Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

TRUST MANAGEMENT SCHEMES FOR PEER-TO-PEER NETWORKS

by
Lin Cai

Peer-to-peer (P2P) networking enables users with similar interests to exchange, or obtain files. This network model has been proven popular to exchange music, pictures, or software applications. These files are saved, and most likely executed, at the downloading host. At the expense of this mechanism, worms, viruses, and malware find an open front door to the downloading host and gives them a convenient environment for successful proliferation throughout the network. Although virus detection software is currently available, this countermeasure works in a reactive fashion, and in most times, in an isolated manner. A trust management scheme is considered to contain the proliferation of viruses in P2P networks. Specifically, a cooperative and distributed trust management scheme based on a two-layer approach to bound the proliferation of viruses is proposed. The new scheme is called double-layer dynamic trust (DDT) management scheme. The results show that the proposed scheme bounds the proliferation of malware. With the proposed scheme, the number of infected hosts and the proliferation rate are limited to small values. In addition, it is shown that network activity is not discouraged by using the proposed scheme. Moreover, to improve the efficiency on the calculation of trust values of ratio based normalization models, a model is proposed for trust value calculation using a three-dimensional normalization to represent peer activity with more accuracy than that of a conventional ratio based normalization.

Distributed network security is also considered, especially in P2P network security. For many P2P systems, including ad hoc networks and online markets, reputation systems have been considered as a solution for mitigating the affects of

malicious peers. However, a sybil attack, wherein forging identities is performed to unfairly and arbitrarily influence the reputation of peers in a network or community. To defend against sybil attack, each reported transaction, which is used to calculate trust values, is verified.

In this thesis, it is shown that peer reputation alone cannot bound network subversion of a sybil attack. Therefore, a new trust management framework, called Sybildefense, is introduced. This framework combines a trust management scheme with a cryptography mechanism to verify different transaction claims issue by peers, including those bogus claims of sybil peers. To improve the efficiency on the identification of honest peers from sybil peers, a k-means clustering mechanism is adopted. Moreover, to include a list of peer's trustees in a warning messages is proposed to generate a local table for a peer that it is used to identify possible clusters of sybil peers. The defensive performance of these algorithms are compared under sybil attacks. The performance results show that the proposed framework (Sybildefense) can thwart sybil attacks efficiently.

TRUST MANAGEMENT SCHEMES FOR PEER-TO-PEER NETWORKS

by
Lin Cai

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering, NJIT

January 2012

APPROVAL PAGE

TRUST MANAGEMENT SCHEMES FOR PEER-TO-PEER NETWORKS

Lin Cai

Roberto Rojas-Cessa, Dissertation Advisor — Date
Associate Professor, Department of Electrical and Computer Engineering, NJIT

Nirwan Ansari, Committee Member — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Sotirios G. Ziavras, Committee Member — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Yanchao Zhang, Committee Member — Date
Associate Professor, School of Electrical, Computer, and Energy Engineering, Arizona State University

---

Guiling Wang, Committee Member — Date
Associate Professor, Department of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Lin Cai

Degree: Doctor of Philosophy

Date: January 2012

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2012
- Master of Engineering in Electrical Engineering,
  Beijing University of Posts and Telecommunications, Beijing, China, 2005
- Bachelor of Engineering in Telecommunications,
  Nanjing University of Posts and Telecommunications, Nanjing, China, 2002

Major:

Presentations and Publications:

L. Cai, R. Rojas-Cessa, "Mitigation of Malware Proliferation in P2P Networks using Double-Layer Dynamic Trust (DDT) Management Scheme," *submitted to Journal of Cyber Security and Mobility,* 2011.

L. Cai, R. Rojas-Cessa, and T. Kijkanjanarat, "Avoiding Speedup from Bandwidth Overhead in a Practical Output-Queued Packet Switch," *IEEE International Conference on Communications,* Kyoto, Japan, June, 2010.

L. Cai, R. Rojas-Cessa, "Three-Dimensional based Trust Management Scheme for Virus Control in P2P Networks," *IEEE International Conference on Communications,* Cape Town, South Africa, May, 2010.

L. Cai, R. Rojas-Cessa, "Bounding Virus Proliferation in P2P Networks with a Diverse Parameter Trust Management Scheme," *IEEE Communication Letters,* Vol. 13, pp. 812-814, 2009.

L. Cai, R. Rojas-Cessa, "Mitigation of Malware Proliferation in P2P Networks using Double-Layer Dynamic Trust (DDT) Management Scheme," *IEEE Sarnoff Symposium,* pp. 198-202, Princeton, NJ, April, 2009.

R. Rojas-Cessa, L. Ramesh, Z. Dong, and L. Cai, "Implementation of a Parallel-Search Trie-based Scheme for Fast IP Lookup and Table Update," *Proceedings of XIV Workshop Iberchip,* Puebla, Mexico, February, 2008.

R. Rojas-Cessa, L. Ramesh, Z. Dong, L. Cai, N. Ansari, "Parallel Search Trie-Based Scheme for Fast IP Lookup," *Global Telecommunications Conference (GLOBECOM),* Washington, DC, November, 2007.

L. Cai, Y. Yang, Y.X. Yang, "A New Idea of E-Learning: Establishing Video Library in University Network League," *IEEE International Conference on E-Commerce Technology for Dynamic E-Business, IEEE Computer Society Press,* pp. 126-129, 2004.

Y. Yang, L. Cai, Y.X. Yang, "Developing Digital Library Network in Asia," *Asian Info-Communications Council, Conference,* Kuala Lumpur, April, 2004.

I dedicate this thesis to my family. Without their support, understanding and patience, the completion of this work would not have been possible.

ACKNOWLEDGMENT

First of all, I would like to thank my advisor Professor Roberto Rojas-Cessa, who gave me the opportunity to study in NJIT, and then constantly gives me invaluable advices, criticisms and encouragements. His ideas and insights on research topics, problems and methodologies are the critical factors of all our accomplishments. It is a great honor to work with and learn from him.

I would also like to thank my dissertation committee members, who in alphabetical order are, Professor Nirwan Ansari, Professor Guiling Wang, Professor Yanchao Zhang, Professor Sotirios G. Ziavras. Their insightful reviews greatly helped me improve my dissertation.

It is a great pleasure to be a member of the Networking Research Laboratory. During these years, I live in a friendly and enjoyable environment, being able to work with many talented colleagues, Chuanbi Lin, Zhen Qin, Ziqian Dong, and Khondaker Salehin.

I need to thank my family, my parents, my husband and my baby, for their years of nurturing and supporting. I am proud of them as they are proud of me. This dissertation, with all my effort and endeavor, is the best present I can give to them.

Finally, I offer my regards and blessings to all of those who supported me in any respect during the completion of the PhD program.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Perhaps the simplest service model of a connection between two internet hosts is the one used in peer-to-peer (P2P) networking, where a host can perform as a client or server of information, depending on his interests and role in a network. Current models in the Internet defines the provider of a service as a host which is used almost exclusively as a server for technical and economical reasons, making the host able to handle a large number of requests as a centralized entity. This allows to localize resources and making the investment a feasible one.

In P2P networks, all peers are able to provide resources, including bandwidth, storage space, and computing power, in a cooperative manner. As peers are added to the network the demand for system resources and services may increase the total system capacity, improving scalability. P2P networks have the potential of converting any host to a source of information and of to allow the dissemination of information without the limitations of using a single (host) interface. The use of P2P for content distribution is currently under the consideration as a substrate for massive applications such as e-commerce [1]-[3] and IPTV [4]-[5], where video sources can rely on intermediate peers for further distribution of content. The potential of these distribution content systems has been demonstrated by some distribution networks for file sharing [6], [7, 8].

To scale up P2P networks, it is important that all clients provide resources, including bandwidth, storage space, and computing power. Thus, as peers arrive and the demand on the system increases, the total capacity of the system also increases. This is not always true for a client-server architecture with a fixed set of servers, in which adding more clients could mean slower data transfers for all users.

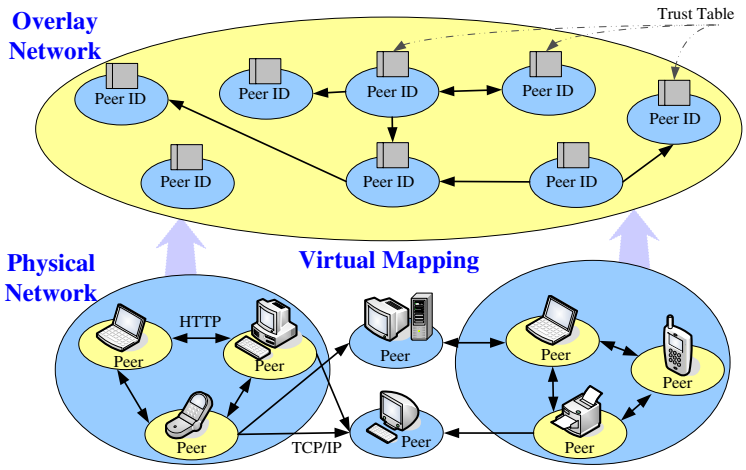


Figure 1.1 Peer to Peer network.

The P2P network consists of numerous participating peers as network peers. There are connecting links between any two peers who know each other: i.e. if a participating peer knows another peer in the P2P network, then there is a directed edge from the former peer to the latter in the overlay network.

The primary advantage of a P2P approach is that it leverages the resources of the many peers to provide the overall application and network services rather than relying on the resources of one or more central servers thus preventing those central servers from becoming a bottleneck for the entire network. A secondary advantage of a P2P approach is that there is no single central authority that can be blocked or removed and cause the collapse of the whole P2P network; this provides fault tolerance and robustness quality that may be desired for various reasons.

1.1 Peer-to-Peer Network Architectures

P2P networking has generated tremendous interest worldwide among both Internet surfers and computer networking professionals. P2P software systems like Kazaa and Napster rank amongst the most popular software applications. The original MP3 file sharing system, Napster [6] became the world's most popular Internet software

application literally overnight. Napster [6] typified a new P2P system defined above: a simple user interface running outside of the browser supporting both file serving and downloads. Most Napster user migrated to the Kazaa and Kazaa Lite software applications, and to the FastTrack network. FastTrack grew to become even larger than the original Napster network. Kazaa has suffered from its own legal troubles, but various other systems, like eDonkey, have continued the legacy of free P2P file sharing software. Based on how peers in an overlay network are linked to each other, the P2P networks are classified as structured or unstructured.

1.1.1 Unstructured P2P Network

An unstructured P2P network is formed when the overlay links are established arbitrarily. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another peer and then form its own links over time. In an unstructured P2P network, if a peer wants to find a desired piece of data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. The main disadvantage with such networks is that the queries may not always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding finds a peer that has the desired data. Flooding also causes a high amount of signaling traffic in the network, which results in very poor search efficiency. Most of the popular P2P networks are unstructured.

In unstructured networks (such as Gnutella), the placement of data (files) is completely unrelated to the overlay topology. Since there is no information about which peers are likely to have the relevant files, searching essentially amounts to

perform a random search, in which various peers are probed and asked if they have any files matching the query. Unstructured P2P networks differ in the way in which they construct the overlay topology and distribute queries from peer to peer. The advantage of such systems is that they can easily accommodate a highly dynamic peer population.

1.1.2 Structured P2P Network

Structured P2P networks employ a protocol to ensure that any peer can efficiently route a search to some other peer that has the desired file, even if the file is hard to find. Such a guarantee necessitates a more structured pattern of overlay links. By far the most common type of structured P2P network is the distributed hash table (DHT) [11, 12], in which a variant of consistent hashing is used to assign ownership of each file to a particular peer, in a way analogous to a traditional hash table's assignment of each key to a particular array slot [9, 10].

Structured networks, (such as Chord, CAN, PAST, and Tapestry) have emerged mainly in an attempt to address the scalability issues faced by unstructured systems. The random search methods adopted by unstructured systems seem to be inherently unscalable, and structured systems were proposed, in which the overlay network topology is tightly controlled and files (or pointers to them) are placed at precisely specified locations. These systems provide a mapping between the file identifier and location, in the form of a distributed routing table, so that queries can be efficiently routed to the peer with the desired file. Structured systems offer a scalable solution for exact-match queries, i.e., queries in which the complete identifier of the requested data object is known (as compared to keyword queries). There are ways to use exact exact-match queries as a substrate for keyword queries, however, it is not clear how scalable these techniques can be in a distributed environment. The disadvantage of

structured systems is that it is hard to maintain the structure required for routing in a very dynamic peer population, in which peers are joining and leaving at high rates.

1.2 Peer-to-Peer Network Threats

Attacks of P2P file sharing networks can be performed in many different ways and all may aim to harm the host computer or some other resources. The open access to these systems makes peers vulnerable to malicious users who can affect data or services to exploit them for personal or commercial gain [18]. Moreover, malicious users may take advantage of the advertisement of popular downloads that potentially encourage users to retrieve files and explore them, creating an incubating environment of malware. P2P file sharing networks can be attacked by spamming, spoofing, and identity theft. In the following sections, three attacks are introduced, virus proliferation, multiple identities, and spyware.

1.2.1 Virus Proliferation

Several interesting studies about virus proliferation have been presented in the literature [13]-[16], [9, 38]. They consider a network topology and features that describe the proliferation profile of a specific virus. Among other properties, viruses tend to increase their spreading rate in highly dense networks. Attacks of P2P file sharing networks can include many different types of viruses. The viruses can attach files on the network and begin to be downloaded to computers all over without anyone being aware of it. Viruses can harm networks and make them to run slowly or to shut down. If the virus is downloaded into other people's computers unknowingly, that virus can spread around their computer and cause major damage. Once in your computer, the virus attacks other files and goes out in emails, attacking other people's computers. Viruses or malware have usually a specific destructive objective, whether they aim to the host computer, to retrieve financial information that can be illegally

profitable, or to affect communication resources (e.g. denial of service). Attacks on P2P file sharing networks by virus can become frequent and they can be very costly.

A popular countermeasure against malware in a host is the use of an anti-virus application, which task can be coarsely divided into detecting a computing threat and removing it from the host. Installing virus protection on a host helps to ward off some of the problems associated with virus attacks. The successful detection by this protection software is based on the knowledge of hazardous files or software and their properties or signature for identification. Therefore, a new virus can be unnoticeable hosted in a peer until the detection program is updated for its identification. During this detection delay, the virus could be downloaded by another peer. Furthermore, after a virus is detected in a downloaded file by a peer, the detection software may remove the threat but this information is kept from other peers as it may be considered information of only local significance. Trust management schemes are a promising technology to detect misbehavior and suppress malware propagation in P2P networks [17, 18, 19, 20, 21, 22, 23].

1.2.2 Spyware and Identity Theft

Spyware, and adware are also common attacks of P2P file sharing networks. Spyware and adware are much like a virus but they attach themselves to the software downloaded in P2P file sharing networks. Spyware or adware tracks visited websites and makes a web browser run slower and cause antivirus software and firewalls not to function properly.

Spyware and adware can be difficult to delete off a host and can remain even after the original software is deleted. Spyware or adware can compromise user's privacy or security on the Internet by tracking website visits leaving the host open to attacks from P2P file sharing networks.

Attacks on P2P file sharing network can also be performed by spamming and spoofing. Spamming is when users of the P2P file sharing network get unsolicited information. This can be any type of information unsolicited but received you keep getting. Spamming can slow down a host and the P2P networks. Spoofing attacks on a P2P file sharing network includes downloading files that are not as they were described. The actual file could be illegal or questionable material. Spoofing may not do a lot of harm to a host but it does waste user's time and resources. It can become aggravating to download files that are not correct and this can become a problem for the P2P network itself if other people leave the network.

1.2.3 Sybil Attack

In the Sybil attack, a malicious peer impersonates a larger number of peers by using stolen and/or non-existing identities. For example, in an online environment, new identities may be created with minimal cost, so that users are not tied to unique identifiers. Therefore, a single user may create enough Sybils [24]. The Sybil peers can use multiple identities to falsely vouch for other Sybils, or to support an identity that would otherwise gain a bad reputation. This attack is a powerful way to get more benefits from a cooperative system than other users, without actually contributing to the networked community. Hence, Sybil attack can easily defeat trust-based management schemes. Each of the defenses against Sybil attack has different tradeoffs. Most defenses are not capable of defending against every type of Sybil attack [24, 25, 26, 27, 28].

Synchronous reputation system is vulnerable to Sybil attack [29, 30, 32]. An attacker that wishes to increase its reputation simply uses Sybil identities to create a copy of the existing graph representing trust relationships. The original peers cannot be distinguished from the impersonated peers. Thus, some Sybil peer has reputations equal or better to any original peer and the system will be subverted in

the end. Asynchronous reputation system is more robust to Sybil attacks, since no Sybil attacker can create a duplicate global graph as explained above in the symmetric case. For example, in [33], a social network is used as the central authority. A node trusts its neighbors. Each node learns about the network from its neighbors. It use randomized routes (a variant of random walks) on a social network topology in order to reject sybils.

When there are Sybil attacks in the network, peer reputation may not be enough to limit their proliferation because the reputation system is the one under attack. A cryptography-based trust management scheme is proposed in this thesis named SybilDefense.

1.3 Trust Management Schemes

Attacks of P2P file sharing networks are common and can be found with almost any computer applications. Trust management systems have been developed to enhance system security and to suppress malicious peers in various scenarios, such as distributed computing, agent technology [19, 21], GRID computing [35, 36], Ad-hoc networks[37, 38, 39], and component software [22, 23], among others. Trust management can help minimize risk and ensure the network activity of benign entities in distributed systems. In addition to use a level of trust for each peer, participants of P2P networks can distribution of trust information about peers in different networks scenarios to decrease the degree of effect of misbehaving hosts [95]-[91] in combination with trust management schemes. Trust information about peers can be built through evaluation of the interaction history of peers [95, 91]. Moreover, trust-based incentive schemes can potentially discourage free-riders and selfish peers by only offering services to cooperative peers [18].

Several of the trust management schemes in P2P network that have been described in the literature can be broadly categorized as globalized (Synchronous

reputation system) or localized schemes (Asynchronous reputation system). This categorization is based upon the approach adopted to evaluate and calculate trust value of the peers. In globalized schemes, the trust value is calculated (or partly calculated) and assigned to each peer by the trust management system [20, 40, 41, 42, 43, 44, 45, 46]. In [20], the trust value assigned to a trust relationship is a function of the combination of the peer's global reputation and the evaluating peer's perception of that peer. While in localized schemes, each peer computes the trust value by itself according to its local history statistics [17],[18].

Globalized management schemes are complex since the schemes need to assign trust value to each peer by collecting statistics and computing a global rating over a long period of time. Recently, a few localized trust management systems have been proposed for supporting trusted collaborations and suppress malware propagation [18, 17]. The scheme in [18] calculates the trust value by getting votes from all peers. For a large scale P2P network, it may be complex to collect votes from the majority of the peers due to practical network constraints of each anticipated peer. The scheme in [17] is based on localized trust evaluation and in warning dissemination to prevent others from downloading a file from a suspicious peer. The scheme aims to limit the proliferation of malware under the assumption that there is no local file infection. In other words, when a malware-free peer downloads a file containing malware, other existing files in the peer are not infected. However, viruses not only could specifically attempt to spread themselves but also infect the other files within the P2P network or pursue further hardware and software damage at the host of network level.

When a network entity establishes trust in other network entities, it can predict the future behaviors of others and diagnose their security properties. This prediction and diagnosis can fully or partially solve the following four important problems. Assistance in decision making to improve security and robustness. With a prediction of the behaviors of other entities, a network entity can avoid collaborating with

untrustworthy entities, which can greatly reduce the chances of being attacked. For example, a peer can choose the most trustworthy route to deliver its packets in a P2P network.

The prediction of peers' future behavior directly determines the risk faced by the network. Given the risk, the network can adapt its operation accordingly. For example, stronger security mechanisms should be employed when the risk is high. Trust evaluation leads to a natural security policy that network participants with low trust values should be investigated or eliminated. Thus, trust information can be used to detect misbehaving network entities. Moreover, with the assessment of trustworthiness of individual network entities, it is possible to evaluate the trustworthiness of the entire network. For example, the distribution of the trust values of network entities can be used to represent the healthiness of the network.

Trust-management engines avoid the need to resolve identities in an authorization decision [47, 48, 49, 50, 51]. Instead, they express privileges and restrictions in a programming language. This allows for increased flexibility and expressibility, as well as standardization of modern, scalable security mechanisms. Further advantages of the trust-management approach include proofs that requested transactions comply with local policies and system architectures that encourage developers and administrators to consider an application's security policy carefully and specify it explicitly [56, 57, 83, 67, 72, 89].

1.4 Motivation and Objects

The security of distributed network is considered since numerous businesses and Web sites have promoted "peer to peer" technology as the future of Internet networking. P2P networking has the potential of providing wide channels for information exchange, especially in the form of files. At the same time, P2P networking is prone to the proliferation of viruses. Although malware detection

software is currently available, this countermeasure works in a reactive approach and, in most cases, isolated manner.

Trust management is a promising proactive mechanism to prevent virus dissemination. Current trust models use peer reputation for this purpose. However, when viruses have infectious properties, peer reputation may not be enough to limit their proliferation. Moreover, peer reputation alone cannot bound epidemics efficiently in an infectious environment. A trust management algorithm is proposed in this thesis which uses the combination of trust values of peers and infection values of both peers and content to bound the proliferation of viruses in P2P networks. The simulation results show that the proposed trust management scheme can bound virus proliferation to a small number of peers, without inhibiting file-download activity. The influence of the propagation delay on the system performance is analyzed and observed, such as how delayed alerts benefit network infection as informed peers cannot prevent clean peers from downloading files from infected peers in a timely fashion.

A Sybil adversary creates a large amount of cheep peers or pseudonyms (Sybils) that act in the systems as separate entities, vouching for each other if necessary to fool the reputation system. By masquerading and presenting multiple identities, the adversary can control the network substantially. The Sybil Attack is a powerful way to get more benefits from a cooperative system than other users, without contributing the corresponding work.

In this thesis, a SybilDefense is presented and illustrated. How a Sybil attack can significantly impact the reputation system of the P2P network is introduced. Then the problem is abstracted theoretically and the necessary conditions to prevent the Sybil attack are pointed out. Finally, a simulation is built under the architecture of distributed P2P network. The results show that SybilDefense framework is effective and efficient in defending Sybil attacks.

CHAPTER 2

RATIO-BASED DOUBLE-LAYER DYNAMIC TRUST MANAGEMENT SCHEME

Trust management schemes aim to distribute reputation information about peers in different networks scenarios to categorize the behavior and contribution of hosts to the P2P community [95]-[91]. A dynamic trust management scheme was proposed [17]. This scheme is based on localized trust evaluations and in dissemination of alert messages to prevent others peers from downloading a file from a suspicious peer. The scheme aims to limit the proliferation of malware under the assumption that there is no local file infection. In other words, when a virus-free peer downloads a file containing a virus, other existing files in the peer are not infected. However, viruses not only attempt to spread themselves but also to infect other files in the host, or to pursue further hardware and software damage at the host or network level. Although the authors didn't assign a name to the scheme in the paper, this scheme is called dynamic threshold management (DTM) in the remainder of this paper, for brevity.

In this paper, the performance of DTM under file infection is discussed and it is shown that file infection has the potential to underscore proliferation countermeasures. To bound virus proliferation, the double-layer dynamic trust (DDT) management scheme is proposed, which uses a two-layer trusting strategy aimed to contain the impact of the internal infection. The results show that the proposed trust management scheme is efficient for bounding the dissemination of viruses in P2P networks under viruses with infectious properties. The proposed scheme uses a rating messaging scheme, used to advertise the undergone experience of a peer after a download. The effect of the propagation delay on the system performance is analyzed, and observed how delayed alerts benefit network infection as informed peers cannot

prevent clean peers from downloading files from infected peers in a timely fashion. Furthermore, it is shown that the adoption of the proposed scheme has a negligible the impact on the downloaded activity by peers.

The remainder of this chapter is organized as follows. Section 2.1 describes the proposed scheme based on dynamic trust management, the terms and the parameters for evaluation of peer trust, and the operation of the proposed management scheme in a P2P network. Section 2.2 presents the theoretical analysis of the number of infection peers under different trust management schemes. Section 2.3 presents a performance study of the proposed scheme, obtained through computer simulation. Section 4.6 presents the conclusions.

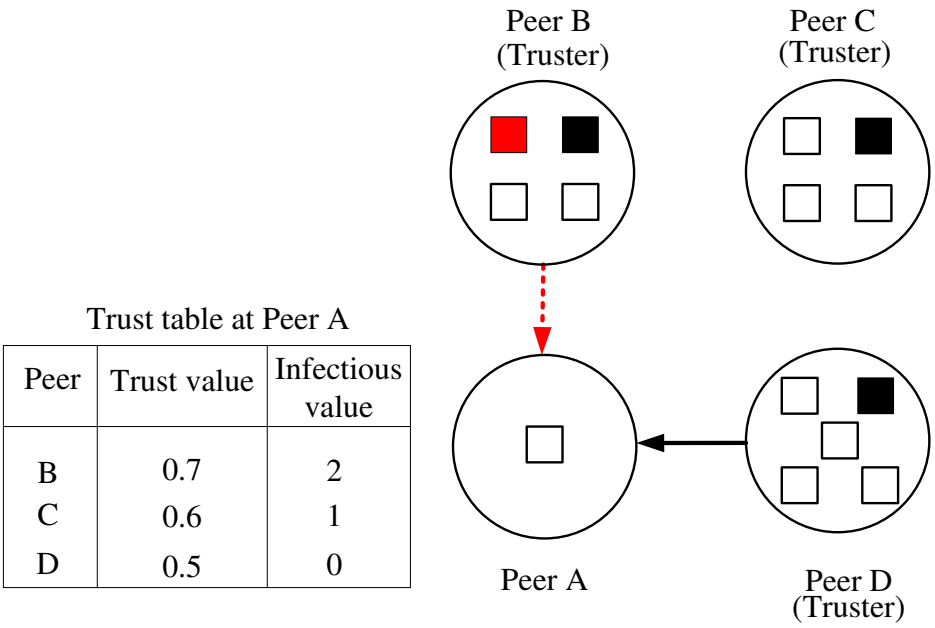
Peer-to-peer (P2P) networking is used by users with similar interests to exchange, contribute, or simply acquire data or information. These computer files (whether music, pictures, or software applications) are saved, and most likely executed at the downloading host. Worms, viruses and intrusion files find an open door in P2P networks, giving place to a very convenient environment for a successful proliferation throughout the network. Although malware detection software is currently available, this countermeasure works in a reactive approach and, in most cases, isolated manner. In this chapter, a proactive approach of trust management to contain the proliferation of malware in P2P networks is considered. Specifically, a cooperative and distributed trust management system is proposed to bound the proliferation of malware.

2.1 Trust Model in Peer-to-Peer Network

In the two-layer approach of the DDT scheme, each peer has a trust table that keeps two main parameters. The first parameter, similarly to that used in DTM, is a trust value about the other peers. A trust value at peer A about peer B indicates the probability that a virus is downloaded from B by peer A. The higher the trust A has on B, the smaller the probability.

Any peer in the system that is trusted by any other peer is called trustee and any peer that trusts a trustee is called truster. The second trust value in this table is designed to prevent internal infection, which is defined as the action of an infected file or malware that infects other files in a host by using this host as a means of local proliferation. This trust value is called the infectious value. The infectious value at peer A about peer B indicates the probability of internal infection from a file downloaded. The larger the infectious value is, the higher the probability of an infection from virus downloaded from B. This means that peer A would less likely to download a file from B.

The following is an example of how the proposed algorithm works. Consider that peer A wants a file and there are three possible trusters B, C, and D who have the desired file. Figure 2.1 shows this example. The black square in the Figure represents the requested file, the red square in peer B represents a virus, which has infected the other files in peer B with probability P_I. In the DTM scheme, peer A chooses the peer that has the highest trust value at A. Peer A then chooses peer B as the downloading source. In the proposed scheme, the higher the infectious value is, the larger the probability that an infection has occurred in the corresponding peer. Peer A then chooses a peer with the smallest infectious value from its eligible trustees. In this example, peer A selects peer D as the downloading source because D's infectious value is 0, which is the smallest among B, C, and D. In this way, the system guarantees that a peer performs a download from the cleanest source. Different from another schemes, it is considered that a file can be infected by a file stored in the same peer. For example, as this Figure shows, peer B has an infected file, which is different from the one requested by A. Therefore, if peer A had selected the sought file from peer B, this file may have been infected and all the files at peer A may become infected in turn.



Trust table at Peer A

Peer	Trust value	Infectious value
B	0.7	2
C	0.6	1
D	0.5	0

Figure 2.1 Example of the proposed scheme using a double-layer trust management

As another example, consider that peer A wants a file and none of its three trusters, B, C, and D, has that file, as Figure 2.2 shows. Peer B, C, and D forward the search request message to their trusters, consequently. Figure 2.2 shows that peer A finds the searched file in peers E and F, the grey colored ones. Peer A calculates the trust values on these two peers, which are $Tv(A,B) \times Tv(B,E) = 0.48$ and $Tv(A,C) \times Tv(C,F) = 0.3$. In the DTM scheme, peer A chooses the peer that has the largest trust value. Peer A then chooses peer E as the downloading source. In the proposed scheme, the larger the infectious value is, the larger the probability that an infection has occurred in the corresponding peer. Peer A then chooses the peer with the smallest infectious value from its possible trustees. The infectious value of peer E and F separately are $Iv(A,B) + Iv(B,E) = 3$ and $Iv(A,C) + Iv(C,F) = 1$. In this example, peer A selects peer F as the downloading source since its Iv is lower than that of peer E.

As Figure 2.3 shows, peer A downloads the file from peer F. When the downloading is finished, peer A checks whether or not the downloaded file has a

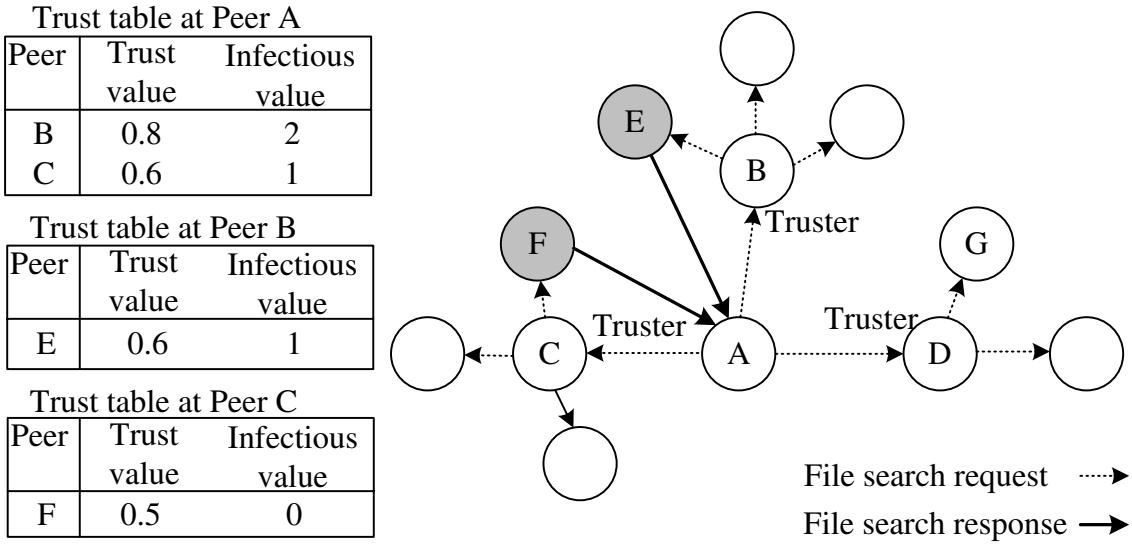


Figure 2.2 File search mechanism, from a peer to its trusters. Trusters forward the search request to their own trusters.

virus. If peer A is not satisfied with the download, it sends a warning message to its trustees as shown in Figure 2.4. When the trustees receive the message, they update their trust value and infectious value about peer E, and they forward the message to their trustees until the time stamp expires.

2.1.1 Trust Model

In the proposed trust management scheme, there are N peers, where each peer has a trust table with $2 \times (N-1)$ entries. The trust value and the infectious value in the trust table are used to select the downloading source. The trust model has the following major components:

- **Trust table**. The trust table in peer i is denoted as $T(i)$. The trust value of peer i on peer j, is denoted as $T_v(i,j)$, where $T_v(i,j) \in [-1,-1]$. For example, $T_v(i,j) = -1$ means that peer i does not trust peer j and any filed downloaded from j would be expected to be a virus with probability 1.0. On the other hand $T_v(i,j) = 1$ means that peer i trusts peer j and any file downloaded from j is

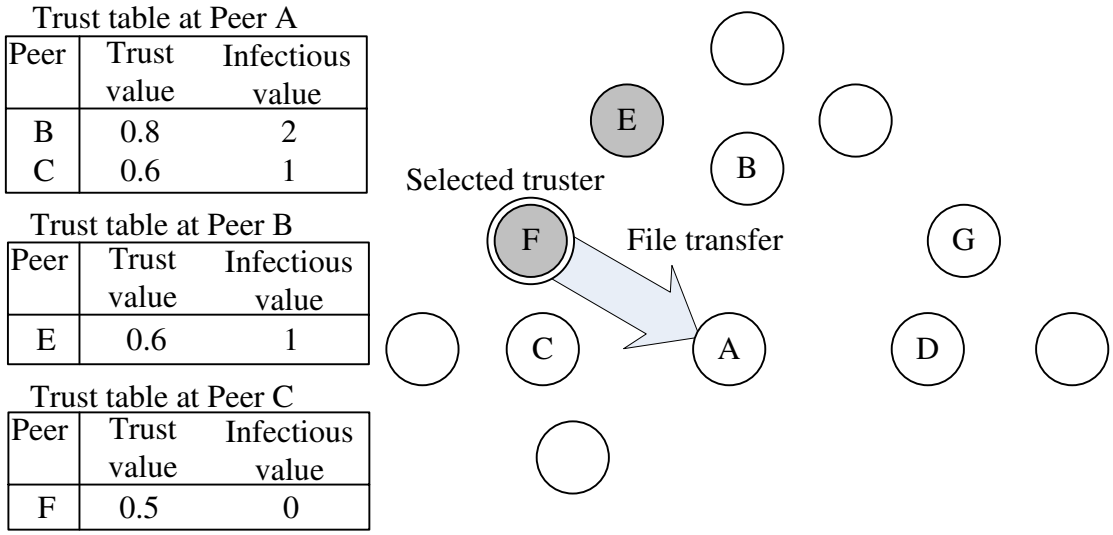


Figure 2.3 File download mechanism, from the downloading source to the file requesting peer.

expected to be innocuous with probability 1.0. Therefore, in the selection of the downloading source, peer j has the top priority to become the downloading source. Peer i updates its trust table after downloading a file from peer j by re-evaluating the trust and infectious values about peer j according to the experienced interactions with peer j, and these are represented as the ratio of downloads of clean files and all downloads from peer j. It is defined *social distance* as the number of peers that a message would traverse to reach a given peer. For example, if a peer forwards a file search request from a truster to its trustee, the social distance that the request travels is two.

- **Infectious value**. The second value in $T(i)$ is the infectious value I_v that represents the possible internal infection degree of peer j. The larger the value of I_v, the larger the probability that peer contains virus. If there are several trustees whose trust value is larger than the threshold for an acceptable trust value, the peer with the smallest I_v is selected as the downloading source. Peer i updates $T(i)$ if it receives an alert from its trustee, peer j.

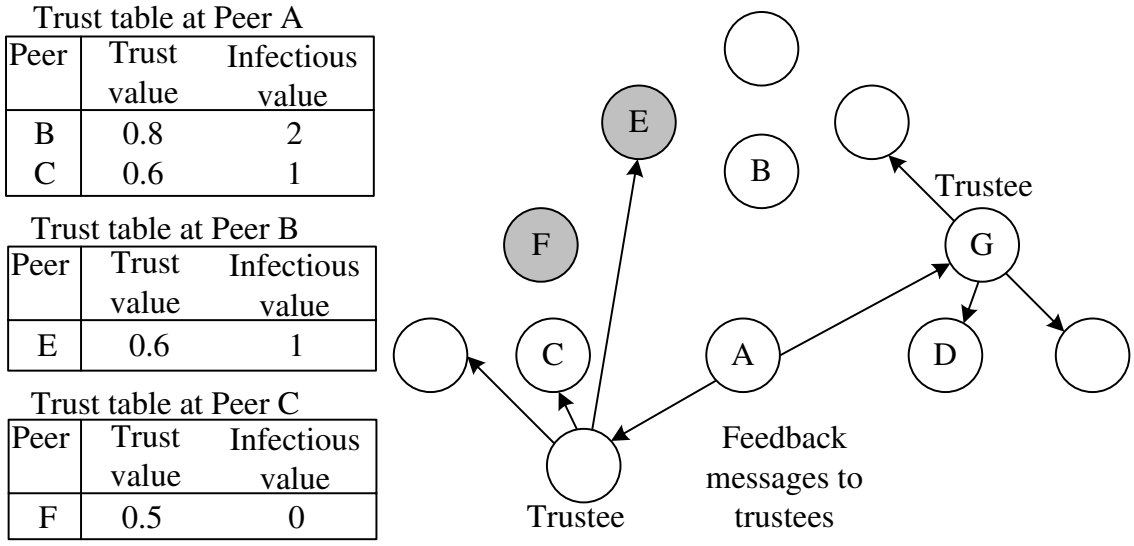


Figure 2.4 Feedback messages.

- **Antivirus software**. In this paper, it is considered that a peer has virus-detection software available. A successful virus detection indicates that a peer has downloaded an infected file, and the antivirus software can identify the file. Therefore, peer i detects a virus with probability $P_d(i)$.

- **Internal infection**. If a clean peer (whose files are virus free), downloads a file containing viruses, other existing files in this peer can possibly get infected with probability P_I. An infected download is defined as a download of a file containing a virus.

- **Propagation delay**. The propagation delay is the time it takes to download a file or the time that a ranking message takes to travel from one peer to another. The units of the propagation time in this dissertation a fixed period of time, called time slot. In this paper, it is assumed that a download takes a time slot. Also, it is assumed that the time that takes for a ranking message to be sent to a peer is one time slot. The propagation delay between peer i and peer j is denoted as $d(i, j)$.

2.1.2 Management Scheme

The trust management scheme works as follows. When peer i searches for file f, it checks the local file's reputation in the file record. If the file's reputation value is found at the database and is above the acceptable reputation threshold, Th_R, then the peer proceeds to find the file source.

The values held by a peer are updated after different actions take place. These are described as follows.

File Search. A peer i sends a request for file f to all trustees whose trust value is above the admissible threshold value Th_T (i.e., trustable trustees). Peer i chooses the peer that has the largest T_v and the lowest infectious value among those who have a copy of the requested file. If the file is not available from peer i's trustable trustees, the peer sends a recursive query for f to all trustees. In this query, the receiving trustee searches for the requested file among its own trustees. This process is performed recursively until either a fruitful search is achieved or there are no more trustees to query. After a recursive query, if peer k is introduced to i, new values are calculated: $T_v(i,k) = T_v(i,j) \times T(j,k)$, and $I_v(i,k) = I_v(i,j) + I_v(j,k)$, then the peer proceeds to the selection of a downloading source.

Post-download update. If the download of f is determined to be clean, $T_v(i,j) = \alpha T_v(i,j)$, where α is the rate of the trust value growth, $\alpha > 1$, while $I_v(i,j)$ remains unchanged. If the download of f is determined infected:

$$\begin{aligned} T_v(i,j) &= \delta T_v(i,j) \\ I_v(i,j) &= I_v(i,j) + 1 \\ F(i,f_l) &= F(i,f_l) + 1 \end{aligned}$$

where δ is the rate of the trust value degradation and $1 > \delta > 0$. During this phase, if $T_v(i,j) < th_w$, where th_w is the threshold to trigger a warning process, peer i issues warning messages to all its trusters. In this way, peers exchange only critical

information about other interacting peers. A warning message has the following format: $\{ID, v_j, f_m, \Delta, d\}$, where ID is the warning identification number, v_j is the identification of the peer that served as the source of a threatening file, f_m is the file's name, Δ indicates the decrement of the trust value at peer i, and d is the maximum number of truster hops the warning message is allowed to propagate.

Post-warning updates. After receiving a warning message from peer k about peer j, peer i updates the trust values. If $T_v(i,k) > Th_T$:

$$\begin{aligned} T_v(i,j) &= T_v(i,j) - \Delta T_v(i,j) \\ I_v(i,j) &= I_v(i,j) + \frac{(d-1)}{d} \\ F(i,f_l) &= F(i,f_l) + \frac{(d-1)}{d} \\ \Delta &= \Delta\frac{d-1}{d}. \end{aligned}$$

Because the forwarding of the warning message is bound by d, this value is also updated as $d = d - 1$. If the updated $d > 1$ and $\Delta T_v(k,i) > th_w$, peer i sends a warning message to its trusters with the updated values.

2.2 Analysis

Probability is used to analyze proliferation of malware over a P2P network and a recursive formula is developed to calculate the number of infected peers in the P2P network.

Consider a distributed trust management system with n legitimate peers and m infected peers uniformly distributed in the network. Each legitimate peer has v trusters in average. Total number of peers in the network is $n+m$. Let $I(t)$ represent the number of infected peers in the P2P network at time t. Therefore, $I(0) = m$. Let $H(t)$ represent the number of legitimate peers in the P2P network at time t. Therefore, $H(t) + I(t) = n + m$ and $H(0) = n$. Let the probability of each peer to

perform a download at time slot t be p. Then, the total number of downloads in a time slot are $(n+m)*p$. The probability of downloading from peer carrying a virus at time slot t is $\gamma(t)$, where $\gamma(0) = I(0)/(n+m)$. Let r the probability of requesting a malicious file of peer r at time slot t is $q(r,t)$. The total number of files in the network is M, and among them, M_f are infected. Therefore, $q(r,t) = \frac{M_f}{M}$.

The average number of infected peers of a P2P network, without using a trust management scheme at time $t+1$ can be expressed as:

$$\begin{cases} I(t+1) &= I(t) + \sum_{i=1}^{n-I(t)} p \times ((1-q(i,t)) \times \frac{I(t)}{n+m} \\ & \quad +q(i,t)) \\ q(i,t) &= \frac{M_f}{M} \end{cases} \tag{2.1}$$

where, $0 \leq I(t) \leq (n+m)$.

The probability of downloading a file from a malicious peer by a P2P network is reduced by using warning messages. Let $N(i,t)$ denote the number of malicious peers recorded by peer i at time slot t. $G(i,t)$ denotes number of legitimate peers in the view of peer i at time slot t, and $G(i,t) = n + m - N(i,t)$.

Consider that at time slot t, a truster peer of peer i, peer k, downloads an infected file from peer j, peer k then sends a warning message to its trustees if the malicious file is detected. Each warning message contains the downloading information such as name of the file, and the ID of peer j. The average number of infected peers at time slot $t+1$ can be expressed as

$$\begin{cases} I(t+1) &= I(t) + \sum_{i=1}^{n-I(t)} p \times ((1-q(i,t)) \\ & \quad \times \frac{I(t)-N(i,t)}{n+m-N(i,t)} + q(i,t)) \\ N(i,t+1) &= N(i,t) + \sum_{k=1}^{v} p \times ((1-q(k,t)) \\ & \quad \times \frac{I(t)-N(k,t)}{n+m-N(k,t)} + q(k,t)) \\ q(i,t) &= \frac{M_f}{M} \end{cases} \tag{2.2}$$

where, $0 < N(i,t) < I(t) < (n+m)$.

Since $N(i,t) > 0$, $\frac{I(t)}{n+m} > \frac{p \times (I(t)-N(i,t))}{n+m-N(i,t)}$. Therefore, the proposed trust management scheme reduces the growth rate of the number of malicious peers in the network.

When file reputation (FR) is used in DDT scheme, $F(i,t)$ denotes the number of malicious peers recorded by peer i at time slot t. The average number of infected peers can be expressed as:

$$\begin{cases} I(t+1) &= I(t) + \sum_{i=1}^{n-I(t)} p \times ((1-q(i,t)) \\ & \quad \times \frac{I(t)-N(i,t)}{n+m-N(i,t)} + q(i,t)) \\ N(i,t+1) &= N(i,t) + \sum_{k=1}^{v} p \times ((1-q(k,t)) \\ & \quad \times \frac{I(t)-N(k,t)}{n+m-N(k,t)} + q(k,t)) \\ q(i,t) &= \frac{M_f - F(i,t)}{M} \\ F(i,t+1) &= F(i,t) + \sum_{k=1}^{v} p \times ((1-q(k,t)) \\ & \quad \times \frac{I(t)-N(k,t)}{n+m-N(k,t)} + q(k,t)) \end{cases} \tag{2.3}$$

where $0 < F(i,t) < M_f$, and $0 < N(i,t) < I(t) < (n+m)$.

Compared to the DDT scheme, $\frac{M_f - F(i,t)}{M} < \frac{M_f - F(i,t)}{M}$ and therefore, the average number of infected peers in the DDT scheme with FR is smaller than that of using the DDT scheme alone.

2.2.1 Simulation Study of the DDT Scheme with and without Warning Messaging

A P2P network is modeled with 100 peers, and three malicious peers. The total number of files is 150, and 10 of them are viruses. The downloading probability p is 0.2. 10 peers are randomly selected as the number of trustees for each peer as initial condition. In Figure 2.5, NTM indicates the performance of DDT with no propagated messages and no trust management scheme. From this Figure, it is shown that the DDT scheme, combined with file reputation, limits the number of infected peers in 30 peers within 30 time slots.

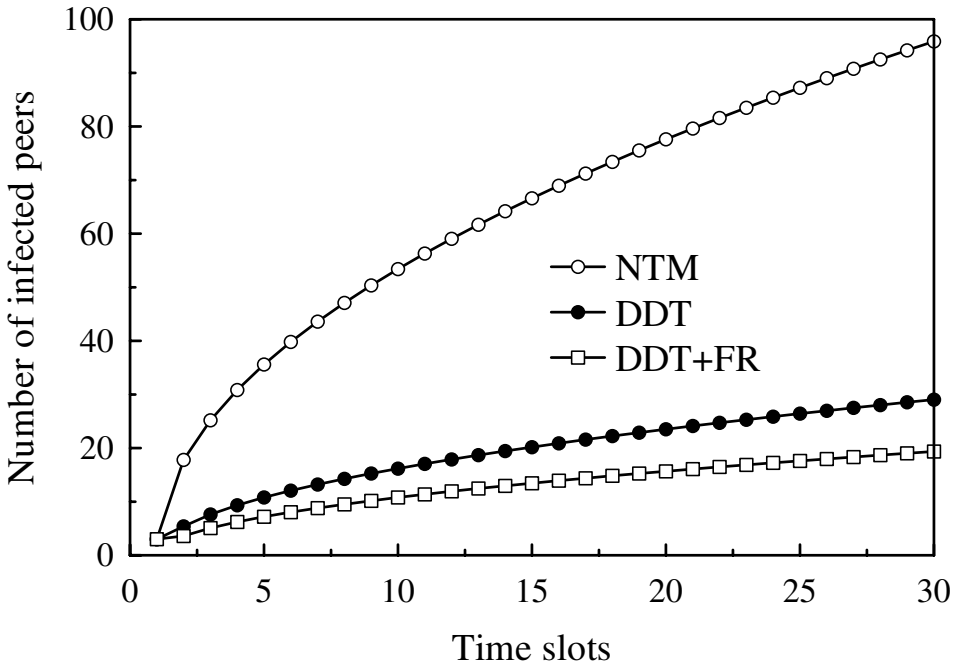


Figure 2.5 Theoretical estimation of proliferation of viruses in DDT, DDT+FR, and without a trust management scheme, NTM.

2.3 Performance

A P2P network is simulated using a mesh topology, with 100 peers randomly placed as active peers in the mesh. An active peer is a host that forwards, stores, or requests files to or from the other peers. The network has 150 existing files with several copies for each file. Files (and copies) are distributed randomly with a uniform distribution among peers. From these files, 60% of them are set as popular files (i.e., requested with high frequency). Among all files, 10 randomly selected files are designated as malware (i.e., virus). After a host downloads a malicious file, there is a probability of detecting it, which is denoted as P_d. Here, it is considered that the minimum time for an event (e.g., a download or a transmission of an alert from one peer to another in the network) is a fixed amount of time or time slot. The total number of infected peers are evaluated at each time slot.

Figure 2.6 shows the performance of the DTM scheme measured in the number of infected peers, where only a trust value per peer is used. In this scheme, the trust value of a peer is evaluated by considering the recorded downloads of a truster from its trustees. This Figure also considers when downloading a file and the broadcasting peers' trust values to trusters, and $P_I = 0$. The Figure shows two curves, one with $P_d = 0.5$, and $P_d = 0.25$. Because the number of infected peers changes significantly from time slot to time slot, the curve for $P_d = 0.25$ converges to 70 infected peers after 20 time slots, while the curve for $P_d = 0.5$ converges to 50 peers after the same time. This shows that the management scheme cannot bound the malware proliferation efficiently.

This Figure shows the performance of the DTM scheme with $P_I = 0.0$ in a network. This Figure shows that for peers with antivirus software with, $P_d = 0.5$, the number of infected peers is 45 after a long period of time (or until the number of downloads reaches 800), and for $P_d = 0.25$, the maximum number of infected peers approaches 70 after 750 downloads. These results show that the delay

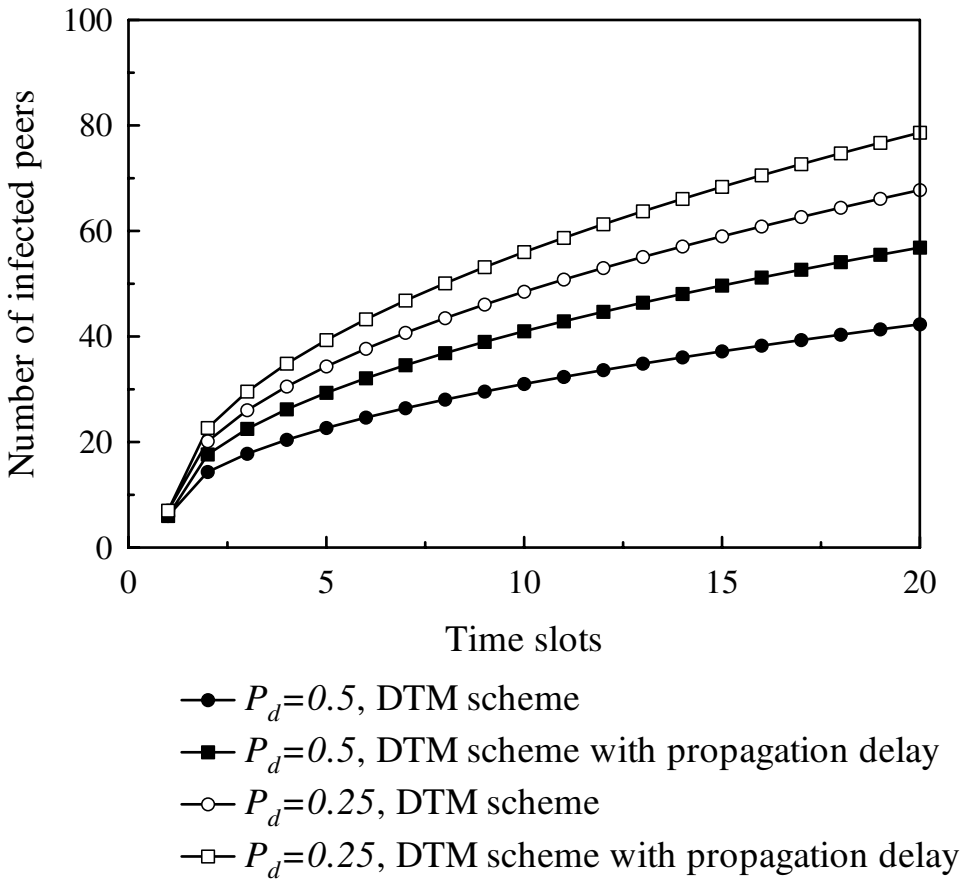


Figure 2.6 Proliferation of malware using T_v and $P_d = \{0.25, 0.5\}$, with no local infection and alert delay.

on disseminating the alert messages allows more high-risk downloads, allows the proliferation of malware.

Figure 2.7 shows the proliferation of the DTM scheme, as in the two cases above but, however, with infection probability ($P_I = 0, 0.25, 0.5$). This case considers no propagation delay for the alert system, and $P_d = 0.5$. This Figure shows that the infection property of viruses increase the effectiveness of malware proliferation, and even with $P_d = 0.5$, all peers in the network would end up infected after 1200 downloads.

Figure 2.8 shows the degree of proliferation of malware using the DTM scheme and the proposed DDT scheme, where I_v is used, under $P_d = 0.5$ with no propagation

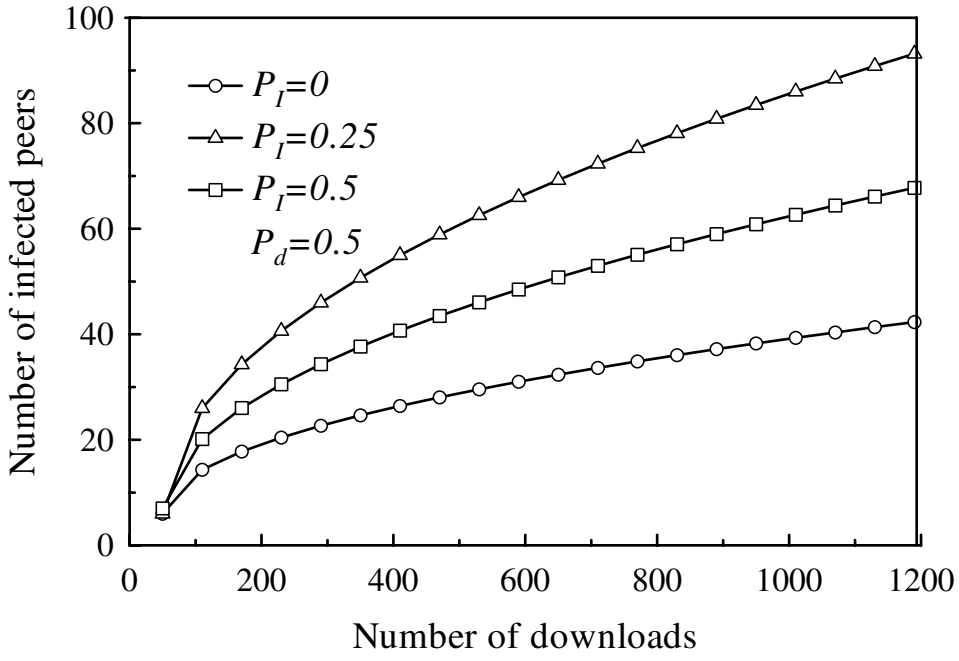


Figure 2.7 Proliferation of malware using DTM scheme, with $P_d = 0.5$ and considering infection probability $P_I > 0$.

delay in the distribution of the alert messages. This figure shows the spreading of viruses in the number of infected hosts per time slots. In this Figure, the performance of the DTM scheme decreases as the P_I increases. On the other hand, with the proposed DDT scheme, the impact of the infection probability is also noticeable but this impact is significantly lower, making the proposed scheme more effective.

These results are also shown in terms of the number of downloads. Figure 2.9 shows the proliferation of malware using the DTM scheme and the DDT scheme under $P_d = 0.5$ with propagation delays for the alert messages. The curves for different P_I, as in Figure 2.8, show a similar performance. The proposed scheme bounds it. In the case of a high P_I value, or $P_I = 0.5$, the number of infected peers drops from 100 peers as in the case of the DTM scheme to close to 30 peers in the DDT scheme.

Figure 2.10 shows the performance of both the DTM and the proposed schemes, measured as the number of infected peers per number of downloads in the network under different P_I values in an infectious environment. The proposed trust

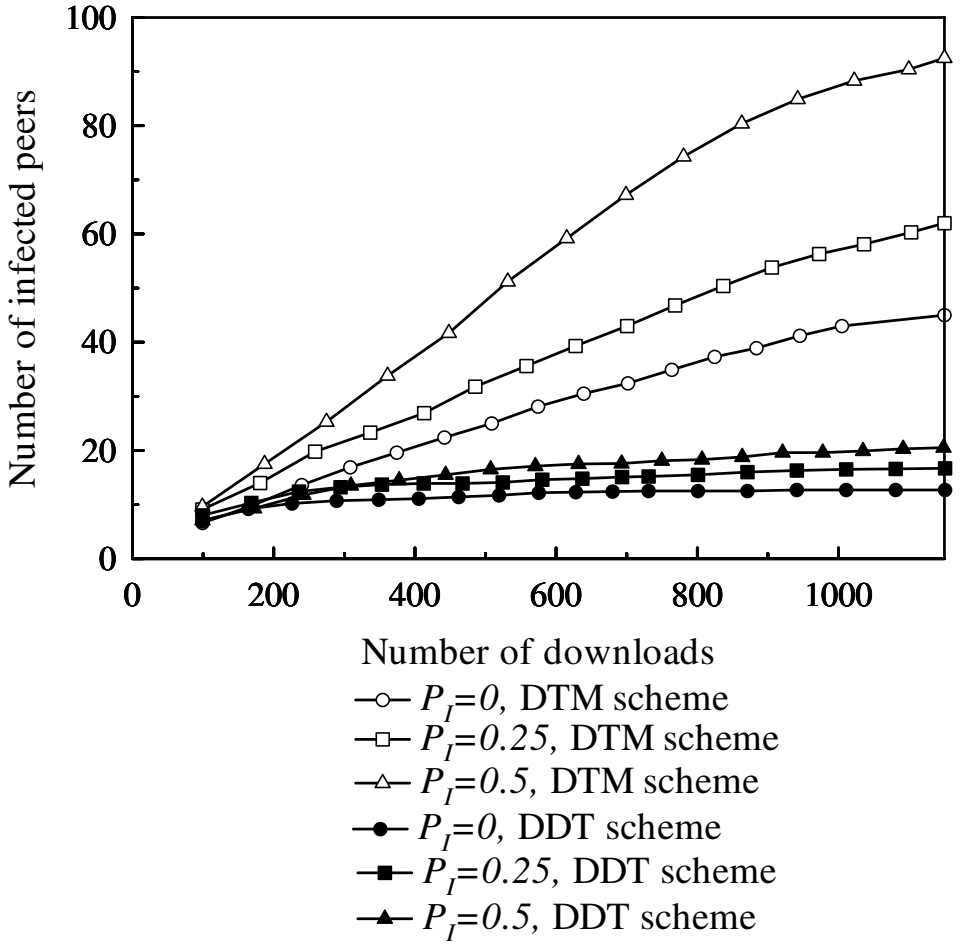


Figure 2.8 Proliferation of malware using the proposed DDT scheme with $Pd = 0.5$ and different P_I values in time slots.

management scheme uses file reputation, labeled FR in the Figure, with and without I_v. Curves a) to d) show that infectious viruses inhibit the effectivity of the DTM scheme as all peers in the network eventually get infected. This occurs because peers may be isolated after viruses have infected some peers. Curves e) to h) show that when file reputation is used, without recurring to I_v, the number of infected peers is bounded as the number of infected files is smaller than the total number of peers in the network. The proliferation is bounded because a peer can now identify a file coming from a peer with a record of no infections, in a proactive way. Curves i) to l) show that the use of a file reputation value in combination with I_v, which is updated

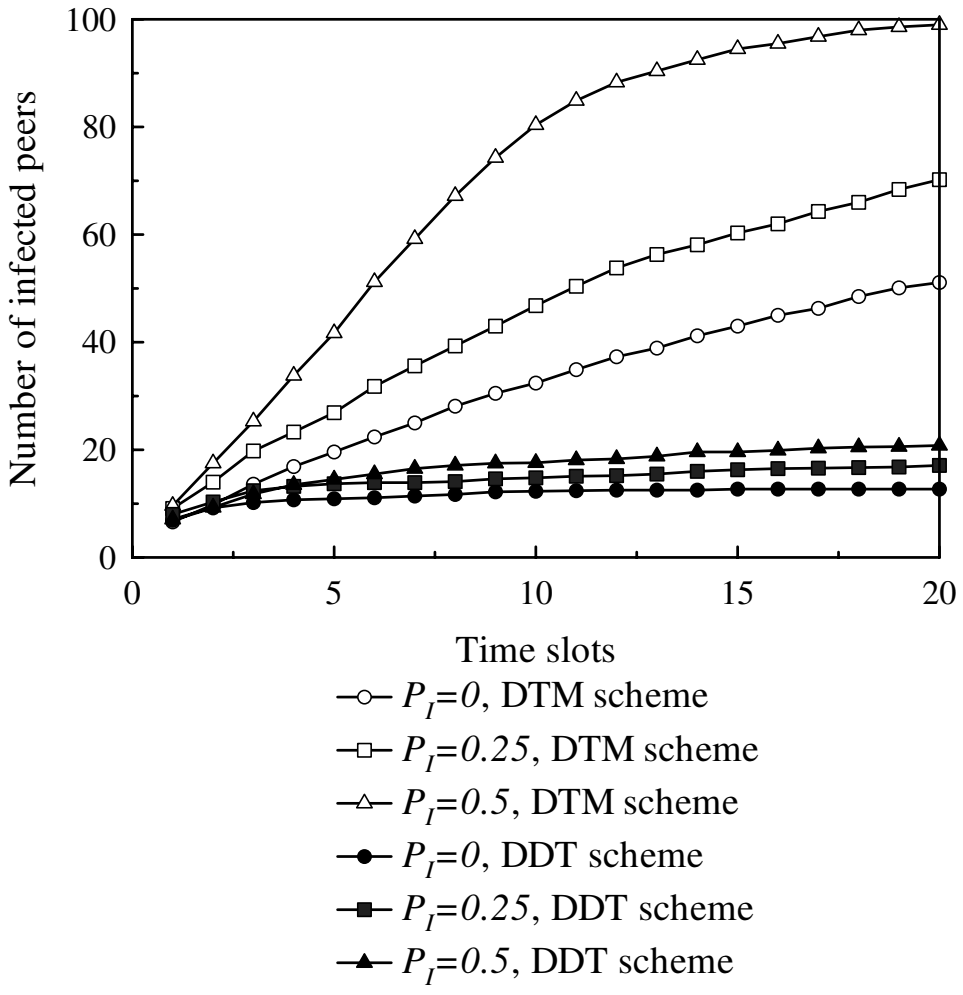


Figure 2.9 Proliferation of malware using the proposed DDT with $Pd = 0.5$ and different P_I values.

based on warning messages among peers, has the highest performance as the number of infected peers decreases to an average of 10. The warning messages then are also used to identify peers with trustable values but that may contain infected files. This is shown under the highest P_I values, $P_I = 0.9$, as the number of infected peers of l) is smaller than those of h).

Increasing the number of trust parameters in the management systems creates the risk of discouraging the download activity. The download activity of the network is evaluated using the same conditions as above. Figure 2.11 shows the download activity of a network using the DDT scheme, in downloads per time slot. The results

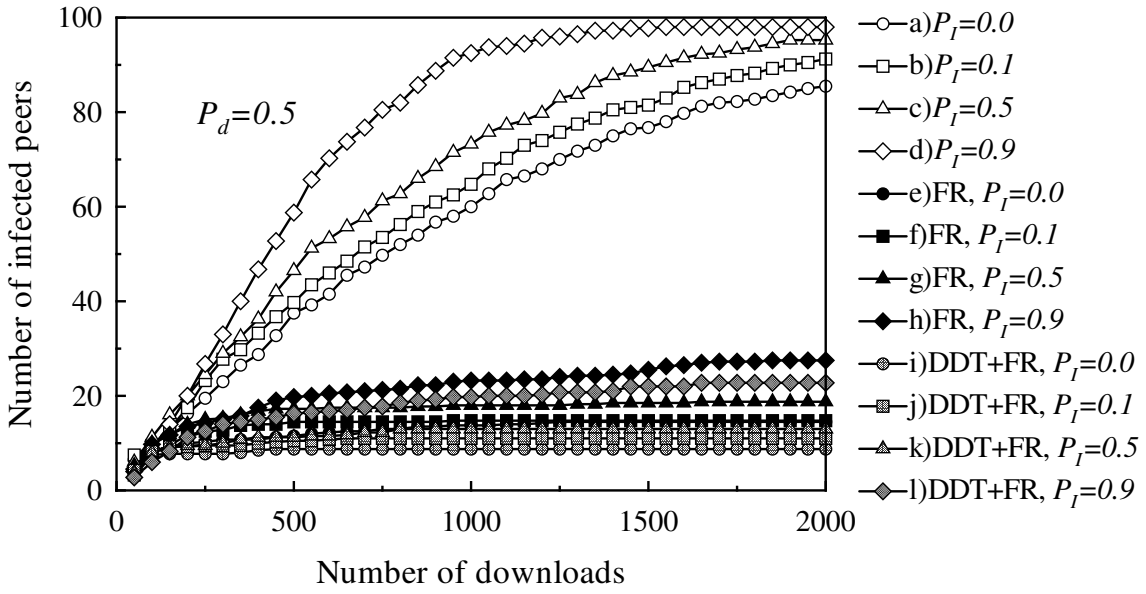


Figure 2.10 Comparison of proliferation of viruses using T_v only and with the proposed scheme.

show that the download activity with different P_I values, which impacts I_v for each peer, has no significant changes. This means that the proposed approach does not discourage network activity.

2.4 Conclusions

Trust management is a promising strategy to bound the proliferation of malware on peer-to-peer networks that can work jointly with virus detection systems. In this paper, it is showed that the use of a single trust value per peer has deficiencies in bounding the proliferation of malware. In most cases, it is highly probable that the majority of peers become infected. By using extra information, based on the infectious value, where the consideration of a peer having hosted an infected file, the proliferation of malware becomes bounded more effectively. By using computer simulation of a mesh peer-to-peer network, the improvement of this proposed approach has been shown. Furthermore, considering that trust parameters to bound proliferation have the potential of discouraging download activity in P2P networks, the impact of using

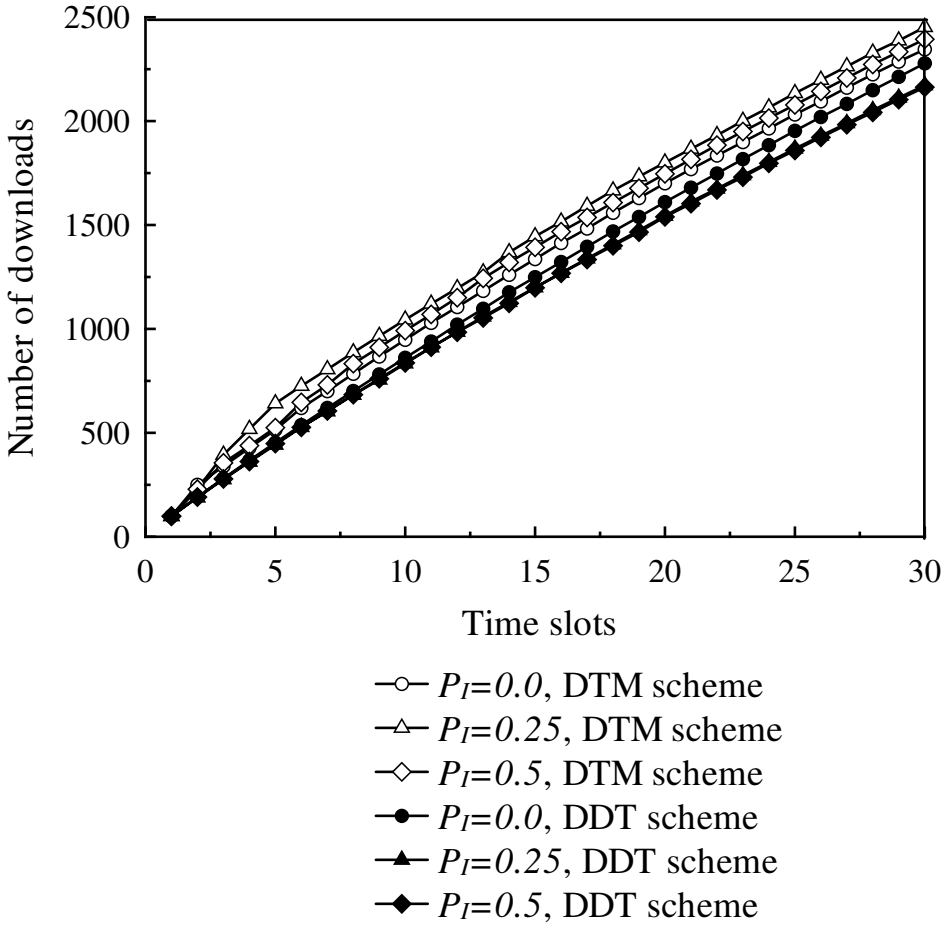


Figure 2.11 Download activity of the network using the proposed DDT scheme.

the proposed DDT scheme is studied. It is showed that the approach has little impact on the download activity of the network.

CHAPTER 3

THREE DIMENSIONAL TRUST MANAGEMENT SCHEME IN PEER TO PEER NETWORK

The calculation and aggregation of the local trust values in distributed environment is challenging and crucial in successfully designing a trust management scheme. Most existing trust management schemes normalize the trust value before further process. However, there can be some drawbacks in some normalizations. Especially, in the case where the attacker may utilize the normalization in ratio based trust management scheme to attack the system. The attacker can offer a high quality download resource in the first iteration to obtain a high trust from the receiver. After that, it can take advantage of the high trust value obtained to be regarded with a priority in the following iterations.

A ratio based trust management scheme is fragile under such kind of attacks. In this chapter, a novel normalization of trust values is proposed.

3.1 Trust Value Normalization

In a distributed environment, peers rate each other after each transaction. For example, in Eigentrust [20], peer i may rate a download as negative if the file downloaded is inauthentic, malicious, or tampered. Each peer i stores the number satisfactory transactions it has had with peer j, $sat(i,\ j)$ and the number of unsatisfactory transactions it has had with peer j, $unsat(i,\ j)$. To aggregate local trust values, Eigentrust used a normalized local trust value c_{ij} with the following format: $c_{ij} = \frac{S_{ij}}{\sum S_{ij}}$, where S_{ij} is defined as $S_{ij} = sat(i,\ j) - unsat(i,\ j)$ and $\sum S_{ij}$ is the number of differential transactions of peer i with all other peers it has interacted. Through the normalization procedure, all values are bounded between 0 and 1. All

previous works in P2P reputation systems **check these references and cite well** [1, 8] have all been based on similar notions of local trust values. In dynamic trust [17], the local trust value is calculated as $c_{ij} = \frac{S_{ij}}{\sum S_{ij}}$. It is observed that all above trust value normalized is obtained through the ratio calculation.

However, there are some drawbacks to normalize in these manners. These c_{ij} values are relative, and there is no absolute interpretation. Such as, if $c_{ij} = c_{ik}$, peer j has the same reputation as peer k as seen by peer i, but don't know if both of them are very reputable, or if both of them are mediocre. The reputed peer and the mediocre peer may have the same trust value. For example, consider that $sat(i,\ j) = 1, total(i,\ j) = 1$, and $sat(i,\ k) = 10000, total(i,\ k) = 1000$, which leads to $c_{ij} = \frac{1}{1} = 1 and c_{ik} = \frac{10000}{10000} = 1$ by calculating $c_{ik} = frac$satisfied transactionstotal transactions. As seen by peer i, the trust value of c_{ij} and c_ik are equal, but it is unfair to peer k since the total number of transaction between peer i and peer k is ten thousand times the total number of transaction between peer i and peer j. Hence, how to calculate and aggregate the local trust values c_{ij} in a distributed environment is challenging and crucial to successfully design a trust management scheme.

3.2 3D Trust Value Normalization

In this section, a new trust value normalization scheme is proposed. Different from the ratio calculation, the calculation is based on the ideas of closeness, used mainly in studying the behavior of functions close to values at which they are undefined. For example, the function $y = \alpha^{-\frac{\beta}{x}}$, where $0 < \alpha < 1$ and $\beta > 1$, is close to 1 with the increasing of variable x. Here, α and β are two parameters that control the approaching speed to 1. For example, Figure 3.1 shows four functions with different β value, $y = e^{-\frac{1}{x}}, y = e^{-\frac{5}{x}}, y = e^{-\frac{10}{x}}, y = e^{-\frac{50}{x}}$. From Figure 3.1, the approaching speed to 1 is reduces from the red line to the green line.

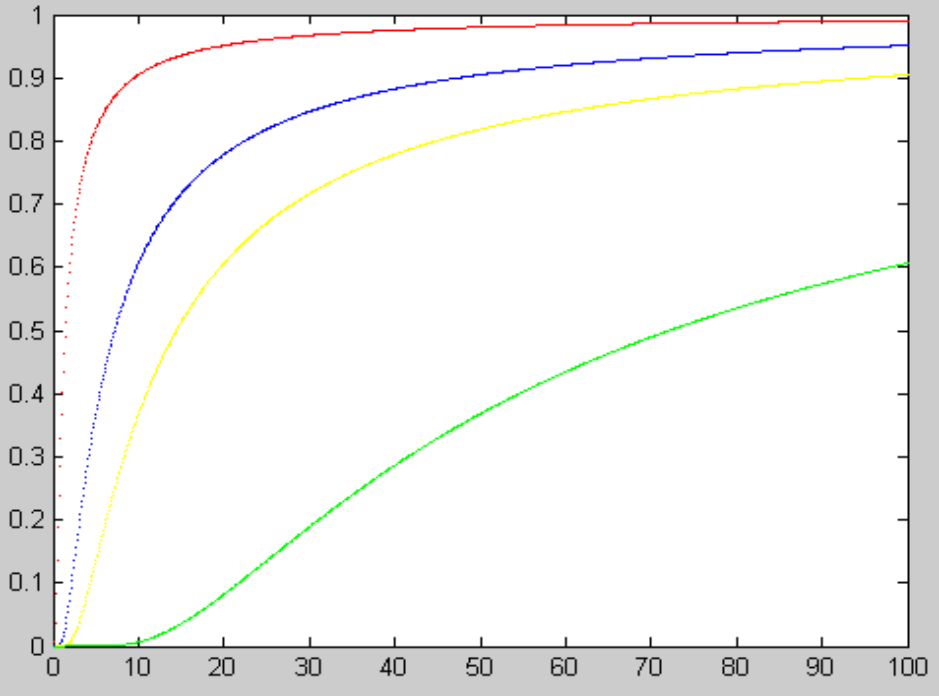


Figure 3.1 Functions with four different β values

Based on $y = \alpha^{-\frac{\beta}{x}}$, the y-axis is the spinning axle and x-axis as the base. After a 360-degree rotation, a three-dimensional surface is obtained. Figure 3.2 shows the surface of the three-dimension function spun from $y = e^{-\frac{5}{x}}$ and $y = e^{-\frac{50}{x}}$. The surface of the first function approaches to its closing limit 1 faster than the second one. Figure 3.2 provides us a view to the surfaces from different angles.

A questions arises: how to express the three-dimensional surface? If the surface is cut vertically from the top, Figure 3(a) displays a vertical cross section of the three-dimension function. The line in Figure 3.3(a) can be expressed as:

$$z = \alpha^{-\frac{\beta}{r}} \tag{3.1}$$

Considering the choosing of a point in the line randomly, where a perpendicular line to the base is drawn, a point A is got shown in Figure 3.3(a). From Figure 3.3(b), x-intercept of point A, y-intercept of point A and r compose a right-angle triangle. According to the Pythagorean Theorem, $r = \sqrt{x^2 + y^2}$ is obtained. $r = \sqrt{x^2 + y^2}$ is

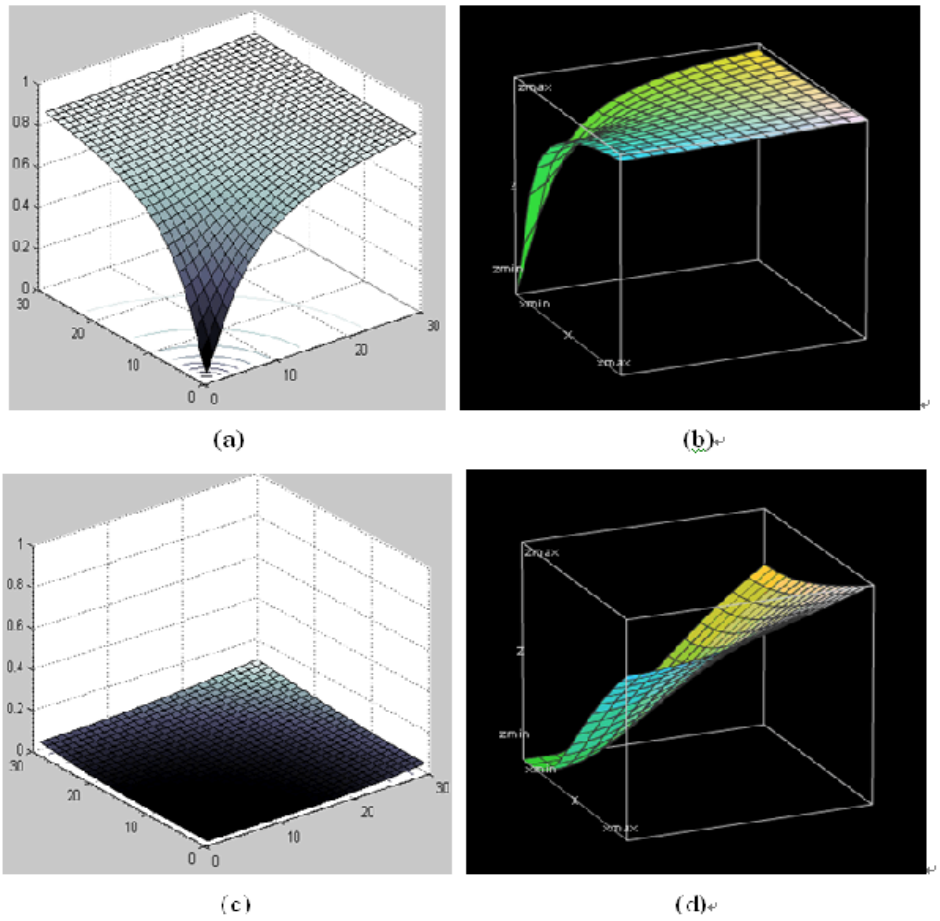


Figure 3.2 Surfaces of the three-dimensional functions.

substituted for r in equation 3.1, and to get it to express a three-dimensional surface $z = \alpha^{-\frac{\beta}{\sqrt{x^2+y^2}}}$. According to this transformation, the surfaces in Figure 3.2 can be expressed as $z = e^{-\frac{5}{\sqrt{x^2+y^2}}}$ and $z = e^{-\frac{50}{\sqrt{x^2+y^2}}}$, separately.

To calculate the trust value, $sat(i,\ j)$ and $tol(i,\ j)$ are used as the input variables x and y. the local trust value is defined as:

$$c_{ij} = \alpha^{-\frac{\beta}{\sqrt{sat(i,j)^2+tol(i,j)^2}}} \tag{3.2}$$

The local trust values is normalized in this manner because it models the trust value aggregation fairly and can reflect the real transition history accurately.

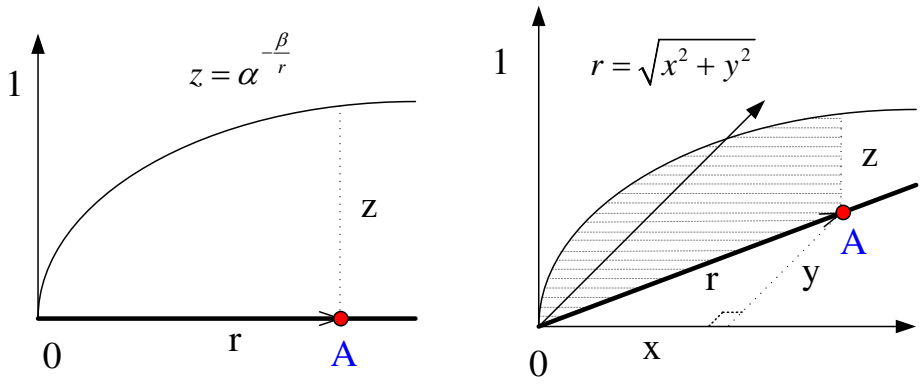


Figure 3.3 The mapping of two dimensions to three dimensions.

Moreover, the boundary of the trust function is between 0 and 1. Hence, the advantages of the previous schemes obtained through trust value normalization are maximally preserved in the new scheme. Nevertheless, the proposed scheme has some new features. First, the trust value is not only related to the proportion of satisfied transactions but also the total number of transaction history. Second, the model is more flexible since different approaching speed through control α and β is adjusted. For example, in the startup period, a small slope is chosen to control the increase speed of the trust value, if a peer constantly provides satisfactory transactions, in the near future, a larger slope which means quicker approaching speed to 1, can be given to the corresponding peer. Through the three dimension trust value management, peers have more flexibility to control the trust value calculation.

3.3 Performance and Results

A P2P network using a mesh topology is simulated, with 100 peers selected randomly as active peers in the mesh. In the beginning, the peers do not know each other. The trust relationship is built through the downloading interactions. There are two trust-value calculation schemes in the trust model. One is a ratio based trust model and the other is a 3D-based trust model. The robustness of the system is shown

by observing the number of peers infected. In the trust model, the attacker joins the system from the third time slot. The attack contains some clean files and some infected files or viruses. A peer will download from a stranger peer if and only if he can not find the downloading source anywhere else. There are 150 files in the mesh network. Among them, twenty percent are popular files. To reveal the effect of the attacker to the network, there are 10 files owned by the attacker only.

Figure 3.4 shows the performance of the ratio-based scheme and the 3D-based scheme with different local virus detection probability, $P_D = 0.25, 0.5$. From this Figure, after the attacker joins the network, the performance of the ratio based scheme degrades quickly. After 1400 downloads, almost all peers are infected. The attacker successfully subverts the system throughout. However, the infection is just partially controlled with the 3D-based scheme.

Figure 3.5 shows the effect of internal infection $P_I = 0,\ 0.25,\ 0.5$.The system performance degrades quickly with the increase of P_I. This case also considers no propagation delay for the alert system and $P_d = 0.5$. These results are also shown in terms of the number of downloads.

In Figure 3.6, the total number of infected peers after each time slot is evaluated. This Figure shows the spreading of the malware in number of infected hosts per time slot. The curves for different P_I in the Figure show a similar performance, as in Figure 3.5. Compared to the ratio-based scheme, the 3D-based scheme not only inhibits the proliferation of malware but also bounds it.

In the end, the download activity of the network using the same conditions as above is evaluated. Figure 3.7 shows the download activity of a network using the 3D based scheme, in downloads per time slot. The results show that the download activity with different P_I values has no significant changes. This means that the proposed approach does not discourage network activity.

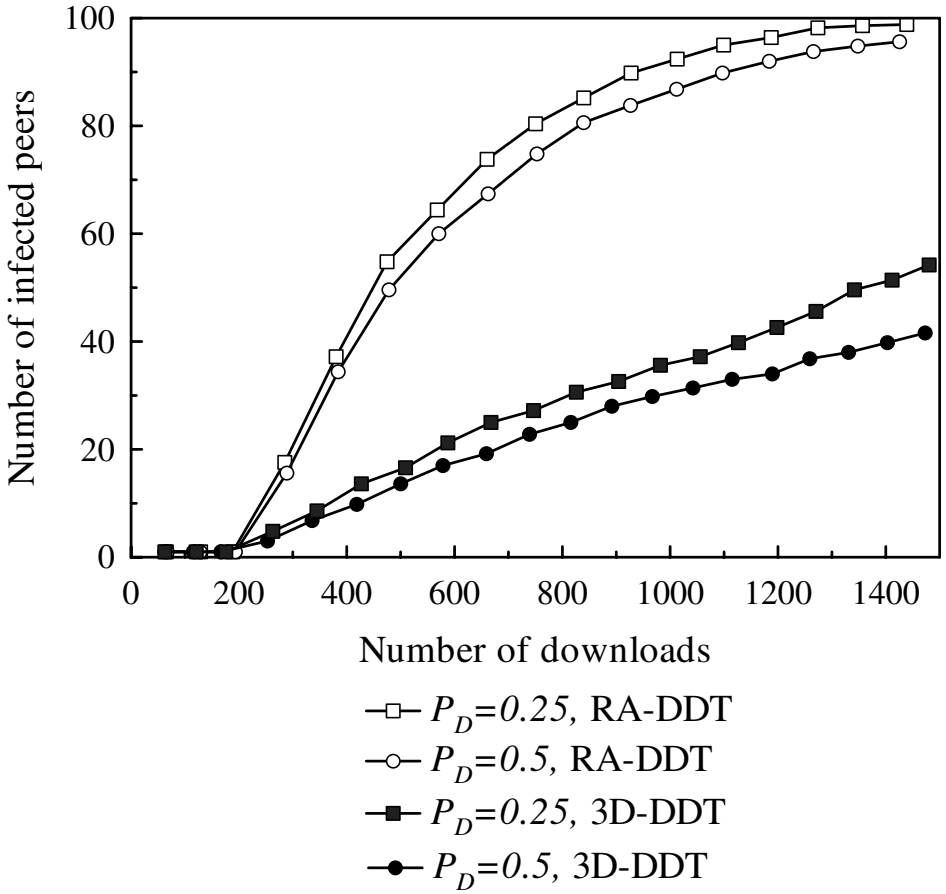


Figure 3.4 Comparison of the ratio-based scheme and the 3D-based scheme, under $P_D = \{0.25,\ 0.5\}$, with no local infection and alert delay.

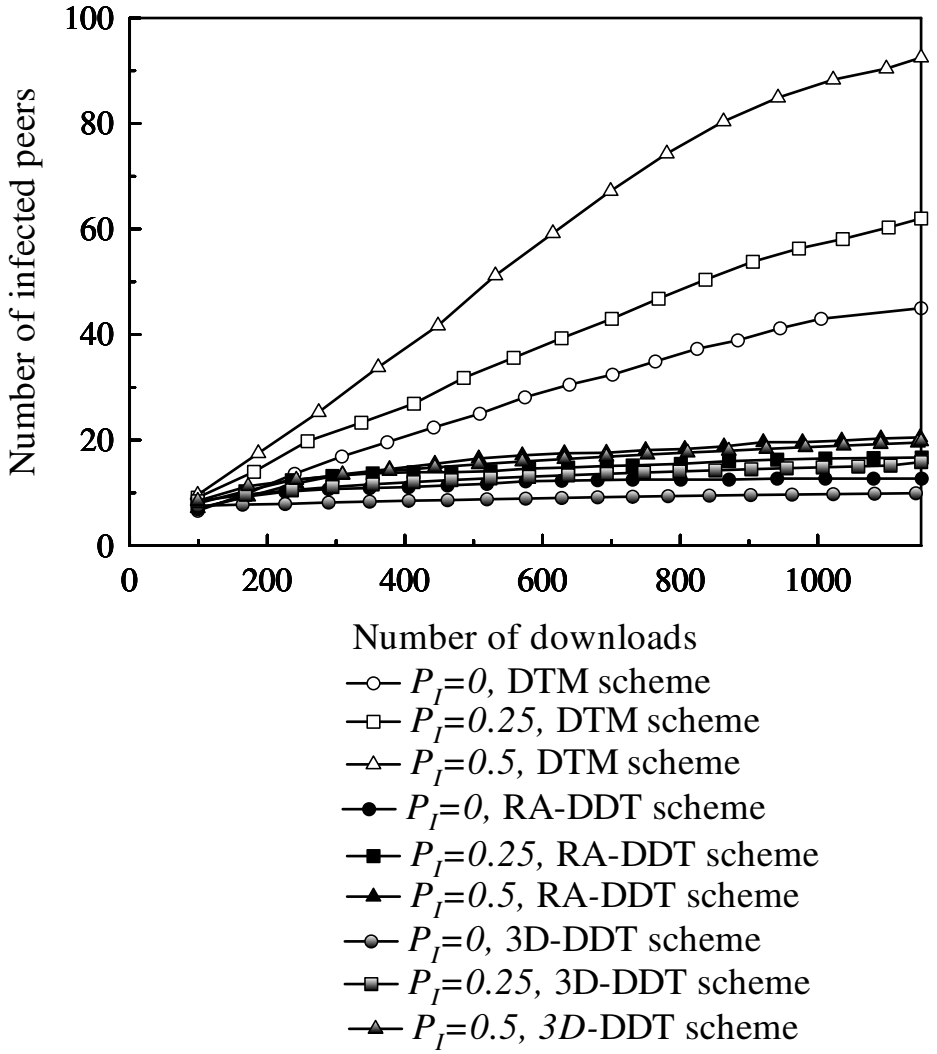


Figure 3.5 Comparison of the ratio-based scheme and the 3D-based scheme, under $P_D = 0.5, P_I = \{0,\ 0.25,\ 0.5\}$, with no local infection and alert delay

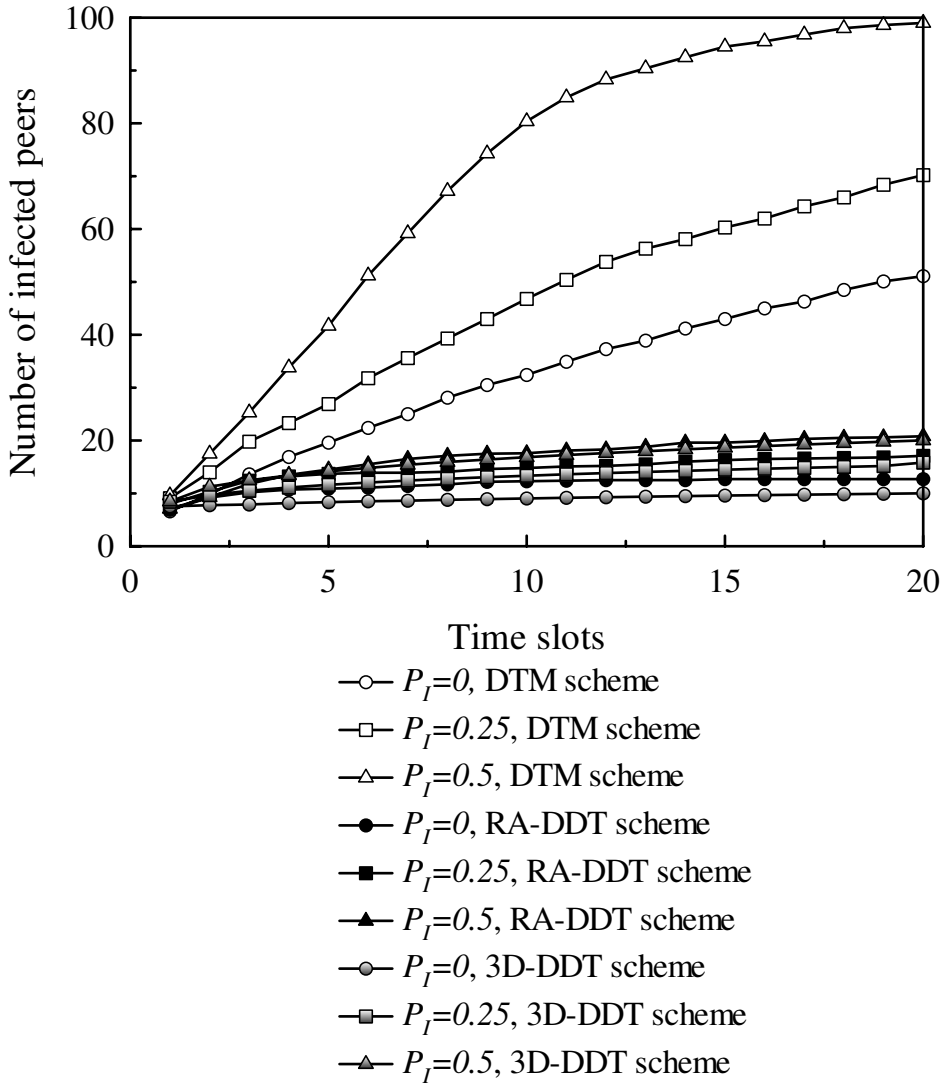


Figure 3.6 Comparison of the ratio-based Scheme and the 3D-based scheme, under $P_D = 0.5, P_I = \{0,\ 0.25,\ 0.5\}$, with no local infection and alert delay.

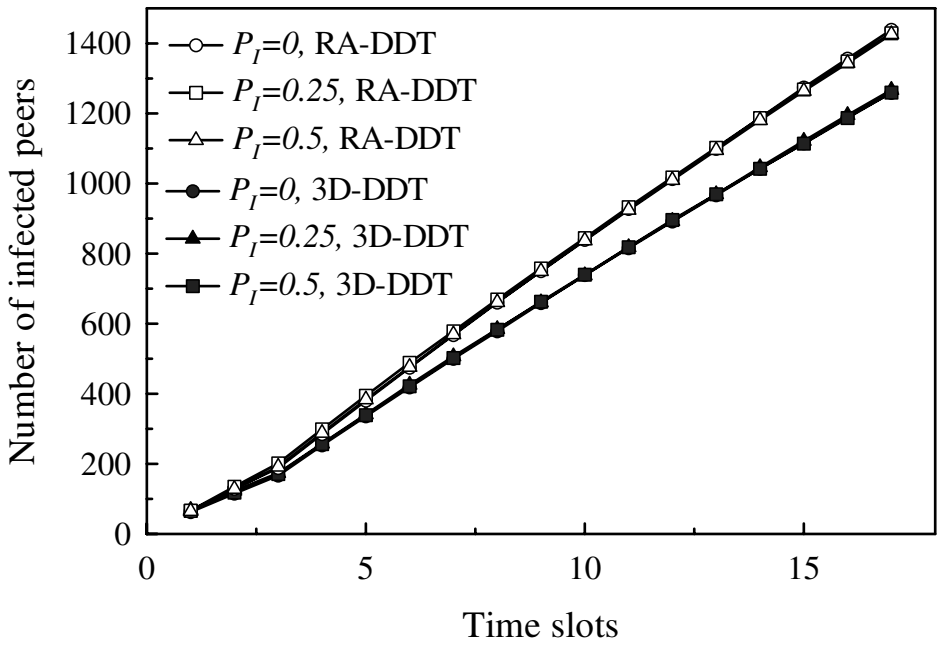


Figure 3.7 Download activity of the network using the 3D-based scheme.

CHAPTER 4

SYBILDEFENSE: DEFENSE OF A TRUST MANAGEMENT SYSTEM IN A P2P NETWORK AGAINST A SYBIL ATTACK

Trust management schemes have been used as proactive mechanisms to prevent virus dissemination in peer-to-peer (P2P) networks [20, 17, 95]. Current trust models are based on evaluating peer reputation as experienced by interacting peers. Reputation can be also applied to the objects exchanged between peers. A system that uses object reputation computes and publishes the reputation scores of the set of objects (e.g. service providers, services, goods or entities) in a P2P community. In these approaches, the peer reputation is set as reputation scores, which are computed using the opinions that peers hold about the objects [22]- [18]. The reputation scores are used by peers to decide whether or not to acquire the object. An object with a high reputation score becomes more attractive than one with a low reputation score.

A reputation system can be classified as synchronous or asynchronous. In synchronous reputation systems, like EigenTrust [20], the trust value is calculated by considering the values estimated by all the peers in the network. Trusted peers are built through the propagation of all reputation values. An identity's reputation depends solely on the topology of the trust table. In asynchronous reputation system, reputation for each peer is calculated locally and independent of any other. Each entity separately computes a trust value along their unique paths to every other identity in the system.

However, peer reputation may not be able to limit the proliferation of viruses in a network under sybil attacks because the reputation system is the one under attack. In a sybil attack, a sybil peer impersonates a larger number of peers by using stolen and/or non-existing identities. For example, in a P2P network, new identities can be

easily created as users are not tied to unique identifiers. Sybil peers take advantage of this and create multiple identities to falsely to support and inflate the reputation of the fake identities. In this way, a single peer may create a number of sybil peers for its own benefit [24]. A peer with a high reputation can obtain more benefits from a cooperative system with a smaller contribution to the network than those of other users. Hence, a sybil attack can potentially defeat trust-based management schemes. The existing defense schemes against sybil attacks consider different tradeoffs and most of them are not capable of defending against all the different sybil attacks [22, 24, 20].

Synchronous reputation systems have been shown vulnerable to sybil attacks [24]. In such systems, an attacker creates sybil identities, as impersonated peers, to create a copy of a graph of existing (trusting) relationships and playing the role of honest peers to increase its own reputation. In this graph, the original peers cannot be distinguished from the impersonated peers. Thus, sybil peers acquire reputations equal to or better than that of a honest peer, and the system is subverted in the end. Asynchronous reputation systems are more robust to sybil attacks, because no sybil attacker can create a duplicate global graph as explained above for the synchronous system case. This trust value can change over time as the entity interacts with and observes the behavior of different identities.

This chapter proposes a framework to bound proliferation of viruses or malware in P2P networks under sybil attacks. Within this framework, a set of necessary conditions are proposed for a trust management scheme needs to follow to defend against sybil attacks. It is shown that several of the previously proposed methods [28],[20] can be effective as they satisfy the proposed conditions. The framework is based in three different defense mechanisms: 1) a local trust table, where are peer records the different trust values, one per peer, 2) a k-means mechanism to

differentiate honest from sybil peers, and 3) a transaction verification scheme to verify reported interactions by all peers.

As part of the transaction verification scheme, a cryptographic trust model is proposed, called sybil defense framework (SDF). Furthermore, the k-means clustering mechanism is it used to identify sybil peers without recurring to threshold based schemes [17], [20], as it is well known that selecting a threshold value is complex.

The remainder of this chapter is organized as follows. Section 4.1 states the model of sybil attack and gives a theoretic formulation of sybil attack under reputation system. Section 4.2 proposes the sybil defense network model assumptions. Section 4.3 introduces the downloading process of the sybil defense algorithm with transaction verification and k-mean clustering scheme in detail. Section 4.4 introduces the trust management model. Section 4.5 shows the performance results obtained through computer simulation. Section 4.6 presents the conclusions.

4.1 Problem Statement

A sybil peer creates a number of virtual peers or pseudonyms (sybils) who behave in the system as separate peers, vouching for each other to artificially increase the reputation of a single or a few sybil peers. By using multiple identities, the adversary attempts to control the trust that other peers observe on it. A sybil attack is a mechanism for the attacking peer to get more benefits from a cooperative system than other users, without contributing to the community.

In this section, the sybil attack model is introduced and illustrated, how a sybil attack can significantly impact the trust management system of the P2P network.

4.1.1 Sybil Attack Model

The sybil attack in this chapter is an attack where one or several peers attempt to subvert the network by forging identities that can help to manipulate the trust

management system. The trust management system is used to rank the level of trust of peers as experienced by interacting peers.

The vulnerability of a trust management system under a sybil attack is determined by how effectively sybils can affect the system, while using strategies to countermeasure attacks and treating all peers without distinction (i.e., fairly).

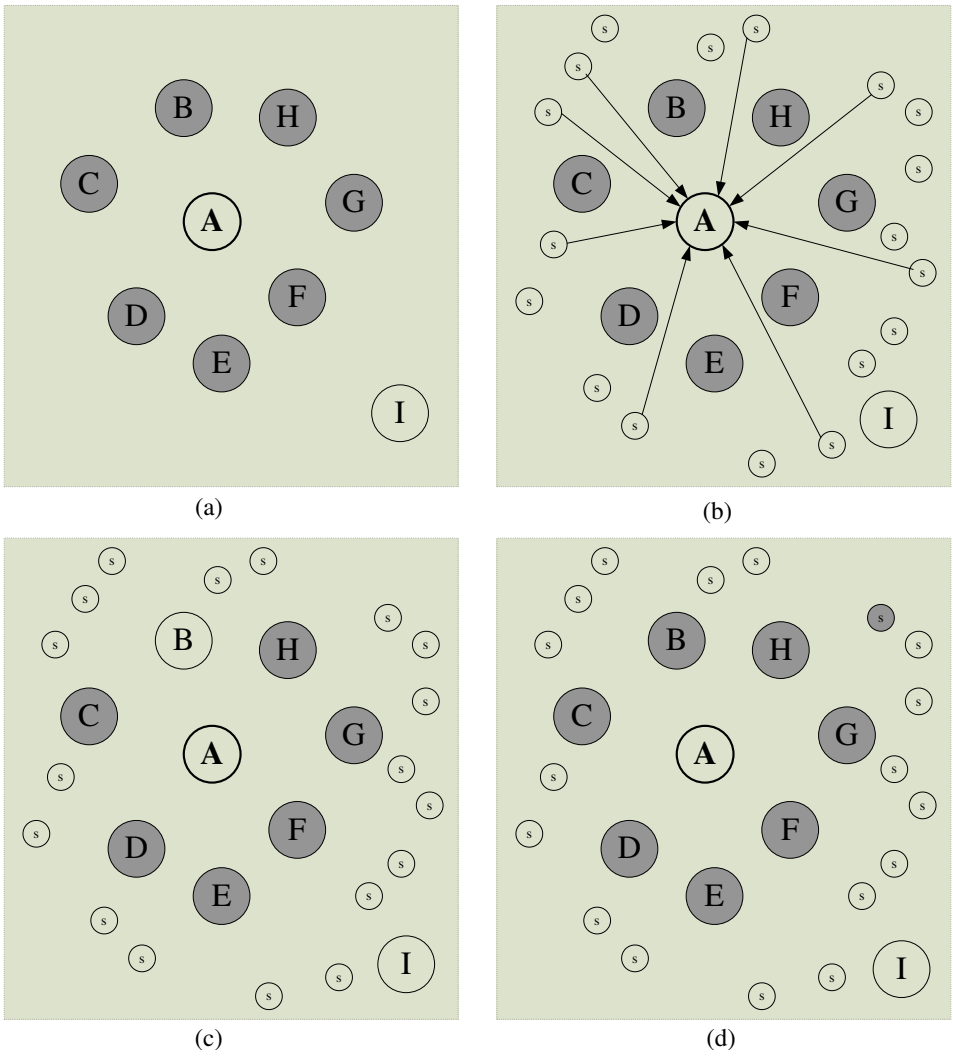


Figure 4.1 Sybil attack.

Figure 4.1(a) shows an distributed and asynchronous trust-management system. In a distributed and asynchronous system there is neither a centralized directory nor any control over the system topology or resource placement. When a new peer joins

the system, it forms connections with other peers freely. In this network, a peer that is trusted by peer i, where $1 \geq i \geq N$, is called *trustee* of peer i, and a peer that trusts peer i is called truster. Peer i has a trust table, which is denoted as $T(i)$. The trust value of peer i on peer j is denoted as $T_v(i, j)$, which is equal to the number of virus-free downloads divided by total number of downloads. A peer sends messages to its trustees to share the experienced downloading interactions. The information on these messages is used by the trustees to compute trust values about some other peers.

When peer i completes a download from peer k, if the download is determined uninfected, peer i sends a message to its trustees with a positive rating. This is called a positive rating message.Otherwise, peer i issues a negative rating message. After receiving the message, peer j, which is the trustee of peer i, calculates the propagated value $P_v(j, k)$. $P_v(j, k)$ is average of the propagated ratings about peer k. Peer j accepts rating messages only from its trusters. Each peer calculates its trust value and propagated value from its downloading history and the message propagated from its trustees timely. There is no central server in this network and each peer is independent and processes information locally.

Two different models of sybil attacks are identified. The first one occurs when sybil peers attempt to inflate the trust value of a single or multiple sybil peers who does not provide actual contribution to the network, with the objective of constructing a high reputation of sybil peers and, therefore, making them attractive for honest peers. In the second attack model, sybil peers attempt to defame one or multiple honest peers through the issuing of messages carrying low scores or (negative) rating messages. An effective attack would then decrease the trust value of the target peers (peers receiving the messages and those being defamed are all victim peers).

Figure 4.1 also shows an example of how a sybil peer can bias the selection of interacting peers for honest peers. There are eight peers in this Figure, where

the dark colored peers represent the trustees of peer A. The darker the color is, the stronger relationship between A and its trustees. Figure 4.1(a) shows that peers B, C, D, E, F, G, and H are trustees of peer A.

The sybil peers may take advantage of the ranting messaging process to gain a high reputation or depreciate reputation of a honest peer. As shown in Figure 4.1(b), sybil peers marked with s broadcast the rating messages to the network. Figure4.1(c) shows the one result of successful attacking. The trust value of peer A on peer B is decreased. Peer B is excluded from the trustee set of peer A. Figure4.1(d) shows another result of successful attacking. One sybil peer becomes trustee of peer A and its rank is improved by sybil strategy.

4.1.2 Model of Sybil Attack on a Trust Management System

Proof against a sybil attack has been formulated using a static graph of trust management to establish the required necessary conditions [24]. In this section, a theoretical analysis of a trust management scheme is introduced to explore necessary conditions for making this scheme immune to sybil attacks. A discrete time, where each event has a fixed duration or time slot, is considered in the following discussion.

In the following discussion, it is considered that each event in the P2P network, such as a download, the sending of a rating message, or the search for a file or peer, takes a constant unit of time, called time slot.

Consider a network using a trust management system with n honest peers and s sybil peers, each honest peer has m trusters in average and each sybil peer has r replications (i.e., sybil peers). Total number of peers in the network is $n + s + s * r$. Let $I(t)$ represent the number of infected peers (i.e., honest that have downloaded a file carrying a virus such that all files uploaded by the peer may also carry the virus) in the P2P network at time t. Let $T(t)$ represent the number of honest peers in the

P2P network at time t, $T(t) + I(t) = n + s + s * r$. The probability that each peer performs a download at the t^{th} time slot is p.

In this case, the total number of downloads in a time slot is $(n + s + s * r) * p$. The probability that a downloading is performed from a sybil peer at time t is γ_t, and $\gamma_0 = I(0)/(n + s + s * r)$, where $I(0) = s + s * r$.

Let $N(i, t)$ denote the number of sybil peers recorded by peer i at time t, and let $G(i, t)$ denote the number of honest peers as determined by peer i at time t, and $G(i, t) = n + s + s * r - N(i, t)$.

Now, consider that at time t, a truster (peer k) of peer i downloads an infected file from peer q. If the downloaded file is detected infected, peer k sends a rating message to its trustees. Each rating message contains the download information, such as name of the file, the download time, and the ID of peer q. $\Omega(k, t)$ represents the average value for each peer calculate from those values in the issued rating message. A large $\Omega(k, t)$ means more information is obtained by the propagated rating messages. With this information, peer i can quarantine identified or suspicious peers more efficiently.

For a distributed P2P network, the average number of infected peers at time $t + 1$ can be expressed as

$$\begin{cases} I(t+1) = I(t) + \sum_{i=1}^{n-I(t)} \frac{p * (I(t) - N(i,t))}{n + s + s * r - N(i,t)} \\ N(i,t+1) = N(i,t) + \sum_{k=1}^{m} \frac{p * (I(t) - N(i,t)) * \Omega(k,t)}{n + s + s * r - N(k,t)} \end{cases} \tag{4.1}$$

where, $0 < N(i, t) < I(t) < (n + s + s * r)$.

Therefore, the effectivity of of this mechanism depends on $\Omega(k, t)$. A conventional trust management scheme quarantines a sybil peer at a time, which means that each propagated rating message carries information about a sybil peer, and $\Omega = 1$ (DTM scheme [**?**]). The Sybildefense framework uses a k-means clustering

algorithm and advertisement of rating messages, which carries information about a possible sybil peer q and its trustees. Hence, the information content in each message is $\Omega = r$, where $r > 1$. When r is a large value, the Sybildefense scheme identifies sybil peers accurately and in a short time.

The number of compromised peers in a network is evaluated with 200 peers, two sybils, and each sybil has five replications. The downloading probability p is 0.5. The average number of trustees for each peer is 10.

Figure 4.2 gives $I(t)$ for different Ωs, estimated with $\Omega = \{0, 1, 10, 80\}$ to 80, and $N(i,t) = 0$ ($\Omega = 0$), which means no rating messaging. As shown in Figure4.2, the number of infected peers decreases as Ω increases. To increase *Omega*, statistics can be collected (and estimated) for longer periods of time. As the Figure shows, with $\Omega = 0$, or no rating messages, all the peers get infected quickly.

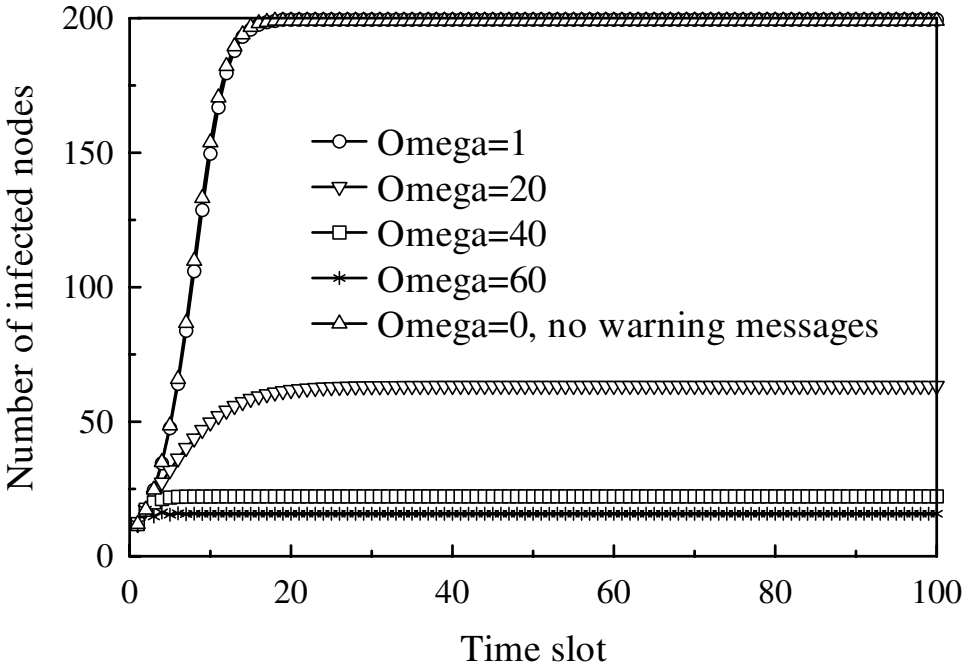


Figure 4.2 Number of infected peers under sybil attack.

The above analysis shows that an efficient and timely distribution of information about sybil peers and files are necessary and sufficient conditions to banish the effect

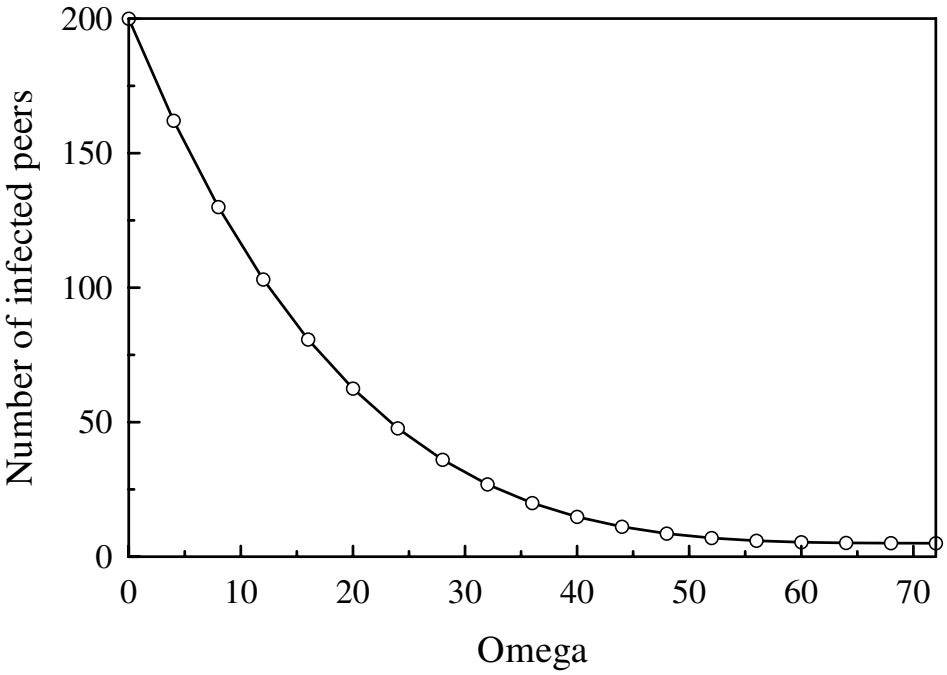


Figure 4.3 Number of infected peers under different Ωs.

of a sybil attack in a dynamic trust management scheme under discrete time. In section 4.2 discusses how to increase Ω by using the proposed framework.

4.2 Clustering of Sybil Peers

A characteristic of sybil attacks is the duplication of peer identities, as discussed before. In general, most sybil peers do not have a large trust value within the set of honest peers right after the sybil peers have been created. As the objective of the sybil peers is to raise their trust values, they may prevent to interact (or report a high interaction rate) and this produces a strong connectivity among sybil peers, creating a sybil cluster. This feature is then used in this dissertation for sybil identification.

A network is represented as a weighted directed graph $G = (V, E)$, with peers represented by graph peers V, and trust relationship between a pair of peers $i, j \in V$ represented as directed edges $e \in E$ between i and j. In this paper, $e(i, j)$ indicates that the edge is directed from peer i to peer j. Peer j is called a truster of peer i, and

peer i is said to be a trustee of peer j. Edges have different weights, and the weight of $e(i, j)$ is equal to the trust value of peer i about peer j, $T_v(i, j)$. Note that an existing edge indicates strong trust, as edges are considered existing only if $e(i, j) > 0$.

The set of edges inter-connecting honest peers a cluster of honest peers are defined. In similar way, the set of edges connecting sybil peers define a cluster of sybil peers. The links connecting a honest cluster to a sybil cluster are called *attack edges*. An edge may exist between a sybil peer and a honest peer if a honest peer decides to download a file from an sybil peer. As shown in Figure 4.4, the central cluster represents the honest-peer cluster. The three clusters around the central cluster represent three sybil-peer clusters. The first and the third sybil cluster have two attack edges to the honest-peer cluster. The second sybil-peer cluster has one attack edge to the honest-peer cluster.

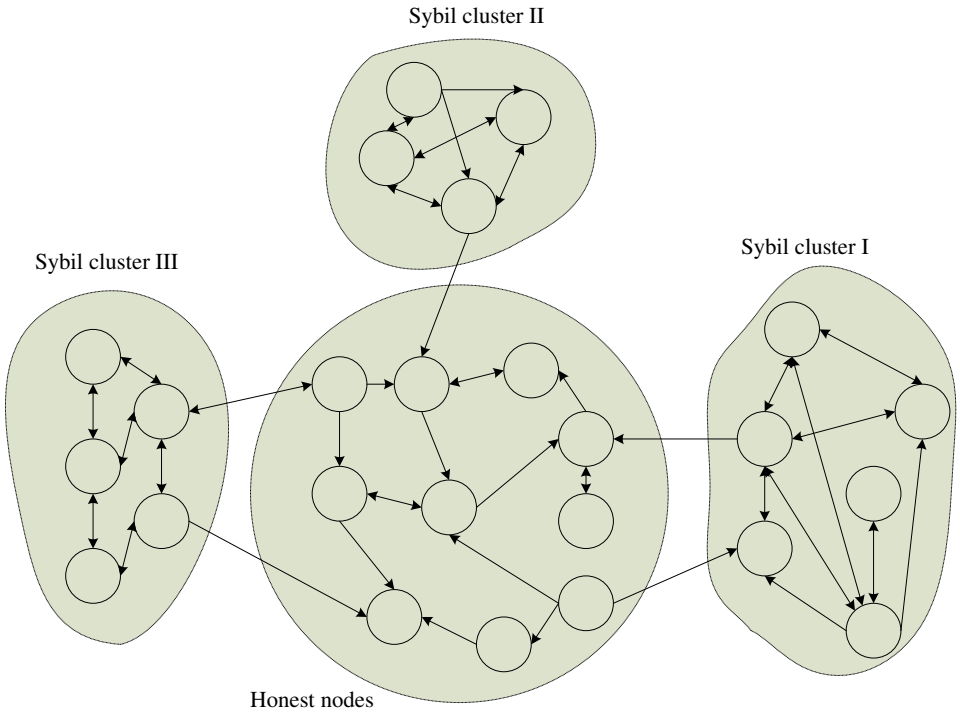


Figure 4.4 P2P network with three sybil clusters.

To gain influence over the P2P network, sybil peers aim to increase the number of attack edges. However, setting an edge might take time and be difficult. Therefore, it is considered that the feasible number of attack edges that can be built is small. Let $F(i)$ denote the set of trusters of peer i, and let $FJ(i,j)$ denote the set of peers that are common set trusters of both peer i and j. Therefore, $FJ(i,j) = F(i) \cap F(j)$. $FM(i)$ represent the trusters as a $1 \times N$ matrix in peer i, as $FM(i) = \{FM(i,1), \ \ldots, \ FM(i,N)\}$, where $FM(i,j) = 1$ if peer j is a truster of peer i and $FM(i,j) = 0$, otherwise. The total number of trusters of peer i is $TN[F(i)] = \sum_{j=1}^{N} FM(i,j)$. If $(FM(i)(j) + FM(k)(j)) = 2$, $\alpha(j) = 1$, otherwise $\alpha(j) = 0$. The total number of common trusters between peer i and peer k is $TN[FJ(i,j)] = \sum_{j=1}^{N} \alpha(j)$.

Figure 4.5 shows a sybil cluster I of Figure 4.4. The Figure shows six sybil peers. To raise the reputation of the pseudo peers in sybil clusters, the sybil peers create high trust values for each other, and exchange this information. As a result, the peers are almost fully connected.

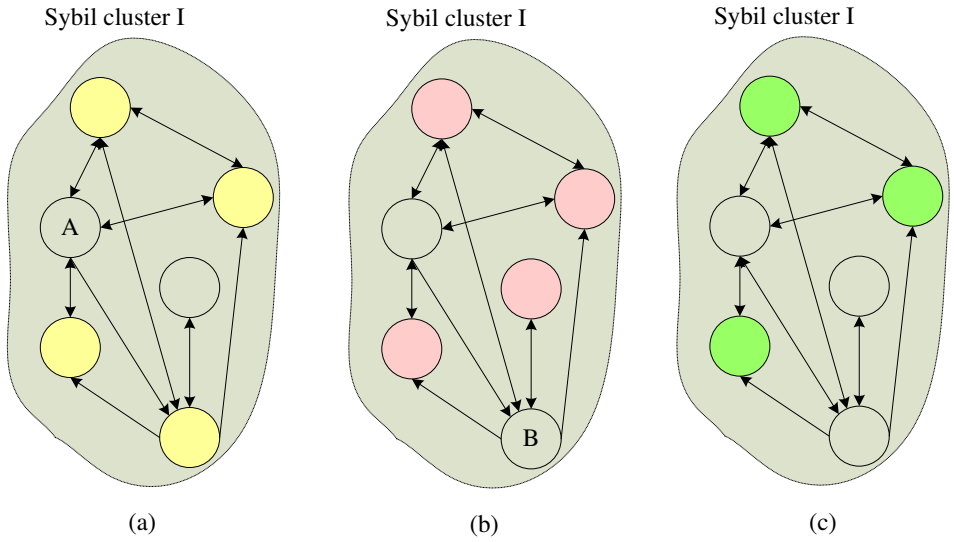


Figure 4.5 Trusters' overlap in sybil cluster.

The yellow peers in Figure 4.5(a) are the trustees of peer A, or $F(A)$, and $TN[F(A)] = 4$. The red peers in Figure 4.5(b) are the trustees of peer B, or $F(B)$, and $TN[F(B)] = 4$. Figure 4.5 shows the overlap of the trustees of peer A and B, $FJ(A, B)$ and $TN[FJ(A, B)] = 3$. In the future, if peer D claims to the system its trustees and it is found that some of the trustees are among the green peers $FJ(A, B)$ and $TN[FJ(A, B, D)] > TN_{thres}$, it is said that this peer is considered as sybil peers with higher probability, where TN_{thres} represents the threshold that determines a peer as a sybil. When the number of peers common to peer D and $FJ(A, B)$ is larger than this threshold, peer D is labeled as sybil. From this, it is seen that, the trusters between different sybil peers usually have many common trusters. The clustering algorithm relies on this property to identify possible sybil peers.

In the proposed Sybildefense framework, each peer has a local table (i.e., database) to store the list of trusters to be marked as possible sibyls peers, so they can be avoided. For peer i, peer j is viewed as a sybil if peer j provide an incomplete or a file with malware to peer i (or to one of its trusters peer k).

$F(j)$ is stored in the local table of peer i, with the format shown in Figure . Here, win_s is the size of the time window in which the rating messages are considered for determination of possible sybils, and N is the total number of peers in the network. The first row is the peer IDs. The last two columns, columns j and k, indicate that the message is sent from peer j to peer k. When $j = i$, it means that it is the local downloading experiences, and $F(k)$ is marked in local database as $T_v(i, k) * FM(k)$. If $j \neq i$, the rating message is sent from j to i pointing to k and the local database is updated as $T_v(i, j) * T_v(j, k) * FM(k)$. In the future, peer i will avoid to downloading files from peers whose local database value is larger than a threshold TN_{thres}. In this example, if $TN_{thres} = 3$, peer $(N - 2), (N - 1)$ and N is quarantined of being the downloading source.

Peer index / Time slot	1	2	· · ·	N-2	N-1	N	Sender j	Target k
t-wins	0	0	· · ·	0.1	0	0	3	5
t-wins+1	0	0.7	· · ·	0.7	0	0.7	N	N-1
t-wins+2	0	0.1	· · ·	0	0	0.1	7	16
t-wins+3	0.2	0	· · ·	0	0	0	N-1	21
⋮	⋮	⋮	⋮ ⋮ ⋮	⋮	⋮	⋮	⋮	⋮
t-4	0.2	0	· · ·	0	0	0.2	6	13
t-3	0	0	· · ·	0.2	0.2	0	7	4
t-2	0	0	· · ·	0	0	0	2	21
t-1	0	0.2	· · ·	0	0	0.2	17	11
t	0.2	0	· · ·	0.2	0	0.2	N	2
Overlap Indicator	1.9	2.7	· · ·	3.9	4.9	5.9		

Figure 4.6 Local database calculation.

4.3 Verification of Reported Transactions

In the trust management system, peer i sends a feedback about the latest transaction to its trustees after each download. Sybil peers recur to this method to post feedbacks among other sybils to increase their reputation and make them attractive to honest peers. However, sybil peers report not-occurred transactions as their objective is to increase their trust values without contributing to the P2P community.

Therefore, a countermeasure against inflating peer reputation is proposed. The scheme is based on providing verification of the reported transactions. In Sybildefense, it is considered not only the reputation and rating messages but also the set of the trustees of each peer as that may define the nature and risks for a peer.

Consider that peer i has a public key and a private key, namely, $< PK_i, RK_i >$, respectively. It is assumed that the public keys are tied to peers using digital certification authorities. Before peer i downloads file f from peer j, peer j sends a transaction guarantee certificate to peer i. This transaction guarantee can be expressed as:

$$TG(j,i) = PK_i\{RK_j\{time,\ f\}\}$$

where the symbol $a||b$ represents the concatenation of a and b. A rating message sent by peer i is accepted by peer j if and only if $TG(j,i)$ is also provided.

By using TG, sybil peers can not issue rating messages with forged ratings to the network.

The basis of this scheme is to generate a proof of an occurred transaction between peers i and j if and only if i and j in fact experienced a transaction with each other. If so, the transaction proof is sent from peer i to peer j after each download. The transaction proof, $TP(i,j)$, is defined as:

$$TP(i,j) = PK_j\{RK_i\{d, time, f, TG(j,i)\}\}. \tag{4.2}$$

where d is the description of a transactions, as satisfying or not satisfying, or 0 and 1, respectively. $TG(j,i)$ is embedded in $TP(i,j)$. With TG, peer j is able to prove that it actually uploaded f to peer i at time slot t. By using $TP(i,j)$, forged ratings can be avoided. The forged references remain between sybil peers.

4.4 Trust Management Scheme: Trust Values

The considered P2P network has n peers and s sybil peers. Each sybil peer has r replication peers, which can attack honest peers through collaboratively forged ratings. Each peer has three tables to maintain, trust table, propagated table and local database. Each peer generates a public/private key part and distributes the

public key. When a peer looks for a downloading source, the peer will make decision by the three tables to choose the downloading source. It is assumed that file identities remain unchanged for long periods of time.

The trust table in peer i is denoted as $T(i)$. The trust value of peer i on peer j, is denoted as $T_v(i,j)$,

$$T_v(i,j) = \frac{\text{number of satisfying downloads}}{\text{total number of downloads}} \tag{4.3}$$

where $T_v(i,j) \in [0,1]$. For example, $T_v(i,j) = 0$ means that peer i expects that any filed downloaded from j would be infected with probability 1.0. On the other hand $T_v(i,j) = 1$ means that peer i trusts peer j and any file downloaded from j is expected to be innocuous with probability 1.0. Therefore, peer j has top priority to become the downloading source. Peer i updates $T_v(i,j)$ in its trust table after downloading a file from peer j, according to the experienced download.

The second value in the trust table is the propagated value $P_v(i,j)$, is calculated from the opinions of trusters of peer i on peer j received through rating messages.

$$P_v(i,j) = \frac{\text{number of positive propagated ratings ratings}}{\text{total number of propagated ratings}} \tag{4.4}$$

where $P_v(i,j) \in [0,1]$. A positive rating is identified by a value 1, and a negative rating by a value of 0. A small $P_v(i,j)$ means a large probability that peer j is a sybil peer. The propagated ratings come from the local download history of those trusters. The total number of propagated ratings include both positive and negative ratings.

When peer i seeks f, it chooses a download source from one of the trusters that owns f. A conventional scheme selects the truster a peer whose $T_v(i,j)$ is larger than trust-value threshold t_h as the file source. As shown in Figure 4.7(a), the six peers above the slope are the trusters. However, two issues arise: 1) the users need to select a threshold value t_h for the management scheme and 2) the classification is not

accurate sometimes. To understand the latter issue, consider Figure 4.7(a), the grey circles correspond to five sybils peers and the light circles 11 honest peers. By using linear comparison, one misclassifies two sybils and seven honest individuals (circled in Figure 4.7(b)).

To correct this problem, the k-means clustering algorithm [55] is used with $T_v(i,j)$ and $P_v(i,j)$ as the x and y axis to group the peers. The red dotted line in Figure 4.7(a) is mapped to the vertical line in Figure 4.7(b). In general, the k-means clustering algorithm aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Given a set of observations $(x_1,\ x_2,\ \ldots,\ x_n)$, where each observation is a d-dimensional vector, the k-means clustering algorithm aims to partition the n observations into k sets $(k \leq n)$, $S = S_1,\ S_2,\ \ldots,\ S_k$ so as to minimize the sum of squares within a cluster :

$$arg_S min \sum_{i=0}^{k} \sum_{x_i \in S_i} \parallel (x_i - \mu_j) \parallel \tag{4.5}$$

where μ_i is the mean of points in S_i. In the proposed algorithm, k is equal to 2, in other words, two clusters are generated. Figure 4.7(b) shows the result of classification by the k-mean clustering algorithm.

Each peer tries to identify the honest-peer cluster and the sybil-peer cluster, each time slot. When peer i request file f, it will send request to the whole network with its private key as following:

$$RK_i\{i,\ f,\ rt\}. \tag{4.6}$$

where rt is the request time, which is the time the request for a file/download is issue. After receiving the request, the peers are able to encrypt the messages with the public key of peer i. The honest peers that have f send feedback information to peer i, including the trustee list and TP associated to the trustee list as a negative

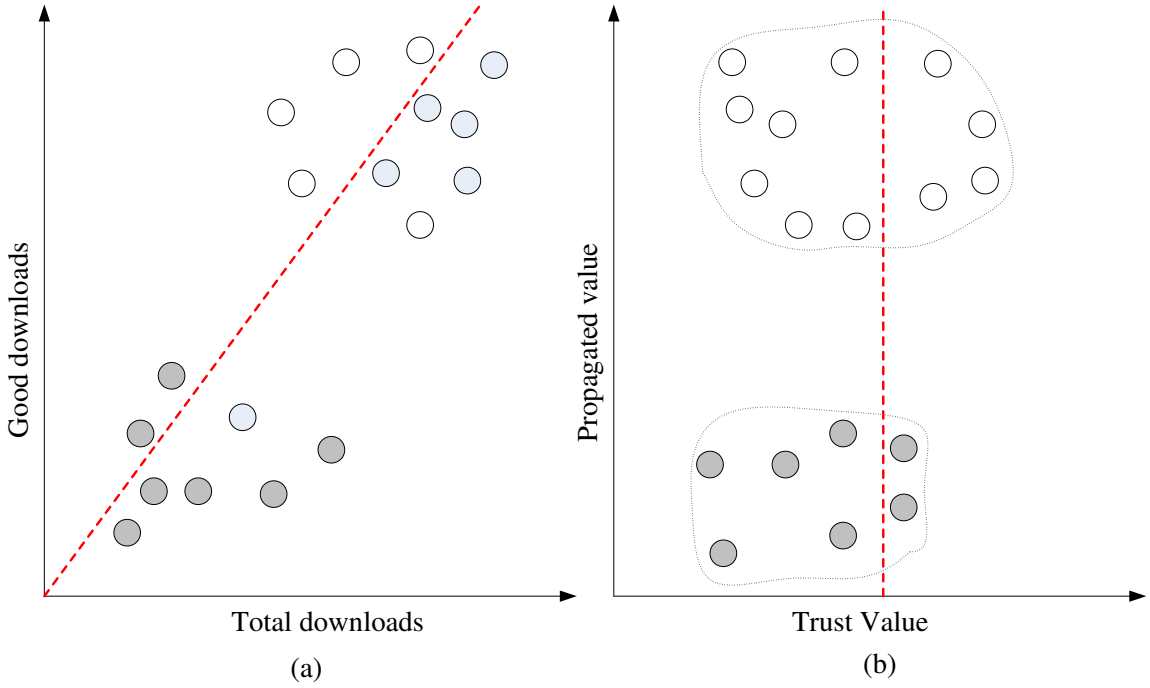


Figure 4.7 K-means clustering on truster classification.

rating message. After receiving the feedbacks, peer i eliminate from its candidates the peers who are not in its truster list. Peer i also eliminate as candidates those peers whose trustees are identified as possible sybils through the local table.

Peer i then chooses peer j who has the highest sum of squared values, $T_v^2(i,j) + P_v^2(i,j)$, as the download source. If peer j agrees to be the downloading source, peer j sends a transaction guarantee $TG(j,i)$ to peer i, and the upload begins in the next time slot.

After f is downloaded from peer j, if the download is satisfactory (i.e., a clean file and complete download), peer i sends a transaction proof TP to peer j, that can be used in future by peer j to prove its upload contribution to peer i. Simultaneously, $T_v(i,j)$ is updated and peer i propagates positive rating message to its trustee set, where the message has the following format:

$$RK_i\{ID_i,\ ID_j,\ \tau,\ TG(j,i)\} \tag{4.7}$$

where τ is the transaction description field, which indicates whether peer i is satisfied with the last download experience by using a flag to indicate either a positive or negative rating message. Peer k receives a propagated rating message from peer i is this peer is a truster of peer k, and then it proceeds to update P_v. The propagated message expires after a specified time, as indicated in the transaction-description field of the message. When peer k receives a message has exceed the expiration time slot, peer k discard the message directly.

After the download, if the download of file f is determined unsatisfied (e.g., detected as malware or the download fails), the transaction proof to peer j is suppressed and peer i records peer j into its local table and updates $T_v(i, j)$. Peer i propagates a negative rating message among its trustee list. The format of the negative rating message is as follows:

$$RK_i\{ID_i,\ ID_j,\ TG(j,i)\} \tag{4.8}$$

4.5 Performance Analysis

A P2P network is simulated by using a mesh topology, with 200 peers selected randomly as active peers in the mesh. There are five different groups of sybils clusters (each contains 11 peers) in the beginning. In the initial status, each honest peer perform clustering to identify honest peers from possible sybils by using the k-means clustering algorithm.

Sybil peers attack the network collaboratively. In the simulation, 30% of sybil peers send rating messages each time slot to raise the reputation of their central peer and their collaborator sybils. Since the sybil peers can not get transaction proofs TP from honest peers, each sybil peers randomly select 6 sybil peers from those in their cluster as their trustees and send forged rating messages to them.

The effectivity of the proposed framework is measured by estimating the number of peers that become compromised as sybil peers in the network. The simulation

results, under a large attack rate of 0.8, are shown in Figure 4.8. The attack rate is the probability that the sybil peers develop attacks, in the form of issuing bogus rating messages that inflate the trust values of other sybil peers, in a time slot. In this Figure, K indicates the use of the k-means clustering algorithm, T indicates the use of transaction proof, and L indicates the use of a local table. The Figure shows six curves, each represents a different combination of the Sybildefense mechanism. The Figure show the effect of using T,K,L separately, and from these, it can be observed the transaction verification scheme T is the mechanism that has the major impact on the efficiency of Sybildefense. This is because the effect of the local table and and k-means clustering schemes are used to determine sybil candidates but it is no guarantee that they actually are sybils. These two mechanism reduce the probability of selecting a sybil as a download source. However, the transaction proof mechanism is a direct countermeasure to the inflation of reputation that sybil peers pursue, and therefore, this mechanism has the largest countermeasure effect.

This Figure also shows that when these three mechanisms are combined, the efficiency of Sybildefense is very high. The results show that the proposed scheme can suppress the sybil attacks efficiently, as the number of honest peers remains at 184, showing that the number of compromised peers is small and it remains at this value indefinitely.

Figure 4.9 shows the same scenarios discussed in Figure 4.8, under, however, a small rate attack of 0.2. This Figure shows the larger effect of the Sybildefense mechanisms under this low-intensivity attack.

Figure 4.10 shows the failure ratio of the k-means clustering methods under local table (L), the transaction (T), and the combination of the two ($L + T$) . The failure ratio is calculated as the number of sybil peers in the truster list divided by the total number of trusters. The failure ratio is calculated in the honest peers only which means that among the two hundred honest peers. From this Figure, it is seen

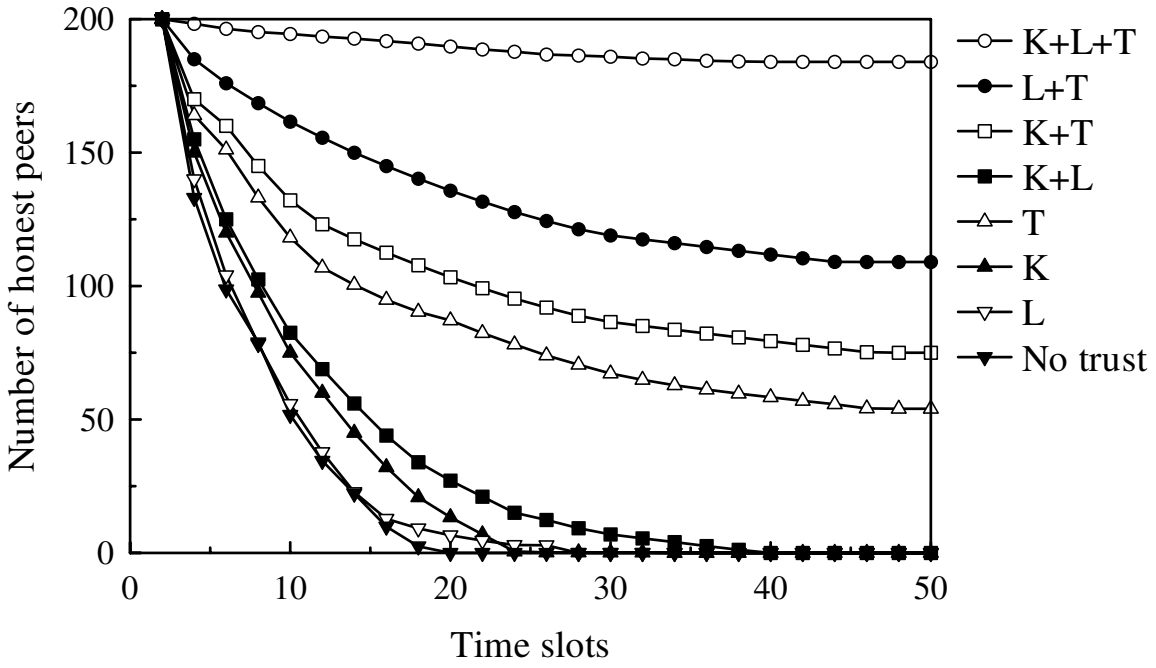


Figure 4.8 Comparison of different the Sybildefense's mechanisms under attack rate 0.8.

that with the using of local table and the transaction verification mechanisms, the ratio false positives is reduced.

4.6 Conclusions

Trust management is a strategy to determine the reputation of peers by determining the level of trustability. However, trust management alone may not be efficiently determine the level of threat of peers when a network is under sybil attacks, as these attacks attempt to undermine peer reputations. Therefore, a framework is proposed for defending peers in a P2P network from sybil attacks to support trust management schemes. The proposed algorithm guarantees that a P2P network may not be compromised under sybil attacks. With the proposed framework, an honest peer can successfully contribute and obtain services other honest peers in the network. The framework consist of three mechanisms, a local table to determine

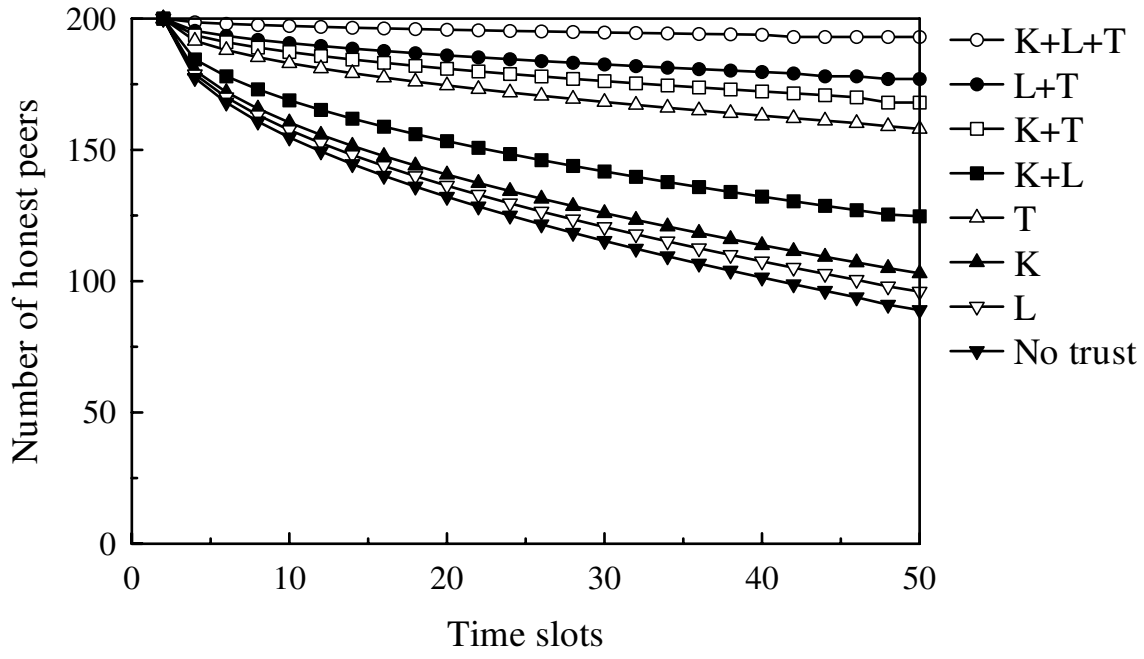


Figure 4.9 Comparison of the different Sybildefense's mechanisms under attack rate of 0.2.

different collaborating groups that could identify sybil cluster, a k-means mechanism to determine which one of the cluster could be a sybil candidate, and a transaction verification mechanism to undermine bogus transaction reports.

Since the attackers are forced to build up trust before effectively launching attacks, the algorithm is designed to mitigate the sybils attack in an anonymous and decentralized fashion.

The presented simulation results show the affectivity of the proposed framework, which limits the number of compromised peer to a small number.

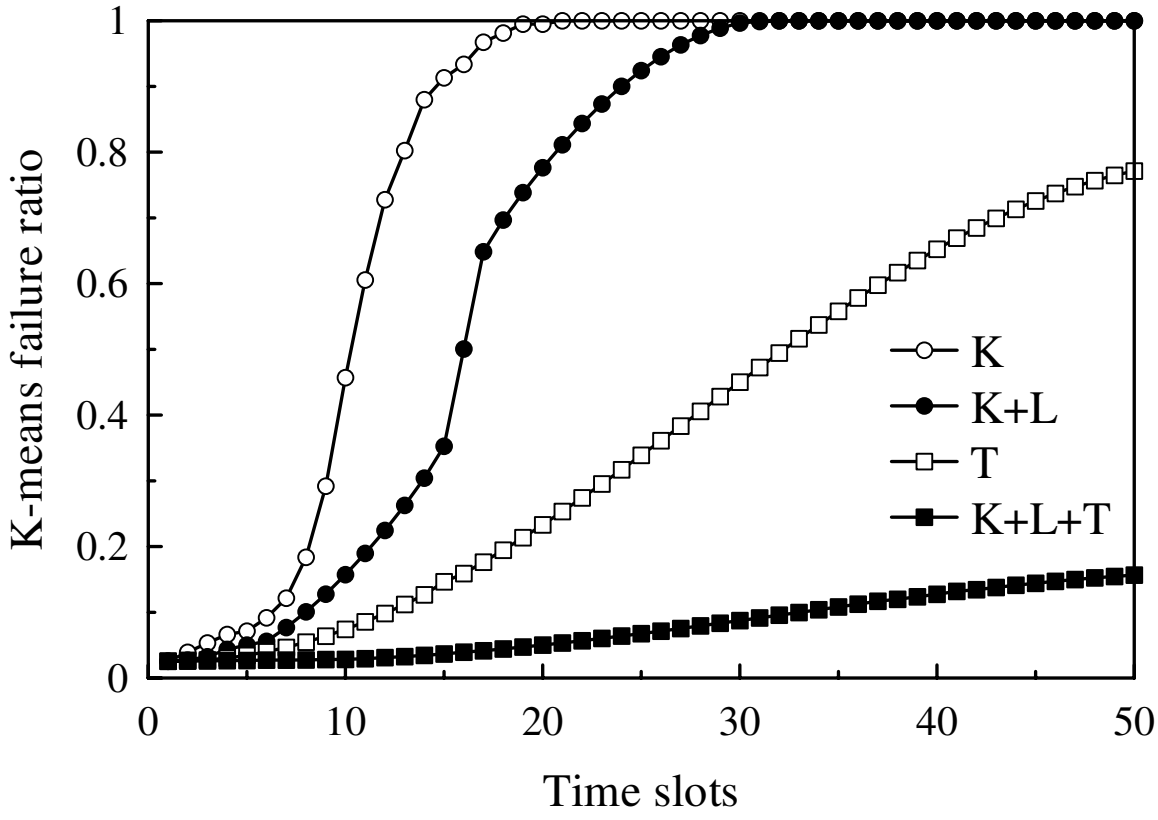


Figure 4.10 K-means clustering failure ratio.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Trust management is a promising strategy to bound the proliferation of malware on peer-to-peer networks that can work jointly with virus detection systems. However, trust management alone can not defense sybil attacks. An asynchronous reputation system is designed in defending two primary attacking developed by sybil peers. The proposed algorithm guarantees that an honest peer accepts, and is also accepted by, most other honest peers with high possibility. An honest peer can successfully obtain service from and provide service to most other honest peers. This algorithm enables a peer to partition the peers into two groups, the honest peers groups and sybil peers groups.

Trust management is a strategy to determine the reputation of peers by determining the level of trustability. However, trust management alone may not be efficiency determine the level of threat of peers when a network is under sybil attacks, as these attacks attempt to undermine peer reputations. Therefore, a framework is proposed for defending peers in a P2P network from sybil attacks to support trust management schemes. The proposed algorithm guarantees that a P2P network may not be compromised under sybil attacks. With the proposed Sybildefense framework, an honest peer can successfully contribute and obtain services other honest peers in the network. The framework consist of three mechanisms, a local table to determine different collaborating groups that could identify sybil cluster, a k-means mechanism to determine which one of the cluster could be a sybil candidate, and a transaction verification mechanism to undermine bogus transaction reports.

Since the attackers are forced to build up trust before effectively launching attacks. The algorithm is designed to mitigate the sybil attack in an anonymous

and decentralized fashion. The presented simulation results show the effectivity of the proposed framework, which limits the number of compromised peer to a small number.

REFERENCES

[1] B.F. Cooper and H.G. Molina, "Peer to peer data trading to preserve information," *ACM Transactions on Information Systems*, Vol. 20, pp. 133-170, New York, NY, April, 2002.

[2] B. Horne, B. Pinkas and T. Sander, "Escrow services and incentives in peer-to-peer networks," *Proc. 3rd ACM Conference on Electronic Commerce*, pp. 85-94, Tampa, Florida, October, 2002.

[3] B. Yang, and H.G. Molina, "Ppay: Micropayments for peer-to-peer systems," *Proc. 10th ACM Conference on Computer and Communications Security*, pp. 300-310, New York, NY, 2003.

[4] X. Xu, Y. Wang, S. P. Panwar and K. W. Ross, "A Peer-to-Peer Video-on-Demand System using Multiple Description Coding and Server Diversity," *Proc. IEEE International Conference on Image Processing*, Vol. 3, pp. 1759-1762, Brooklyn, NY, October, 2004.

[5] X. Hei, C. Liang, Y. Liu and K.W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Trans. on Multimedia*, Vol. 9, Issue 8, December, 2007.

[6] M. Macedonian, "Distributed File Sharing: Barbarians at the Gate?," *IEEE Computer*, Vol. 33, Issue 8, pp. 99-101, August, 2000.

[7] Y. Wang, X. Yun and Y. Li, "Analyzing the Characteristics of Gnutella Overlays," *Proc. IEEE IV International Conference in Information Technology*, pp. 1095-1100, Las Vegas, NV, April, 2007.

[8] R. Matei, A. Iamnitchi, and P. Foster, "Mapping the Gnutella network," *Internet Computing, IEEE.*, Vol. 6, pp. 50-57, 2002.

[9] O.K. Won, S. Lee and J. Kim, "FileTrust: Reputation Management for Reliable Resource Sharing in Structured Peer-to-Peer Networks," *IEICE Transaction on Communications*, pp. 826-835, April, 2007.

[10] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Oper. Syst. Rev.*, Vol. 36, pp. 299-314, 2002.

[11] K. Hwang, M. Cai, S. Song and Y. Chen, "DHT-based Security Infrastructure for Trusted Internet and Grid Computing," *International Journal of Critical Infrastructures*, Vol. 2, pp. 654-662, November, 2006.

[12] C. LesniewskiLaas, "A Sybil-proof one-hop DHT," *SocialNets: Proceedings of the 1st Workshop on Social Network Systems*, New York, NY, pp. 19-24, December, 2008.

[13] X. Zhang, D. Saha and H.H. Chen, "Analysis of virus and anti-virus spreading dynamics," *IEEE Global Telecommunications Conference*, pp. 1822-1826, St. Louis, MO, November, 2005.

[14] L.C. Chen and K.M. Carley, "The Impact of Countermeasure Propagation on the Prevalence of Computer Viruses," *IEEE Trans. on System, Man, and Cibernetics*, Vol. 34, pp. 823-833, April, 2004.

[15] P. Dhungel, X. Hei, K.W. Ross and N. Saxena, "The pollution attack in P2P live video streaming: measurement results and defenses," *Proc. Sigcomm P2P-TV Workshop*, pp. 987-993, Kyoto, Japan, 2007.

[16] R. Kumar, A. Bagchi, K.W. Ross and D. Rubenstein, "Fluid modeling of pollution proliferation in P2P networks," *Performance Evaluation Review*, Vol. 34, pp. 335-346, New York, NY, June, 2006.

[17] X. Dong, W. Yu and Y. Pan, "A Dynamic Trust Management Scheme to Mitigate Malware Proliferation in P2P network.," *Proc. IEEE International Conference on Communications*, pp. 1605-1609, Beijing, China, May, 2008.

[18] S. Marti and H.G. Molina, "Limited Reputation Sharing in P2P Systems," *Proc. of the 5th ACM Conference on Electronic commerce*, pp. 91-101, New York, NY, March, 2004.

[19] P. Resnick and R. Zeckhauser, "Trust Among Strangers in Internet, Empirical Analysis of eBay's Reputation System," *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce*, Vol. 11, pp. 127-157, 2002.

[20] S.D. Kamvar, M.T. Schlosser and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," *Proc. 12th International World Wide Web Conference*, pp. 785-791, Budapest, HUNGARY, May, 2003.

[21] P. Herrmann, "Trust-Based Procurement Support for Software," *Proc. 4th International Conference of Electronic Commerce Research*, pp. 505-514, Dallas, Texas, November, 2001.

[22] E.K. Lua, J. Crowcroft, R. Sharma and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Comm. Survey and Tutorial*, Vol. 7, pp. 72-93, 2005.

[23] K. Walsh and E.G. Sirer, "Fighting Peer-to-Peer SPAM and Decoys with Object Reputation," *Proc. Third Workshop on the Economics of Peer-to-Peer Systems*, pp. 138-143, New York, NY, 2005.

[24] A. Cheng and E. Friedman, "Sybilproof Reputation Mechanisms," *Proceedings of the ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pp. 128-132, Philadelphia, Pennsylvania, August, 2005.

[25] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis and Defences," *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 259-268, April, 2004.

[26] B. Neil Levine, C. Shields, and N.B. Margolin, "A Survey of Solutions to the Sybil Attack," *Tech. rep. University of Massachusetts*, October, 2006.

[27] A. Josang, "Challengs for Robust Trust and Reputation Systems," *in Proceedings of the 5th International Workshop on Security and Trust Management*, Saint Malo, France, September, 2009.

[28] D. Quercia, and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," *INFOCOM*, pp. 1-5, San Diego, CA, March, 2010.

[29] B. Xiao, B. Yu, and C. Gao, "Detection and localization of Sybil peers in VANET," *Proceedings of the workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pp. 72-93, 2006.

[30] G. Guette, and B. Ducourthial, "On the Sybil attack detection in VANET," *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007.

[31] T. Zhou, R. Choudhury, P. Ning and K. Chakrabarty, "Privacy-Preserving Detection of Sybil Attacks in vehicular ad hoc network," *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, 2007.

[32] N. Margolin, and B. N. Levine, "Quantifying Resistance to the Sybil Attack," *Financial Cryptography and Data Security*, 2008.

[33] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks." Proceedings of ACM SIGCOMM Conference, pp. 267-278, New York, NY, September 2006.

[34] A. Cheng and E. Friedman, "Manipulability of PageRank under Sybil strategies," *First Workshop on the Economics of Networked Systems*, 2006.

[35] R. Zhou and K. Hwang, "Trust overlay networks for global reputation aggregation in P2P grid computing," *Parallel and Distributed Processing Symposium*, pp. 311-335, 2006.

[36] S. Song, K. Hwang and Y.K. Kwok, "Trusted Grid Computing with Security Binding and Trust Integration," *Journal of Grid Computing*, Vol. 3, pp. 127-157, June, 2005.

[37] G. Theodorakopoulos and J.S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 24, pp. 318-328, Feburary, 2006.

[38] P. Li, Z. Wang and X. Tan, "Characteristic Analysis of Virus Spreading in Ad Hoc Networks," *Proc. IEEE Workshop in Computationall Intelligence and Security*, pp. 538-541, December, 2007.

[39] F. Li and J. Wu, "Mobility reduces uncertainty in MANETs," *Proceedings of IEEE INFOCOM*, 2007.

[40] Y. Wang, and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks," *Third International Conference on Peer-to-Peer Computing*, 2003.

[41] S. Song, K. Hwang, R. Zhou, and Y.K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *Internet Computing, IEEE*, Vol. 9, pp. 24-34, November, 2005.

[42] T. Dimitriou, G. Karame, and I. Christou, "SuperTrust: A secure and efficient framework for handling trust in super peer networks," *Proceedings of ACM PODC*, 2007.

[43] R. Morselli, J. Katz, and B. Bhattacharjee, "A game-theoretic framework for analyzing trust-inference protocols," *Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[44] S. Ba, and P. Pavlou, "Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior," *MIS Quarterly*, Vol. 26, pp. 243-268, 2002.

[45] R. Aringhieri, E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati, "Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems," *J. Am. Soc. Inf. Sci. Technol.*, Vol. 57, pp. 528-537, 2006.

[46] M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks," *Proceedings of the 14th international conference on World Wide Web*, pp. 422-431, 2005.

[47] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks," *Third European Symposium on Research in Computer Security*, 1994.

[48] R. Levien, "Attack Resistant Trust Metrics," *PhD thesis, University of California at Berkeley*, 2003.

[49] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," *Proceedings of the tenth international conference on Information and knowledge management*, pp. 310-317, 2001.

[50] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, Vol. 43, pp. 618-644, May, 2007.

[51] A. Singh, and L. Liu, "TrustMe: anonymous management of trust relationships in decentralized P2P systems," *Third International Conference on Peer-to-Peer Computing*, pp. 142-149, September, 2003.

[52] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Comm. Survey and Tutorial*, Vol. 7, Isssue 2, pp. 72-93, 2005.

[53] G.J. Foschini and M.J. Gans, "On limits of wireless communications in a fading environment," *Wireless Pers. Commun.*, Vol. 6, pp. 311-335, 1998.

[54] K. Hoffman, D. Zage, and C. Nitarotaru, "A survery of attack and defense technique for reputation systems," *ACM Computing Surveys (CSUR) Surveys*, Vol. 42, Issue 1, December, 2009.

[55] J.B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297 , 2009.

[56] S. Buchegger, and J. Y. Le Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[57] L. Xiong, L. Liu, and M. Ahamad, "Countering sparsity and vulnerabilities in reputation systems," *Tech. Rep. TR-2005-017-A Emory University*, 2005.

[58] A. Nandi, T.W. Ngan, A. Singh, P. Druschel, and D. S. Wallach, "Scrivener: Providing incentives in cooperative content distribution systems," *ACM/IFIP/USENIX 6th International Middleware Conference*, Vol. 1, pp. 270?C291, Springer Berlin, Heidelberg, November, 2005.

[59] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 207-216, New York, NY, 2002.

[60] A. Altman and M. Tennenholtz, "On the axiomatic foundations of ranking systems," *Proc. 19th International Joint Conference on Artificial Intelligence*, Vol. 6, pp. 917-922, 2005.

[61] K. Walsh, and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," *Symposium on Networked System Design and Implementation (NSDI)*, May, 1998.

[62] M. Ham, and G. Agha, "ARA: a robust audit to prevent free-riding in P2P networks," *Peer-to-Peer Computing*, pp. 125-132, August, 2005.

[63] S. Lee, R. Sherwood, and B. Bhattacharjee, "Cooperative peer groups in nice," *IEEE Infocom.*, April, 2003.

[64] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," *Proceedings of the 5th ACM conference on Electronic commerce*, Vol. 1, pp. 102-111, 2004.

[65] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker,"A scalable content addressable network," *Tech. Rep. TR-00-010, UC Berkeley*, 2000.

[66] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *Proceedings of the 2001 ACM SIGCOMM Conference*, pp. 149-160, 2001.

[67] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," *Tech. Rep. UCB/CSD-01-1141, UC Berkeley*, April, 2001.

[68] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," *ACM International Conference on Distributed Systems Platforms*, Vol. 11, pp. 329-350, 2001.

[69] S. Dahan and M. Sato , "Survey of six myths and oversights about distributed hash tables' security," *Distributed Computing Systems Workshops, 27th International Conference on, IEEE Computer Society*, 2007.

[70] P. Eugster, S. Handurukande, R. Guerraoui, A.M. Kermarrec, and P. Kouznetsov, "Lightweight probabilistic broadcast ," *The International Conference on Dependable Systems and Networks*, 2001.

[71] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 8, pp. 300-304, 1960.

[72] Q. Lian, Z. Zhang, M. Yang, B. Zhao, Y. Dai, and X. Li, "An empirical study of collusion behavior in the Maze P2P file-sharing system," *Distributed Computing Systems. ICDCS. 27th International Conference*, 2007.

[73] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, "Introduction to Algorithms," *MIT Press*, 2001.

[74] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for cooperation in peer-to-peer networks," *in Workshop on Economics of Peer-to-Peer Systems*, 2003.

[75] E. J. Friedman and P. Resnick, "The social cost of cheap pseudonyms," *Economics and Management Strategy*, Vol. 10, pp. 173-199, 2001.

[76] R. A. Bazzi, and G. Konjevod, "On the establishment of distinct identities in overlay networks," *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, Vol. 6, pp. 312-320, 2005.

[77] D. J. Zage and C. Nita-Rotaru, "On the accuracy of decentralized network coordinates in adversarial networks," *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007.

[78] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "A near-optimal social network defense against sybil attacks," *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008.

[79] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," *Stanford Digital Library Technologies Project*, 1998.

[80] P. Flocchini, A. Nayak, and M. Xie, "Enhancing peer-to-peer systems through redundancy," *Selected Areas in Communications, IEEE Journal on*, Vol. 25, pp. 15-24, 2007.

[81] P. Michiardi and R. Molva, "CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pp. 107?C121, 2002.

[82] B. Yu and M. P. Singh, "A social mechanism of reputation management in electronic communities," *Proceedings of the 4th International Workshop on Cooperative Information Agents IV*, Vol. 1, pp. 154-165, 2000.

[83] E. Damiani, S. De Capitani Di Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servants reputations in p2p systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, pp. 840-854, 2003.

[84] L. D. Zhou, L. T. Zhang, F. Mcsherry, N. Immorlica, M. Costa, and S. Chien, "A first look at peer-to-peer worms: threats and defenses," *Proceedings of the 4-th International Workshop on Peer-To-Peer Systems*, Vol. 6, pp. 311-335, 2005.

[85] W. Yu, C. Boyer, S. Chellappan, and D. Xuan, "Peer-to-peer system based active wor attacks: modeling and analysis," *Proceedings of the IEEE International Conference on Communications*, 2005.

[86] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "The pollution attack in p2p live video streaming: measurement results and defenses," *Proceedings of Sigcomm P2P-TV Workshop*, 2007.

[87] R. Thommes and M. Coates, "Epidemiological modeling of peer-to-peer viruses and pollution," *Proceedings of IEEE INFOCOM*, 2006.

[88] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," *Proceedings of IEEE International Confernece on Multimedia Computing and Networking*, 2002.

[89] E. Friedman, P. Resnick, and R. Sami, "Algorithmic Game Theory," *Cambridge University Press*, 2007.

[90] C. Dellarocas, "The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms," *Management Science*, Vol. 49, pp. 1407-1424, October, 2003.

[91] J. Shin, T. Kim, Taehoon, and S. Tak, "A Reputation Management Scheme Improving the Trustworthiness of P2P Networks," *Proc. IEEE International Conference on Convergence and Hybrid Information Technology*, pp. 92-97, Daejeon, Korea, May, 2008.

[92] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent?," *Proceedings of the Fourth USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[93] T. Khopkar, X. Li, and P. Resnick, "Self-selection, slipping, salvaging, slacking, and stoning: the impacts of negative feedback at eBay," *Proceedings of the 6th ACM conference on Electronic commerce*, pp. 223-231, 2005.

[94] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 50, pp. 472-484, March, 2006.

[95] E. Damiani, D.C. Vimercati, S. Paraboschi and F. Violante, "A Reputation-based Approach for choosing Reliable Resources in Peer-to-Peer Networks," *Proc. of the 9th ACM conference on Computer and communications security*, pp. 207-216, Chicago, IL, October, 2002.