}_j = \{k | (j, k) \in E \text{ or } (k, j) \in E\}$ .

If the factor graph has no cycles, then it is efficient to begin the message computation from the leaves and to successively compute message as their required “input” messages become available. In this way, each message is computed exactly once and we are able to get the *a posteriori* marginal probability of all  $s_i$ ’s simultaneously. Figure 3.8 is an example of trust graph whose corresponding factor graph is a tree, as shown in Figure 3.9. Suppose the trust values are binary, i.e.  $s_i \in \{-1, 1\}$ ,

the *a priori* probability of a node being GOOD is known to be 0.75, and the conditional probability of  $d_{ij}$  is defined as the binary example in Sec. 3.5.1, which is shown in Table 3.1. In the table, we assume the right hand side in Eqn. (3.2) is

Table 3.1: The Conditional Probability  $\Pr(d_{ij}|s_i, s_j)$

		$(s_i, s_j)$			
		(1,1)	(1,-1)	(-1, 1)	(-1,-1)
$d_{ij}$	1	0.8	0.2	0.5	0.5
	-1	0.2	0.8	0.5	0.5

equal to 0.8.

The message passing procedure of this tree example is shown in Figure 3.10. The message  $\mu$  are represented as  $\begin{pmatrix} \mu(1) \\ \mu(-1) \end{pmatrix}$ , scaled such that  $\mu(1) + \mu(-1) = 1$ . The final result in Step 6 is the *a posteriori* probability  $\Pr(s_i|d_{ij}, (i, j) \in E)$  for  $i = 1, \dots, 4$ .

However, trust graphs are usually cyclic, thus the factor graph generally has cycles. We can use iterative algorithms in this case. All messages are repeatedly updated, according to a certain schedule. The computation stops when the available time is over or when some other stopping condition is satisfied (e.g., the trust values change below a threshold). The results with cycles will usually be only an approximation. Nevertheless, it has been shown that the message passing algorithm could achieve suboptimum in sparse graphs without short cycles (such as sparse graphs with girth greater or equal to 6).

The message passing approach we have discussed in this subsection is a good approximation. In the followings, we introduce a stochastic voting scheme which provides the exact optimal solution [7, 32].

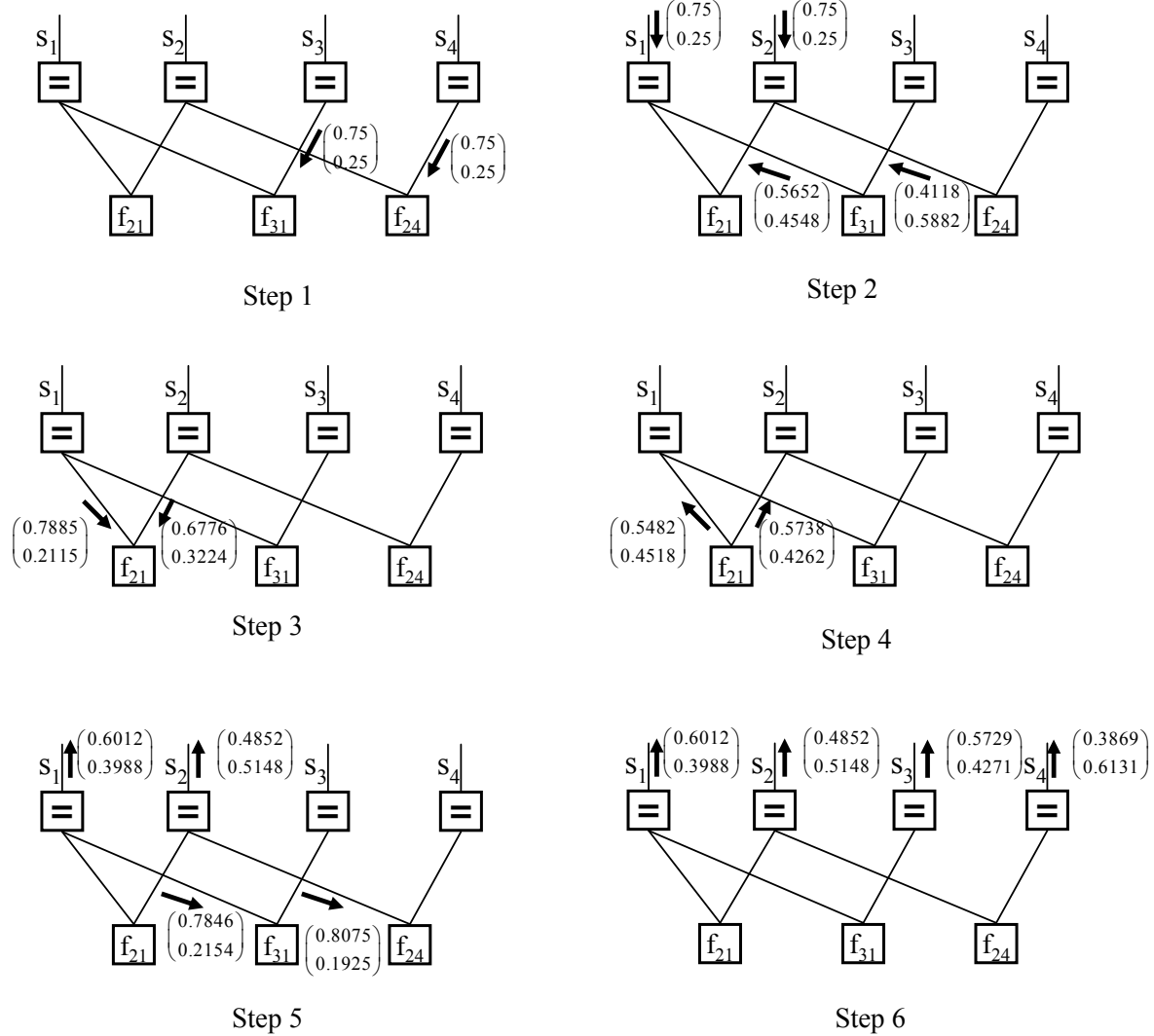


Figure 3.10: Message Passing for the Tree Factor Graph

### 3.5.4 Markov Random Field

The distribution of  $S$  in fact presents a Markovian type property, as stated in the following lemma.

**Lemma 5** *The distribution of the random vector  $S$  satisfies that the probability of the estimated trust value for a certain node  $i$ ,  $s_i$ , given the estimated trust values of all the other nodes in the network, is the same as the probability of  $s_i$ , given only the estimated trust values of the neighbors of  $i$ .*

$$\Pr[s_i|S/\{s_i\}] = \Pr[s_i|s_j, \forall j \in \hat{\mathcal{N}}_i]. \quad (3.30)$$

**Proof** According to the definition of conditional probability

$$\Pr[s_i|S/\{s_i\}] = \frac{\Pr[S]}{\Pr[S/\{s_i\}]}.$$

Substitute from Eqn. (3.25) to get

$$\Pr[s_i|S/\{s_i\}] = \frac{\Pr[d_{kl}, (k, l) \in E|s_k, k \in V] \prod_{k \in V} \Pr[s_k]}{\sum_{s_i} \Pr[d_{kl}, (k, l) \in E|s_k, k \in V] \prod_{k \in V} \Pr[s_k]}. \quad (3.31)$$

According to the definition of conditional probability, we have that

$$\Pr[d_{kl}, (k, l) \in E|s_k, k \in V] = \quad (3.32)$$

$$\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i|s_k, k \in V] \times \Pr[d_{kl}, (k, l) \in E, k \neq l \neq i|s_k, k \in V, d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i].$$



Because of the independence assumption in Eqn. (3.23), we have that

$$\Pr[d_{kl}, (k, l) \in E | s_k, k \in V] = \quad (3.33)$$

$$\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | s_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \times \Pr[d_{kl}, (k, l) \in E, k \neq l \neq i | s_k, k \in V, k \neq i].$$

Therefore, Eqn. (3.31) can be written as

$$\Pr[s_i | S / \{s_i\}] = \frac{\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | s_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[s_i]}{Z_i}, \quad (3.34)$$

where  $Z_i = \sum_{s_i} \Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | s_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[s_i]$ . The last equation shows that the conditional probability only depends on trust values of nodes that are neighbors of node  $i$ . ■

As opposed to the Markov chain, which has the Markov property with respect to time, Eqn. (3.34) shows Markovian property in the space dimension. A distribution with such property is called a *Markov random field* (MRF).

The well-known Hammersley-Clifford theorem [41] proves the equivalence between a MRF on a graph and the Gibbs distribution. We can write the distribution of  $S$  in Gibbs distribution form as following

$$\Pr[S] = \exp\left\{ \sum_{(i,j) \in E} \ln \Pr[d_{ij}, d_{ji} | s_i, s_j] + \ln \Pr[s_i] \right\} / Z. \quad (3.35)$$

### 3.5.5 Stochastic Voting

We have proved that the PMF of  $S$  is a Markov random field. Thus we define the stochastic rule as the following:

$$\Pr \left[ s_i(k+1) | s_j(k), j \in \hat{\mathcal{N}}_i \right] = \frac{\prod_{j \in \hat{\mathcal{N}}_i} \Pr[d_{ij}, d_{ji} | s_i(k+1), s_j(k)] \Pr[s_i(k+1)]}{Z_i(k)} \quad (3.36)$$

where  $Z_i(k)$  is the normalization factor

$$Z_i(k) = \sum_{s_i(k+1)} \prod_{j \in \hat{\mathcal{N}}_i} \Pr[d_{ij}, d_{ji} | s_i(k+1), s_j(k)] \Pr[s_i(k+1)]. \quad (3.37)$$

#### 3.5.5.1 Update Sequence

Our evaluation rule is essentially an updating rule. Another component of an updating rule is the update sequence. We list three possible types of update sequences

- *Synchronous updates*: Trust values of all nodes are updated at the same time.
- *Ordered asynchronous updates*: Trust values are updated one at a time in a predefined order.
- *Random asynchronous updates*: Trust values are updated one at a time. The updated node is randomly chosen following a distribution.

In the autonomic environment, it is very difficult to achieve synchronicity, which involves complicated algorithms and substantial control messages. Thus the

system should only use asynchronous updates. The ordered updates require extra control as well, so we only study random asynchronous updates, but the analyses in the following are also valid for the ordered asynchronous updates with only a few modifications. In random asynchronous updates, the probability that node  $i$  is chosen as the target is defined as  $q_i$ , and  $\sum_{i \in V} q_i = 1$ .

### 3.5.5.2 Markov Chain Interpretation

Our trust evaluation rule in the form of Eqn. (3.36) has the obvious Markov property: the value of  $s_i$  at time  $k + 1$  only depends on  $s_j, j \in \hat{\mathcal{N}}_i$  at time  $k$  and is independent of all the rest history.

The state of the Markov chain at time  $k$  is a configuration of  $S$ , and at time  $k + 1$ ,  $s_i$  changes while all other nodes keep unchanged. Combining Eqn. (3.36) and the random asynchronous updates, the stochastic voting rule can be considered as a Markov chain, where the states are composed of all the possible configurations of vector  $S$ . Suppose at time  $k$ , we are at state  $S = [s_1, \dots, s_i, \dots, s_N]$ . Denote the transition probability from state  $S$  to state  $\bar{S}^i = [s_1, \dots, \bar{s}_i, \dots, s_N]$  as  $p_{S, \bar{S}^i}$ , where  $\bar{s}_i = -s_i$ , then we have

$$p_{S, \bar{S}^i} = q_i \Pr[\bar{s}_i | S]. \quad (3.38)$$

Then the probability of staying still at  $S$  at time  $k + 1$  is

$$p_{S, S} = 1 - \sum_{i \in V} p_{S, \bar{S}^i}. \quad (3.39)$$

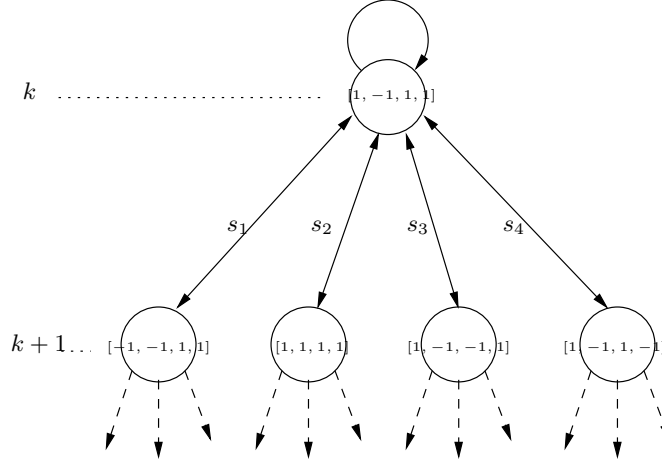


Figure 3.11: Part of the Markov Chain. Suppose  $S(k) = [1, -1, 1, 1]$ , then  $S(k + 1)$  either flips one of the element in  $S(k)$  or stays at the state of  $S(k)$ .

Figure 3.11 is an example of the Markov chain for a network with four nodes, such as the one in Figure 3.2. We only depict one state transition due to space constraints. Suppose at time  $k$ ,  $S(k) = [1, -1, 1, 1]$ , i.e. the state at the top of Figure 3.11. Each outgoing edge represents a state transition. The one with  $s_i$  on the edge means transition from  $S$  to  $\bar{S}^i$ . For instance, the leftmost transition is to flip the value of  $s_1$ .

### 3.5.5.3 Convergence

Having developed the probabilistic interpretation of our local voting rule in terms of a Markov chain, in this subsection, we study convergence of the Markov chain. We will show that the Markov chain converges and give the explicit formula for the stationary distribution. First, we give a lemma used for the proof of convergence.

**Lemma 6** *The Markov chain with transition probability defined as in Eqns. (3.38)*

and (3.39) is irreducible and aperiodic, given  $q_i > 0, \forall i \in V$ .

This lemma is easy to verify by proving that all states have a self-loop with nonzero probability and all are connected with each other under the condition for  $q_i$ .

Furthermore, the Markov chain is finite with dimension  $2^N$ , so it is a regular Markov chain. It is known that there is a unique stationary distribution  $\pi$  for a regular Markov chain. Therefore, our voting rule does converge under the trivial condition  $q_i > 0, \forall i \in V$ . In order to derive the stationary distribution, we introduce the notion of a *reversible* Markov chain.

**Definition 1** *A Markov chain is reversible if*

$$\pi_S p_{S,R} = \pi_R p_{R,S} \text{ for all } R, S, \quad (3.40)$$

where  $R, S$  are the states and  $\pi_S$  is the probability of being in state  $S$  at the steady state.

In other words, for a reversible Markov chain at the steady state, the process looks the same forward as well as backward. Hence it is said to be *reversible*<sup>5</sup>. Furthermore, the following theorem which is the reverse of Definition 1 provides a way to compute the stationary distribution [4].

**Lemma 7** *Suppose that  $\pi$  is a probability distribution satisfying Eqn. (3.40), then  $\pi$  is the unique stationary distribution and the chain is reversible.*

---

<sup>5</sup>Notice that not all regular Markov chains are reversible.

**Proof** This is true because Eqn. (3.40), sometimes called the *detailed balance equation*, implies

$$\sum_S \pi_S p_{S,R} = \pi_R \sum_S p_{R,S} = \pi_R \text{ for all } R \quad (3.41)$$

and therefore  $\pi$  satisfies the *balance equation* of the stationary distribution. ■

Define

$$M(S) = \prod_{(i,j) \in E} \Pr[d_{ij} | s_i, s_j] \prod_{i \in V} \Pr[s_i] \quad (3.42)$$

and a distribution  $\pi$  on the states of the Markov chain as following

$$\pi_S = \frac{M(S)}{Z}. \quad (3.43)$$

Then we have the following theorem.

**Theorem 8** *For the stochastic voting rule defined by Eqn. (3.36) and using random asynchronous updates, if  $q_i > 0, \forall i \in V$ , we have that*

1. *the stochastic voting rule converges to the steady state with a unique stationary distribution;*
2. *the distribution  $\pi_S = \frac{M(S)}{Z}$  is the unique stationary distribution.*

**Proof** The convergence has been verified based on Lemma 6. To prove  $\pi$  is the stationary distribution, we just need to check if  $\pi$  satisfies Eqn. (3.40) according to Lemma 7.

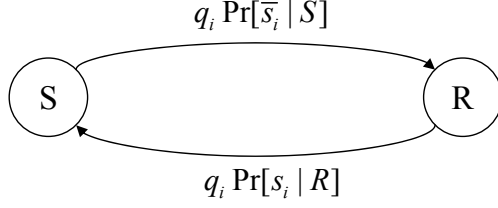


Figure 3.12: Transitions between  $S$  and  $R$ .  $p_{S,R} = q_i \Pr[\bar{s}_i | S]$  and  $p_{R,S} = q_i \Pr[s_i | R]$ .

Consider now two adjacent states  $S$  and  $R$  in the Markov chain.

$$S = [s_1, \dots, s_i, \dots, s_N] \text{ and } R = [s_1, \dots, \bar{s}_i, \dots, s_N].$$

We know that

$$p_{R,S} = q_i \Pr[s_i | R] = q_i \frac{\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | s_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[s_i]}{Z_i}, \quad (3.44)$$

$$p_{S,R} = q_i \Pr[\bar{s}_i | S] = q_i \frac{\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | \bar{s}_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[\bar{s}_i]}{Z_i}. \quad (3.45)$$

Notice that since the transitions between  $R$  and  $S$  are just flipping  $s_i$ , the normalization factor  $Z_i$  is the same for  $p_{R,S}$  and  $p_{S,R}$ . Fig. 3.12 shows the transitions between  $R$  and  $S$ . We need only verify Eqn. (3.40), i.e.

$$\frac{M(S)}{Z} p_{S,R} = \frac{M(R)}{Z} p_{R,S} \quad (3.46)$$

or

$$\frac{p_{S,R}}{p_{R,S}} = \frac{M(R)}{M(S)} \quad (3.47)$$

We know that

$$\begin{aligned} \frac{p_{S,R}}{p_{R,S}} &= \frac{\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | s_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[s_i]}{\Pr[d_{ij}, d_{ji}, j \in \hat{\mathcal{N}}_i | \bar{s}_i, \{s_j, j \in \hat{\mathcal{N}}_i\}] \Pr[\bar{s}_i]} \\ &= \frac{M(R)}{M(S)}. \end{aligned}$$

Thus, Eqn. (3.40) is satisfied and  $\pi$  is the stationary distribution of the stochastic voting rule. ■

Let vector  $SS$  be equal to the trust vector  $S$  at the steady state. The objective of trust evaluation is to find  $SS$  that is as close as to the real trust vector  $T$ .

### 3.5.6 Binary Example

In this section, we consider a special model: the binary-error model. Assume that  $t_i, s_i \in \{1, -1\}$ , i.e., a node in the network is either GOOD or BAD. Assume the *a priori* probability of nodes being good is known as  $p_G$ , i.e.,  $\Pr[s_i = 1] = p_G$  and the probability that a GOOD node has a wrong direct trust on its neighbors is  $p_e$ . For BAD nodes, we assume that they always provide opposite direct trust with the probability of errors being  $p_e$  as well. In other words, suppose node  $i$  is BAD and  $j$  is GOOD, then  $d_{ij} = -1$  with probability  $1 - p_e$ . Therefore, we can get a generalized direct trust model:

$$\Pr[d_{ij} \neq s_i s_j] = p_e \tag{3.48}$$

for both GOOD and BAD nodes.



If  $i$  has correct direct trust on  $j$ , we have that  $d_{ij}s_i s_j = 1$ , and  $d_{ij}s_i s_j = -1$  if  $i$  has wrong direct trust on  $j$ . If we define parameter  $a, b, \zeta$  and  $\eta$ , such that

$$\begin{aligned} a + b \times 1 &= \ln(1 - p_e) \\ a + b \times (-1) &= \ln p_e. \end{aligned} \tag{3.49}$$

and

$$\begin{aligned} \zeta + \eta \times 1 &= \ln(p_G) \\ \zeta + \eta \times (-1) &= \ln(1 - p_G). \end{aligned} \tag{3.50}$$

The stochastic voting rule Eqn. (3.36) can be written as following

$$\Pr [s_i(k+1) | s_j(k), j \in \mathcal{N}_i] = \frac{\exp\{\sum_{j \in \hat{\mathcal{N}}_i} (a + b(d_{ij} + d_{ji})s_i(k+1)s_j(k)) + \zeta + \eta s_i\}}{Z_i(k)}, \tag{3.51}$$

where  $Z_i(k)$  is the normalization factor. By solving the linear equations Eqn. (3.49) and (3.50), we have that

$$a = (\ln(1 - p_e) + \ln p_e)/2, b = (\ln(1 - p_e) - \ln p_e)/2,$$

and

$$\zeta = (\ln(1 - p_G) + \ln p_G)/2, \eta = (\ln(1 - p_G) - \ln p_G)/2,$$

Assume for any node  $i$ ,  $|\hat{\mathcal{N}}_i|$  is identical. For instance,  $G$  is an undirected 2-D lattice,

then  $|\hat{\mathcal{N}}_i| = 4, \forall i \in V$ . The stochastic voting rule Eqn. (3.51) can be written as

$$\Pr [s_i(k+1)|s_j(k), j \in \mathcal{N}_i] = \frac{\exp\{\sum_{j \in \hat{\mathcal{N}}_i} b(d_{ij} + d_{ji})s_i(k+1)s_j(k) + \eta s_i(k+1)\}}{Z_i(k)}, \quad (3.52)$$

We investigate properties of the estimated trust values when the voting rule reaches the steady state. In the binary case, the performance criterion can be defined in a more straight way, as the percentage of correct estimation  $P_{correct}$

$$\begin{aligned} P_{correct} &= \{ \text{Expected \# of } SS_i = T_i \} / N \\ &= \mathbf{E} \left[ 1 - \frac{\|SS - T\|_1}{2N} \right]. \end{aligned}$$

where  $\|SS - T\|_1 = \sum_{i \in V} |SS_i - T_i|$ . The last equation is true because  $SS, T \in \{1, -1\}$ . Before discussing properties of the steady state, we introduce an important model that models local interactions of magnets in physics – the Ising model.

### 3.5.6.1 Ising Model and Spin Glasses

For those familiar with statistical physics, it is very easy to link the stochastic voting rule Eqn. (3.52) with the Ising model [70] in statistical physics. The Ising model describes interaction of magnetic moments or “spins” of particles, where some particles seek to align with one another (ferromagnetism), while others try to anti-align (antiferromagnetism). In the Ising model,  $s_i$  is the orientation of the spin at particle  $i$ .  $s_i = 1$  or  $-1$  indicates the spin at  $i$  is “up” or “down” respectively. A

Hamiltonian, or energy, for a configuration  $S$  is given by

$$H(S) = - \sum_{(i,j)} J_{ij} s_i s_j - mH \sum_i s_i. \quad (3.53)$$

The first term represents the interaction between spins. The second term represents the effect of the external (applied) magnetic field. Then the probability of configuration  $S$  is given by

$$\Pr[S] = \frac{e^{-\frac{1}{kT}H(S)}}{Z}, \quad (3.54)$$

where  $T$  is the temperature and  $k$  is the Boltzmann constant. In the Ising model, the local interaction “strengths”  $J_{ij}$ ’s are all equal to a constant  $J$ , which is either 1 or  $-1$ . The Ising model has been extensively studied ever since Ernst Ising published his results on this model in 1920s. In recent years, an extension of the Ising model called the Edwards-Anderson model of spin glasses is used to study local interactions with independently random  $J_{ij}$  [55], which corresponds to  $d_{ij} + d_{ji}$  in our voting rule. The rich literature in statistical physics will no doubt help us to understand our voting model at the steady state.

### 3.5.6.2 Virtuous Network

Now go back to our discussion on trust at the steady state. We start with the simplest case: a virtuous network, where all nodes are good and they always have full confidence on their neighbors, so  $t_i = 1, \forall i \in V$  and  $d_{ij} = 1, \forall (i, j) \in E$ . Then

the stationary distribution  $\pi$  is exactly the same as the one in the Ising model with

$$b = \frac{1}{2kT} \text{ and } \eta = \frac{mH}{kT}. \quad (3.55)$$

Since all nodes are good with  $t_i = 1$  and  $SS_i$  is either 1 or  $-1$ , the percentage of correct estimation can also be written as

$$P_{correct} = \frac{\mathbf{E}[\langle SS \rangle] + N}{2N},$$

where  $\langle SS \rangle = \sum_{i \in V} SS_i$ . In the terminology of physics,  $\langle SS \rangle$  is called the total magnetization. It is known that when the external field  $H > 0$ ,  $\mathbf{E}[\langle SS \rangle]$  is positive and when  $H < 0$ , it is negative.

We use simulations to study the value of  $P_{correct}$  with respect to parameters  $\eta$  and  $b$ . In this section, the network topology for all the simulations is a two-dimensional lattice with periodic boundary. The number of nodes is 100 and each takes four nearest nodes as their neighbors. We chose the lattice because most theoretical results for the Ising model are for the 2-D lattice. In Sec. 3.5.6.4, we will discuss the effect of network topology.

Fig. 3.13 and Fig. 3.14 represent the percentage of correct estimation as a function of  $b$  for  $\eta$  being negative, positive or zero. While  $\eta > 0$  ( $p_G < 0.5$ ), the percentage of correct estimation is less than half. Such a result is obviously undesired, since it is even worse than randomly choosing the trust values, which can achieve the mean 0.5. Therefore, it is infeasible to use a positive threshold. This

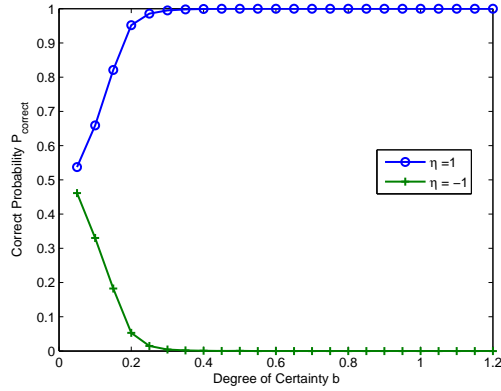


Figure 3.13:  $P_{correct}$  vs.  $b$  with  $\eta < 0$  and  $\eta > 0$ .

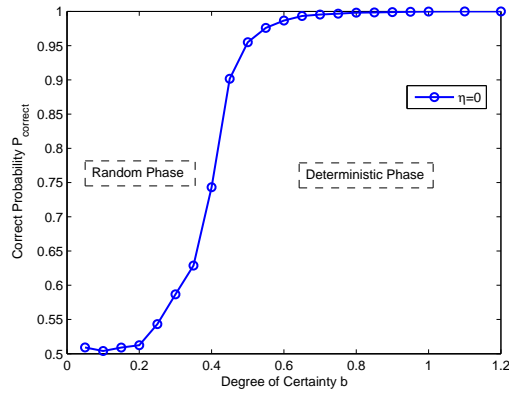


Figure 3.14:  $P_{correct}$  vs.  $b$  with  $\eta = 0$ .

conclusion is rather surprising given the natural thought of setting a threshold when using a voting rule. On the other hand, for  $\eta = 0$  ( $p_G = 0.5$ ) or  $\eta < 0$  ( $p_G > 0.5$ ), if the value  $b$  satisfies that  $b > 0.6$ ,  $P_{correct}$  is close to 1. Therefore, the threshold  $\eta$  must be non-positive.

Without our effort of deriving the stationary distribution  $\pi$ , the conclusion that  $\eta$  cannot be greater than zero is not an obvious fact if we only look at the local voting rule. This simple result elucidates the power and necessity of conducting analytical studies.

The other interesting property is the phase transition phenomenon observed in Fig. 3.14 when  $b$  is in  $[0.4, 0.5]$ . Phase transition has been extensively studied by physicists in the Ising model. For the Ising model on a two-dimensional lattice with  $H = 0$ , there exists a so-called *critical temperature*  $T_c$ . When the temperature  $T$  is above  $T_c$ , all the spins behave nearly independently (no long-range correlation), whereas when temperature is below  $T_c$ , all the spins tend to stay the same (i.e., cooperative performance). The above observation is expressed in the following in terms of the total magnetization  $\langle SS \rangle$ ,

$$\mathbf{E}[\langle SS \rangle] \begin{cases} = 0, & \text{if } T > T_c \\ \neq 0, & \text{if } T < T_c \end{cases}. \quad (3.56)$$

For a 2-D lattice, the value of the critical temperature can be accurately calculated as  $2kT_c = \frac{2}{1+\sqrt{2}} = 2.269$ , which is consistent with our simulation result where the critical value of  $b$ , denoted as  $b_c$ , is in  $[0.4, 0.5]$ , given the relation of  $b$  and  $T$  in Eqn. (3.55).

If we look closer into the interval when  $b < 0.4$ , the estimated trust value of each node is changing all the time and looks like random jumping. While when  $b$  is above the critical value, all values converge steadily to 1. So we call the first interval the *random phase*, while the second is the *deterministic phase*.

The discovery of phase transition in our voting rule is quite surprising given that the rule itself is very simple. More importantly, the fact that a small change in the parameter might result in a totally opposite performance of our voting rule

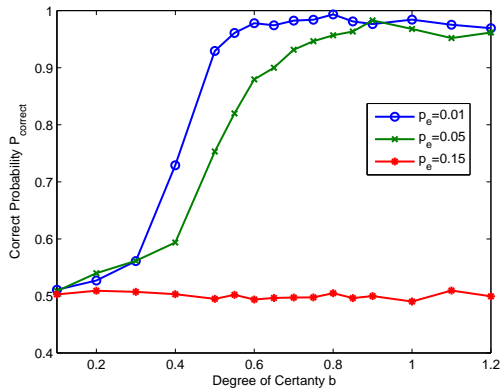


Figure 3.15:  $P_c$  vs.  $b$  with link errors  $p_e$ .  $\eta = 0$ .

proves the necessity of doing more analyses before applying any distributed algorithms. A rule with arbitrary or unverified parameters may ruin performance of the whole system.

As we have discussed, due to uncertainty and incompleteness of trust evidence,  $d_{ij}$  should be modeled as a random variable rather than being always 1. Thus in a virtuous network, the distribution of  $d_{ij}$  is

$$\Pr(d_{ij} = 1) = 1 - p_e; \quad \Pr(d_{ij} = -1) = p_e, \quad (3.57)$$

which is simplified as  $d \sim (1 - p_e, p_e)$ . This model corresponds to the  $\pm J$  model in spin glasses, where  $J \sim (p, 1 - p)$ .

We again investigate the phase transition. As shown in Fig. 3.15, the phase transition still happens when  $\eta = 0$ . However, as  $p_e$  increases, the wrong votes with value  $-1$  gradually destabilize the votes of value 1. Thus it is harder to keep  $s_i$ 's equal to 1, which means that  $b_c$  becomes larger and the system more probably stays

in the random phase given a high link error  $p_e$ . When  $p_e$  is large enough, as shown in the figure, where  $p_e = 0.15$ , the system always stays in the random phase.

In [55], the authors theoretically studied phase transitions between random and deterministic phases, and introduced the replica symmetry method to solve them analytically. Based on this method, very good approximations of values, such as  $\mathbf{E}[\langle SS \rangle]$  and  $\mathbf{E}[SS_i^2]$ , can be derived. The mathematical manipulation of the replica symmetry method is beyond the scope of this work, but it is definitely a very good direction for our future work. Given explicit expressions for these values, they will provide even better guide for network management and control design.

### 3.5.6.3 Adversary Model

The motivation of trust evaluation is to be able to identify good nodes and detect bad nodes. In this section, we study a network in the presence of adversaries, i.e., bad nodes. Three types of adversaries are considered:

- **Independent:** adversaries do not collude with each other and have normal power. We assume their probabilities of wrongly identifying whether a node is good or bad are  $p_e$  the same as good nodes. Once they identify a node as their friend, i.e. a bad node as well, they rate it with value 1 and vice versa.
- **Collusive:** adversaries know each other. They always vote for their friends with value 1 and vote for good nodes with value  $-1$ .
- **Random:** to confuse the evaluation rule, they randomly assign confidence values on others.



The voting rule used here is the one with threshold  $\eta = 0$ , while the following results can be extended to the case where  $\eta < 0$ . First consider independent adversaries. Define a new network  $G^*$  which has the same topology as the current network  $G$  and same probability of error  $p_e$ , but with all nodes being good. Suppose the percentage of correct estimation for  $G^*$  is  $P_{correct}^*$ , we have the following theorem:

**Theorem 9** *The percentage of correct estimation in the presence of independent adversaries  $P_{correct}$  is equal to the percentage of correct estimation  $P_{correct}^*$  for the network  $G^*$  with all good nodes.  $P_{correct}$  is independent of the number of adversaries.*

**Proof** We know that the distribution of  $d_{ij}^*$  is  $d_{ij}^* \sim (1 - p_e, p_e)$ , while the distribution of  $d_{ij}$  is

$$d_{ij} \sim \begin{cases} (1 - p_e, p_e) & \text{if } t_i \cdot t_j = 1 \\ (p_e, 1 - p_e) & \text{if } t_i \cdot t_j = -1 \end{cases}.$$

Therefore we have

$$d_{ij}^* \sim d_{ij} t_i t_j. \tag{3.58}$$

By the definition,

$$P_{correct} = \mathbf{E} [1 - \|S - T\|_1 / (2N)].$$

In probability theory, we know that

$$\mathbf{E}[X] = \mathbf{E} [\mathbf{E}[X|Y]].$$

Now let's set  $X = \|S - T\|_1$ ,  $Y = \{d_{ij}t_it_j\}_{(i,j) \in E}$  and  $X^* = \|S^* - T^*\|_1$ ,  $Y^* = \{d_{ij}^*\}_{(i,j) \in E}$ . So we are trying to prove

$$\mathbf{E}[\mathbf{E}[X|Y]] = \mathbf{E}[\mathbf{E}[X^*|Y^*]] \quad (3.59)$$

From Eqn. (3.58), we have  $Y \sim Y^*$ , so if we can prove

$$\mathbf{E}[X|Y] = \mathbf{E}[X^*|Y^*],$$

given  $Y = Y^*$ , then Eqn. (3.59) is true. First consider the left hand side,

$$\begin{aligned} \mathbf{E}[X|Y] &= \sum_S \|S - T\|_1 e^{b \sum d_{ij} s_i s_j} \\ &= \sum_S \sum_i |s_i - t_i| e^{b \sum (d_{ij} t_i t_j) (s_i t_i) (s_j t_j)} \end{aligned}$$

The last equation is because  $t_i^2 = 1, \forall i$ . Notice that

$$|s_i - t_i| = |t_i| |s_i - t_i| = |s_i t_i - t_i^*|$$

because  $|t_i| = 1$  and  $t_i^* = 1$ . If for all  $i$ , we take  $s_i^* = s_i t_i$ , and  $d_{ij} t_i t_j = d_{ij}^*$  (which is the assumption), we have

$$\mathbf{E}[X|Y] = \sum_{S^*} \sum_i |s_i^* - t_i^*| e^{b \sum d_{ij}^* s_i^* s_j^*} = \mathbf{E}[X^*|Y^*].$$

■

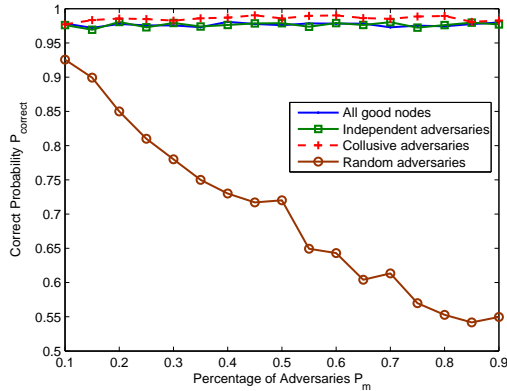


Figure 3.16:  $P_c$  vs. the percentage of adversaries  $P_m$ . The link error  $p_e = 0.05$  and the degree of certainty  $b = 1$ .

Therefore, the performance of our voting rule is independent of the number of adversaries in the case where all adversaries are independent. This is a very valuable property for a trust evaluation rule, and we can apply all the analyses in the last two subsections to the network with independent adversaries.

We use simulations to compare the three types of adversaries. In each simulation, the only difference is the behavior of bad nodes. Fig. 3.16 shows the simulation results. In order to verify Theorem 9, the curve for all good nodes is plotted as well. The curves of all good nodes and independent adversaries nearly overlap except for some small random perturbations, and the performance of independent adversaries does not change with respect to the percentage of adversaries in the network. Both verify the conclusion of Theorem 9.

The curve of collusive adversaries performs better than the ones for all good nodes and independent adversaries. This is because when we define the direct trust model for BAD nodes, we assume independent adversaries. The worst performance is the one of random adversaries, because our voting rule does not capture such

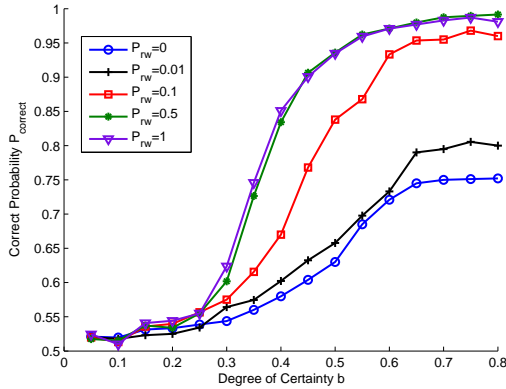


Figure 3.17: The effect of network topology.  $P_{rw}$  is the percentage of shortcuts in WS small world model. The network with  $P_{rw} \in [0.01, 0.1]$  has the small world property.  $p_e = 0.1$ .

malicious behavior.

### 3.5.6.4 Network Topology

So far the network topology being used is the two-dimensional lattice. While in reality, a trust graph is not just a lattice. In this section we further investigate our trust evaluation rule with respect to network topology.

We use the rewiring small-world model proposed by Watts and Strogatz in [69](WS model). In the WS model, we start from a ring lattice with the number of nodes  $N = 100$  and degree of each node  $k = 4$ . Each edge is rewired at random so as to create shortcuts with the percentage of shortcuts being the parameter  $P_{rw}$ . This construction models the graph transition between regular lattices ( $P_{rw} = 0$ ) and chaotic random graphs ( $P_{rw} = 1$ ).

Our simulation results are shown in Fig. 3.17, where different curves represent different shortcut percentages  $P_{rw}$ . We observe that the performance improves as

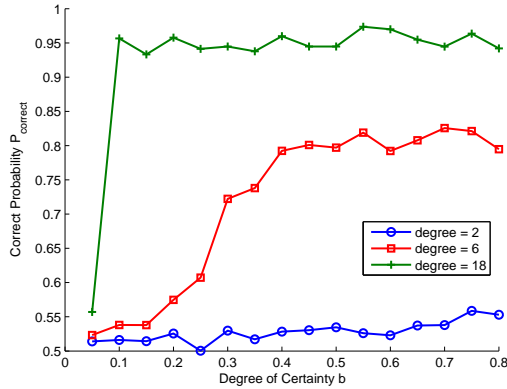


Figure 3.18: The effect of average degree with  $p_e = 0.1$ .

the model changes from regular lattices to random graphs. For instance, as  $b = 0.4$ ,  $P_{correct} = 0.55$  for regular lattices, while  $P_{correct} = 0.85$  for random graphs. This is because a more random network has shorter average distance between any two nodes in the network. Therefore, the number of hops which trust information takes to reach any node in the network is small. The accuracy of trust information degenerates over the path length, so short spreading paths have more accurate information and lead to good results. In particular, the most obvious improvement happens when  $P_{rw}$  increase from 0.01 to 0.1, which corresponds to the small-world topology. Therefore, a few shortcuts in the network will greatly improve performance of the trust evaluation rule.

We also investigated the effect of node degree, as shown in Fig. 3.18. The network is a 100-node ring lattice without shortcuts. The neighbors of a node are those within  $k$  hops distance, where degree  $k$  varies. The simulations show that the performance improves as the degree increases. One reason is still that the path length reduces between any two nodes by increasing the number of their neighbors.

From the above discussion, clearly the network topology has great influence on the system performance. As our future work, it is interesting to study our trust evaluation rule under real trust network topologies, and to investigate what kind of network topology has the best performance in terms of trust evaluation.

## Chapter 4

### Applications of Trust

What is it that trust management is supposed to achieve? It is uncontroversial, I believe, to claim that the ultimate goal is to enable the development of mutually beneficial cooperation. It is natural to suggest, furthermore, that the role of trust management is to maximise trust between the parties and thereby provide a basis for cooperation to develop. The aim of trust management must be to facilitate beneficial cooperation and to avoid interactions that are not beneficial. Secondly, trust is often an important basis for cooperation. This idea is found in much of the trust literature. For example, in social science, Misztal points out three basic things that trust does in the lives of people: It makes social life predictable, it creates a sense of community, and it makes it easier for people to work together [51]. Trust also plays important role in the cooperation of biological systems. The work by Nowak [56] states that cooperation among cells and genes is essential for life to evolve to a new level of organization. He modeled the cooperation as a Prisoner's Dilemma like game in which players acquire reputations. He found that if reputations spread quickly enough, they could increase the chances of cooperation. In this chapter, we investigate the evolution of cooperation in self-interested autonomic networks with the help of trust.

Besides cooperation, trust can also be used to the utility optimization prob-

lems. Many resource allocation and functionality allocation problems can be formulated as a constrained maximization of some utility function. The key mechanism of recent advances in Network Utility Maximization (NUM) applies optimization techniques so as to achieve decomposition by duality or the primal-dual method. The NUM approach is particularly useful for problems in multihop wireless networks, which helps design the most appropriate distributed algorithm and provides the analytical foundation for the cross-layer protocol design. In this dissertation, we apply nodes' trust values in the optimizations. The effect of these trust values on the resulting protocols is that in the routing and scheduling problems the trustworthiness of the node will be automatically considered and used. For example packets will not be routed as frequently to suspicious nodes, or suspicious nodes will not be scheduled by the MAC protocol.

#### 4.1 Trust and Cooperation: the Game Theoretic View

Autonomic networks rely on the collaboration of participating nodes for almost all their functions, for instance, to route data between source and destination pairs that are outside each other's communication range. However, because nodes are resource constrained, we deal with networks composed of users who are trying to maximize their own benefit from participation in the network. In the case of packet forwarding, the fundamental user decision is between forwarding or not forwarding data packets sent by other users. Given the constraints (mostly related to energy) that the user faces, there is a very real cost incurred when choosing to forward. So,



all users would like to send their own data packets, but not forward those of other users. Unfortunately, if all users were to do that, the network would collapse.

We assume that users want to be connected to as many other users as possible, directly (one-hop) or indirectly (multi-hop, through other users). In other words, by activating a communication link towards one of their neighbors, they gain by having access to the users with which that neighbor has activated *his* links, and so on, recursively. In the mean while, activation of links introduce cost. The more neighbors a user is connected to, the more packets he has to forward, which results in more communication cost. Our work is inspired by this fundamental tradeoff between gain and cost in the context of user collaboration in autonomic networks [33].

The conflict between the benefit from collaboration and the required cost for collaboration naturally leads to game-theoretic studies, where each node strategically decides the degree to which it volunteers its resources for the common good of the network. A user's payoff depends not only on whether he decides to collaborate or not, but also on whether his neighbors will decide to collaborate. For instance, Srinivasan et al. [64] address the problem of cooperation among energy constrained nodes and devised behavior strategies of nodes that constitute a Nash equilibrium. In [35], there is a link between two nodes if they agree to cooperate. These links are formed through one-to-one bargaining and negotiation.

Different with previous work in the literature, we study collaboration based on the notion of *coalitions* [8]. The concept of users being connected to each other, and acquiring access to all the other users that each of them had so far access to,

can be well captured by coalitional game theory (also known as cooperative game theory [57]). In coalitional game theory, the central concept is coalition formation, i.e., subsets of users that join their forces and decide to act together. Players form coalitions to obtain the optimum payoffs. The key assumption that distinguishes cooperative game theory from non-cooperative game theory is that players can negotiate collectively [53].

#### 4.1.1 Problem Formulation

Suppose there are  $N$  nodes<sup>1</sup> in the network. Define the set of nodes  $V = \{1, 2, \dots, N\}$ . The communication structure of the network is represented by an *undirected* graph  $G(V, E)$ , where a link between two nodes implies that they are able to directly communicate. Undirected links model the symmetric communications between neighboring nodes, because the willingness of both the two nodes is necessary to establish and maintain a link. Thus the undirected links are also called *pairwise links*. For instance, in wireless networks, reliable transmissions require that the two nodes interact in order to avoid collisions and interference.

A communication link is established only if two end nodes agree to collaborate with each other, i.e., they are *directly* connected with each other in  $G$ . Once the link is added, two end nodes join one coalition and they agree to forward all the traffics from each other. Note that *indirect* communication between two players require that there is a path connecting them. A path in  $G$  connecting  $i_1$  and  $i_m$  is a set of distinct nodes  $\{i_1, i_2, \dots, i_m\} \subset V$ , such that  $\{i_1i_2, i_2i_3, \dots, i_{m-1}i_m\} \subset E$ .

---

<sup>1</sup>In this chapter, the terms node, player and user are interchangeable.

### 4.1.2 Coalitional Games

A coalition is a subset of nodes that is connected in the subgraph induced by the active edges. In other words, two users are in the same coalition, if and only if there exists a path of active edges between them. The communication structure  $G$  gives rise to a partition of the node set into groups of nodes who can communicate with each other. A *coalition* of  $G$  is a subgraph  $G' \subset G$ , where  $\forall i \in V(G')$  and  $j \in V(G')$ ,  $i \neq j$ , there is a path in  $G'$  connecting  $i$  and  $j$ , and for any  $ij \in G$  implies  $ij \in G'$ .

If two users of separate coalitions join, then the two coalitions *merge* into one. We are interested in the total productivity of the coalition formed by selfish nodes, how this is allocated among the individual nodes and the stability of the coalition. These notions are captured by coalitional games.

Coalition formation has been widely studied in economics and sociology in the context of coalitional game [63, 20]. In our game, some nodes are not directly connected with each other; therefore the game we consider has to take the communication constraints into consideration. Myerson [52] was the first to introduce a new game associated with communication constraints, the *constrained coalitional game*, which incorporates both the possible gains from cooperation as modeled by the coalitional game and the restrictions on communication reflected by the communication network.

An important concept in coalitional games is the characteristic function  $v$  [22]. Since the game we study has communication constraints. The characteristic

function  $v$  is defined on a particular network rather than on a set of nodes in general coalition games, i.e.,  $v : G \rightarrow \mathbb{R}$  with convention:  $v(\emptyset) = 0$ . Notice that in our work the empty set  $\emptyset$  represents a graph where there is no link between any two nodes in the graph.  $v(G)$  is interpreted as the maximum payoff the network  $G$  can get given the network structure.

In our case, the value of  $v$  is the maximum aggregate of the net payoffs from all nodes in the graph

$$v(G) = \sum_{i \in G} v_i(G) \quad (4.1)$$

where  $v_i(G)$  is the payoff node  $i$  can get when the network is  $G$ .

An payoff allocation rule  $x : G \rightarrow \mathbb{R}^N$  describes how the value ( $v(G)$ ) associated with each network is distributed to the individual nodes.  $x_i(G)$  is the payoff of node  $i$  from network  $G$  and under the characteristic function  $v$ . For a graph  $G'$ , which is a subgraph of  $G$ , define

$$x(G') = \sum_{i \in G'} x_i(G).$$

The payoff allocation is *feasible* if  $x(G) \leq v(G)$  and *efficient* if  $x(G) = v(G)$ . In our case, the payoff may not be transferable, so the payoff allocation rule represents the payoff that each node receives from the network, i.e.,  $x_i(G) = v_i(G)$ . It is easy to show that such payoff allocation rule is feasible and efficient. We will discuss the stability of the constrained coalitional game in details in Sect. 4.1.3.

As we have discussed, users obtain benefits by joining a coalition. A user's

gain is explicitly defined on how he is connected to other users in the coalition. We assume that nodes always have information sent to other nodes in the network. Thus we assume node  $i$  potentially offers benefits  $x_{ij}$  to node  $j$  if  $(i, j) \in E$ . Then we could easily find the characteristic function of the coalitional game as:

$$v(G) = \sum_{(i,j) \in E} x_{ij} + x_{ji}. \quad (4.2)$$

Notice that  $\forall i, v(\{i\}) = 0$ . We denote the cooperative game defined from (4.2) as  $\Gamma = (G, v)$ .

In the next section, we will investigate stable solutions for enforcing cooperation among nodes, and demonstrate two efficient methods for achieving such cooperation: negotiation and trust mechanism.

### 4.1.3 Cooperation in games

#### 4.1.3.1 Cooperative games with negotiation

In this section, we investigate the impact of the games on the collaboration in a network. First we start with a simple fact.

**Lemma 10** *If  $\forall i, j, x_{ij} + x_{ji} \geq 0$ , then  $\Gamma = (G, v)$  is a superadditive game*

**Proof** Suppose  $S$  and  $T$  are two disjoint sets ( $S \cap T = \emptyset$ ), then

$$\begin{aligned} v(S \cup T) &= \sum_{i,j \in S \cup T} x_{ij} = \sum_{i,j \in S} x_{ij} + \sum_{i,j \in T} x_{ij} + \sum_{i \in S, j \in T} (x_{ij} + x_{ji}) \\ &= v(S) + v(T) + \sum_{i \in S, j \in T} (x_{ij} + x_{ji}) \geq v(S) + v(T). \end{aligned}$$

The last inequality holds by our assumption that  $x_{ij} + x_{ji} \geq 0$ . ■

The main concern in cooperative games is how the total payoff from a partial or complete cooperation of the players is divided among the players. A *payoff allocation* is a vector  $\mathbf{x} = (x_i)_{i \in N}$  in  $\mathbf{R}^N$ , where each component  $x_i$  is interpreted as the payoff allocated to player  $i$ . We say that an allocation  $\mathbf{x}$  is *feasible for a coalition*  $S$  iff  $\sum_{i \in S} x_i \leq v(S)$ .

When we think of a reasonable and stable payoff, the first thing that comes to mind is a payoff that would give each coalition at least as much as the coalition could enforce itself without the support of the rest of the players. In this case, players couldn't get better payoffs if they form separate coalitions different from the grand coalition  $N$ . The set of all these payoff allocations of the game  $\Gamma = (G, v)$  is called the *core* and is formally defined as the set of all  $n$ -vectors  $\mathbf{x}$  satisfying the linear inequality:

$$\mathbf{x}(S) \geq v(S) \quad \forall S \subset G, \tag{4.3}$$

$$\mathbf{x}(G) = v(G), \tag{4.4}$$

where  $\mathbf{x}(S) = \sum_{i \in S} x_i$  for all  $S \subset G$ . If  $\Gamma$  is a game, we will denote its core by  $C(\Gamma)$ . It is known that the core is possibly empty. Therefore, it is necessary to discuss existence of the core for the game  $\Gamma$ . We first give the definition of a family of common games: convex games [22]. The convexity of a game can be defined in terms of the *marginal contribution* of each player, which plays the role of first difference of the characteristic function  $v$ . Convexity of  $v$  can be defined in terms

of the monotonicity of its first differences. The first difference (or the marginal contribution of player  $i$ )  $d_i : 2^N \rightarrow \mathbf{R}$  of  $v$  with respect to player  $i$  is

$$d_i(S) = \begin{cases} v(S \cup \{i\}) - v(S) & \text{if } i \notin S \\ v(S) - v(S \setminus \{i\}) & \text{if } i \in S \end{cases}$$

A game is said to be *convex*, if for each  $i \in N$ ,  $d_i(S) \leq d_i(T)$  holds for any coalition  $S \subset T$ .

**Lemma 11**  $\Gamma(G, v)$  is a convex game.

**Proof** For  $\Gamma$ ,  $d_i(S) = \sum_{j \in S, j \neq i} (x_{ji} + x_{ij})$ . Take two sets  $S \subset T$ ,

$$d_i(T) - d_i(S) = \sum_{j \in T \cap S^c} (x_{ji} + x_{ij}) \geq 0$$

■

The core of a convex game is nonempty ([22]), thus  $C(\Gamma) \neq \emptyset$ . By Lemma 11, we have the following theorem,

**Theorem 12**  $\Gamma = (G, v)$  has a nonempty core.

Now let's find one of the payoff allocations that are in the core. For any pair of players  $(i, j)$ , suppose the payoff allocation of the game between  $i$  and  $j$  is  $(\hat{x}_{ij}, \hat{x}_{ji})$ .

Then we have the following

**Corollary 13** If the payoff allocation satisfies  $\hat{x}_{ij} \geq 0$  and  $\hat{x}_{ji} \geq 0$ , then the payoff allocation  $\hat{x}_i = \sum_{j \in \mathcal{G}_i} \hat{x}_{ij}$  is in the core  $C(\Gamma)$ .

**Proof** Take an arbitrary subset  $S \subset N$ ,

$$\hat{x}(S) = \sum_{i \in S} \hat{x}_i = \sum_{i, j \in S} \hat{x}_{ij} + \sum_{i \in S, j \notin S} \hat{x}_{ij} \geq \sum_{i, j \in S} \hat{x}_{ij} = v(S)$$

the inequality holds because  $\hat{x}_{ij} \geq 0, \forall i, j \in N$ . ■

Because we only consider transferable utility games,  $\hat{x}_{ij} + \hat{x}_{ji} = x_{ij} + x_{ji} \geq 0$ .

Therefore  $(\hat{x}_{ij}, \hat{x}_{ji})$  could be constructed in the following way:

$$\hat{x}_{ij} = \begin{cases} x_{ij} & \text{if } x_{ij} \geq 0, x_{ji} \geq 0 \\ x_{ij} + \lambda_{ij}x_{ji} & \text{if } x_{ij} < 0, x_{ji} > 0 \\ (1 - \lambda_{ij})x_{ji} & \text{if } x_{ij} > 0, x_{ji} < 0 \end{cases}$$

where  $0 \leq \lambda_{ij} = \lambda_{ji} \leq 1$ , and  $\hat{x}_{ij} \geq 0$  is achieved by carefully choosing  $\lambda_{ij}$ .

Obviously, the payoff allocation we provided in Corollary 13 is a set of points in the core, while there generally exist more points in the core that are not covered in the Corollary. However, this solution indicates a way to encourage cooperation in the whole network. The players which have positive gain can negotiate with their neighbors by sacrificing certain gain (offering their partial gain  $\lambda x_{ji}$ ). Though they cannot achieve their best possible payoff, they can set up a cooperative relation with their neighbors. This is definitely beneficial for the players who negotiate and sacrifice, since without cooperation they cannot get anything. This solution is also efficient and scalable, because players only need to negotiate with their direct neighbors.



Thus we established cooperative games among nodes in the network, and described an efficient way to achieve cooperation throughout the network. In the next section, we are going to discuss solutions by employing trust mechanisms, which do not require negotiation and the assumption on  $x_{ij} + x_{ji} \geq 0$  can also be relaxed.

#### 4.1.3.2 Trust mechanism

Trust is a useful incentive for encouraging nodes to collaborate. Nodes who refrain from cooperation get lower trust value and will be eventually penalized because other nodes tend to only cooperate with highly trusted ones. From Fig. 4.1 and the corresponding system equations, the trust values of each node will eventually influence its payoff. Let's assume, for node  $i$ , that the loss of not cooperating with node  $j$  is a nondecreasing function of  $x_{ji}$ , because the more  $j$  loses, the more effort  $j$  undertakes to reduce the trust value of  $i$ . Denote the loss for  $i$  being non-cooperative with  $j$  as  $l_{ij} = f(x_{ji})$  and  $f(0) = 0$ . For simplicity, assume the characteristic function is a linear combination of the original payoff and the loss, which is shown as

$$v'(S) = \sum_{i,j \in S} x_{ij} - \sum_{i \in S, j \notin S} f(x_{ji}) \quad (4.5)$$

The game with characteristic function  $v'$  is denoted as  $\Gamma'(G, v')$ . We then have

**Theorem 14** *If  $\forall i, j, x_{ij} + f(x_{ji}) \geq 0$ ,  $C(\Gamma') \neq \emptyset$  and  $x_i = \sum_{j \in V} x_{ij}$  is a point in  $C(\Gamma')$ .*

**Proof** First we prove  $\Gamma'$  is a convex game, given  $x_{ij} + f(x_{ji}) \geq 0$ .  $\forall i \in V$  in  $\Gamma'$ ,

$$d_i(S) = \sum_{j \in S, j \neq i} (x_{ij} + x_{ji}) - \sum_{k \notin S} f(x_{ki}) + \sum_{j \in S, j \neq i} f(x_{ij})$$

Let  $S \subset T$ ,

$$\begin{aligned} d_i(T) - d_i(S) &= \sum_{j \in T \cap S^c} (x_{ij} + x_{ji}) + \sum_{k \in T \cap S^c} f(x_{ki}) + \sum_{j \in T \cap S^c} f(x_{ij}) \\ &= \sum_{j \in T \cap S^c} ((x_{ij} + f(x_{ji})) + (x_{ji} + f(x_{ij}))) \geq 0. \end{aligned}$$

Therefore  $C(\Gamma')$  is nonempty. Next, we verify that  $x_i = \sum_{j \in \mathcal{N}_i} x_{ij}$  is in the core.

For any  $S \in G$ ,

$$\begin{aligned} \sum_{i \in S} x_i - v(S) &= \sum_{i \in S} \sum_{j \in V} x_{ij} - \left( \sum_{i, j \in S} x_{ij} - \sum_{i \in S, k \notin S} f(x_{ki}) \right) \\ &= \sum_{i \in S, j \notin S} (x_{ij} + f(x_{ji})) \geq 0. \end{aligned}$$

■

Apparently, the payoff  $x_i = \sum_{j \in \mathcal{N}_i} x_{ij}$  does not need any payoff negotiation.

Thus we showed that by introducing a trust mechanism, all nodes are induced to collaborate with their neighbors without any negotiation.

In this section, we introduced two approaches that encourage all nodes in the network to cooperate with each other: 1) negotiation among neighbors; 2) trust mechanism. We proved that both approaches lead to non-empty core for the cooperative game played in the network. However, we have only considered these two

approaches separately, and the results are based on static settings. The more interesting problems are how these two intertwine and how the dynamics between the two approaches converge to a cooperative network – these are discussed in the next section.

#### 4.1.4 Dynamics of Cooperation

We have analyzed the effect of a trust mechanism on the formation of cooperation. However, what we concentrated on in the previous section is the final impact of trust on the payoffs at the steady state. In this section, we are going to discuss two dynamic behaviors in the system: trust evaluation and game evolution.

##### 4.1.4.1 System model

In our model, each node has a self-defined playing strategy, which is denoted by  $\gamma_i$  for node  $i$ . Another characteristic of each node is its trust values, which are dependent on the opinions of other nodes. Trust values of a node can be different for different players. For instance,  $t_{ji}$  and  $t_{ki}$  are the trust values of  $i$  provided by distinct player  $j$  and  $k$ , and possibly  $t_{ji} \neq t_{ki}$ . Fig. 4.1 is a block graph demonstrating how nodes interact among their neighbors, where the payoff of node  $i$  after playing games is represented as  $x_i$ . The procedure is summarized as the following three rules:

- Strategy updating rule: as shown in Figure 4.1, nodes update strategies based on their own payoffs. They tend to choose rules that obtain the maximum

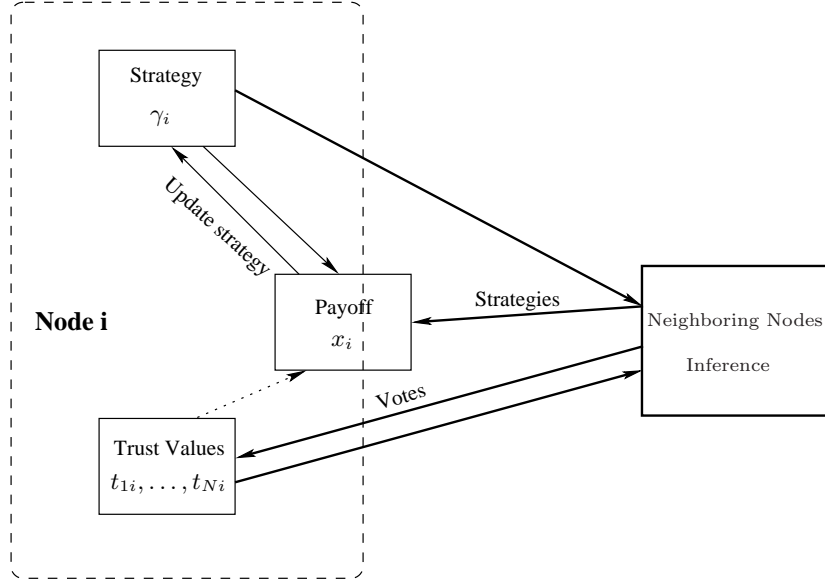


Figure 4.1: System operation block-graph for a typical node

payoffs.

- Payoff function: the payoffs are functions of the strategies of all participants. For a specific node, the payoff only depends on strategies of its neighbors and itself.
- Trust evaluation rule: trust values are computed based on votes, which are provided by neighbors and are related to the history (strategies and trust values) of the target node. Since trust values eventually have impact on the payoff of the node, there is a dotted line in Figure 4.1 from trust values to payoff to represent their implicit relation.

For simplicity, we assume the system is memoryless. All values are dependent only on parameter values at most one time step in the past. Therefore, the system

can be modeled as a discrete-time system:

$$\gamma_i(t+1) = f^i(x_i(t), \gamma_i(t), \gamma_j(t), t_{ij}(t)) \quad (4.6)$$

$$t_{ik}(t) = g^i(t_{ij}(t), v_{jk}(t)) \quad \forall k \in N \quad (4.7)$$

$$x_i(t) = h^i(\gamma_i(t), \gamma_j(t)) \quad (4.8)$$

$$v_{ij}(t) = p^i(\gamma_j(t), t_{ji}(t)) \quad (4.9)$$

where  $j$  stands for all neighbors of  $i$ , and  $v_{ij}$  is the value node  $i$  votes for  $j$ .

#### 4.1.4.2 Game evolution

As shown in Sect. 4.1.3.2, the trust mechanism drives selfish nodes to sacrifice part of their benefits and thus promotes cooperation. In this section, the procedure and dynamics of such cooperation evolution are studied.

In this section, we assume that nodes either cooperate or do not cooperate with neighbors.  $\gamma_{ij} = 1$  denotes that node  $i$  cooperates with its neighbor  $j$ , and  $\gamma_{ij} = 0$  denotes that it does not cooperate with  $j$ . We assume that the payoff when one of them does not cooperate is fixed as  $(0, 0)$ , and as  $(x_{ij}, x_{ji})$  when both cooperate. If  $x_{ij} < 0$ , we call the link  $(i, j)$  a *negative link* for node  $i$ , and when the opposite holds a *positive link*. Since all nodes are selfish, nodes tend to cooperate with neighbors that are on positive links, while they do not wish to cooperate with neighbors on negative links. In the mean while, the trust mechanism is employed, which aims to function as the incentive for cooperation. In this part, we assume that revocation

and nullification of revocation can propagate through out the network.

In our evolution algorithm, each node maintains a record of its past experience by using the variable  $\Delta_i(t)$ . First define  $x_{a,i}(t)$  as the payoff  $i$  gains at time  $t$  and  $x_{e,i}(t)$  as the expected payoff  $i$  can get at time  $t$  if  $i$  always chooses cooperation with all neighbors. Notice that the expected payoff can be different each time, since it depends on whether the neighbors cooperate or not at the specific time. Then compute the cumulative difference

$$\Delta_i(t) = \Delta_i(t - 1) + (x_{a,i}(t) - x_{e,i}(t)), \quad (4.10)$$

of the total payoff in the past minus the expected payoff if the node always cooperates. Each node chooses its strategy on the negative links by the following rule:

- if  $\Delta_i(t) < 0$ , node  $i$  chooses to cooperate, i.e.,  $\gamma_{ij} = 1, \forall j \in \mathcal{N}_i$ .
- if  $\Delta_i(t) \geq 0$ ,  $\gamma_{ij} = 0$ , if  $j \in \mathcal{N}_i$  and  $x_{ij} < 1$ .

Notice that at time 0,  $\Delta_i(0) = 0$ . That is to say initially all nodes choose not to cooperate on the negative links, since they are inherently selfish. There are two other conditions that force non-cooperation strategies:

- nodes do not cooperate with neighbors which have been revoked.
- nodes do not cooperate with non-cooperative neighbors.

To summarize, as long as one of those aforementioned conditions is satisfied, nodes choose not to cooperate.

Consider node  $i$ , and the initial settings are as follows:

- all the trust states are set to  $s_{ij} = 1, \forall j \in N$ ;
- the variable  $\Delta_i(0) = 0$ .

Node  $i$  chooses strategies and updates variables in each time step for  $t = 1, 2, \dots$ :

1. The strategy on the game with neighbor  $j$  is set according to the following rule:
  - for negative links ( $x_{ij} < 0$ ), choose non-cooperation strategy ( $\gamma_{ij} = 0$ ) if  $\Delta_i(t-1) \geq 0$ ;
  - if  $s_{ij} = 0, \gamma_{ij} = 0$ ;
  - for all neighbors,  $\gamma_{ij} = 0$  iff  $\gamma_{ji} = 0$  (cooperation is bilateral);
  - otherwise  $\gamma_{ij} = 1$ .
2. For all  $j \in \mathcal{N}_i$ , update the trust state  $s_{ij}$  if one of the following three conditions is satisfied, otherwise keep the previous state
  - if  $i$  accepts a revocation on node  $j$ ,  $s_{ij} = 0$ ;
  - if the revocation has been nullified for more than  $\tau$  consecutive steps, set  $s_{ij} = 1$ ;
  - if  $\gamma_{ji} = 0$ , set  $s_{ij} = 0$ ;
3. Compute the actual payoff  $x_{a,i}(t)$  and expected payoff  $x_{e,i}(t)$ , then get the cumulative difference  $\Delta_i(t)$  by Eqn.( 4.10).

Figure 4.2: Algorithm for game evolution modeling trust revocation

Since we allow and encourage nodes to rectify, i.e., to change their strategies from non-cooperation to cooperation, we define a temporal threshold  $\tau$  in the trust propagation rule. Instead of always keeping 0 once the state is switched to 0, we allow the nullification of revocation (switch back to state 1) under the condition that the revocation has been nullified for  $\tau$  consecutive time steps.  $\tau$  also represents the

penalty for being non-cooperative.  $\tau$  needs to be large so that the non-cooperative nodes would rather switch to cooperate than get penalized. However, large  $\tau$  will also reduce the payoff.

The detailed algorithm is shown in Fig. 4.2.

Suppose the total payoff of node  $i$ , if every node cooperates, is  $x_i = \sum_{j \in \mathcal{N}_i} x_{ij}$ .

We have the following

**Theorem 15**  $\forall i \in N$  and  $x_i > 0$ , there exists  $\tau_0$ , such that for a fixed  $\tau > \tau_0$ :

1. *The iterated game converges to Nash equilibrium.*
2.  $\Delta_i(t)/t \rightarrow 0$  as  $t \rightarrow \infty$ .
3.  *$i$  cooperates with all its neighbors for  $t$  large enough.*

**Proof** Nodes without negative links, will always cooperate, thus  $\Delta_i \equiv 0$ . Therefore, we only consider nodes with negative links. First we prove that for  $t$  large enough  $\Delta_i(t) < 0$ . Define for node  $i$ , the absolute sum of positive payoffs and negative payoffs as  $x_i^{(p)}$  and  $x_i^{(n)}$  respectively. Then

$$x_i = x_i^{(p)} - x_i^{(n)}$$

Therefore the first payoff for node  $i$  is  $x_{a,i}(1) = x_i^{(p)} > 0$  and  $\Delta_i = x_i^{(n)}$ . Define  $T_{max}$  as the maximum propagation delay in the network. Then at  $t = T_{max}$  all  $i$ 's neighbors revoke  $i$  because at time  $t = 1$ ,  $i$  didn't cooperate, and the payoff now is  $x_{a,i}(T_{max}) = 0$  and  $\Delta_i(T_{max}) = \Delta_i(T_{max} - 1) - x_i$ .  $i$  continues to get 0 payoff till all



neighboring nodes have used the penalty interval  $\tau$ . It's easy to show that as  $\tau$  is set large enough,  $i$  eventually gets negative  $\Delta_i$ .

If  $i$  follows the strategy rules in Fig. 4.2,  $i$  starts to cooperate with all neighbors. The difference of the actual payoff and expected payoff is 0 from then on. Therefore  $\Delta_i(t)/t \rightarrow 0$  as  $t \rightarrow \infty$ .

Assume node  $i$  deviates to non-cooperation, then it will get negative cumulative payoff difference as discussed above. So node  $i$  has no intention to deviate from cooperation. Therefore the game converges to its Nash equilibrium with all nodes cooperating. ■

We have also performed simulation experiments with our evolution algorithm. In the simulations, we didn't assume the condition that  $\forall i, x_i > 0$ , instead the percentage of negative links is the simulation parameter. We can report that without this condition, our iterated game with the trust scheme can still achieve very good performance. Figure 4.3 shows that cooperation is highly promoted under the trust mechanism. In Figure 4.4, the average payoffs between the algorithm with strategy update and without strategy update are compared, which explains the reason why nodes converge to cooperation.

## 4.2 Trust Aware Cross-Layer Optimization

The problem of resource allocation in wireless networks has been a growing area of interest over the past decade. Recent advances in the area of NUM driven cross-layer design[16] have led to current efforts on top-down development of next

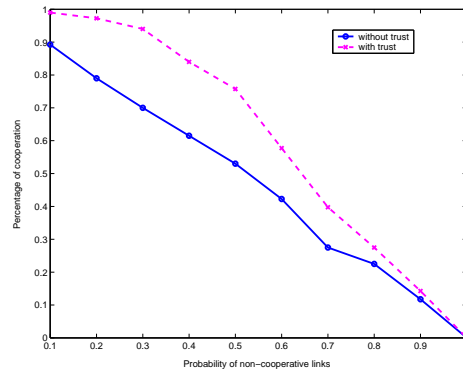


Figure 4.3: Percentage of cooperating pairs vs. negative links

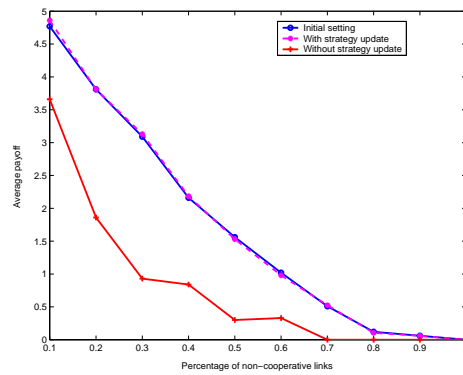


Figure 4.4: Average payoffs vs. negative links

generation wireless network architectures. By linking decomposition of the NUM problem to different layers of the network stack, we are able to design protocols, based on the optimal NUM derived algorithms, that provides much better performance gain over the current network protocols.

There are new challenges to apply such approaches to wireless networks comparing to wireline networks. Wireless spectrum is scarce, thus it is important to use the wireless channel efficiently. The wireless channel is a shared medium where the transmissions of users interfere with each other. The channel capacity is “elastic” (time-varying) and the contention over such shared network resources provides a fundamental constraint for resource allocation. Furthermore, for multihop wireless networks, it is critical to perform resource control in a distributed manner. However, because of the interference among wireless channels, a scheduling policy has to resolve the contention between various attempting transmissions, which require global information. All these challenges cause interdependencies across users and network layers. In spite of these difficulties, there have been significant recent developments in optimization-based approaches that results in loosely coupled cross layer solution[45].

In recent years, network security becomes more and more important. Security in wireless networks is even more critical, because communication channels are shared media. We will use “trust weights” in the optimizations and shows that trust helps to achieve better performance in the presence of adversaries. These trust weights will be developed by our community based monitoring approach and will be disseminated via efficient methods so that it is timely available to all nodes that

need it.

We consider a set of flows that share the resources of a fixed wireless network. Each flow is described by its resource-destination node pair, with no *a priori* established routes. The effect of these trust weights on the resulting protocols is that in the scheduling problems (whether they are at the MAC or the routing protocol) the trustworthiness of the node will be automatically be considered and used. For example packets will not be routed as frequently to suspicious nodes. Or suspicious nodes will not be scheduled by the MAC protocol.

We present decentralized algorithms which obtain a set of feasible solutions for the NUM problem, and compare the decentralized solutions with results of a centralized algorithm which is optimal and solves the NUM problem. Furthermore, we compare our results with schemes which do not take trust into consideration.

#### 4.2.1 System Model

We consider a multihop wireless network with  $N$  nodes. Let  $\mathcal{L}$  denote the set of links  $(i, j)$  such that the transmission from node  $i$  to node  $j$  is possible. The interference constraints among transmission links will be described later. The wireless network can be described as a graph  $(\mathcal{N}, \mathcal{L})$ , where  $\mathcal{N}$  is the set of  $N$  nodes.

##### 4.2.1.1 Trust

We study the utility optimization problem with consideration of trust. All previous work on the NUM problems assumes that nodes function correctly. For

instance, intermediate nodes always successfully forward all packets, and they follow the routing and scheduling protocols if they are given. However, nodes do not always function correctly in reality. They may be compromised by attackers, their communications may be blocked or interfered by attackers, or they may just simply malfunction. Wireless networks are especially vulnerable to attacks because of the inherent properties of the shared wireless media. Therefore, we believe it is important to take the trust into consideration for the NUM problems. We define the trust value of node  $i$  as  $v_i$ .

#### 4.2.1.2 Data flow and utility function

There are  $F$  flows that share the network resources and each flow  $f$  is associated with a source node  $s_f$  and a destination node  $d_f$ . The set of all  $F$  flows is denoted as  $\mathcal{F}$ . Let  $x_f$  be the rate with which data is sent from  $s_f$  to  $d_f$  over possibly multiple paths and multiple hops.

In our model, the flow can use different routes in the network. Suppose the set of routes for flow  $f$  is  $R_f$ . A route  $r \in R_f$  is a set of nodes on the route, i.e.,  $i \in r$  means that node  $i$  is on route  $r$ . We define the trustworthiness of route  $r$  as the multiplication of the trust values of nodes on the route, that is

$$v_r = \prod_{i \in r} v_i.$$

Suppose the percentage of data transmitted on route  $r$  is denoted as  $p_r$ . The aggre-

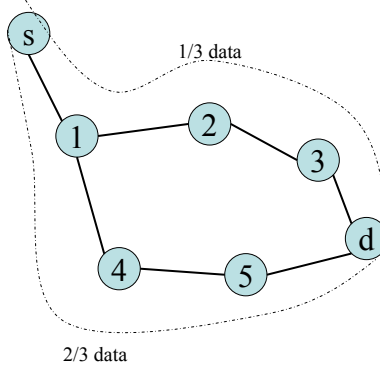


Figure 4.5: An example network with 2 routes for flow from  $s$  to  $d$ .

gate trust value flow  $f$  passes is defined as

$$g_f = \sum_{r \in R_f} p_r v_r. \quad (4.11)$$

For instance, Fig. 4.5 shows flow  $f$  from node  $s$  to  $d$ . The flow uses two routes. One third passes through nodes 1, 2 and 3, while the rest passes through nodes 1, 4 and 5. We have the aggregate trust value flow  $f$  passes as the following

$$g_f = v_1 \left( \frac{1}{3} v_2 v_3 + \frac{2}{3} v_4 v_5 \right). \quad (4.12)$$

$g_f$  represents of the quality of routes that flow  $f$  passes.

Users prefer data to be transmitted on paths with high trust values. Therefore, we define the utility function for flow  $f$  as a function of the data rate  $x_f$  and the trust values of paths it travels. Each flow is associated with a utility function defined as  $U_f(x_f g_f)$ , which reflects the “utility” to the flow  $f$  when it can successfully transmit at data rate  $x_f$ . We assume that  $U_f(\cdot)$  is strictly concave, nondecreasing

and continuously differentiable. For two flows of the same data rate, the one passes paths with better trust values has higher utility. Define  $\hat{x}_f = x_f g_f$  as the effective data rate for flow  $f$ .

### 4.2.1.3 Interference model and stability

In this subsection, we describe the interference model. The data rate of each link depends on the transmission power and interference of other transmissions. We denote  $u(G, P, BW, N_0)$  to be the rate function, which models the rate a particular link transmits, given the power assignment  $P = [P_{ij}, (i, j) \in \mathcal{L}]$ , the gain matrix  $G$ , the noise power  $N_0$ , and the link bandwidth  $BW$ . We assume  $G, BW$  and  $N_0$  are constants so the rate function depends only on  $P$ .

We let  $\mu = \{\mu_{ij}\}_{(i,j) \in \mathcal{L}}$  denote the rate vector at which data can be transferred over each link  $(i, j) \in \mathcal{L}$ . Then  $\mu = u(P)$ . We let  $\Gamma$  denote a bounded region of  $|\mathcal{L}|$  dimensions. It represents the set of  $\mu$  that can be achieved in a given time slot due to the interference constraints. Notice that  $\Gamma$  may not be convex. We let  $\hat{\Gamma} := Co\{\Gamma\}$  the convex hull of  $\Gamma$ .  $\hat{\Gamma}$  can be achieved by timesharing between different rate vectors in  $\Gamma$ .  $\hat{\Gamma}$  is convex, closed and bounded.

We define the *capacity region*  $\Lambda$  of the network as the largest set of rate vectors  $\mathbf{x}$  such that  $\mathbf{x} \in \Lambda$  is a necessary condition for network stability under *any* scheduling policy. We assume that nodes keep a queue for each flow. Let  $\mu_{ij}^f$  denote the amount of capacity on link  $(i, j)$  that is allocated for flow  $f$ . We have the following definition of the capacity region.

**Definition 2 (Capacity Region)** *The capacity region,  $\Lambda$ , of the network contains the set of flow rate  $\mathbf{x}$  for which there exists  $\mu$  that satisfies the following[54].*

1.  $\mu_{ij}^f \geq 0$  for all  $(i, j) \in \mathcal{L}$  and for all  $f \in \mathcal{F}$ .
2. for all  $f \in \mathcal{F}$  and for all  $i \in \mathcal{N}$ , we have

$$\sum_{j:(i,j) \in \mathcal{L}} \mu_{ij}^f - \sum_{j:(j,i) \in \mathcal{L}} \mu_{ji}^f - x_f \mathbf{1}(i = s_f) \geq 0. \quad (4.13)$$

3.  $\{\sum_f \mu_{ij}^f\} \in \hat{\Gamma}$ .

$\mathbf{1}(i = s_f)$  is an indicator function.  $\mathbf{1}(i = s_f) = 1$  if  $i$  is the source of flow  $f$ , and  $\mathbf{1}(i = s_f) = 0$  otherwise. The condition 2) is the flow constraint that must hold at each node, and the condition 3) captures the interference constraints.

#### 4.2.2 Utility Optimization and Dual Decomposition

Our goal is to design a scheduling mechanism such that the flow rate  $x_f$  solves the following optimization problem:

$$\text{maximize}_{\hat{\mathbf{x}}} \quad \sum_f U_f(\hat{x}_f) \quad (4.14)$$

$$\text{subject to} \quad \mathbf{x} \in \Lambda \quad (4.15)$$

$$\hat{x}_f = g_f \cdot x_f \text{ for all } f \in \mathcal{F} \quad (4.16)$$

We refer the above as the *primal problem*. Due to the strict concavity assumption of  $U_f(\cdot)$  and the convexity of the capacity region, there exists a unique optimizer of



the primal problem, which we refer as  $\hat{\mathbf{x}}^*$ .

In order for us to provide a distributed solution to the problem we use the technique of dual decomposition[12]. By decomposing the optimization problem, we provide decentralized algorithms which solve the underlying optimization problem. As we will see some of the optimal algorithms require centralized information and thus are not feasible to implement in a distributed way.

Notice that the variables  $g_f$  and  $\hat{x}_f$  are coupled by the second constraint Eqn. (4.16). We take the log of variables to decouple  $g_f$  and  $\hat{x}_f$  and log change of variables and constants:  $\hat{x}'_f = \log \hat{x}_f$ ,  $g'_f = \log g_f$ ,  $x'_f = \log x_f$ , and  $U'_f(\hat{x}'_f) = U_f(e^{\hat{x}'_f})$ . Now the primal problem is separable.

After the log change, we decompose the primal problem by defining Langrange multipliers  $\lambda_i^f$  and  $\nu_f$  that are associated with the stability constraints stated in Eqn. (4.15) and (4.16). We get the following dual function:

$$L(\lambda, \nu, \hat{\mathbf{x}}, \mathbf{x}, \mu, \mathbf{g}) = \sum_f \max_{x'_f} \left\{ \nu_f x'_f - \lambda_{s_f}^f e^{x'_f} \right\} + \sum_f \max_{\hat{x}'_f} \left\{ U'_f(\hat{x}'_f) - \nu_f \hat{x}'_f \right\} \quad (4.17)$$

$$+ \max_{\mathbf{g}'} \sum_f \nu_f g'_f \quad (4.18)$$

$$+ \max_{\mu \in \hat{\Gamma}} \sum_{(i,j) \in \mathcal{L}} \sum_{f \in \mathcal{F}} \mu_{ij}^f (\lambda_i^f - \lambda_j^f). \quad (4.19)$$

The dual objective function is

$$h(\lambda, \nu) = \sup_{\substack{\mathbf{x} \in \Lambda \\ x_f = g_f \cdot x_f}} L(\lambda, \nu, \hat{\mathbf{x}}, \mathbf{x}, \mu, \mathbf{g}) \quad (4.20)$$

For a given  $\lambda$  and  $\nu$ , there are now three, decoupled, maximization problems we

can solve separately, i.e., source rate control, routing and scheduling. By solving for these independently, we can produce  $\hat{\mathbf{x}}'^*(\lambda, \nu)$ ,  $\mathbf{x}'^*(\lambda, \nu)$ ,  $\mu^*(\lambda, \nu)$  and  $\mathbf{g}^*(\lambda, \nu)$ . Given these values, we can then solve the dual problem by minimizing  $h(\lambda, \nu)$  over  $\lambda, \nu$ . Because the capacity region  $\Lambda$  is a convex set, there is no duality gap between the primal and the dual. In the rest of this section, we discuss the dual decomposition techniques in details.

The data rate of flow  $f$  is determined by the maximization over  $x'$  in Eqn. (4.17). It is important to note that each source node adjusts its rate using only local information  $\lambda_{s_f}^f$  and  $\nu_f$ , thus the source rate control can be distributed across source nodes.

Eqn. (4.18) determines the route.

$$\mathbf{g}' = \operatorname{argmax}_{\mathbf{g}'} \sum_f \nu_f g'_f. \quad (4.21)$$

We have that the optimal routing for flow  $f$  is to always choose the route with the highest trust value. Since the trust values are fixed, routing can be decided off-line.

Eqn. (4.19) determines the schedule.

$$\mu = \operatorname{argmax}_{\mu \in \hat{\Gamma}} \sum_{(i,j) \in \mathcal{L}} \sum_{f \in \mathcal{F}} \mu_{ij}^f (\lambda_i^f - \lambda_j^f) \quad (4.22)$$

To maximize Eqn. (4.22), the term inside the second summation should take one

single flow  $f$  such that  $(\lambda_i^f - \lambda_j^f)$  is maximized. Therefore we have that

$$\mu_{ij}^f = \mu_{ij} \text{ if } f = \operatorname{argmax}_f (\lambda_i^f - \lambda_j^f), \quad (4.23)$$

and  $\mu_{ij}^f = 0$  otherwise. The schedule of link rates in Eqn. (4.23) is the same as the *back-pressure scheduler* introduced by Tassiulas[68]. Notice that the maximization in Eqn. (4.23) is performed over  $\hat{\Gamma}$ , which requires centralized knowledge. In the next section, we propose a distributed algorithm which optimizes the network utility.

### 4.2.3 Distributed Algorithm

Scheduling sub-problem discussed in the last section requires global knowledge on the rate vector, which becomes the bottleneck for solutions in wireless networks. In this section, we study the distributed implementation of the maximization problem. We assume that time is divided into slots. At each time slot, source nodes choose the flow data rate and the scheduling policy will select data to be forwarded on each link.

The source node of each flow uses its local multiplier and the utility function associated with that flow to update the flow rate in an iterative manner. One example of the rate controller is directly derived from Eqn. (4.17):

$$x'_f[t+1] = \operatorname{argmax}_{x'_f} \left\{ \nu_f[t] x'_f - \lambda_{s_f}^f[t] e^{x'_f} \right\}, \quad (4.24)$$

$$\hat{x}'_f[t+1] = \operatorname{argmax}_{\hat{x}'_f} \left\{ U'_f(\hat{x}'_f) - \nu_f[t] \hat{x}'_f \right\}. \quad (4.25)$$

The subgradient of  $h(\lambda, \nu)$  is given by

$$\frac{\partial h}{\partial \lambda_i^f} = \sum_{j:(i,j) \in \mathcal{L}} \mu_{ij}^f - \sum_{j:(i,j) \in \mathcal{L}} \mu_{ji}^f - e^{x'_f} \mathbf{1}(i = s_f), \quad (4.26)$$

$$\frac{\partial h}{\partial \nu_f} = g'_f + x'_f - \hat{x}'_f. \quad (4.27)$$

We can then use the following subgradient method to solve the dual problem.

$$\lambda_i^f[t+1] = \left\{ \lambda_i^f[t] - \beta[t] \left( \sum_{j:(i,j) \in \mathcal{L}} \mu_{ij}^f[t] - \sum_{j:(i,j) \in \mathcal{L}} \mu_{ji}^f[t] - e^{x'_f[t]} \mathbf{1}(i = s_f) \right) \right\}^+, \quad (4.28)$$

$$\nu_f[t+1] = \left\{ \nu_f[t] - \beta[t] (g'_f[t] + x'_f[t] - \hat{x}'_f[t]) \right\}^+, \quad (4.29)$$

where  $\beta[t]$ ,  $t = 1, 2, \dots$  is a sequence of positive step sizes.  $\mu_{ij}^d[t]$ ,  $g'_f[t]$ ,  $\hat{x}'_f[t]$  and  $x'_f[t]$  are defined as earlier with time slot  $t$ . According to Theorem 2.3 in [62], we know that there is no duality gap, the converging results of  $\mathbf{x}'$  solves the maximization problem and such  $\mathbf{x}'$  is the unique optimal solution.

Notice that the scheduling problem of Eqn. (4.22) requires global information. We consider the solution to the scheduling problem that is coupled to the power control problem, as in Lin and Shroff [44]. The link transmission powers determine an explicit form for the allowable rate region  $\hat{\Gamma}$ . The scheduling problem is posed as an optimization problem, whose solution gives us the transmission powers. At optimality, each node should transmit at full power or shut off, and also should transmit to at most one neighbor - the solution is thus distributed.

## BIBLIOGRAPHY

- [1] K. Aberer. P-Grid: A self-organized access structure for p2p information systems. In *Proceedings in 9th International Conference on Cooperative Information Systems*, 2001.
- [2] Daniel Aguayo, John Bicket, Sanjit Biswas, and Glenn Judd and Robert Morriss. Link-level measurements from an 802.11b mesh network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 121–132, Portland, Oregon, USA, 2004.
- [3] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.
- [4] David Aldou and James Allen Fill. Reversible Markov chains and random walks on graphs. monograph in preparation.
- [5] Autonomic communication. <http://www.autonomic-communication.org/>.
- [6] Dirk Balfanz, D. K. Smetters, Paul Stewart, and H. Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of Symposium on Network and Distributed Systems Security (NDSS'02)*, San Diego, California, February 2002.
- [7] John S. Baras and Tao Jiang. Cooperative games, phase transitions on graphs and distributed trust in manet. In *Proceedings of 2004 IEEE Conference on Decision and Control (CDC)*, pages 93–98, Bahamas, December 2004.
- [8] John S. Baras and Tao Jiang. *Cooperation, Trust and Games in Wireless Networks*, pages 183–202. Birkhause, June 2005.
- [9] Thomas Beth, Malte Borcherdig, and Birgit Klein. Valuation of trust in open networks. In *Proceedings of 3rd European Symposium on Research in Computer Security – ESORICS'94*, pages 3–18, 1994.
- [10] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, pages 185–210, 1999.
- [11] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of 1996 IEEE Symposium on Security an Privacy*, pages 164–173, May 1996.
- [12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge, Cambridge, MA, 1st edition, 2004.

- [13] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Texts in Applied Mathematics; 31. Springer-Verlag New York, Inc., 1999.
- [14] Sergey Brin, Lawrence Page, R. Motwami, and Terry Winograd. The PageRank citation ranking: bringing order to the web. Technical Report 1999-0120, Computer Science Department, Stanford University, 1999.
- [15] S. Capkun, L. Buttyán, and J. P. Hubaux. Small worlds in security systems: an analysis of the PGP certificate graph. In *Proceedings of The ACM New Security Paradigms Workshop 2002*, pages 28–35, Norfolk, Virginia Beach, USA, September 2002.
- [16] Mung Chiang, Steven H. Low, A. Robert Calderbank, and John C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, January 2007.
- [17] Philip Chou, Yunnan Wu, and Jain Kamal. Network coding for the internet. In *Proceedings of IEEE Communication Theory Workshop*, Capri, 2004.
- [18] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46, 2001.
- [19] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, Philadelphia, PA, USA, 2004.
- [20] Bhaskar Dutta and Matthew O. Jackson, editors. *Networks and Groups: Models of Strategic Formation*. Studies in Economic Design. Springer-Verlag Berlin Heidelberg, 2003.
- [21] Laurent Eschenauer. On trust establishment in mobile ad-hoc networks. Master’s thesis, University of Maryland, College Park, 2002.
- [22] Ferenc Forgo, Jenő Szep, and Ferenc Szidarovszky. *Introduction to the Theory of Games: Concepts, Methods, Applications*. Kluwer Academic Publishers, 1999.
- [23] G. David Forney, Jr. Codes on graphs: Normal realizations. *IEEE Transactions on Information Theory*, 47(2):520–548, February 2001.
- [24] Diego Gambetta. Can we trust trust? electronic edition, <http://www.sociology.ox.ac.uk/papers/trustbook.html>, 2000. In *Trust: Making and Breaking Cooperative Relations*, Gambetta, Diego (ed.), Department of Sociology, University of Oxford.
- [25] Christos Gkantsidis and Pablo Rodriguez. Network coding for large scale content distribution. In *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

- [26] V. D. Gligor, S.-W. Luan, and J.N. Pato. On inter-realm authentication in large distributed systems. In *Proceedings of the IEEE Conference on Security and Privacy*, pages 2–17, 1992.
- [27] Virgil D. Gligor. Security of emergent properties in ad-hoc networks. In *Proceeding of the Security Protocols Workshop*, Sidney Sussex College, Cambridge, UK, April 2004.
- [28] Gnutella. <http://www.gnutella.com>.
- [29] Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16, 2000.
- [30] IEEE Std 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.
- [31] Tao Jiang and John S. Baras. Autonomous trust establishment. In *Proceedings of 2nd International Network Optimization Conference (INOC)*, Lisbon, Portugal, February 2005.
- [32] Tao Jiang and John S. Baras. Trust evaluation in anarchy: A case study on autonomous networks. In *Proceedings of 2006 INFOCOM*, Barcelona, Spain, April 2006.
- [33] Tao Jiang and John S. Baras. Fundamental tradeoffs and constrained coalitional games in autonomic wireless networks. In *Proceedings of 5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, April 2007.
- [34] Tao Jiang and John S. Baras. Trust document distribution in manets. Submitted to *Milcom*, 2007.
- [35] R. Johari, S. Mannor, and J.N. Tsitsiklis. A contract-based model for directed network formation. *Games and Economic Behavior*, 2006.
- [36] C.M. Jonker and J. Treur. Formal analysis of model for the dynamics of trust based on experiences. In *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering*, pages 221–231, 1999.
- [37] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, Bled, Slovenia, 2002.
- [38] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.
- [39] Audun Jøsang, Claudia Keser, and Theo Dimitrakos. Can we manage trust? In *Proceedings of the Third International Conference on Trust Management (iTrust'05)*, pages 93–107, 2005.

- [40] M. Kinatered and K. Rothermel. Architecture and algorithms for a distributed reputation system. In P. Nixon and S. Terzis, editors, *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, pages 1–16, Crete, Greece, 2003. Springer-Verlag.
- [41] Ross Kindermann and J. Laurie Snell. *Markov Random Fields and Their Applications*, volume 1 of *Contemporary mathematics*. American Mathematical Society, 1980.
- [42] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001.
- [43] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. In *Proceedings of the 13th ACM Symposium on Operating Systems Principles*, pages 265–310, Oct 1991.
- [44] Xiaojun Lin and Ness B. Shroff. Joint rate control and scheduling in multihop wireless networks. In *Proceedings of 48th IEEE Conference on Decision and Control*, pages 1484–1489, Paradise Island, Bahamas, December 2004.
- [45] Xiaojun Lin, Ness B. Shroff, and R Srikant. A tutorial on cross layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, August 2006.
- [46] Hans-Andrea Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, January 2004.
- [47] IETF Mobile Ad-hoc Networks (MANET) working group. <http://www.ietf.org/html.charters/manet-charter.html>.
- [48] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 255–265, Boston, Massachusetts, United States, 2000. ACM Press.
- [49] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com communicty. In *Proceedings of 25th AAAI Conference*, 2005.
- [50] Ueli Maurer. Modelling a public-key infrastructure. In *Proceedings of 1996 European Symposium on Research in Computer Security – ESORICS’96*, pages 325–350, 1996.
- [51] Barbara Misztal. *Trust in Modern Societies: The Search for the Bases of Social Order*. Polity Press, 1995.



- [52] Roger B. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2:225–229, 1977.
- [53] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
- [54] Michael J. Neely, Eytan Modiano, and Charles E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *Proceedings of IEEE INFOCOM*, volume 1, pages 745–755, April 2003.
- [55] Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, 2001.
- [56] Martin A. Nowak. *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard University Press, 2006.
- [57] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [58] Michael K. Reiter and Stuart G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security*, 2(2):138–158, 1999.
- [59] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In P. Herrmann et al., editor, *Proceedings of the 3rd International Conference on Trust Management (iTrust 2005)*, volume 3477 of *LNCS*, pages 77–92. Springer-Verlag, 2005.
- [60] J. Sabater. *Trust and Reputation for Agent Societies*. PhD thesis, Institut d’Investigaci en Intelligncia Articial, Bellaterra, 2003.
- [61] Eugene Seneta. *Non-negative matrices and Markov chains*. Springer series in statistics. Springer-Verlag New York Inc., 2nd edition, 1981.
- [62] Naum Zuselevich Shor. *Minimization Methods for Non-Differentiable Functions*. Berlin: Springer-Verlag, 1985.
- [63] Marco Slikker and Ann van den Nouweland. *Social and Economic Networks in Cooperative Game Theory*, volume 27 of *SERIES C: GAME THEORY, MATHEMATICAL PROGRAMMING AND OPERATIONS RESEARCH*. Kluwer Academic Publishers, 2001.
- [64] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM’03*, pages 808–817, San Francisco, March 30 - April 3 2003.
- [65] Frank Stajano and Ross J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, 2000. Springer-Verlag.

- [66] Jennifer Steiner, Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *USENIX Workshop Proceedings, UNIX Security Workshop*, pages 191–202, 1988.
- [67] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, pages 149–160, San Diego, California, August 2001.
- [68] L. Tassiulas. Scheduling and performance limits of networks with constantly varying topology. *IEEE Transaction on Information Theory*, 43(3):1067–1073, May 1997.
- [69] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
- [70] S.T. Wierzchon. Ising model. Eric Weisstein's World of Physics, <http://scienceworld.wolfram.com/physics/IsingModel.html>, 2002.
- [71] B. Yu and M.P. Singh. An evidential model of distributed reputation management. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 294–301, Bologna, Italy, 2002. ACM Press.
- [72] Yongguang Zhang, Wenke Lee, and Yi-an Huang. Intrusion detection techniques for mobile wireless networks. *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, 9(5), September 2003.
- [73] P. Zimmermann. *PGP User's Guide*. MIT press, 1994.