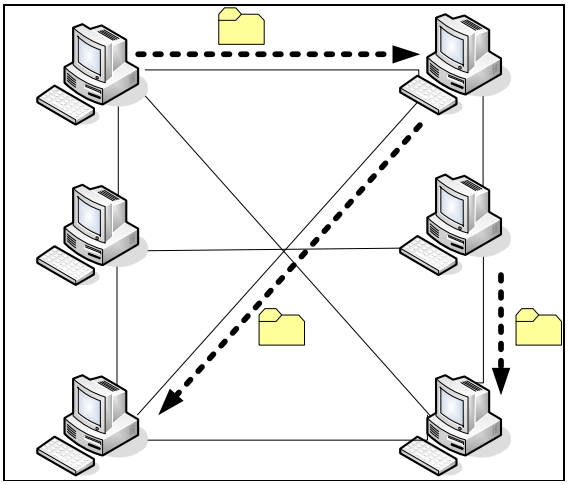
TU Delft

Master Thesis

Trust Algorithm in Files Sharing P2P Network



Indira Nurtanti
Student Number: 1119230
Chairman: Jan van den Berg
First Supervisor: Semir Daskapan
Second Supervisor: Ana Cristina da Costa
Faculty of System Engineering, Policy Analysis and Management
Delft University of Technology
Delft, the Netherlands
Email: I.Nurtanti@student.tudelft.nl

Executive Summary

The rapid growth of the internet technology that is spurred by peer to peer network (p2p network) has created many developments and innovations. This p2p revolution is started when the software for files sharing was introduced among the internet users in the 90s. By downloading this software, the users are able to join a p2p network and connect directly with other users through the internet to share various types of files. In a p2p network, the peers act both as client and server in the network in order to provide service and content that is offered by the network such as file sharing activity (Kellerer, 1998). Thus, in p2p network the peers represent the users behind the computers with their wishes and expectations from the network, which basically are to acquire the files that they are interested in. Moreover, these peers expect not only to receive the good files (good quality files and free from virus) but also a simple and fast process to get the files. In order to achieve these goals, a peer who wants a certain file must know how and where to get it. In other word a peer must choose from a peer that it trusts can deliver a good result which means that it has to find a trustworthy peer. Several trust calculation schemes have been proposed in the past few years. The scheme, known as trust algorithm, is a system where trust is defined and computed based on some trust factors. We come across several algorithms that correspond with this research study. These current algorithms are evaluated based on literature study and the simulation tool. Based on the evaluation, the algorithms have shortcomings in the trust calculation process. Therefore, the research question is formulated as follows:

"Which improvements can be made to the trust algorithm for measuring trust value in file sharing p2p network which in turn will help the peers in p2p network for a better selection mechanism i.e. choosing the right peer during file sharing activity?"

The improvements are used as the inputs for designing a new algorithm which is given the name Hybrid algorithm. To guide the evaluation process there is a list of design requirements for trust algorithm that has been developed. This list can be stated as follows:

Trust algorithm should: incorporate non-symmetric element, incorporate conditional transitive element, incorporate contextual element, incorporate reflexive element, incorporate dynamic element, incorporate self-regulated system to cope with network scalability, maintain the peer anonymity, enforce minimum load through the whole process in order to improve network and peer performance, implement a user intervention system and include pre-trusted peer recommendation in the trust value calculation process.

Based on the evaluation result, the Hybrid algorithm is designed and its performance is compared with the performance of the current algorithms. The Hybrid algorithm has the best performance in most of the design requirements. The basic conclusion of the research study is the contribution of Hybrid algorithm that lies on the combination between implementing a user intervention system and the usage of pre-trusted peer experience in the trust calculation process. Some current algorithms depend completely on pre-trusted peer while others ignore their presence. On the other hand, almost the entire current algorithm is automatically set up by the network which does not allow its users to find their best preferences. The Hybrid algorithm balances these two features by combining them together to improve the algorithm's performance.

Preface

The rapid development of file sharing p2p network application has always amazed me as the regular user of this technology. Therefore, it is a privilege for me to be able to have a chance to do a research study on this field. I can only hope what I have done can contribute something valuable for the future development.

Firstly, I would like to pay my respect to our dear departed Prof. Dr.Ir. Renee Wagenaar which has given me a kick off start to officially begin this research. Even though it was such a short time, but his input has been always valuable to me.

Most of all, I would like to say my biggest gratitude to my supervisor and my guidance during this research, Semir Daskapan; that amazingly not only always has the patience to bear with all my missing deadlines, my clumsiness and my irregular working rhythm during the research process but also still manage to give me a pep-talk every time I feel that I was about to hit a wall. I will always highly appreciate it all.

Many many thanks also for my other supervisors in this thesis project Jan van den Berg and Anna Cristina Costa. Even though I have only limited time to discuss my project with them, it never becomes a problem to them to always give jack-pot comments about my thesis.

I also never forget all of my friends: Aldo my private technical supervisor, Kum and Bollie my (extended) housemates, Deasy, Leonie, Daniel, Henny, Kaewta, Steven, Faber, Edison and many more that I am sure I forget to mention here, who stick with me through all this time. Thank you for all the help and support guys.

Last but not least, I want to dedicate this thesis to my parents. The people who I care and love the most. Thank you for all of your days and nights prays for me. I hope I don't disappoint you.

And thank you GOD for giving me a special place among these lovely people.

Delft, 21 December 2007

Table of Content

List of Figure

List of Tables

Chapter 1

Introduction

1.1 Introduction to Trust and File Sharing P2P Network

The rapid growth of the internet technology that is spurred by peer to peer network (p2p network) has created many developments and innovations. This p2p revolution is started when software for sharing files was introduced among the internet users in the 90s. By downloading this software, the users are able to join a p2p network and connect directly with other users through the internet to share various types of files. The first file sharing p2p software application was initiated by Shawn Fanning who founded Napster in 1999 (Oram, 2001). This software application has shown that simple personal home computers could perform tasks more than browsing websites and exchanging emails. Through this software, they are able to form groups in large scale and collaborate to become user-created search engines, file systems and virtual supercomputer (Minar and Hedlund, 2001). Examples of file sharing p2p applications (i.e. successor of Napster) are KaZaa, Gnutella, LimeWire and Torrent.

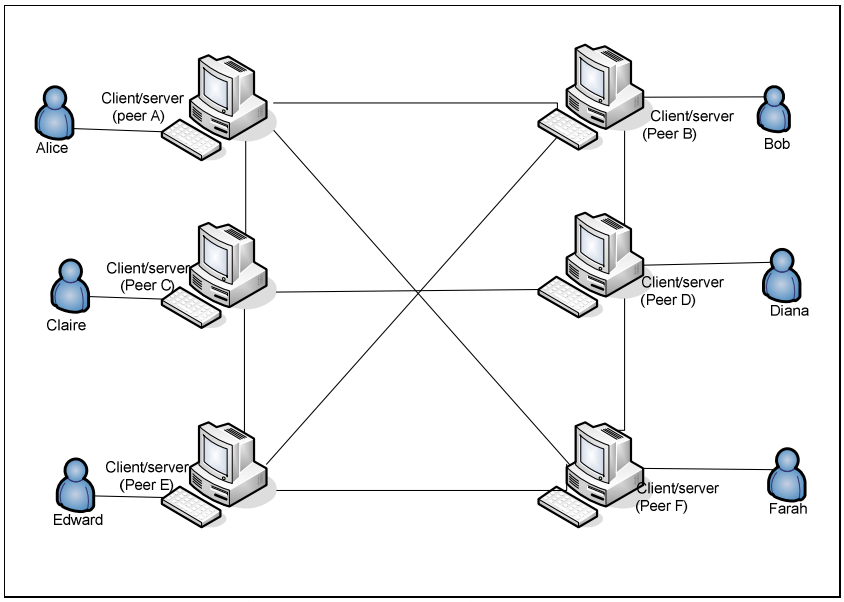


Figure 1. 1 Topology of P2P Network

Figure 1.1 displays the topology of p2p network which shows a group of peers that is connected to one another as a part of the network. These peers act both as client and server in the network in order to provide service and content that is offered by the network such as file sharing (Kellerer, 1998). The peers also represent the users behind the computers with their wishes and expectations from the network, which basically are to acquire the files that they are interested in. Moreover, these peers expect not only to receive the good files (good quality files and free from virus) but also simple and fast process getting the files. In order to achieve these goals, a peer who wants a certain file must know how and where to get it. The activity starts when, for example, Alice joins a file sharing p2p network after installing a file sharing software application and wants to have a certain file x. She does not know which peers have the file that she wants. Even if there are several peers who have the x file, Alice will not know which one to choose.

In order to receive a good file, Alice must choose from a source peer that she trusts can deliver a good result, or in other word she has to find a trustworthy peer. The problem with p2p network is that the peers within the network are anonymous. If in the daily life Alice knows and trusts Bob, she will not know that Bob is actually peer B and vice versa, as shown in figure 1.2.

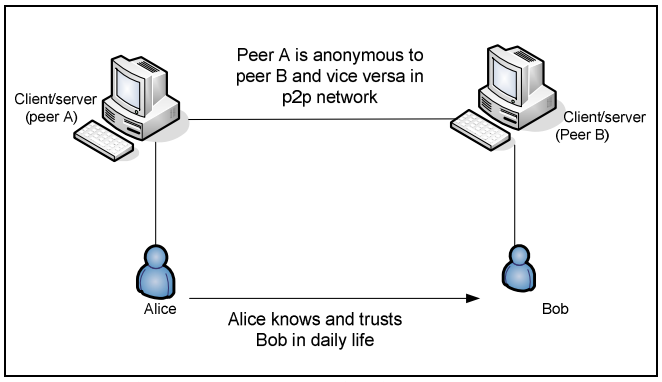


Figure 1.2 Trust Relation in P2P Network

Thus, the challenge here is to develop a mechanism that allows Alice (and other users) to find a trustworthy peer within the network.

1.2 Research Problem

Trust issue is one of many subjects that we are facing in the daily life. As part of the social community, we have to communicate or make transaction with other community member. Some of them are the people that we are familiar with, but often we are dealing with less familiar people or even with complete strangers. If we are dealing with familiar people that we already have experiences with, it is easier for us to decide whether we could trust them with some information or conduct business with them. But if we are dealing with unfamiliar people we might trust them less and therefore we are not always willing to exchange information or conduct business with them. Diego Gambetta (1990) defines trust as:

"A particular level of the subjective probability with which an agent will perform a particular action, both before (we) can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects (our) own action"

Based on the definition, Abdul-Rahman and Hailes (1997) conclude several notions about trust: trust is (1) subjective, (2) affected by action that we can not monitor, (3) depending on how our actions are in turn affected by the agent's action and (4) based on a specific context.
Thus, referring back to figure 1.2, how Alice trusts Bob is related to few factors such as:

- How much Alice trusts Bob is not the same with how much Claire trusts Bob.
- How much Alice trusts Bob is affected by Bob's action that can not always be monitored by Bob.
- How much Alice trusts Bob is depending on how Bob's action is in turn affected by Alice's action.
- Alice trusts Bob for a specific subject, for example: Alice trusts Bob for sharing movie files but not for sharing music files.

These factors characterize the trust relationship between Alice and Bob in the daily life. In the p2p network activity, the peers who represents Alice and Bob (and other users as well)

encounter the similar situation where the peers have to communicate and share files among one another. Often, these peers are strangers to one another, in the sense that most of them have never connected before (Garcia-Molina, 2003). Naturally, in p2p network, not all of the peers are good peers (which provide good and honest service), some of them are malicious peers with the bad intention towards other peers and the network (i.e. spreading virus, creating unnecessary traffic, creating a network failure, etc). Peer A (Alice) has to be able to distinguish whether peer B (Bob) or other peers in the network are good peers or malicious peers in order to establish a trust relationship between them. If the daily life, trust is expressed with Alice's feeling toward Bob, the p2p network has to transfer the "trust feeling" into a value that can be calculated by the peers' computers. Several trust calculation schemes have been proposed in the past few years. The scheme, known as trust algorithm, is a system where trust is defined and computed based on some trust factors. The outcome of the computation process is expressed in number and labelled as trust value (which describes peer's trustworthiness level). A number of algorithms are already proposed to address the trust problems within the file sharing p2p networks. Based on the literature study, we come across several algorithms that correspond with this research. These algorithms are all applicable for p2p network but using various theory foundations.

Some algorithms such as Marsh Algorithm (1994), Abdul-Rahman and Hailes Algorithm (1997) and Jin et al. Algorithm (2005) are relying on so-called central peer that manages the data for trust computation within the network. Algorithm Aberer and Despotovic (2001) is an example of a decentralised algorithm, where there is no central peer needed. This algorithm relies on complaints from other peers which make this algorithm very sensitive to a situation where low number of transactions occurs among peers within the network. Furthermore, there are also some algorithms that are based on global trust value (one global trust value for each peer for the entire network) and local trust value computation (every single peer is able to compute other peer's trust value which results in various local trust value for each peer in the network). The drawback of global trust algorithm is that there is no differentiation between various trust elements which can result in a less accurate outcome of trust value. On the other hand, local trust algorithm requires large capacity as every peer will have to acquire its own computation data that can result in large overhead of data management which limits its application on real p2p network. Based on the information of the current algorithms, the research problem can be stated as:

> ***"Based on the literature evaluation, the current trust algorithms for file sharing p2p network have a number of shortcomings that can limit their trust value calculations"***

Based on this research problem, the research objective will be developed as follows:

> ***"To develop a trust algorithm that can improve the result of the trust value calculation which in turn will help the peers in p2p network for a better selection mechanism i.e. choosing the right peer during file sharing activity "***

Based on the research problem and research objective, the main research question is formulated as follows:

> ***"Which improvements can be made to the trust algorithm for measuring trust value in file sharing p2p network which in turn will help the peers in p2p network for a better selection mechanism i.e. choosing the right peer during file sharing activity?"***

1.3 Research Methodology

There are numbers of methods that available as guide for the research process. However, there are reasons that have to be taken into account in choosing a research method. The research method first of all has to correspond with the way of how the research is working, because research can start and take place in various situation and environment. Secondly, a research method has to correspond also with the way of research modelling and design support tools, because the modelling and designing of the system is the essence of the research process. Based on these reason, a waterfall model is chosen. This model was introduced in 1992 and it has been frequently used in system engineering research development (Sage, 1992). This model has basically three major phases (formulation, analysis and interpretation) that implemented in several steps. The following illustration shows and gives explanation about the method that has been expanded into 6 steps:

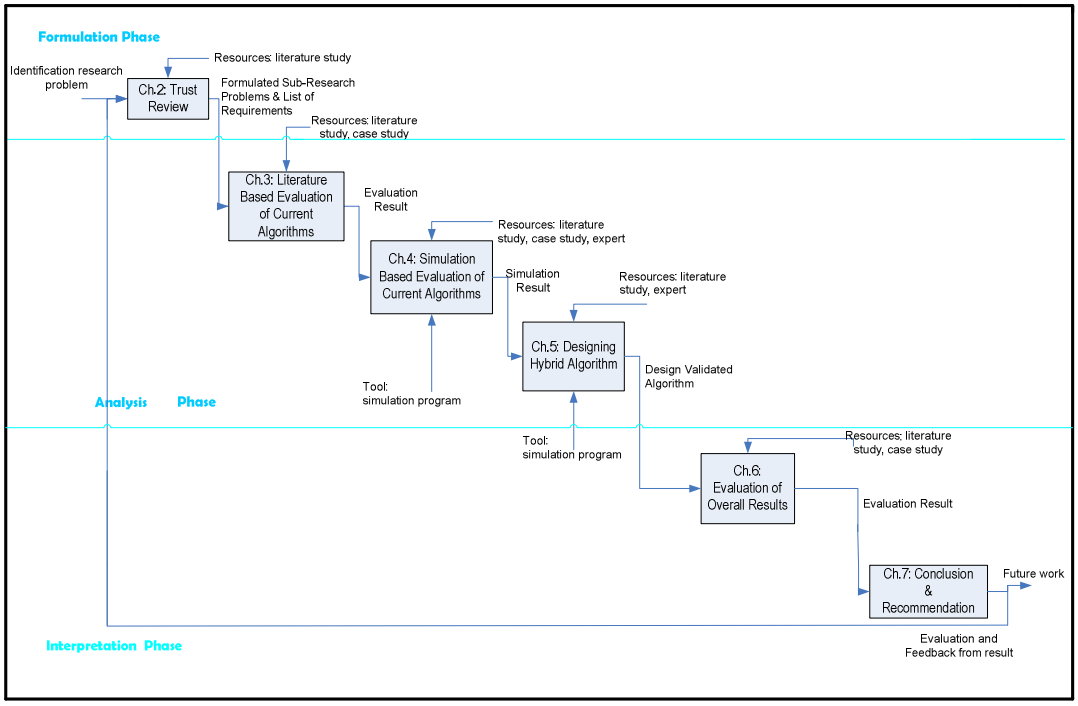


Figure 1.3 Research Methods

- Chapter 2: Trust Review

This chapter provides a depth description about trust. The objective of this chapter is to provide sufficient information on how trust is formalized and operationalized in p2p network with the help of context scenario. Based on the provided information, sub-research questions and a list of requirements will be developed to guide the research process.

- Chapter 3: Literature Based Evaluation of Current Algorithms

This step is covering the third chapter with the objective of conducting evaluation based on literature study on the current algorithms to recognize its drawbacks and advantages. All of this information will be summarized and further evaluated in the next chapter by means of simulation tool.

- Chapter 4: Simulation Based Evaluation of Current Algorithms

In this chapter the current algorithms will be further evaluated by means of simulation tool to confirm the result from the previous chapter. The objective of this chapter is to formulate a list of inputs that can contribute to the designing process of the new algorithm.

- Chapter 5: Designing Hybrid Algorithm

The objective of this step is to design the new algorithm based on the formulated design requirements and the information that is obtained from previous chapters. The new algorithm is given the name Hybrid Algorithm as it is fundamentally based on various current algorithms. The next step is to validate the simulated Hybrid algorithm. This validation process is conducted with the help of query cycle simulation tool and the result of validation will be evaluated in the end of the chapter.

- Chapter 6: Evaluation of Overall Results

This chapter will evaluate the result from simulation test that has been conducted on the algorithms, including the Hybrid Algorithm. The overall results will be presented, compared and evaluated.

- Chapter 7: Conclusion and Recommendation

In this chapter, there are several processes to be identified. The first one is to conclude the result of the research that has been done in previous chapters. The second one is to evaluate whether the result has met the objective of the research which means that problem has been solved. This is also called feed-back process where the result is brought back to the beginning of the research process to be studied and analyzed. The last one in this step is to identify any possibility of further development in the future.

1.4 Structure of the Report

Chapter two deals providing depth description about trust and its connection with p2p network to develop sub-research questions and list of requirements. Chapter three will make use of the requirements to conduct evaluation of the current algorithms, based on literature research. The objective is to obtain a list which contains information about the drawbacks and advantages on the algorithms.
Chapter four is dealing with evolution about the current algorithms with the help of simulation tool. This is the part where the result from the previous chapter can be compared with the result of the simulation.
Chapter five deals with the construction process of the Hybrid algorithm. This process is based on the information that has been previously obtained which result in a design concept.
This concept will be validated in the next step by means of simulation tool. The result will be presented and evaluated in the end of the chapter.
Chapter six will present the overall simulation results from the current algorithms and Hybrid algorithm to be compared and evaluated.
The last chapter will discuss the overall evaluation and conclusion of the research process, based on the research problem and research question that have been formulated in the beginning of the research.

Chapter 2

Trust Review

This chapter will provide a depth description about trust. The objective of this chapter is to provide sufficient information on how trust is formalized and operationalized in p2p network. The first following section will provide detail description on trust relation in file sharing p2p network and its characteristics. The next section is dealing with threats possibilities in connection with trust and p2p network. A context scenario and list of design requirements will be developed in the last section to guide the research process.

2.1 Trust Relation in File Sharing P2P Network

As previously mentioned in the introduction, the peers in the file sharing p2p network represent the users behind the system. They are allowed to join in the network after they install a software application for file sharing into their computers. Once they are in the network, the peers can start to find the files that they are interested in. This is done by "asking" other peers through submitting a message to the network. Because there are many peers (which are representing different users' preferences), there will be lots of messages that are submitted to the network. The other peers in the network which are in possession of the requesting files are welcome to respond. Usually, there will be several responses available so that a peer has to choose which one is the best choice. If the source peer is chosen, the file sharing activity will take place, as illustrated in the following figure.

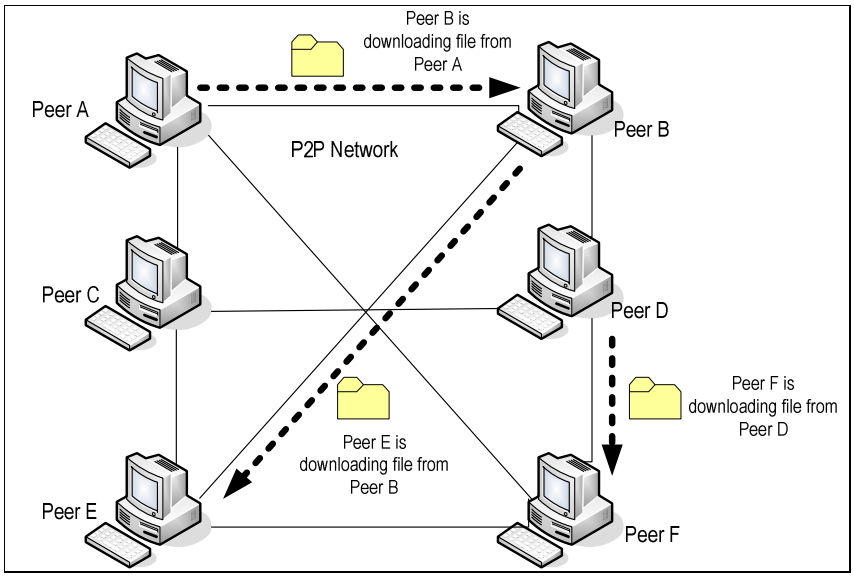


Figure 2.1 File Sharing Activity in P2P Network

The process of choosing the right source peer brings us back to the trust issue. There are several trust relations that can be derived in p2p network, as shown in the following figure:

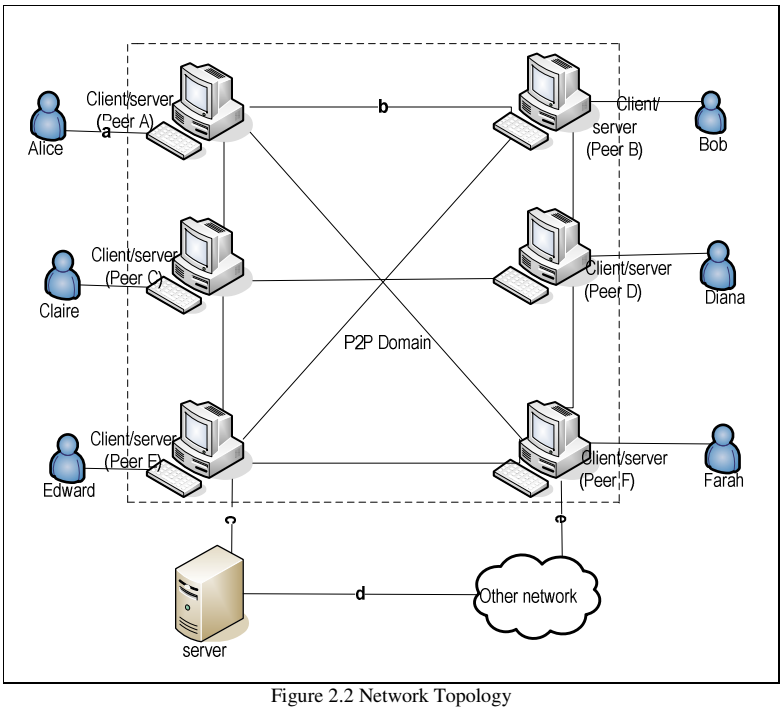


Figure 2.2 Network Topology

According to figure 2.2, there are 5 kinds of trust relations that exist within the networks:

- The first one is trust between *user and system*, marked with "a" in figure 2.
- The second one is trust between *computer peers* in the network (p2p relation), marked with "b" in figure 2.
- The third one is trust between *computer peer and server*, marked with "c" in figure 2.
- The fourth one is trust between *server and network*, marked with "d" in figure 2.
- The last one is trust between *computer peer and network*, marked with "e" in figure 2.

This research will focus on second trust, namely trust between peers in the network. Other kind of trust will be outside the scope of this research with the assumption that other trust relationships are well functioned. However, we must not forget that peers represent the users behind the system. Thus, trust between peers in the network in fact is an extension trust between the users in the network. Figure 2.3 describes different types of trust that are involved in the file sharing p2p network activity.

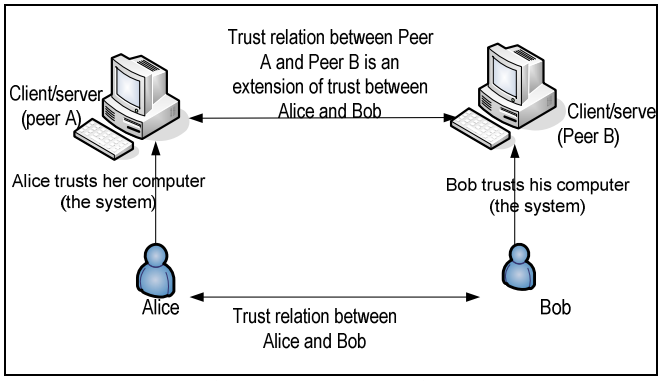


Figure 2.3 Trust Relations

In the daily life, trust is built gradually through time based on the history and experiences. In the case of Alice and Bob, good experiences will raise the level of trustworthiness between them while bad experiences will lower it. In p2p network, trust is also built based on good and bad experiences. A peer with good behaviour will be rewarded with a high trust value while malicious peer will be punished with a low trust value. Therefore, it is crucial to define which factors that characterize a trust relation in order to be able to transfer the trust relation between Alice and Bob to the trust relation between peer A and peer B properly. Almenarez et. al. (2003) defines several properties that are characterize a trust relation:

- Non-symmetric property

This property defines that a trust relation is not a two ways street. If Alice trusts Bob, it does mean that Bob also trusts Alice. Thus, if A and B are two different peers, a non-symmetric relationship means that if A trusts B, it does not automatically mean that B trusts A. This property is important in order to have an accurate trust value calculation in p2p network because if peer A calculates peer B's trust value, the result might be different is peer A calculates peer B's trust value, hence the non-symmetric property.

- Conditional Transitive property

Conditional Transitive means that if Alice trusts Bob, and Bob trusts Claire, it does not mean automatically that Alice trusts Claire. However, Alice may use information from Bob regarding Claire in order to have more accurate decision. In p2p network, if A, B and C are three different peers, which A trusts B and B trusts C, it does not automatically mean that A will trust C. Peer A may use information from peer B to help the decision process. This property is also needed to generate an accurate result of trust calculation. If peer A wants to calculate peer C's trust value and it does not implement this property, which means it will trust completely on peer B's calculation, this can result in an error trust calculation. If peer B and peer C are malicious peers and peer B gives false information to peer A regarding peer C, peer A will end up having transaction with malicious peer.

- Context property

This property means if Alice trusts Bob for baby sitting, it does mean that Alice trusts Bob for driving a car. In p2p network, if A and B are two different peers; z and w are two different type of files, a contextual relationship means that if A trusts B for z, it does not mean automatically that A trusts B for w. This property is important to enhance the accuracy of trust value calculation. Because if peer A calculates peer B's trust value based on x type of files, the result might be different if peer A calculates peer B's trust value based on y type of files. Therefore, it is important to specify the context of trust value to obtain an accurate result.

- Reflexive property

This property means that before trusting anybody else, Alice has to trust herself first. In p2p network, if A is a peer in the network, a reflexive property applies when peer A trusts itself before starts calculating another peer's trust value. Or in other word, it finds its own system to be trustworthy. This property is important for a peer to have a confident on its own system and the result of its own calculation.

- Dynamic

Trust is a dynamic value. It means that it changes through time. Alice might trust Bob more or less through a period of time. If A trusts B for a certain value at t_0; this value is possible to change at t_1. Therefore, it is crucial to take into account this dynamic property to get an up to date trust value result.

Beside the trust properties, trust is also closely related with the term reputation and recommendation. The term reputation is the common opinion that people have about a subject (party) based on prior actions (Cambridge Dictionary, 2007). The party with a good reputation has a higher level of trust from other party. This reputation could be transferred also to another party or community by means of recommendation. Ruohomaa (2005) defined recommendation as an effort to transfer a reputation from one community to another. With recommendation, a peer that is about to perform a transaction with an unknown peer, is able to make a correct decision based on the recommendation. How a peer receives recommendation from other peer (recommender) is depending on how trust is provided within the network i.e. the trust model of the network. There are several trust models that can be employed in a network (Daskapan, 2005):

- Central Hierarchy: trust depends on one central authority; thus, one or more peers grant credentials to other peers.
- Central Peer: trust according to central peer principle, with the use of some peers as mediator for distributing credentials to other peers in the network.
- Decentral Peer: in this model, all of the peers within the network can act as end-peer or recommender for other peers. This is a simple but the least reliable model since there is no need for any recommenders in the system to prove the recognition of its recommendation.
- Meshed Hierarchy: this is a model where trust is built by joining together some of recommender peers that have higher status in the network and they function as bridges for distributing trust through the network.

Research study shows that in order to be able to calculate trust value, it is also important to learn the characteristic p2p network and the issues around it, in connection with trust algorithm. Based on literature study, there are several factors as the main issues in managing trust in p2p network (Abdul Rahman and Hailes, 1997 & Garcia-Molina et. al. 2003):

- No central coordination in the network: Because all of the peers in p2p networks serve the same function on the same level, there is no peer that serves as a central of the network to monitor and coordinate the network. But this unique characteristic is also the power point of p2p network where could make the scale of the network unlimited.
- Unreliability: This term is closely related to the dynamic character of the network. A peer can not fully rely on and trust the neighbours since most of the time they do not know one another.
- Anonymity: Just as mentioned above, the users in the network do not know one another. Anonymity can be both an advantage and disadvantage factor of the system. It can protect the real identity of the peer but at the same time it gives possibilities to negative actions from other peer.
- Lack of prior experience: To build trust among the peers in the network, a reputation is usually used as a base of connection. A problem occurs when a peer is dealing with a new peer with no previous experience. In this case it is difficult to know whether the peer should or should not trust the new peer.

2.2 Threats to Trust

Beside the trust properties and the several issues that are related to trust which have been previously mentioned, there are also a number of threats that are involved in trust relation in trust relation in file sharing p2p network. Referring to the last example of Alice

and Bobs' trust relation, where Alice trusts Bob; the threats to their trust relation are involving the following situations :

- Alice trusts Bob, where Bob's trustworthiness is built gradually through time based on the experience between them. But at a certain time, Alice has a bad experience with Bob it will automatically decrease Bob's trustwothiness (situation A).
- Alice does not know Bob, but Alice needs to have a transaction with Bob. If Bob is not trustworthy, then Alice will end up with unsuccessful transaction with Bob (situation B).
- Alice has a recommendation from Diana about Bob's reputation. But Alice does not know that Diana is untrustworthy who gives a misleading information to Alice about Bob which can result in unsuccessful transaction between Alice and Bob (situation C).

In file sharing p2p environment, the similar situations apply. There are two types of peers that exist in the network: good peers and malicious peers. The good peers are by definition the peers that provide good and honest service in sharing files with other peers in the network. These peers do not pose threats. The other type of peers are the source of the threats in the network. These peers spread malicious files to other peers that eventually can lead to network failure. Besides spreading malicious files, malicious peers also give false recommendation to ruin other peers' reputation in the network. Some time they operate together with other malicious peer (collusion malicious peer) but often they operate on their own (single malicious peer). Based on this description, we can apply the three previous situations on the p2p network activity as follows:

- For situation A: Peer A and peer B are both good peers. Peer A trusts peer B and the trust relation is based on the prior experiences between these two peers. If in a certain time, peer B suddenly turns malicious, it not possible for peer A to know before having a transaction with peer B. This will cause an unsuccessful transaction result (malicious file).
- For situation B: Peer A and peer B never have prior experiences between them (never have files sharing acitivity). Peer A is a good peer but peer B is a malicious one. Because peer A does not have any information regarding peer B and peer B has the requested file, peer A will agree to have a transaction with peer B which will result in a unsuccessful transaction (malicious file).
- For situation C: Peer A is a good peer but peer B and peer D are malicious peers. In this case, peer A receives recommendation from peer D regarding peer B's reputation. Because peer D is a malicious peer, as well as peer B, it will give false recommendation that will mislead the decision process of peer A whether it should trust B or not. If peer B and peer D are cooperating together to deceive peer A, it is called a collusion malicious peer; but if they are acting alone, it is called a single malicious peer. In this research we will only use one type of malicious peer, namely single malicious peer for the shake of simplification.

In the following section, we will apply these situations in the scenario context of the research study.

2.3 Context Scenario and List of Requirements

This scenario is introduced to illustrate the context of the research and p2p environment where the trust issue exists. To join a p2p network, peers have to install sharing files software to allow them to connect with other peers. When the software is installed, the network will group the peers in several small clusters and every peer will have access to the

directory of the network. A directory is an index that serves as the information bank of the network. It keeps network statistics such as network traffic, network diameter, peer cluster, etc. The following section will provide a description on how the files sharing take places among the peers.

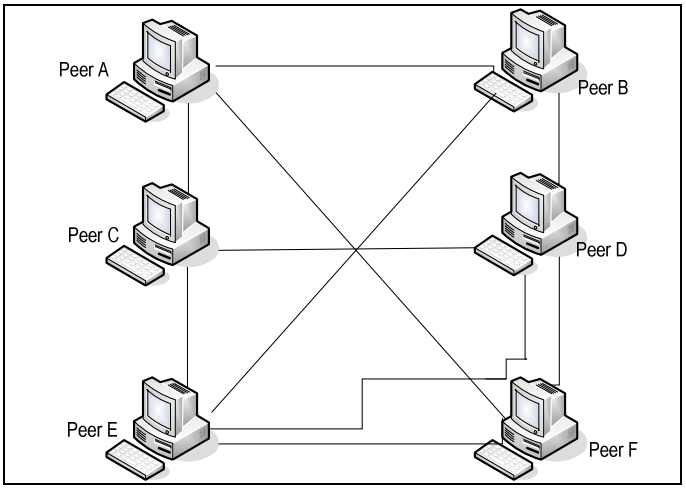


Figure 2.4 P2P Network

Figure 2.2 describes an example of topology of a file sharing p2p network where it is frequently found on the internet. In this figure, six different peers are identified in the cluster:

- Peer F is the oldest member and the founder of the network. It has experiences with almost every peer in the network and has a high trust value. Some of the algorithms give the name pre-trusted peer for this kind of peer.
- Peer A and B are regular members of the network. They have large number of experience with several peers in the network. They are not pre-trusted peers but they are good peers (they do not have malicious intention towards other peers or the network) and identified also with fairly high trust value.
- Peer C is also a good peer but it is a new member with limited experience with other peers within the network, thus its trust value is still low.
- Peer D and E are malicious peers with intention to spread malicious files among other peers to fail the network. Their trust values are also very low or in most of cases will be zero.

Between the peers, there are lines that illustrate the possible connections in the network when the files sharing activity takes place. Files sharing means that the peer will make a certain document available for others to download. The type of the files itselfs are numerous, such as: movie files, music files, software files, document files and etc. And within a type of file, for example: movie files, there are still numbers of different movies files that can be found. Thus, a p2p network deals not only with (large number of) peers, but also with (large numbers of) various types of files. Sharing file activity starts when, for example, peer A, submits a request query for a certain file: Simpsons. The peers will forward this query to its direct neighbours within the cluster, and they will forward the query throughout the network. The peers which posses the file and receive the query will send response queries back to peer A through the directory. The responses are displayed under the search result as seen in the following figure.

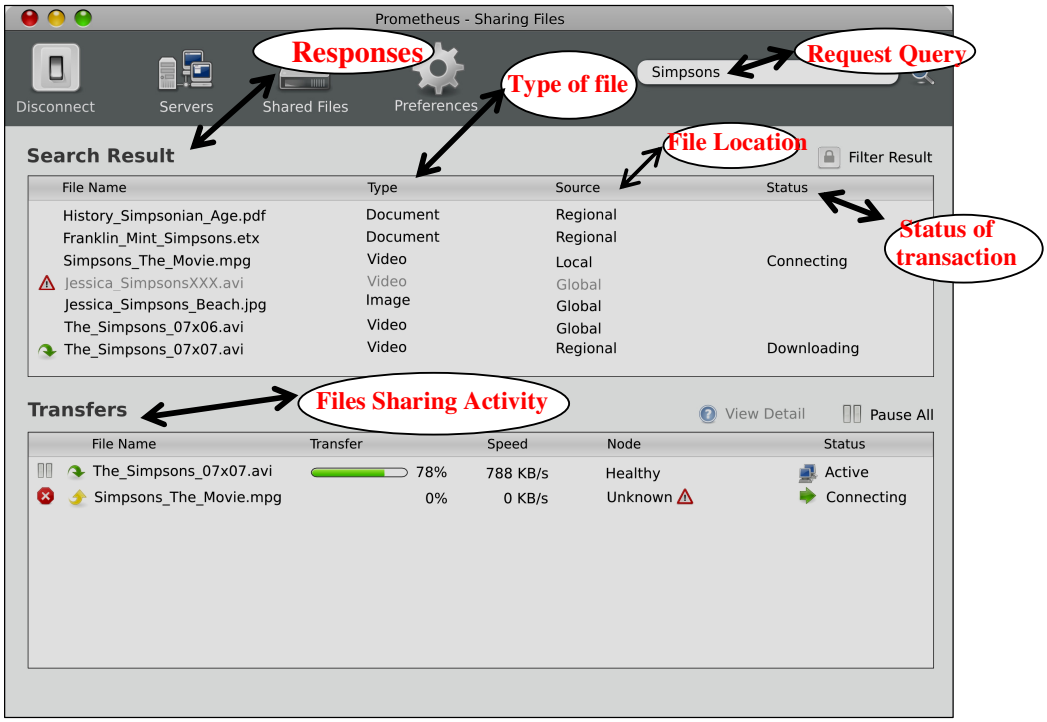


Figure 2.5 Search Result of Simpsons File

In the figure, there are several numbers of responses files that are associated with the requested files: Simpsons. Next to the files name are the types of the files and the source of the file. “Local” means that the file is owned by peer A and other peer is trying to download it. “Regional” means that it comes from a peer the same cluster as peer A. “Global” means it comes from a distant peer that is not in the same cluster. Next to the source is the status of the activity of the files sharing (“Downloading” means the files sharing is taking place and “Connecting” means that another peer is initiating a connection with peer A). Next, peer A will choose one (or more) file from the search result. And once a file is selected (in the figure 2.3, file The_Simpsons_07x07.avi is selected), trust value calculation is taking place between Peer A and the peer which has the selected file, for example peer F. The result of the trust value calculation is shown under “Node” sign. “Healthy” indicates that the files is originated from a good peer (Or in other word, peer F has high trust value based on peer A calculation) and “Unknown” means that peer A does not have information regarding trust value from the owner of this file. This is because peer A is the owner of file Simpsons_The_Movie.mpg and other peer is trying to connect to peer A to get the file. Thus, peer A’s trust value is being assessed. However, some algorithms have earlier trust value calculation. In this case, from the moment peer A receives responses regarding the request query; the system will automatically start to compute the trust value of the responding peers. Thus, the search result will only display the files that originate from peers with known trust value.

Based on information that has been just described, it is very important to calculate trust value as accurately as possible so that peer A can obtain the file from a trustworthy (source) peer. In the previous part, it has been described several trust properties that are characterize the trust relation between the peers in the network. These properties can be used to improve the trust value calculation so that the peers will achieve their objectives to receive the requested files in

simple, safe and fast way. There are five trust properties that can be applied in the context scenario as follows:

- Non-symmetric: As previously mentioned, this element defines that if peer D trusts peer A, it does not automatically mean that peer A trusts peer D back. This property is needed to create an accurate trust value calculation. In the last section about threats, there are three situations described. This particular property is applied to avoid these three situations where a trust value of a certain peer can be unexpectedly change. Thus, with the non-symmetric property applied in the context scenario, peer A will have to calculate peer D's trust value when it is needed even though it is established before that peer D trusts peer A.
- Conditional Transitive: This trust property is used to enhance the accurary of trust value calculation, especially when a third party is involved. Based on the context scenario, if peer A trusts peer D and peer D trusts peer E, peer A will have to calculate peer E's trust value before peer A decides whether it trusts peer D or not. During the calculation process, peer A is allowed to use a recommendation from peer D regarding peer E as part of the input variable. By implementing this trust property, peer A will be able to generate a more accurate trust value and avoid the malicious activity (situation C) as previously described.
- Contextual: This property is used to enhance the trust value calculation when there are several types of files are available in the network. To specify the trust value among the various types of files, the trust value of each peer is set apart for each type of file. Thus, if peer A wants to download two different types of files (for example: music file and movie file), it will have to perform two trust value calculations in order to get more accurate results.
- Reflexive: This property is needed for each peer to have a confidence upon its own system before it starts to calculate another peer's trust value. If a peer is confident to have a trustworthy system, it will trust the result that the system delivers.
- Dynamic: As mentioned before, trust evolves through time. This property is applied in the trust calculation system to cope with trust evolution process in order to generate an accurate trust value result.

These five elements of trust should be included in designing trust algorithm to obtain an accurate computation. However, because the algorithm computation takes place in p2p environment, we have to take into account the characteristic of the network, so that the algorithm can utilize it. In section 2.1, several issues regarding the characteristic of p2p network are already mentioned. Based on this information, we can summarize the following points:

- P2P network needs trust algorithm that can be self regulated, which means that every peer is allowed to make their own judgment, and not depend on a central authority. Based on this system, trust algorithm will be able to cope with the distributed character of the network, so that if the network is dynamically changing, the scalability of the network would not create a problem.
- During the computation process, there will be often recommendation from other peers or data storage needed. To obtain accurate, fair and safe trust value computation, this process should be conducted without any knowledge from the peer that is being assessed. The objective of this system is to maintain anonymity of every peer in the network.

- As previously mentioned, typical p2p network consists large number of peers and various kind of sharing files. In order to utilize the network, trust algorithm should enforce minimum load on the whole process so that it does not influence the network and peer performance. The process of trust value computation is including: messaging load, computation process load and data storage.
- Users' preferences which are represented by the peers in the network should be taken into account in the trust calculation process in order to have a result that corresponds with each user's preference. Therefore, it is required to develop a trust algorithm where a user (a peer) can actively participate in the process (user intervention system). Because most of the current algorithms are fully automated by the system, where the users (the peers) can not influence the trust value calculation process.
- Because trust value is calculated based on the peer prior experiences. A peer with more experiences can calculate more accurate trust value than a peer who has limited experiences. Based on the context scenario, pre-trusted peer F is the peer with the largest number of experiences. And since peer F is the founder of the network, it is very unlikely it would turn malicious. The idea is to use peer F's experiences to help other peers in the network by giving the pre-trusted peer F a role in the trust value calculation process in order to generate more accurate trust value.

These five features have to be included as well in the designing process of trust algorithm, beside the five elements. Thus, combining all of the factors, a list of design requirements for designing algorithm can be developed as follows:

Trust algorithm should:

1. incorporate non-symmetric element.
2. incorporate conditional transitive element
3. incorporate contextual element.
4. incorporate reflexive element.
5. incorporate dynamic element
6. incorporate self-regulated system to cope with network scalability.
7. maintain the peer anonymity.
8. enforce minimum load through the whole process in order to improve network and peer performance.
9. implement a user intervention system.
10. include pre-trusted peer recommendation in the trust value calculation process.

These design requirements will be used as the research guide to conduct the evaluation process on the next following chapter in order to achieve the research objective.

2.4 Research Sub-questions

Based on the description about trust that is previously provided, the research's main question can be broken down into several sub-questions that will assist the process to achieve the research objective. There are three sub-questions that are defined here:

1. What are the shortcomings of the current algorithms to measure trust in p2p networks?
2. Which beneficial factors can be employed in the construction process of the Hybrid algorithm based on the result of the evaluation of the current algorithms?
3. To what extent does the Hybrid trust algorithm meet the requirement criteria?

The first sub question deals with current algorithms (literature based and simulation based evaluation of current algorithms). They are already briefly mentioned before, but this part of the research will study these algorithms in detail in order to find their advantages and disadvantages so that the Hybrid algorithm can benefit from this study. Based on the test result, the second sub question will deal with gathering inputs for designing the Hybrid algorithm based on the result of the previous evaluation. The last sub question will test the newly designed trust algorithm how far it can satisfy the list of the requirements.

Chapter 3

Literature Based Evaluation of Current Algorithms

The objective of this chapter is to assemble a list that contains the drawbacks and advantages of the current algorithms. This is done by conducting an advance literature study on the current algorithms. This section will start with reviewing the current algorithms that will give more insight about the detail on the algorithms. The review process will be conducted based on the design requirements that have been described in the previous chapter. By the end of the chapter, the advantages and disadvantages of the existing algorithms are expected to be singled out.

3.1 Introduction

In this section, we will review the previous works that have been proposed by various scientists over the past decade. There are numbers of trust algorithms that are introduced to measure trust value for different applications and environments. We have done literature study and narrowed them down into seventeen algorithms that correspond with this research. These algorithms will be categorized into two different groups based on the algorithm type. Within these groups, the algorithms will be divided into sub-category in order to be able to review and examine more thoroughly. The reviewing and examining process will be conducted based on the design requirements of trust algorithm that have been discussed previously. Based on these processes, there will be four final algorithms that are chosen to represent each category. These last four will be closely observed and tested to analyse their drawbacks and advantages that can benefit for the designing process of Hybrid algorithm.

The categorization criterion of the algorithms is the first crucial subject to be made in this process. There are number of possibilities of categorization that can be applied on the algorithms but not all of them are optimal categorization. An example of categorization criterion is based on the structure of the trust model. According to this categorization, the algorithms can be grouped based on whether the trust model incorporates decentralized system or centralized system (which is recognized with the presence of CA). Unfortunately, the result from the literature study shows that more than sixteen algorithms incorporate decentralized system which leaves only two algorithms that apply centralized system with CA. This causes an extreme unbalance in the group categorization. Another example of possible categorization is based on the scope of the algorithm, whether it is a global trust algorithm or a local trust algorithm. Global trust algorithm applies when a peer use all of possible recommendations from other peers in the network in addition to it own data history to compute trust value. And a local trust algorithm applies when a peer use only its own data to compute trust value without feedback of recommendation from other peers. And once again the study shows that all of the algorithms use feedbacks of recommendation from other peers to measure trust value which leaves no algorithm that only incorporates a local trust algorithm, which also makes an unbalance categorization. Another possibility of criterion is to categorize the algorithms based on the fundamental computation theory they use in the algorithms, but the result is extremely varied, making it difficult to classify them into few groups.

Finally, after a thorough review of the algorithms, we select the optimal categorization for these algorithms. The selected categorization is similar with one of the previous example with a modification. The algorithms are categorized based on scope of how the algorithms are built, namely global and partial trust algorithm. The first type has been explained previously

and the second one applies when a peer uses only some of the recommendations to measure trust value of another peer. See illustration below to get a better picture of the algorithm types:

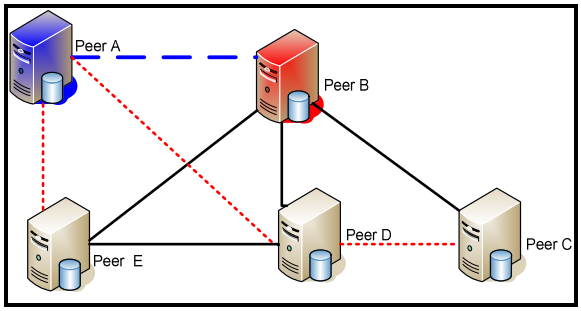


Figure 3.1 Global Algorithm

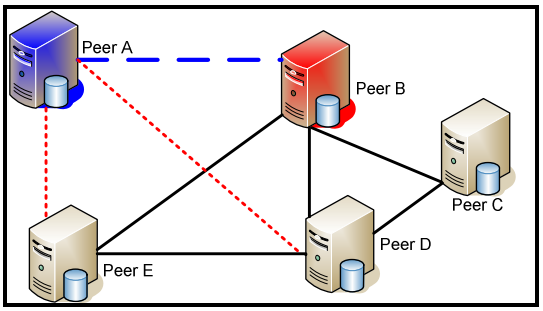


Figure 3.2 Partial Algorithm

The first illustration shows that peer A receives feedbacks from other peers (the short dotted lines), in addition to its own data base (the long dotted line), and uses it all to compute trust value of peer B. In the second illustration, peer A still uses its own data base (the long dotted line) but it only uses some of the recommendations of other peers to measure peer B (the short dotted lines).

3.2. Categorization Process

In this section, the categorization will begin with dividing the algorithms into two categories, namely global trust algorithms and partial trust algorithm. Within this category, the algorithms will be sub-divided into two smaller groups. This sub-division will be applied to both two categories and is based on whether there is distinction made in the algorithm between direct and indirect trust. Direct trust is trust value that is obtained through a direct interaction and stored in the peer's own data base, where indirect peer is obtained from other peers by means of feedbacks or recommendation. Some of the algorithms do not make distinction between these two values and consider them as one type of trust value in the computation. Other algorithms propose that these values are different, and therefore must be considered and treated in different manners. These algorithms incorporate different weights or different formula for these values in the computation process in order to obtain more an accurate outcome. With this procedure, the algorithms will be divided into four different groups and they will be further assessed each one specifically.

The design requirements are used in this process as guidance for the assessment. Algorithm which meets the requirements for trust algorithm the most will be selected to represent each category. In the end, there will be four selected algorithms to be tested later on to provide some information for the designing process of Hybrid algorithm.

This list will be applied on every single algorithm within each sub-category. For detail on each of the algorithm, see appendix 1 because we present only the selected algorithms in this section.

Categorization:	**Partial Algorithm**	**Global Algorithm**

Distinction between direct and indirect trust	Category 1a: - Trust algorithm based on Fuzzy Set (Shuqin et. al, 2004) - Trust algorithm with Confirmation Theory (Mengshu et. al, 2002) - Trust algorithm with Community Peers (Agostini and Moro, 2004) - Trust algorithm with Distributed System (Abdul-Rahman and Hailes, 1997) - Trust algorithm with Evidential Model (Yu and Singh, 2002) - Trust algorithm with Statistical Foundation (Shi et. al, 2003)	Category 2a: - Trust algorithm with Bayesian network (Wang and Vassileva, 2004) - Trust algorithm with Beta Reputation System (Josang and Ismail, 2002) - Trust algorithm based on Similarity Measure of Vector (Guo et. al, 2005) - Trust algorithm with Ad-Hoc Environment (Almenarez et. al, 2003)
No distinction between direct and indirect trust	Category 1b: - Trust algorithm with Information System (Aberer and Despotovic, 2001) - Trust algorithm with Information Based Model (Sierra and Debenham, 2005)	Category 2b: - Trust algorithm in Decentralized Community (Xiong and Liu, 2004) - Trust algorithm in Decentralized Environment (Wang and Varadharajan, 2004) - Trust algorithm in Pure Ad-Hoc Network (Pirzada and Mc.Donald, 2004) - Trust algorithm with Community Based Model (Jin, 2005) - Trust algorithm with EigenTrust System (Garcia-Molina et. al, 2003)

Table 3.1 Algorithm Categorization

3.3 Sub-Category 1a: Partial Algorithm with Distinction Between Direct and Indirect Trust Value

There are in total six algorithms in this sub-category. All six of them have various procedures to calculate the trust value. Based on design requirements, these algorithms are carefully examined and Algorithm Mengshu is chosen as the representative algorithm. This algorithm is based on the notion of decentralized system where there is no central authority needed. The basic foundation of the algorithm Mengshu lies on confirmation theory to calculate the trust value. This theory is based on the notion belief and disbelief that can be loosely translated to successful transaction (leading to belief) and unsuccessful transaction (leading to disbelief), which will result in certainty factor C (Mengshu et. al, 2002). This C will later define the trust value of the peer in the network. If there are multiple certainty factors available, these will be combined to produce a single C value as follows (for description of each variable in the following figure, see legend in pages 25 and 26):

$$C_{ACDFB} = \begin{cases} C_{ACB} + C_{ADB} + C_{AFB} - C_{ACB} \times C_{ADB} \times C_{AFB} & if\ C_{ACB} \geq 0, C_{ADB} \geq 0 \text{ and } C_{AFB} \geq 0 \\ C_{ACB} + C_{ADB} + C_{AFB} + C_{ACB} \times C_{ADB} \times C_{AFB} & if\ C_{ACB} < 0, C_{ADB} < 0 \text{ and } C_{AEB} < 0 \\ \dfrac{C_{ACB} + C_{ADB} + C_{AFB}}{1 - \min\{|C_{ACB}|, |C_{ADB}|, |C_{AFB}|\}} & if\ C_{ACB} < 0 \ or\ C_{ADB} < 0 \ or\ C_{AFB} < 0 \end{cases}$$

Figure 3.3 Certainty Factor Formulas

Context Scenario:

Following the example from the previous context scenario, peer *A* sends a request query for Simpsons file to the network through the directory. To spread the query to the other peers in the network, the query is duplicated and sent to peer A direct neighbours, in this case it will be sent to peer C. Peer C will duplicate this query again and send it to its neighbours.

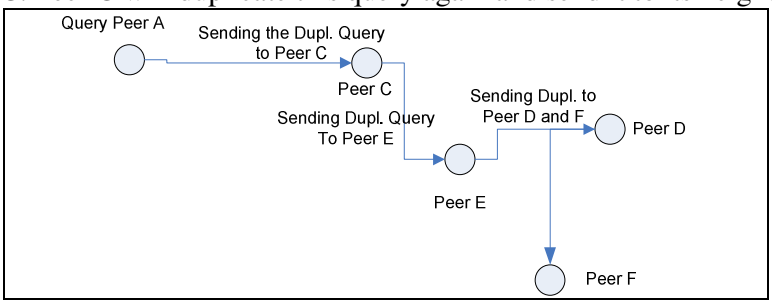


Figure 3.4 Query Propagation Mengshu Algorithm

Based on this system, the algorithm will be able to cope with both large and small network. This activity will continue until the query TTL (Time To Live) is up. Query TTL is a limit on the period of time that each query can experience before it will be discarded. When the TTL is up, the peers that have received the (duplicates of) query and posses the Simpsons file will start to respond. In the case of Mengshu algorithm, all of the responses will be filtered before being displayed to peer A, except for the uploading file (uploading means that peer A is being assessed by other peer in the network). In this case, peer A trust value is being assessed by other peer, hence the "unknown" sign under "Node".

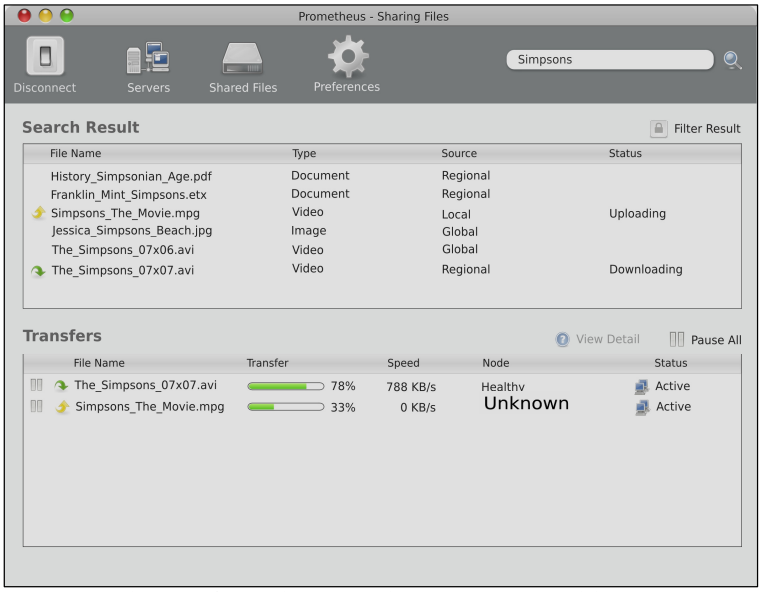


Figure 3.5 Search Result with Mengshu Algorithm

Trust Calculation:

Figure 3.5 shows the filtered query responses of peer A which means that when these responses are received, the trust value calculation will take place for each of the responding peer. For example, let say that peer B responds the query which means that peer A will calculate peer B's trust value. In order to be able to do it, peer A has to calculate first the total number of successful transactions (S_{AB}) and unsuccessful transactions (F_{AB}) between peer A and B, denoted as C_{AB} :

$$C_{AB} = \frac{S_{AB} - F_{AB}}{S_{AB} + F_{AB}} \quad \text{.....(I)}$$

After getting the value of C_{AB}, peer A will have to find some other peers who has recommendation about peer B. To do this, peer A will have to send a second query regarding recommendation about peer B. If, for example peer C, peer D and peer F are responding the query, peer A will have to evaluate the trustworthiness of these three peers by calculating their certainty factor (defining the value, C_{AC}, C_{AD} and C_{AF}). This is done by applying equation (I). If a peer's certainty factor is less than 0, suppose $C_{AD} < 0$, than peer D will be ignored. In this case, peer D is a malicious peer, thus its certainty factor will be lower than zero. For the new peer C, even though it is not a malicious peer but peer C still has limited experience with other peer. This may lead to a low certainty factor for peer C (certainty factor close to zero). On the other hand, peer F will have high certainty factor because of its pre-trusted history. This is the procedure where peer A is calculating the recommendation value from peer C, peer D and peer F (applying the conditional transitive factor of the design requirements). In the following formula, C_{ADB} is obtained through calculation between recommendation value from peer D to peer A about peer B (denoted with C_{DB}) and the maximum value between 0 and C_{AD} which is denoted with $\max\{0, C_{AD}\}$. The similar procedure also applies for obtaining C_{ACB} and C_{AFB}.

$$\begin{aligned} C_{ACB} &= C_{CB} \times \max\{0, C_{AC}\} \\ C_{ADB} &= C_{DB} \times \max\{0, C_{AD}\} \\ C_{AFB} &= C_{FB} \times \max\{0, C_{AF}\} \end{aligned} \quad \text{.....(II)}$$

The next step is to combine all of the recommendation value which is obtained from equation (II) by applying formula from figure 3.3. Even though some of the certainty factors from equation (II) will be ignored because it might result in value that is lower than zero, we will present the combination of all three certainty factors. If the combination value of the certainty factors2 (C_{ACDFB}) is acquired, the trust value of peer B, T_{AB}, can be calculated by the following formula:

$$T_{AB} = \alpha C_{AB} + (1-\alpha) C_{ACDFB} \quad \text{.....(III)}$$

Legend	
Variable	**Description**
A	peer in the network who wants to calculate other peer trust value (requestor peer)
B	peer whose trust value is being assessed (requested peer)
D	recommender peer (a peer which has peer B trust value and recommend it to peer A)
C	recommender peer (a peer which has peer B trust value and recommend it to peer A)
F	recommender peer (a peer which has peer B trust value and recommend it to peer A)
C_{AB}	Certainty Factor of peer B, calculated by peer A

S_{AB}	Number of successful transactions between peer A and peer B
F_{AB}	Number of unsuccessful transactions A and peer B
$\max\{0, C_{AC}\}$	Maximum value between 0 and C_{AC}
$\max\{0, C_{AD}\}$	Maximum value between 0 and C_{AD}
$\max\{0, C_{AF}\}$	Maximum value between 0 and C_{AF}
C_{AC}	Certainty Factor of peer C by peer A
C_{AD}	Certainty Factor of peer D by peer A
C_{AF}	Certainty Factor of peer F by peer A
C_{ADB}	Certainty Factor of peer B by peer A with recommendation from peer D
C_{AEB}	Certainty Factor of peer B by peer A with recommendation from peer E
C_{AEB}	Certainty Factor of peer B by peer A with recommendation from peer E
C_{ACDFB}	Final Certainty Factor that obtained through calculation between C_{ACB}, C_{ADB} and C_{AFB}
T_{AB}	Trust value of peer A by peer B
α	a constant between 0-1. This constant represents the weight balance between recommendation and private experience. This value is set up by the system.

Table 3.2 Legend

Based on the algorithm evaluation, it can be seen that Mengshu Algorithm has fairly simple procedure but it meets almost all of the design requirements, therefore this algorithm is chosen to represent this sub-category. The factors that this algorithm lacks are the reflexivity factor and the context factor. Mengshu never clearly stated about the reflexive factor nor the context factor in the algorithm process. As previously mentioned, the context or classification is important to distinguish trust value of a peer from various types of sharing files to produce more accurate trust value. And regarding the reflexive factor, it is important for every peer to trust itself before start to question about another peer's trust value. The result of the examination is presented in the following table.

Design Requirements:	**Availability**
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	**X**
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	✓
Maintaining peer anonymity	**X**
Performance on: - Computation process - Data storage - Message complexity.	✓
Implementing a user intervention system	**X**
Take into account the role of pre-trusted peer	**X**

Table 3.3 Table Algorithm Mengshu

3.4 Sub-category 1b: Partial Algorithm with No Distinction Between Direct and Indirect Trust Value

There are two algorithms for this sub-category, namely algorithm from Sierra and Debenham (2004) and algorithm from Aberer and Despotovic (2001). The similar evaluation procedure is applied for evaluation process of this sub-category. For these two algorithms, algorithm Aberer and Despotovic is chosen to represent this sub-category.

This algorithm is based on decentralized system where there is no central authority needed and allows the peers to measure trust on the ground of previous experience with other peers. This data from the previous experience will be stored in a decentralized storage system called P-Grid. The following illustration explains how to get the information with P-Grid method and how to compute trust from this information.

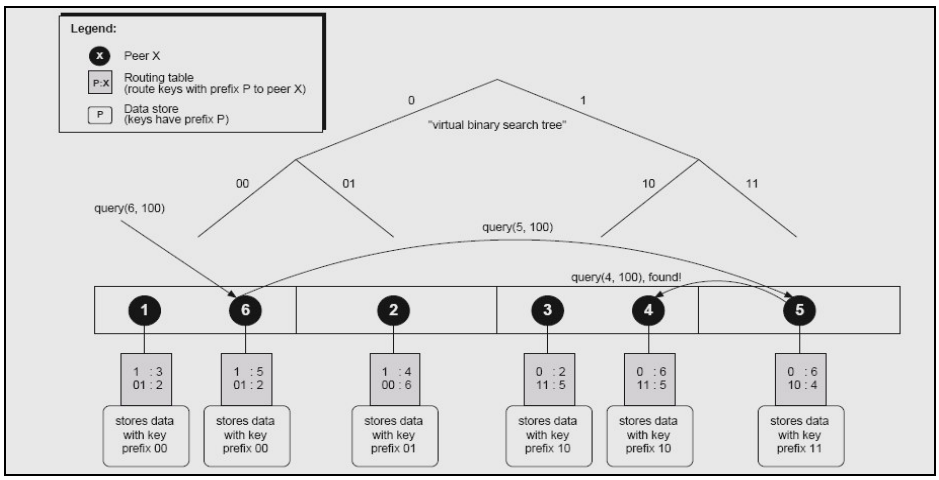


Figure 3.6 Example P-Grids (Aberer and Despotovic, 2001)

The figure above shows an example of P-Grid system with six peers. A request of information from a peer to other peer can be done by means of query. Each peer in the system can serve any query by giving the information or forward the query to other peer if it does not possess the required information. For example of processing a query on a peer, Aberer and Despotovic (2001) give a simple model of query process. The process starts with query (6,100) to peers 6, but peer 6 is not associated with the key starting 1. So it searches in its route table peer 5, which can forward the query to other peer. Peer 5 can not process a query that starts with 10, and forward the query to peer 4 that has the necessary information, because it stores information keys that start with 10. From the illustration above, we can see that the same information on the same peer can be stored in several peers to prevent failures in the network.

This algorithm is the only algorithm that stores complaints as the only data that is stored and used to calculate trust value of a peer. Using the P-Grid to store and manage the data system, a peer can compute trust value in quite small of time. The outcome of the computation process is a binary figure 0 and 1 which indicates whether a peer is either trustworthy or not trustworthy.

Context Scenario:

Supposed that peer *A* (based on the context scenario) sends query for Simpsons file to the network. Peer A will send this query randomly to its neighbour, let say peer C. Peer C will receive this query and forward it again randomly to its neighbour until the query TTL is up.

Because of the randomness of the query, the number of response can be limited for this algorithm.

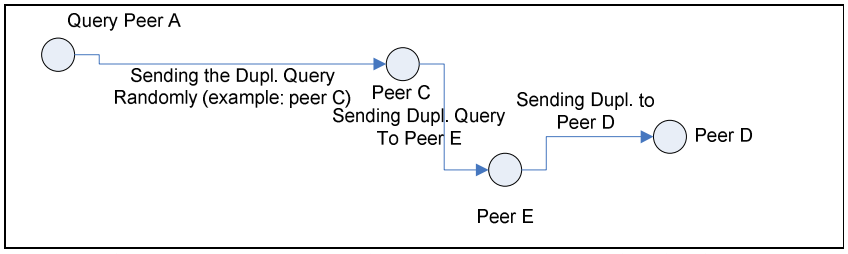


Figure 3.7 Query Propagation Aberer and Despotovic Algorithm

When the responses are received by peer A, it will have to choose one of the responses, and calculate the responding peer's trust value. It can be seen that this algorithm has trust value calculation later than other three previous algorithms.

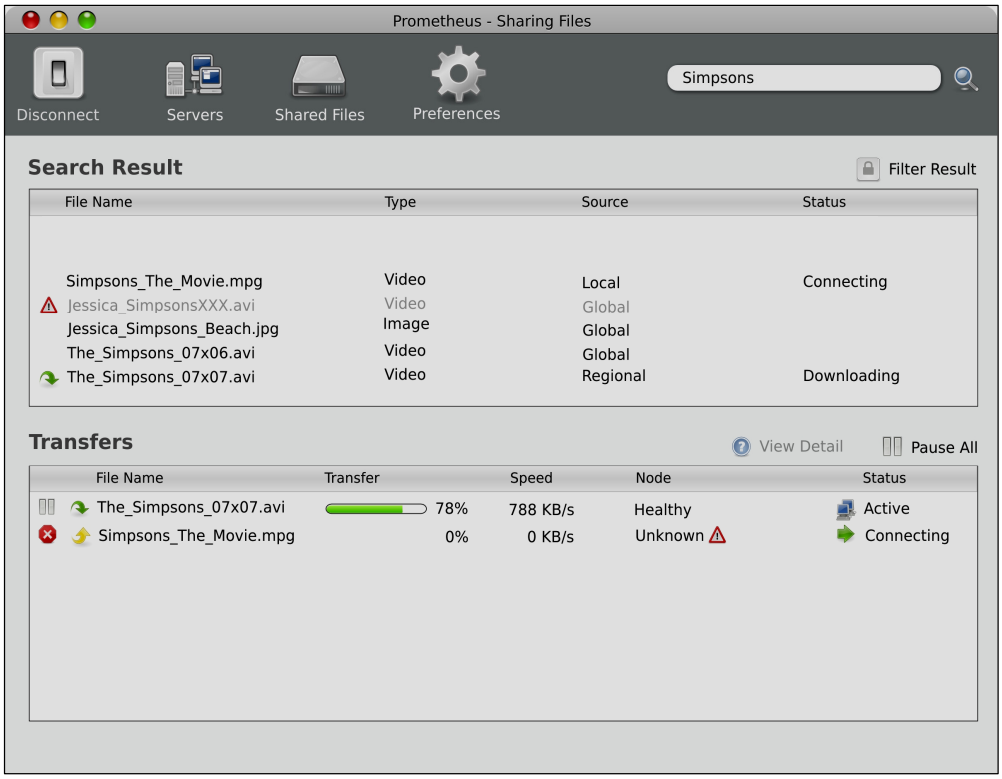


Figure 3.8 Search Result Aberer and Despotovic Algorithm

Trust Calculation:

Figure 3.12 shows the search result of the requested file from peer A. If peer choose to share file with peer B, it will have to calculate first peer B's trust value. It is done by sending a random query to its neighbour. The propagation will be similar with the description of figure 3.10 on P-Grid system.

Usually, a query will be sent several times, let say s time to get multiple data (Aberer and Despotovic, 2001). From this query, let say that peer C are found as recommender of peer B. If the recommender is found, some variables can be defined, such as: number of complaints about peer B that peer A received from various recommenders peer C, CR_{CB}, and number of

complaints that peer B ever filed about other random peers i, which is stored by peer C, CF_{Bi}. In the practice, different frequency f of recommenders will be found due the dynamic structure of the P-Grid. Therefore, there is a possibility that this frequency will be too low. Aberer and Despotovic (2001) proposed a normalization of the values by using the frequency observed during querying as follows:

$$CR_{CB}^{norm} = CR_{CB}(1-(\frac{s-f}{s})^s) \quad \text{.....(I)}$$

$$CF_{Bi}^{norm} = CF_{Bi}(1-(\frac{s-f}{s})^s) \quad \text{.....(II)}$$

Once the value of CR_{CB}^{norm} and CF_{Bi}^{norm} are obtained, peer A will proceed the calculation with finding variables CR_A^{avg} and CF_A^{avg} which respectively are the average of complaints that peer A ever received about all peers and average number of complaints that peer A ever filed or report about other peers in the whole period that peer A exists in the network which is stored in peer A itself. Thus, peer A stores also its own statistic from the time it joins the network until present time.

Based on these variables, peer A can decide whether peer B is trustworthy or untrustworthy (1=trustworthy; 0 = not trustworthy) with the following function:

If $CR_{CB}^{norm} CF_{Bi}^{norm} \le (\frac{1}{2} + \frac{4}{\sqrt{CR_A^{avg} CF_A^{avg}}})^2 CR_A^{avg} CF_A^{avg}$, then 1 else 0 .....(III)

Based on the calculation, it can be seen that this algorithm applies no conditional transitive factor. There is no distinction between private experience and recommendation in the calculation process.

Legend	
Variable	**Description**
A	Peer in the network who wants to calculate other peer trust value (requestor peer)
B	Peer whose trust value is being assessed (requested peer)
C	Peer Recommender for peer A in the network
s	Number of query sent to the network
w	Total number of various peer recommender found
f	Frequency of recommender that is found during the query propagation
CR_{CB}	Number of complaints received about peer B from peer C
CF_{Bi}	Number of complaints filed by peer B about other peer that is stored by peer C
CR_{CB}^{norm}	Normalized number of complaints received about peer B
CF_{Bi}^{norm}	Normalized number of complaints filed by peer B
CR_A^{avg}	Average number of complaints received by peer A about other random peer that is stored in peer A itself.
CF_A^{avg}	Average number of complaints filed by peer A about other random peer that is stored in peer A itself.

Table 3.4 Legend

This algorithm actually contains more drawbacks that other selected algorithms. The reason why this algorithm is selected is because this algorithm is a pioneer on field of the trust data management. While some of algorithms provided only the calculation process of the trust value, this algorithm is actually providing a system how to get and manage the essential

information in order to be able to calculate and store trust value in the network. The advantages of this algorithm lies on the implementation of P-Grid system that makes the information acquiring process and the information storing less complicated. Most of the algorithms are dealing with problem on keeping the information about peer's reputation in order to be able to appropriately compute the trust value. Another shortcoming of this algorithm is that it is very sensitive to p2p network with skewed distribution i.e. the transaction frequency among peers is not high enough. Based on how the algorithm works, this can cause an unreliable trust value calculation due to extremely low number of recommenders that can be found, and therefore is very sensitive for malicious peer's activity.
The result of the evaluation of this algorithm is provided as follows:

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	✓
The presence of dynamic factor.	✓
The presence of scalability factor.	✓
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	X
Implementing user intervention	X
Take into account the role of pre-trusted peer	X

Table 3.5 Requirements Algorithm Aberer and Despotovic

3.5 Sub-category 2a: Global Algorithm with Distinction Between Direct and Indirect Trust Value

There are four algorithms in this sub-category where all of them are examined and assessed to find the representative algorithm. From the examination and assessment process, algorithm Almenarez is chosen to represent the rest of the algorithms. This algorithm is built based on two notions of trust, direct trust and indirect trust, whereas the first trust is provided by the peer's nature or past interaction (private experience) and the second one is given from opinion by recommender peers (Almenarez, et. al, 2002). The calculation of trust algorithm is made of combination between direct and indirect trust with weight mechanism, which means that the algorithm is giving different weight to different trust value (trust value based on private experience and trust value from recommendation). On other word, this algorithm implement conditional transitive factor in the algorithm process.

Context Scenario:
Based on the context scenario, when a peer, let say peer A, sends a request file query to the network for a certain file: Simpsons, peer A will send large number of duplicates of the query to its neighbours (flooding the network). The neighbours will receive the duplicates, make more duplicates (of the first duplicates) and forward further throughout the network. In a

network with limited number of peers, this system can work. On the other hand, in a network with large number of peers, this system will cause network traffic that can limit the number of responses that will be received by peer A.

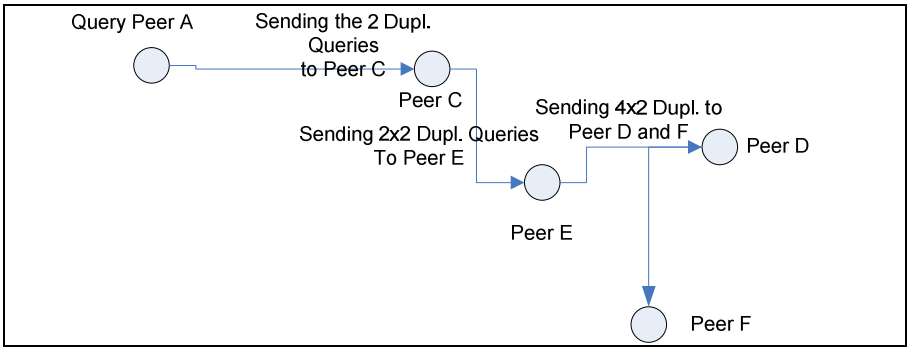


Figure 3.9 Query Propagation Almenarez Algorithm

Figure 3.6 shows the propagation of this algorithm, which a query can flood the network because of the large of message traffic. If the query TTL is up, the responses will start to arrive at peer A. Similar as the two previous algorithms, the trust value calculation will occur at this moment. The following figure displays the search result after the trust value calculation.

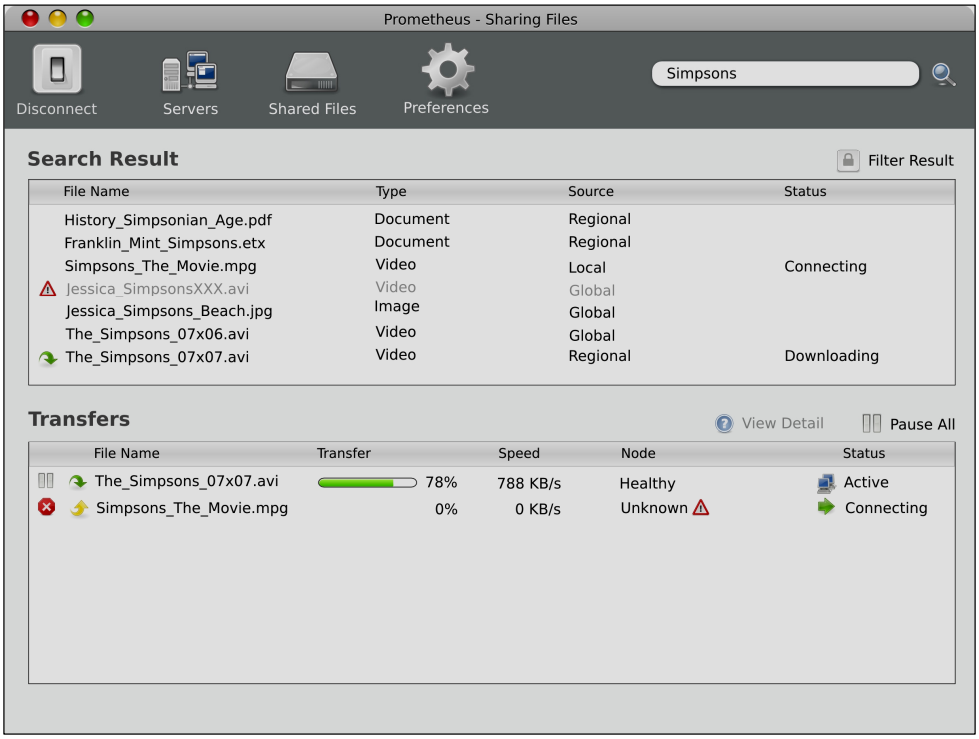


Figure 3.10 Search Result Almenarez Algorithm

Trust Value Calculation:

Peer A will calculate the trust value of all the responding peers. If , peer B respond to peer A, the trust value calculation will be as follows:

- Calculating peer B trust value

To calculate trust value of peer B, peer A will need recommendation from other peer to obtain more accurate result. In order to receive recommendation, peer A sends another query to the network. The second query will propagate in the similar way as the first query (see figure 3.6). Let say, peer C responds to the query and sends its recommendation to peer A (T_{CB}) . The calculation of peer B trust value by peer A with recommendation from peer C, T_{ACB}, will be the result between recommendation from peer C, T_{CB}, times the trust value of peer C by peer A, T_{AB}, as follows:

$$T_{ACB} = T_{CB} \times T_{AC} \quad \text{.....(I)}$$

- Introducing security level m

This algorithm introduces new variable called security level. This variable defines how high the security the peers wish. The variable is not used in the equation (I), but every peer has to define its security level by the time the peer joins the network. This is the first algorithm which allows user intervention in the trust value calculation. The security level can range from 1 (the lowest) to 9 (the highest). The rule is if a peer needs high security, it can assign a high number of security level and vice versa. Within the p2p network, a peer with higher security level will not have transaction with a peer with lower security level. Thus, before peer A executes the equation (I), it will check first whether peer B and peer C have higher or equal security level. In the case where this is not satisfied, peer B and peer C will be ignored and peer A will have to find another peer.

- Up-dating the trust value of peer B

Once the trust value T_{ACB} is acquired, peer A will decide whether it will perform the transaction (file sharing) with peer B. If the file sharing takes place between these two peers, another calculation will be needed if the transaction is completed. This is called trust value evolution (updating trust value). The calculation begins with calculation of action value V which will define a peer reputation based on the past behaviour. This is where the security level plays an important role. A peer with high security level, but abuses it, for example: peer D sets a high security level so that it has transactions (files sharing) with a lot of peers, but use the transaction to spread malicious will be considered as abusing the security level, and it will be severely punished with a very low action value and finally will lead to a low new trust value T_{ACB}^{new}.

Action value V is defined by number of unsuccessful transaction between peer A and peer B, F_{AB}, total transactions between peer A and peer B, $S_{AB} + F_{AB}$, action weight W and security level m. Action weight is variable that gives weight to the transaction that has just been completed between peer A and peer B. If the transaction is successful, W is set to one. On the hand, if the transaction is not successful, W will be set to zero.

$$V = (1 - \frac{F_{AB}}{S_{AB} + F_{AB}}).W^{(m)} \quad \text{.....(II)}$$

From equation (II) it can be seen if the action weight is set to zero, the action value will result in zero and so will be the new trust value.

If the action value is calculated, peer A will be able to adjust trust value of peer B :

$$T_{ACB}^{new} = \begin{cases} V.\beta + T_{ACB}.(1-\beta) & V > 0 \\ 0 & \text{else} \end{cases} \quad \text{.....(III)}$$

Every time a new transaction takes place, the similar routine will occur in every peer within the network. Thus, every peer retains always an up to date trust value information.
The following figures will show two search results of peer A with two different security levels. The first figure shows search result with security level is set to five while the other figure shows search result with security level is set to four. The one with higher security level has less number of responses but it comes from more trustworthy peers than the second figure with more search result (the previous figure under context scenario has security level 4). The peers have to define themselves which security level is suitable for them. It can be done by changing the security level parameter under "Preference" sign until the peers find the most suitable one.

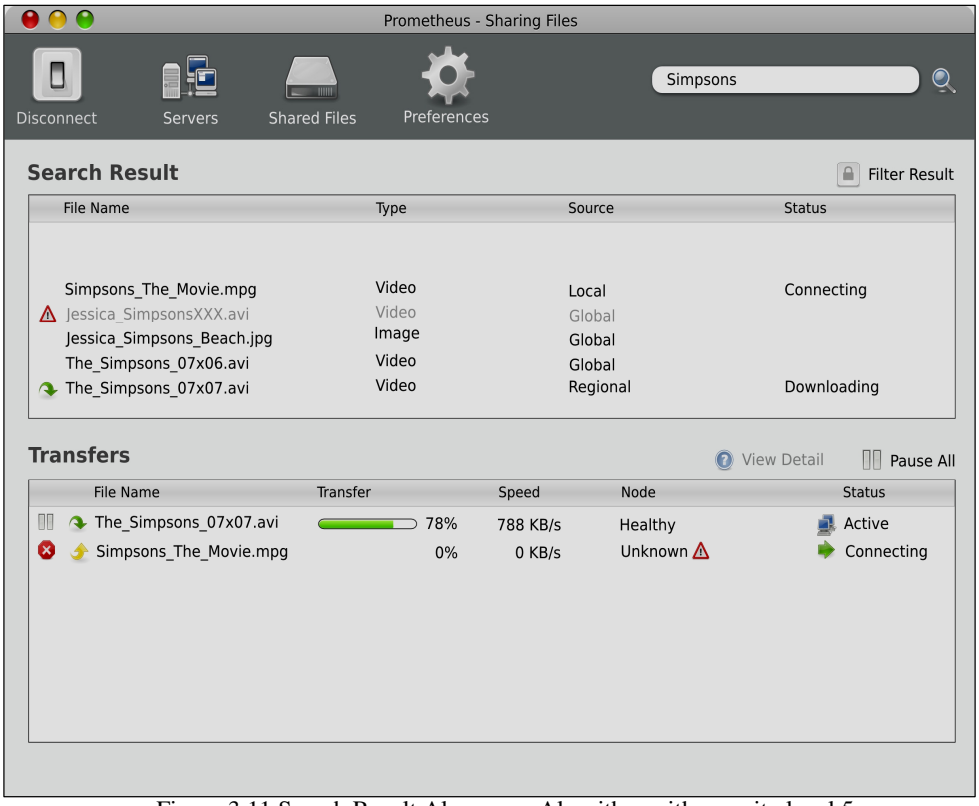


Figure 3.11 Search Result Almenarez Algorithm with security level 5

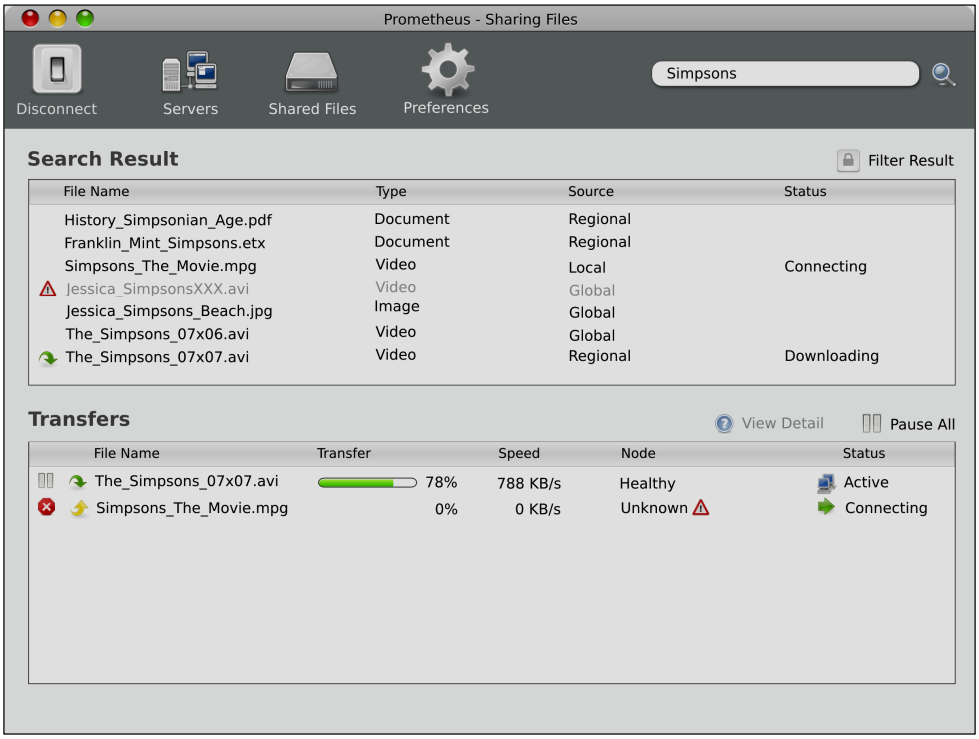


Figure 3.12 Search Result Almenarez Algorithm with security level 4

Legend	
Variable	**Description**
A	peer in the network who wants to calculate other peer trust value (requestor peer)
B	peer whose trust value is being assessed (requested peer)
C	peer in the network that give recommendation to peer A (recommender)
T_{AC}	Trust value of peer C by peer A
T_{CB}	Trust value recommendation of peer B by peer C
T_{ACB}	Trust value of peer B that being calculated by peer A based on recommendation from peer C
V	Action Value
S_{AB}	Number of successful transaction between peer A and peer B
F_{AB}	Number of unsuccessful transaction between peer A and peer B
$F_{AB} + S_{AB}$	Number of total transaction between peer A and peer B
W	Action Weight; defined by transaction that just completed. With successful transaction = 1and by unsuccessful transaction = 0.
m	Security level. This variable is defined by peer itself.
β	Constant between 0-1. This variable is defined by the system to find tune (give balance) between old trust value and the new one.
T_{ACB}^{new}	The up-dated trust value calculation of peer B by peer A

Table 3.6 Legend

Algorithm Almenarez is one of the algorithm that meets almost all of the requirements for a p2p algorithm as previously described, therefore the algorithm is chosen to represent this sub-category. The major drawback from this algorithm is that the algorithm process needs a lot of messaging that flooding the network. The average number of messages (including number of

queries from peer) is much higher than that other algorithms. This could influence the network traffic if there are large numbers of peers in the network. The result of the evaluation of this algorithm is provided in the following table:

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	✓
The presence of dynamic factor.	✓
The presence of scalability factor.	✓
Maintaining peer anonymity	**X**
Performance on: - Computation process - Data storage - Message complexity.	**X**
Implementing a user intervention system	✓
Take into account the role of pre-trusted peer	**X**

Table 3.7 Requirements Algorithm Almenarez

3.6 Sub-category 2b: Global Algorithm with No Distinction Between Direct and Indirect Trust Value

There are five algorithms in this sub-category with various algorithmic procedure for trust value calculation. The same design requirements are used to guide the examination process. Based on the result of the examination process, we single out one algorithm, namely algorithm Garcia-Molina. This algorithm is based on single global trust value for every peer in the network. This global trust value is measured based on aggregation of several local trust values. Referring to the context scenario, when peer A joins the network, it will be automatically placed in one of several clusters (small group of peers) within the network. The system will automatically assign a peer manager to every cluster, namely a pre-trusted peer (in the context scenario, peer F will be the peer manager) that will coordinate the trust value calculation process and the data storage.

Context Scenario:

Based on the context scenario, if peer A sends a request query for Simpsons file to the network, peer A will send a query first to pre-trusted peer F (and other pre-trusted peer as well) whether it has the requested peer. In the case that peer F (or other pre-trusted peer) does not have the file, peer A will send duplicates of the query to its direct neighbours in the similar way as Mengshu algorithm. Because of the waiting time for the response from pre-trusted peer, this process takes longer than that of Mengshu and could result in less number of responses.

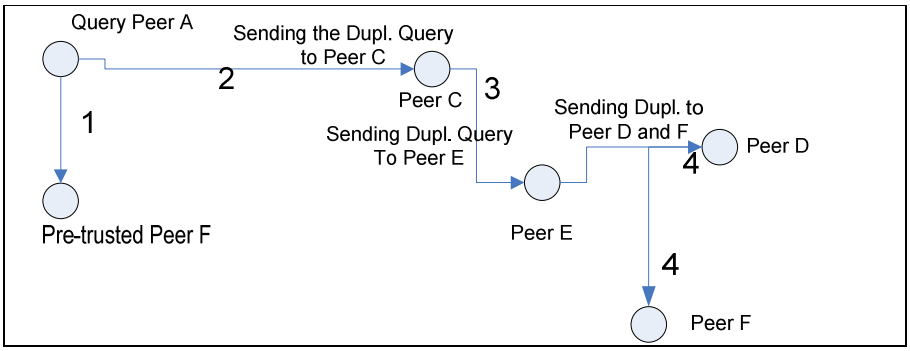


Figure 3.13 Query Propagation Garcia-Molina Algorithm

Figure 3.5 shows the query propagation that occurs in the network. The number indicates the order of the peer destinations.

In this algorithm, the trust value calculation also takes places in the same order as that of Mengshu algorithm. Thus, after peer A receives responses regarding the requested file, it will send query to pre-trusted peer to request for trust values from the responding peers. The result in the display will be already filtered from the malicious peers, as shown in the following figure.

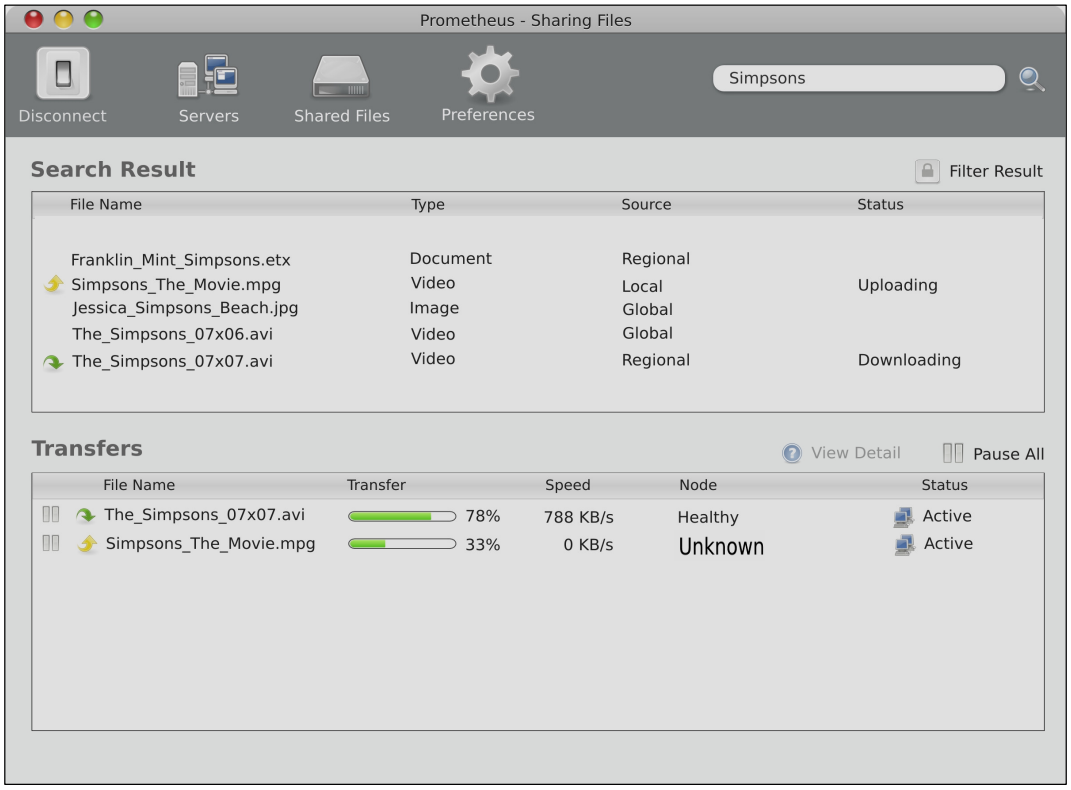


Figure 3.14 Search Result with Garcia-Molina Algorithm

The first step to calculate global trust value is to calculate the local trust value. In this algorithm trust value calculation does not start when a peer wants to measure another peer's trust value, but rather an ongoing process in the network. The peer manager will assign every peer in its cluster to calculate their neighbours' trust values. One peer will be assessed by

several peers in the cluster. The peer manager will aggregate the local trust value to generate a single global trust value. The pre-trusted peer will store this data and duplicate it to send to other pre-trusted peer. To get a clear description on the calculation process, we will provide an example based on the context scenario. Let say, peer F assigns peer A and peer B to calculate peer C's local trust value.

- Calculating the local trust value.
  Since global trust value is an aggregation of several local trust values, peer A and peer B will calculate peer C local trust value C_{AC} and C_{BC} by combining the total number of successful transactions (S_{AC} and S_{BC}) and unsuccessful transactions (F_{AC} and F_{BC}). Thus, local trust value of peer C by peer A and peer B will be:

  $$C_{AC} = S_{AC} - F_{AC}$$
  $$C_{BC} = S_{BC} - F_{BC} \quad \text{..... (I)}$$

- Normalizing the local trust value
  Before aggregating the local trust value, it has to be normalized first in order to avoid false local trust value that submitted by malicious peer (Garcia-Molina et al. 2003). The normalization process is:

  $$C_{AC}^{norm} = \frac{\max\{0, C_{AC}\}}{\sum_C \max\{0, C_{AC}\}}$$

  ……...…….(II)

  $$C_{BC}^{norm} = \frac{\max\{0, C_{BC}\}}{\sum_C \max\{0, C_{BC}\}}$$

  In this normalization process, the maximum value of local trust value between peer C by peer A (C_{AC}) will be divided by total local trust value of peer C by other peer in the cluster. This will give result to C_{AC}^{norm} . The similar process will apply to peer B calculation that will result in C_{BC}^{norm} . The results from peer A and peer B will be sent to peer F to be further processed. Every time a new transaction occurs, the local trust value will be up-dated.

- Aggregating normalized local trust value t_{ABC} by peer F.

  $$t_{ABC} = \frac{1}{n}(C_{AC}^{norm} + C_{BC}^{norm}) \quad \text{..... .(III)}$$

  Peer F will aggregate all of the normalized local trust value of peer C and divide them with the total number of normalized local trust value (n). In this process, it clear that the algorithm does not make distinction between private experience and recommendation. Thus, there is no conditional transitivity factor in this algorithm.

- Incorporating pre-trusted peer & calculating global trust value.
  Garcia-Molina algorithm is only algorithm in p2p network in this research that exploits pre-trusted peer in the calculation process. This algorithm takes advantage of long experience of the pre-trusted peer to improve the trust value robustness against malicious peer. Thus, the calculation of global trust value T_{ABC} of peer C will be:

  $$T_{ABC} = \tfrac{(1-a)}{n}(C_{AC}^{norm} + C_{DC}^{norm}) + aC_{FC} \quad \text{.....(IV)}$$

  where C_{FC} is local trust value peer C that is assigned by pre-trusted peer F and a is the constant between 0-1 to keep the balance between calculation from equation (III) and trust value from pre-trusted peer F.

- When a peer does not trust anybody.
  This algorithm provides also an option when a peer chooses not to trust anybody in the network. This option can be based on the fact the peer is a new member (thus, it does not now anybody) or because the network has high number of malicious peers. According to this algorithm, if peer C as new peer in the network, for example, wants to measure peer B's trust value but it does not trust any peers in the network, the safest way to do is to trust pre-trusted peer:

$$C_{CB} = \begin{cases} \dfrac{\max\{0, C_{CB}\}}{\sum_B \max\{0, C_{CB}\}} & \text{if } \sum_C \max\{0, C_{CB}\} \neq 0 \\ C_{FB} & \text{otherwise} \end{cases} \qquad \text{.....(V)}$$

Legend	
Variable	**Description**
A	peer in the network which is assigned to calculate peer C's trust value
B	peer in the network which is assigned to calculate peer C's trust value
C	peer whose trust value is being assessed
F	Pre-trusted peer
F_{AC}	Number of unsuccessful transactions between peer A and peer C
F_{BC}	Number of unsuccessful transactions between peer B and peer C
S_{AC}	Number of successful transactions between peer A and peer C
S_{BC}	Number of successful transactions between peer B and peer C
C_{AC}^{norm}	Normalized local trust value of peer C by peer A
C_{BC}^{norm}	Normalized local trust value of peer C by peer B
$\max\{0, C_{AC}\}$	Maximum value between interval 0 and C_{AC}
$\max\{0, C_{BC}\}$	Maximum value between interval 0 and C_{BC}
$\sum_C \max\{0, C_{AC}\}$	The sum of maximum value within range 0 and C_{AC}
$\sum_C \max\{0, C_{BC}\}$	The sum of maximum value within range 0 and C_{BC}
C_{CB}	Normalized local trust value of peer C by peer B
t_{ABC}	Aggregated local trust value of peer C based on peer A and peer B local trust value calculation
n	Number of normalized local trust value
T_{ABC}	Global trust value of peer C based on peer A and peer B local trust value calculation
a	Constant between 0-1. Set by the p2p network.
C_{FB}	Trust value of peer B that assigned by pre-trusted peer F.

Table 3.8 Legend

This is the algorithm that provides the most detail and complete information about the trust value computation and its implementation. This is the reason why this algorithm is chosen to represent the others. However, there are also several drawbacks identified from this algorithm such as: this algorithm need high coordination with the network and peers which makes it complicated to implement; also the way how global trust value calculated does not include conditional transitivity factor which can decrease the accuracy of trust value calculation; the last but not least is that this algorithm has high dependency on pre-trusted peer which makes

the pre-trusted peer very attractive for malicious attacks. The result of analysis about this algorithm is presented in the following table.

Design Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	X
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	✓
Maintaining peer anonymity	✓
Performance on: - Computation process - Data storage - Message complexity.	✓
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	✓

Table 3.9 Requirements Algorithm Garcia-Molina

3.7 Summary

From the categorization there are four final chosen algorithms as the representatives:
Trust algorithm with Confirmation Theory (Mengshu et. al, 2002)
Trust algorithm with EigenTrust System (Garcia-Molina et. al, 2003)
Trust algorithm with Ad-Hoc Environment (Almenarez, 2003)
Trust algorithm with Information System (Aberer and Despotovic, 2001)

From these four algorithms, we can summarize the result:

Requirements	Algorithm			
	Mengshu et. al. (2002)	Garcia-Molina et. al. (2003)	Almenarez et. al. (2003)	Aberer and Despotovic (2001)
The presence of non-symmetric factor.	✓	✓	✓	✓
The presence of conditional transitivity factor	✓	X	✓	X
The presence of context or classification.	X	✓	✓	X
The presence reflexivity factor	X	✓	✓	X
The presence of dynamic factor.	✓	✓	✓	✓
The presence of scalability factor.	✓	✓	✓	✓
Maintaining peer anonymity	X	✓	X	X
Performance on: -computation process, -data storage, -message complexity.	✓	X	X	✓
Implementing a user intervention system	X	X	✓	X
Take into account the role of pre-trusted peer	X	✓	X	X

Table 3.10 Final Summary

From the illustration above shows which algorithms can meet the corresponding requirements. The following part will provide description about the result of the table above:

- Requirement: presence of non-symmetric factor

Non-Symmetric means, as previously mentioned, that when peer *p* trusts peer *q*, it does not automatically mean that peer *q* trusts peer *p*. Based on the result of the evaluation, it can be seen that all of the algorithms have incorporated this requirement.

- Requirement: presence of conditional transitivity factor

Conditional Transitivity applies in the algorithm when it makes distinction between peer's own judgment and recommendations that are received from other peers. Mengshu (2003) and Almenarez (2003) they both have this feature in their algorithms. But Garcia-Molina (2003) and Aberer and Despotovic do not have this feature in their algorithms because Garcia-Molina uses only one global trust value for every peer in the network and Aberer and Despotovic uses only one single type of data in the network (complaints) and all peers in the network treat these complaints the same way (no value distinction for each peer). These two different ways of trust value calculation should give impact on the performance of the algorithm. During the simulation test, we will compare which one will deliver a better method.

- Requirement: presence of context or classification factor

Two algorithms clearly mentioned this matter except for Algorithm Aberer and Despotovic and Algorithm Mengshu. Implementing this in the algorithm has a significant added value to the robustness of the algorithm, because there are various types of files that could be shared within the network which means that one peer is possible to have different type of trust value at the same time based on the number of types of files.

- Requirement: presence of reflexivity factor

As previously mentioned in the first chapter, reflexive denotes that before trusting another peer, a peer must trust it self or in other word it trusts its own system. This means that a peer must trust how the algorithm defines and calculates trust value of all peers within the p2p network (Mezzetti, 2003). Since this is an implicit trust (we do not see it in the algorithm process) the only way to show it is to declare it in the beginning of the algorithm. Two of the algorithms (Algorithm from Almenarez (2003) and Algorithm from Garcia-Molina (2003)) have clearly declared this in their algorithms but the two other algorithms did not do it.

- Requirement: present of dynamic factor

Because trust value evolves from time to time, every peer needs to update this value to get an accurate result. All four algorithms have this feature but they implement it in various ways. Algorithm from Mengshu, Aberer and Despotovic and Garcia-Molina are using the new trust value completely and discard the old one. On the other hand, algorithm from Almenarez combines the new and the old trust value. Because Almenarez considers that history is an important part of trust value calculation. By doing the simulation test, we can compare the diversity of the methods and the impact on the result of trust value calculation.

- Requirement: presence of scalability factor

The presence of this requirement is very crucial since the algorithms are designed for p2p network which by nature is a distributed system. All peers in this network have equal rights or privileges to join or leave the network and to initiate or terminate a connection with other peers. All four algorithms implement distributed system in various methods to compute trust value. Algorithm from Aberer and Despotovic (2001) uses p-grid system to store their data (complaints) which can be rapidly accessed by all peers who need it. Algorithm from Mengshu makes use of the closest peers' neighbors to obtain recommendation and to submit the query when a peer wants to request a certain file. Algorithm from Garcia-Molina (2003) implement distributed system by dividing peers into small clusters and incorporate a "neighbourhood watch" for each and every one of them. The last algorithm from Almenarez is simply distributed computation task to each peer in the network so that each peer who needs to compute trust value can obtain their data by "asking" other peer in the network. All of these approaches have benefit and weakness which will be distinguished in by means of simulation test to define which algorithm has the optimal distributed system implementation.

- Requirement: maintaining peer anonymity

Peer anonymity is one important characteristic of p2p network. The trust algorithm for p2p network has to be able to maintain this characteristic in order to keep peer's privacy and security. All four algorithms have implemented this feature in their algorithms in different methods but basically they all have similar idea that when a peer *p* is being assessed by peer *q*, peer *p* does not know where peer *q* gets its information from. Mengshu (2003) does this by choosing peer random recommenders (which is also being assessed in order to get credible recommenders) to acquire accurate recommendations. Peer *p* which is being assessed, does not know who these peer recommenders are and how many recommenders that peer *q* uses in the trust calculation process. Almenarez (2003) uses similar idea with that of Mengshu, only with different formula. Garcia-Molina (2003) which has one global trust for every peer in the network implements this anonymity feature during the calculation of local trust value. In order to calculate peer *p*'s global trust value, the local trust value has to be calculated first. This job is assigned to some peers that have been selected by the peer manager in each cluster. Peer *p* does not know which peers are assigned to do this calculation. Therefore, the peer anonymity is still maintained. The last algorithm from Aberer and Despotovic (2001) has the lowest performance on anonymity because it is possible for every peer in the network to where their complaints data is stored and therefore is very sensitive to malicious peers. Unfortunately, because of the limitation feature of the simulation, this requirement will not be able to be tested.

- Requirement: Performance on computation process, data storage and message complexity

Not only that the algorithms should not give heavy load onto the network, but moreover, the algorithm has to be able also to have minimum load to peers in the network where most of the processes take place so that the network and peer performance can be maintained. All four algorithms have different process and formula to calculate the trust value. This naturally will have different load on the network and peers. Based on the evaluation, two algorithms, namely Algorithm Garcia-Molina and Almenarez give indication that they require heavier load than other two algorithms because of the complex processes. With the help of the simulation test it will be able to be further analyzed which algorithm is able to deliver the best result.

- Requirement: Implementing a user intervention system

As previously mentioned, the peers in the network represent the users' preferences. And as the preferences may vary, the peers should be given the possibility to choose the preferences that are suitable for them. Therefore, the system should give an active role for each peer to give input to the trust value calculation process. From all four algorithms, algorithm Almenarez is the only one that implements this requirement.

- Requirement: Take into account the role of pre-trusted peer.

The valuable experiences of pre-trusted peer can be used to help other peers in the network during trust value calculation in order to obtain more accurate result. From all four algorithms, algorithm Garcia-Molina is the only algorithm that implements this system.

The algorithms have been analyzed based on the theoretical evaluation on their algorithms process with the help of requirements list that has been previously created. The result of each algorithm is already presented and will be studied further by means of simulation in the following chapter. The result of the simulation test will be compared with this evaluation result to provide inputs for designing the hybrid algorithm.

Chapter 4

Simulation Based Evaluation of Current Algorithms

In this chapter the current algorithms that have been discussed in the previous chapter will be once more evaluated based on the simulation result. This is accomplished by implementing the algorithms into p2p network environment simulator with the help of query cycle simulation tool. The performance of the algorithms will be further evaluated and compared with the result from the previous chapter. The simulation tool that is used in this process is called Query Cycle Simulator. The goal of this process is to gather the inputs that can be used for designing the Hybrid Algorithm in the next chapter.

4.1 Query Cycle Simulator

The query-cycle simulator is developed by Condie, Kamvar and Schlosser (2003) under the umbrella of Stanford P2P Sociology Project. This project is dedicated to addressing the issue on the development of p2p technology (Stanford P2P Sociology Project, 2002). One of the issues on the p2p development, according to the Stanford P2P Sociology Project, is the lack of sufficient means for testing a new p2p algorithm. And since testing on a real p2p network is high cost experimentation, the experts from the Project have designed a p2p network simulator that could help to facilitate this need. In addition, the Project has also decided to make the source code widely available on their website in order to give other interesting party an easy access. This following section will provide a short description about the query cycle simulator. Detail information about the simulator is provided in appendix 2.

This simulator is a p2p file sharing network where a peer can submit a request query for a file and other peer can respond to the query, and concluding the process with file sharing between the entities (Condie, Kamvar and Schlosser 2003). The simulation process is taking places in cycles of query. A cycle begins with queries from random peers within the network. Any peers in the network are welcome to respond the queries. The peers that sent the queries will then decide based on the trust algorithm calculation, which entities (that responded the queries) they wish to have transactions with. A cycle of queries is completed when the all entities that sent queries to the network, are able to download the requested file (Condie, Kamvar and Schlosser 2003). Thus, a cycle is a period from when a random number of peers in the network start its activities of sending queries to the network until the time when all peers have downloaded the request files.

The design of the simulator is based on two major types of parameters, namely content distribution and peer behaviour (Condie, Kamvar and Schlosser 2003). The first parameter is dealing with volume and type of the data that are shared within the network and the second one is dealing with all of the activities that the peers perform, such as issuing a query, responding a query, the duration of the session and the online-time of each peer in the network. In order to be able to design a p2p network that similar with a real one, these two main parameters have to be carefully and precisely modelled.

The four chosen algorithms from the previous chapter will be implemented in this simulator separately. Each of these algorithms will be run in a number of cycles and the output data that generated will be compared and evaluated based on the design requirements that are going to be presented in the next section.

The following figure illustrates the query cycle simulator. The first figure shows the initialization phase and the second one shows the result of simulation after a complete cycle.

On the left side of the simulator, a row of attribute panels is available for setting the simulation process.

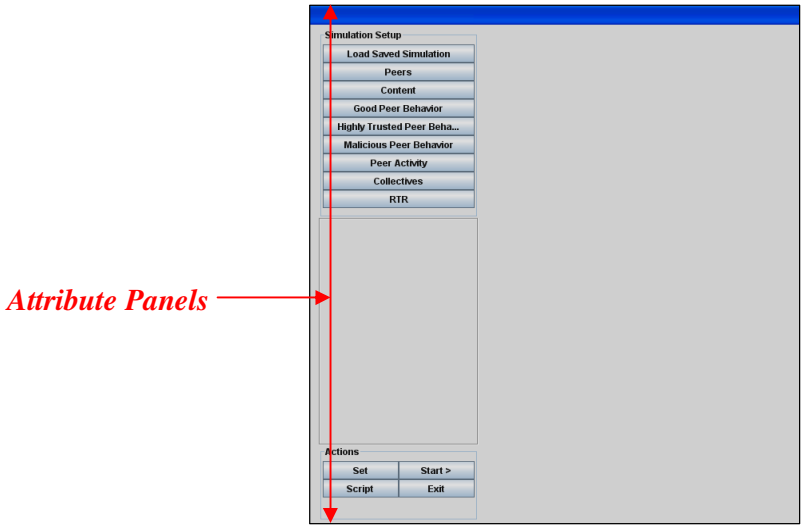


Figure 4.1 Query Cycle Simulator

The initialization phase begins with setting the attribute panels on left side of the simulation bar. There are several panels available for the simulation:

- Peers: this panel is used for setting the number of good, pre-trusted and malicious in the network and number of its neighbours
- Content: setting the distribution of files within the network and setting the number of category of the distributed files
- Good, Malicious and Highly Trusted Peer Behaviour: these three panels are used for setting the good, pre-trusted and malicious peers preferences about the files category and setting the TTL of the query
- Peer activity: this panel is used for setting the duration of peer activity whether they are always online, random active or non active.
- Collective and RTR panels are not used in this simulation because these panels have not perfected yet by the designers of the simulator.

Once these panels are set, the simulation process can be started. An illustration after a complete cycle is provided below.

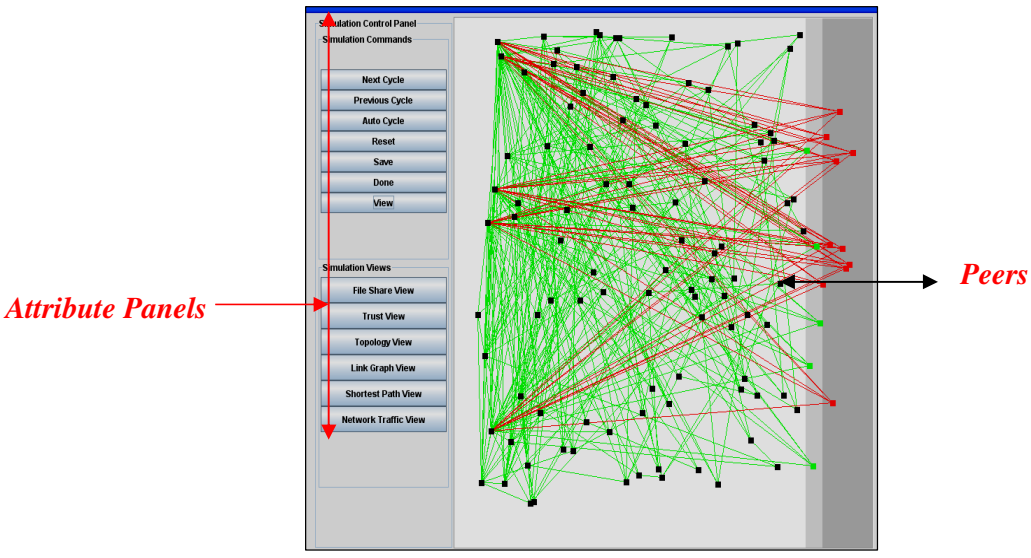


Figure 4.2 Result after a Complete Cycle (Condie, Kamvar and Schlosser, 2003)

The illustration above shows how a p2p network is connected. The dots represent peers in the network. The red dots are the entities with very low trust value (malicious entities); whereas the black and green ones are the average and high trust value peers respectively. Correspondingly, the red lines represent the unsatisfied transactions and the green lines are showing successful transactions. To get a better description about the simulation process, the following part will explain the simulation procedure:

1. Setting the attribute panels (left side of the display).
2. A peer submits a file-request query to the network
3. A peer or more will respond the query.
4. The peer that submits the query receives response from other peer(s).
5. The peer that submits the file-request query will calculate the trust value of the responding peer(s).
6. In order to calculate trust value, the peer needs recommendation from other peers in the network.
7. To obtain recommendation(s), the peer will have to submit another query to the network, namely recommendation-request query, asking for recommendation(s) regarding the responding peer(s).
8. Other peer(s) in the network which has requested recommendation(s) will respond and send its recommendation(s).
9. Based on the recommendation(s) and its own judgment, the peer will be able to calculate trust value.
10. Based on the calculation, the peer will decide which responding peer it chooses to be the source of the requesting file.
11. The transaction (file sharing) can be started between the peers.

4.2 Validation of Simulated Algorithms

This section will explain the validation process of the simulated algorithm. Which means the algorithm which is being implemented in the simulation tool will be tested whether it has been correctly implemented. This process is done by comparing the result from the mathematical formula that provided by the algorithms' designers with the result from the simulation tool. These two results will have to match to get a valid result. In order to obtain similar result, we need to define similar network setting (such as: number of good peers, number of malicious peers, number of peer's neighbour and number of transactions between peers) for both formula calculation and simulation result. During the validation process, the network setting will be frequently changed to generate variation in the outcomes.

4.2.1 Algorithm from Aberer and Despotovic (2001)

For the algorithm from Aberer and Despotovic (2001), we design various network settings starting from the minimum number of peer (setting A) to see if the simulation can produce a valid result. Beside the minimum number, there are also an extreme number of peers (all good peers or all malicious peers) to see if the simulation will still behave accordingly.

Two important notes for this validation process:

- The observation of the simulation is performed during cycle 1.
- To show the process of mathematical calculation and simulation result, we provide an example of a setting (setting D) with step by step calculation and simulation result. The rest of results of other settings are provided in the table.

- Because there are more than one peers in the network, the peer that is being assessed in the validation is always peer 1 (see figure 4.1 below for more information).

Example of Calculation: Setting D

Setting D for this algorithm is:
- total number of peer: 3
- total number of good peer: 2
- total number of malicious file: 1

- ***Calculation from mathematical formula:***

Define the value of several variables:

$f = 5$; $s = 5$

$CR_{CB} = 1$; $CF_{CB} = 1$

$CR_{Ai} = 1$ (number of complaints that peer A received from other peer i which is stored at peer A during cycle 1)

$CF_{Ai} = 1$ (number of complaints that peer A filed regarding other peer i which is stored at peer A during cyle 1)

$CR_{Ai}^{avg} = 1/1=1$ (CR_{Ai} divided by number of cycle)

$CF_{Ai}^{avg} = 1/1=1$ (CF_{Ai} divided by number of cycle)

Normalization of complaint filed and complaint received:

$CR_{CB}^{norm} = CR_{CB}(1-(\frac{s-f}{s})^s)$

$CF_{CB}^{norm} = 1(1-(\frac{5-5}{5})^5 = 1$

$CF_{CB}^{norm} = CF_{CB}(1-(\frac{s-f}{s})^s) = 1(1-(\frac{5-5}{5})^5) = 1$

Calculate trust value:

$cr_i^{norm}(B)cf_i^{norm}(B) \le (\frac{1}{2} + \frac{4}{\sqrt{cr_A^{avg} c_A^{avg}}})^2 cr_A^{avg} cf_A^{avg}$

$1*1 \le (\frac{1}{2} + \frac{4}{\sqrt{1*1}})^2 *1*1$

$1 \le 20.5$.....then 1 else 0 Thus, trust value peer $q = 1$

- ***Result from simulation:1***

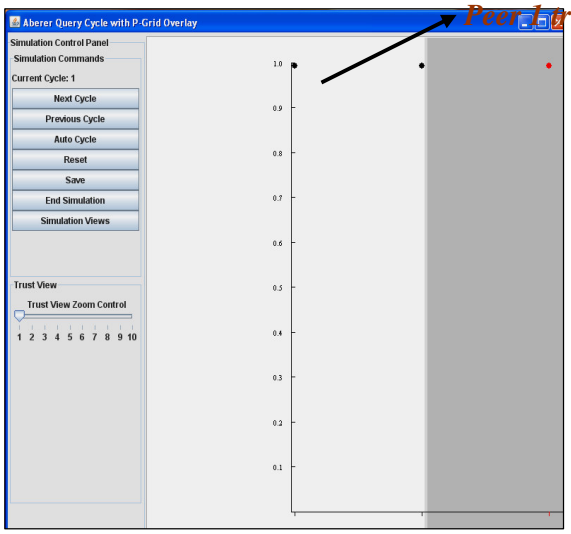


Figure 4.3 Result Simulation Aberer and Despotovic

The figure above shows the result of the validation according to setting D where there are in total three peers in the network (two good peers and one malicious peer). The y-axis in the illustration denotes the trust value while the x-axis shows the number of the peer in the network. The first dot on the x-axis represents peer1, the middle dot represents the second peer and the last red dot is of course the third malicious peer. As previously explained, peer 1 is being assessed in this validation, and the result can be seen here that peer 1 (first dot on the left) has a trust value 1. Beside peer 1, we can see the trust values of peer 2 and peer 3 which happen to be the same value as that of peer 1.

The trust value of peer 1 according to the other settings can be seen in the following table:

CYCLE	RESULT	SETTING					
		A Total peer: 2 Good peer: 1 Malicious: 1 Neighbour: 1	B Total peer: 2 Good peer:2 Malicious: 0 Neighbour: 1	C Total peer: 2 Good peer: 0 Malicious: 2 Neighbour: 1	D Total peer: 3 Good peer: 2 Malicious: 1 Neighbour: 1	E Total peer: 3 Good peer: 3 Malicious: 0 Neighbour: 1	F Total peer: 3 Good peer:0 Malicious: 3 Neighbour:1
1	Math formula	1	1	-	1	1	-
	Simulation	1	1	-	1	1	-
2	Math formula	0	1	-	1	1	-
	Simulation	0	1	-	1	1	-
3	Math formula	0	1	-	1	1	-
	Simulation	0	1	-	1	1	-
4	Math formula	0	1	-	1	1	-
	Simulation	0	1	-	1	1	-
5	Math formula	0	1	-	1	1	-
	Simulation	0	1	-	1	1	-

Table 4.1 Validation Result Aberer and Despotovic

From the result table above, we can see that the result from formula calculation corresponds well with the result from the simulation. For the first cycle which is right after the initialization phase, the peer trust value is still one for all setting (except for setting C and F) because there is still no complaint filed from other peers. Setting C and F which contain only malicious files generate network failure directly after the initialization phase. On the other hand, for setting B and E where there are no malicious peers found, the peer trust value remains stable because of 100% probability of having successful transactions with other.
One last important note is that this is only an observation for a limited number of cycles (cycle 1 to cycle 5), still an early stadium. As the cycle continues, the network reaches more stable stadium which naturally will yield different result.

4.2.2 Algorithm from Mengshu (2003)

For the algorithm from Mengshu, we design various network settings starting from the minimum number of peer (setting A) to see if the simulation has can produce a valid result. Beside the minimum number, there are also an extreme number of peers (all good peers or all malicious peers) to see if the simulation will still behave accordingly.

Two important notes for this validation process:

- For the sake of simplicity, during this validation, the constant α has been set to 0.2, 0.5 and 0.8 instead of random number.
- The observation of the simulation is performed during cycle 1 of the simulation.
- To show the process of mathematical calculation and simulation result, the first setting (setting A) is provided with step by step calculation and simulation result. The rest of results of other settings are provided in the table.
- Because there are few number of peers in the network, the peer that is being assessed in the validation is always peer 1 (see figure 4.2 below for more information) whether it is good or malicious peer

Example of Setting A

Setting A for this algorithm is:

- total number of peer: 3
- total number of good peer: 2
- total number of malicious file: 1

- ***Calculation from mathematical formula:***

Define variables:

$S_{AB} = 1$
$F_{AB} = 1$
$C_{AC} = 0$
$C_{AD} = 0$
$C_{CB} = 0$
$C_{DB} = 0$

Define local trust from private experience:

$$C_{AB} = \frac{S_{AB} - F_{AB}}{S_{AB} + F_{AB}} = (1\text{-}1)/(1+1) = 0$$

Define certainty factor from other peers:

$C_{ACB} = C_{CB} \times \max\{0, C_{AC}\}$
$C_{ADB} = C_{DB} \times \max\{0, C_{AD}\}$
$C_{ACB} = 0$ And $C_{ADB} = 0$

Define certainty factor:

$CF(D)_{AB} = CF_{DA}(H) + CF_{DB}(H) - CF_{DA}(H) \times CF_{DB}(H) = 0+0\text{-}0x0 = 0$

Calculate trust value (with α: 0.5)

$T_{DE} = \alpha CF(D,C) + (1-\alpha) CF(D)$
$= (0.5x0) + (0.5x0) = 0$

- ***Simulation result: 0***

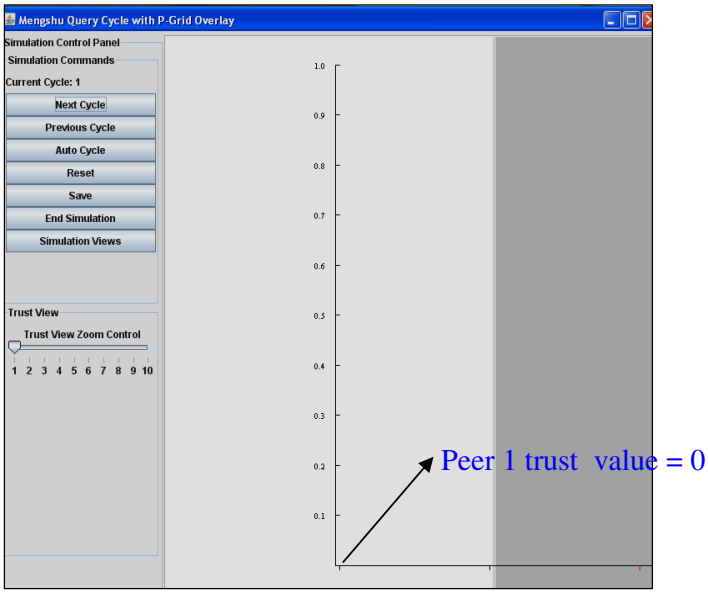


Figure 4.4 Simulation Result Mengshu

The figure above shows the result of the validation according to setting A where there are in total two peers in the network (two good peers and one malicious peer). The y-axis in the illustration denotes the trust value while the x-axis shows the number of the peer in the network. The first dot on the x-axis represents peer 1 and the next dot represents the second peer. As previously explained, peer 1 is being assessed in this validation, and the result can be seen here that peer 1 has a trust value 0. Beside peer 1, we can also see the trust values of peer 2 which happens to be the same value as that of peer 1.

The trust value of peer 1 according to the other settings can be seen in the following table:

α	Result	Setting					
		A Total peer: 2 Good peer: 1 Malicious: 1 Neighbour: 1	B Total peer: 2 Good peer: 2 Malicious: 0 Neighbour: 1	C Total peer: 2 Good peer: 0 Malicious: 2 Neighbour: 1	D Total peer: 3 Good peer: 2 Malicious: 1 Neighbour: 1	E Total peer: 3 Good peer: 3 Malicious: 0 Neighbour: 1	F Total peer: 3 Good peer:0 Malicious: 3 Neighbour: 1
0.2	Mathematical calculation	0	0.2	-	0	0.2	-
	Simulation	0	0.2	-	0	0.2	-
0.5	Mathematical calculation	0	0.5	-	0	0.5	-
	Simulation	0	0.5	-	0	0.5	-
0.8	Mathematical calculation	0	0.8	-	0	0.5	-
	Simulation	0	0.8	-	0	0.5	-

Table 4.2 Validation Result Mengshu

From the table above, we can see that the result from formula calculation corresponds well with the result from the simulation. We can also see that for setting B and E trust value of the peer is directly going up even though it is observed during the first cycle, unlike other setting that still yield low trust value. This is understandable because all of these setting have only limited numbers of peers which make those peers do not have many choices to have transaction with other peers. On setting B and E, there are no malicious peers which give probability of having 100% successful transactions. This process leads to higher trust value for the peers within the network. On the other hand, setting C and F which have no good peers lead to network failure because no peers will be able to have transaction due to low trust value. One last important note is that this is only an observation for a cycle (cycle 1), an early stadium. As the network reaches later stadium the result can change.

4.2.3 Algorithm from Almenarez (2003)

For the algorithm from Almenarez (2003), we design various network settings starting from the minimum number of peer (setting A) to see if the simulation has can produce a valid result. Beside the minimum number, there are also an extreme number of peers (all good peers or all malicious peers) to see if the simulation will still behave accordingly.

Two important notes for this validation process:

- For the sake of simplicity, during this validation, the security level m has been set to 2, 5 and 8 instead of random number.
- For the sake of simplicity, during this validation, the constant β has been set to 0.5
- The observation of the simulation is performed during cycle 1 of the simulation.
- To show the process of mathematical calculation and simulation result, the first setting (setting A) is provided with step by step calculation and simulation result. The rest of results of other settings are provided in the table.
- Because there are few number of peers in the network, the peer that is being assessed in the validation is always peer 1 (see figure 4.3 below for more information) whether it is good or malicious peer

1. Setting A:

- ***Calculation from mathematical formula:***

Define variable value:
$T_{CB} = 0.5$
$T_{AC} = 0.5$
$F_{AB} = 0$
$S_{AB} = 1$
$m = 2$
$\beta = 0.5$

Thus, the trust calculation of peer C by peer A will be:
$T_{ACB} = T_{CB} \times T_{AC}$ = 1/1 (0.5*0.5) = 0.25

- ***Result from simulation:0.25***

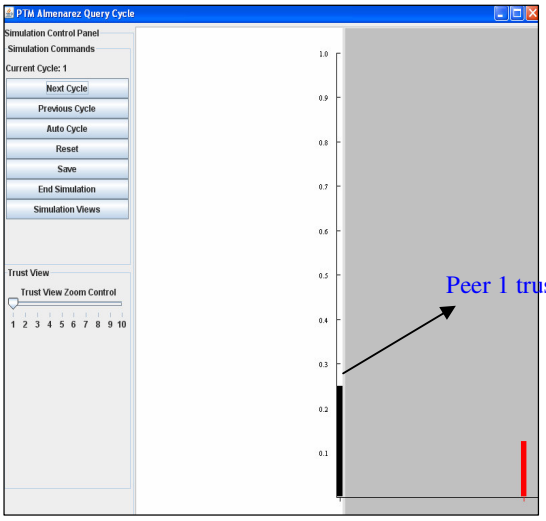


Figure 4.5 Simulation Result Almenarez

If the transaction is successfully completed, the trust value T_{ACB} will be up dated with:

$$V = (1 - \frac{F_{AB}}{S_{AB} + F_{AB}}).W^{(m)} = (1 - \frac{0}{1+0}).1^2 = 1$$

$T_{ACB}^{new} = V.\beta + T_{ACB}.(1-\beta) = (1 \times 0.5) + (0.25 \times 0.5) = 0.625$ (this value is not displayed in the table).

The figure above shows the result of the validation according to setting A where there are in total two peers in the network (two good peers and one malicious peer). The y-axis in the illustration denotes the trust value while the x-axis shows the number of the peer in the network. The first dot on the x-axis represents peer 1 and the next dot represents the second peer. As previously explained, peer 1 is being assessed in this validation, and the result can be seen here that peer 1 (first dot on the left) has a trust value of 0.25. Beside peer 1, we can also view the trust values of peer 2 which has lower trust value (0.15).
The trust value of peer 1 according to the other settings can be seen in the following table:

m	Result	Setting					
		A Total peer: 2 Good peer: 1 Malicious: 1 Neighbour: 1	B Total peer: 2 Good peer: 2 Malicious: 0 Neighbour: 1	C Total peer: 2 Good peer: 0 Malicious: 2 Neighbour: 1	D Total peer: 3 Good peer: 2 Malicious: 1 Neighbour: 1	E Total peer: 4 Good peer: 4 Malicious: 0 Neighbour: 1	F Total peer: 4 Good peer:0 Malicious: 4 Neighbour: 1
2	Mathematical calculation	0.25	1	-	0.25	1	-
	Simulation	0.25	1	-	0.25	1	-
5	Mathematical calculation	0.25	1	-	0.25	1	-

	Simulation	0.25	1	-	0.25	1	-
8	Mathematical calculation	0.25	1	-	0.25	1	-
	Simulation	0.25	1	-	0.25	1	-

Table 4.3 Validation Result Almenarez

From the table above, we can see that the result from formula calculation corresponds well with the result from the simulation. We can also see that for setting B and E trust value of the peer is directly going up even though it is observed during the first cycle, unlike other setting that still yield low trust value. This is understandable because all of these setting have only limited numbers of peers which make those peers do not have many choices to have transaction with other peers. On setting B and E, there are no malicious peers which give probability of having 100% successful transactions. This process leads to higher trust value for the peers within the network. On the other hand for setting C and F which contain no good peers, the network is directly going down because no peers will be able to have transaction because of the low trust value. Setting A and D which contain good and malicious peer, the trust value is lying between one and zero because having transaction with malicious peer can lead to a lower trust value. One last important note is that this is only an observation for a cycle (cycle 1), an early stadium. As the network reaches later stadium the result are going to change, except for setting C and F.

4.2.4 Algorithm with EigenTrust from Garcia-Molina (2003)

For the algorithm from Garcia-Molina (2003), we design various network settings starting from the minimum number of peer (setting A) to see if the simulation has can produce a valid result. Beside the minimum number, there are also an extreme number of peers (all good peers or all malicious peers) to see if the simulation will still behave accordingly.
Three important notes for this validation process:

- For the sake of simplicity, during this validation, the constant *a* has been set to 0.2, 0.5 and 0.8 instead of random number.
- The observation of the simulation is performed during cycle 1 of the simulation.
- To show the process of mathematical calculation and simulation result, the first setting (setting A) is provided with step by step calculation and simulation result. The rest of results of other settings are provided in the table.
- Because there are few number of peers in the network, the peer that is being assessed in the validation is always peer 1 (see figure 4.4 below for more information) whether it is good or malicious peer.

Example of Setting A:

- ***Calculation from mathematical formula*** *(a =0.5)* **:**

Define variables:

$S_{AC} = 3;\ S_{BC} = 3$

$F_{AC} = 1;\ F_{BC} = 3$

$\sum_C \max\{0, C_{AC}\} = 4$

$\sum_C \max\{0, C_{BC}\} = 4$

$C_{FC} = 0.5$
$n = 2$
$C_{AC} = S_{AC} - F_{AC} = 3\text{-}1 = 2$
$C_{CB} = S_{BC} - F_{BC} = 3\text{-}1 = 2$

Define local trust value:

$$C_{AC}^{norm} = \frac{\max\{0, C_{AC}\}}{\sum_C \max\{0, C_{AC}\}}$$
$$= 2/4 = 0.5$$

$$C_{BC}^{norm} = \frac{\max\{0, C_{BC}\}}{\sum_C \max\{0, C_{BC}\}}$$
$$= 2/4 = 0.5$$

Calculate trust global trust value:

$$T_{ABC} = \frac{(1-a)}{n}(C_{AC}^{norm} + C_{BC}^{norm}) + aC_{FC} \quad = \frac{(1-0.5)}{2}(0.5+0.5) + (0.5*0.5) = 0.5$$

- ***Result from simulation:0.5***

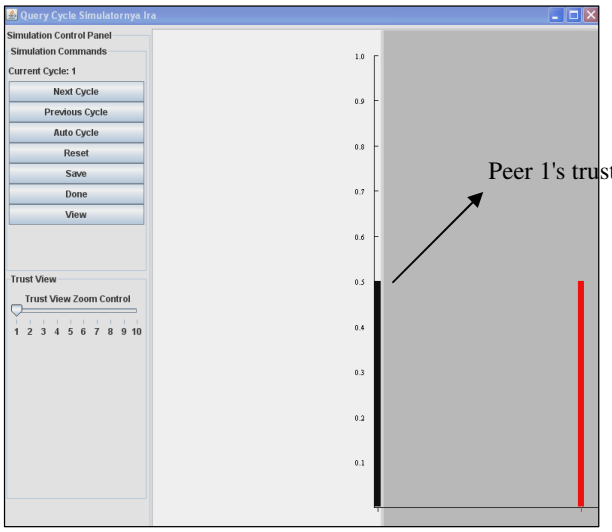


Figure 4.6 Simulation Result Garcia-Molina

The figure above shows the result of the validation according to setting D where there are in total three peers in the network (two good peers and one malicious peer). The y-axis in the illustration denotes the trust value while the x-axis shows the number of the peer in the network. The first dot on the x-axis represents peer1 and the next dot represents the second peer. As previously explained, peer 1 is being assessed in this validation, and the result can be seen here that peer 1 (first dot on the left) has a trust value 0.5. Beside peer 1, we can also see the trust value of peer 2 which happens to be the same value as that of peer 1.
The trust value of peer 1 according to the other settings can be seen in the following table:

a	Result	Setting					
		A Total peer: 2 Good peer: 1 Malicious: 1 Neighbour: 1	B Total peer: 2 Good peer: 2 Malicious: 0 Neighbour: 1	C Total peer: 2 Good peer: 0 Malicious: 2 Neighbour: 1	D Total peer: 3 Good peer: 2 Malicious: 1 Neighbour: 1	E Total peer: 4 Good peer: 4 Malicious: 0 Neighbour: 1	F Total peer: 4 Good peer:0 Malicious: 4 Neighbour: 1
0.2	Mathematical calculation	0.2	0.2	-	0.2	0.2	-
	Simulation	0.2	0.2	-	0.2	0.2	-
0.5	Mathematical calculation	0.5	0.5	-	0.5	0.5	-
	Simulation	0.5	0.5	-	0.5	0.5	-
0.8	Mathematical calculation	0.05	0.05	-	0.05	0.05	-
	Simulation	0.05	0.05	-	0.05	0.05	-

Table 4.4 Validation Result Garcia-Molina

From the table above, we can see that the result from the formula calculation corresponds well with the result from the simulation. We can also see that for all setting the trust value is set to 0.5, because this is still the first cycle, right after the initialization phase where according to this algorithm, the trust value is set to 0.5. On the other hand for setting C and F which contain no good peers, the network is directly going down because this algorithm is very dependent to the presence of high trust peer or good peer which usually will initiate network transaction and break up malicious file (Garcia-Molina et. al, 2003). If there is no high trust peer, the network must have at least one good peer to make the network function.
One last important note is that this is only an observation for a cycle (cycle 1), an early stadium. As the network reaches later stadium the result are going to stabilize, except for setting C and F

4.3 Simulation Setting

This section will describe the setting procedure of the simulation process, which will begin with some explanation about elements in the simulation tool and their role in the process. This will be followed by information about reduction and simplification measure during the simulation. If this step is done, the final step in the setting process will be providing the feeding data for the simulation.

4.3.1 Defining Simulation Elements

Within the simulation display (see figure 4.2), there are elements that can be recognized, such as independent variables (input variables), dependent variables (output variable) and control variable. Independent variables or Input variables are variables where the data that will be used for the simulation, is fed into. Just like the name, these variables are not dependent on other variables. Control variables are variables that are used for managing the simulation process to produce the desirable outputs. And finally the output variables are the variables which produce output data.

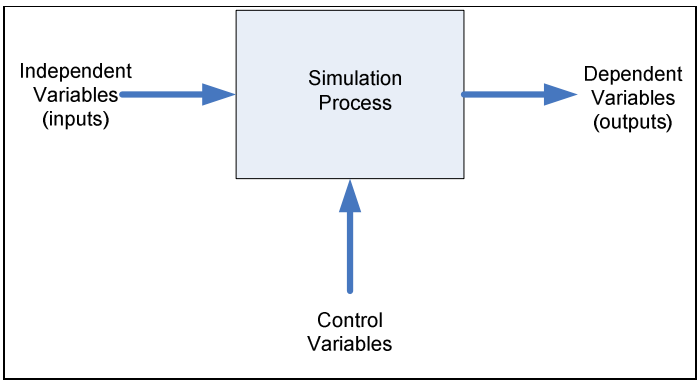


Figure 4.7 Simulation Elements

Independent Variables (Input Variables)

- Variable Number of Peers: through this variable the total number of peers in the network.
- Variable Number of Good and Malicious peers: variable that sets the number of good and malicious peers in the network.
- Variable Number of Peers' Neighbours: define the number of peers' neighbours within the network.

Dependent Variables (Output Variables)

- Variable Number of Responses. This will display the total number of good response and malicious response.
- Variable Files Shared. This will provide information on total number of files that have been shared (downloaded).
- Variable Trust Value. This will display the trust value of each single peer in the network.
- Variable Number of Infected Files. This will display total files malicious files that have been shared (downloaded) during the simulation.
- Variable Computation Time. This variable will display total time of each peer to compute trust value in every cycle during simulation.

Control Variable

- Variable Number of Cycles: through this variable we can control the number of cycle, when it starts and when it should stop.
- Variable Animate and Colour Trust: through this variable we can control whether if we want to see the connection process and whether if we want to see the difference peers' trust value through the cycle. There are three colours, namely red (malicious), black (neutral) and green (high trust value).
- Variable Reset, Save and Done: through these four variables we can reset the simulation display, back to the previous cycle; or save the simulation data; and finally exiting the simulation process (back to initialization phase).

4.3.2 Rules of Simplification

During the simulation process, there will be some simplifications in the simulation tool because it is likely impossible to simulate a full real p2p network environment. Not only complex and large but building the "real" p2p network simulation is highly expensive.
The simplifications in the simulation are:

- There are many forms of malicious activities that take place in the p2p network. But we only define one type of malicious activity in the simulation, namely the single malicious peer, where a malicious peer is acting on its own, without any support or collaboration with other malicious peer in the network.
- The number of peers within the real p2p network is very dynamic. They come and go unexpectedly. For the sake of simplicity, the number of entity in the simulation remains the same within a series of cycles.
- There are endless types of data files that are distributed within the p2p network. In the simulation, a choice of maximum 20 types of categories is available. For example: if the peers are sharing music files, there are maximum 20 various kind of music files available within the network.
- The query TTL is always set to medium.

4.4 Test Case

This section will provide information on the testing procedure on each algorithm. There will be several test cases presented here; each case will explain the test procedure for each requirement that have been discussed before. There are in total ten requirements: reflexive, distributed system, context/classification, maintaining anonymity of the peers, dynamic, conditional transitivity, non-symmetric, performance, implement a user intervention system and take into account the role of pre-trusted peer. As also previously mentioned, not every requirement will be simulated, due to the nature of requirement or the limited feature of the simulation tool.

4.4.1 Test Case 1: Reflexive

As mentioned before, this test case needs no simulation, because it is not possible (within the scope of this research) to analysis the difference between the algorithms that clearly satisfy this requirement and algorithms that only implicitly mention this requirement in the algorithm. To test the differences among them wider scope of research will be needed. The test will be conduct by theoretical discussion of the algorithm.

4.4.2 Test Case 2: Scalability

As previously discussed, the presence of a distributed system in trust algorithm is very crucial since the algorithm is designed for p2p network which by nature is a distributed system. An algorithm that is suited for p2p network should be able to cope with dynamic number of peers in the network, for both limited and large number of peers. The goal is to compare among four algorithms to evaluate which algorithm has the best scalability performance. This test case is using all of the all three input variables (see page 57 about input variables) and using the output variable: number of total responses from all peers in the network in each cycle. Higher number of peer responses shows that the algorithm has large network coverage, and therefore has better scalability performance than algorithm that has

lower peer responses. The similar rule applies for the second output variable where the number of files shared shows the ability of the algorithms to cope with the network environment. Higher number of shared files illustrates better performance than algorithms that only generates limited shared files.

4.4.3 Test Case 3: Context/Classification

Context is important part of trust element that will define as the accuracy of the trust value computation. Because of various types of files that are being shared in p2p network, the trust value calculation should distinguish the calculation based on the each type of file, because a peer which has high trust value in music files type does not necessarily have high trust value on other type of files. Mengshu and Aberer do not mention this requirement in their design, but Almenarez and Garcia-Molina include this requirement, although they do not provide a detail description during the designing and validation process. However, the simulation has limited ability regarding this test case. There is only one context that can be simulated during the test. Therefore, there will be no simulation test for this test case, but a brief discussion about the algorithm instead.

4.4.4 Test Case 4: Dynamic

The fourth test case is dealing with dynamic of the trust value. Because trust value evolves through time, every peer needs to get the most updated information to be able to get an accurate result. The measurement for this test case is based on the three input variables (see page 57 about input variables) and output variable: trust value. This test will focus on a certain peer in the network based on its trust value evoluation. A well-defined trust algorithm should have computation process that results in trust value that corresponds with the peer behavior. A peer with good behavior should be rewarded with high trust value and peer with bad history (malicious peer) should be punished with low trust value. The various calculation processes from four different algorithms will yield different results which can be seen and evaluated later after the simulation process.

4.4.5 Test Case 5: Conditional Transitivity

The fifth test case is dealing with conditional transitivity which applies in the algorithm when it makes distinction between peer's private experience and recommendations that are received from other peers. The influence of these different implementations will be measured based on the three input variables (see pages 57 about input variables) and output variable: number of infected files. This test will observe the total number of infected files in the network through the cycles. The simulation will create a p2p network that has an extremely high number of malicious peers (hostile environment) which will give false recommendations during the trust value calculation. This will give impact to the trust value which will be indicated with the number of infected files shared. If the algorithm computation is accurate, the number of infected files shared should be getting lower with each going cycle.

4.4.6 Test Case 6: Peer Anonymity

The sixth test case is dealing with peer anonymity which is one important characteristic of p2p network. The trust algorithm for p2p network has to be able to maintain this characteristic in order to keep peer's privacy and security. The peer anonymity in this research is defined as the confidentiality of the information which is needed during the trust

value computation, or in other words, peer A that is being assessed by peer B, does not know where peer B gets its information from. Because of the limited capability of simulation, we will conduct the test in different way, namely by observing the algorithm formula itself that will be presented in the evaluation section.

4.4.7 Test Case 7: Non-Symmetric

This is one of the requirements that also need no simulation, because the test case can be conducted by analysing the algorithm mathematical formula. All of the algorithms that are being evaluated in this research satisfy this non-symmetric requirement. The discussion about this test case is presented in the following section.

4.4.8 Test Case 8: Performance on Computation Process, Data Storage and Message Complexity

Not only that the algorithms should not give heavy load onto the network, but more over, the algorithm has also be able to have minimum load to peers in the network where most of processes will take place. All four algorithms have different process and formula to calculate the trust value. This naturally will have different load on the network and peers. The test uses all of the three inputs variables (see page 57 about input variables) and output variable: computation time. This test will compare which algorithm is able to deliver the best result. This is done by comparing the time that the algorithms need for their computation process and also an evaluation on the mathematical formula.

4.4.9 Test Case 9: Implementing A User Intervention System

This test case is dealing with the role of the peers in the network during the trust value calculation. This test case needs no simulation and will be conducted by analyzing the formula of each algorithm.

4.4.10 Test Case 10: Take Into Account The Role of Pre-Trusted Peer

The test case is dealing with the role of pre-trusted peer in the algorithm. Because this can be done by analyzing the formula, there will be no simulation test needed for the test case.

4.5 Result and Evaluation

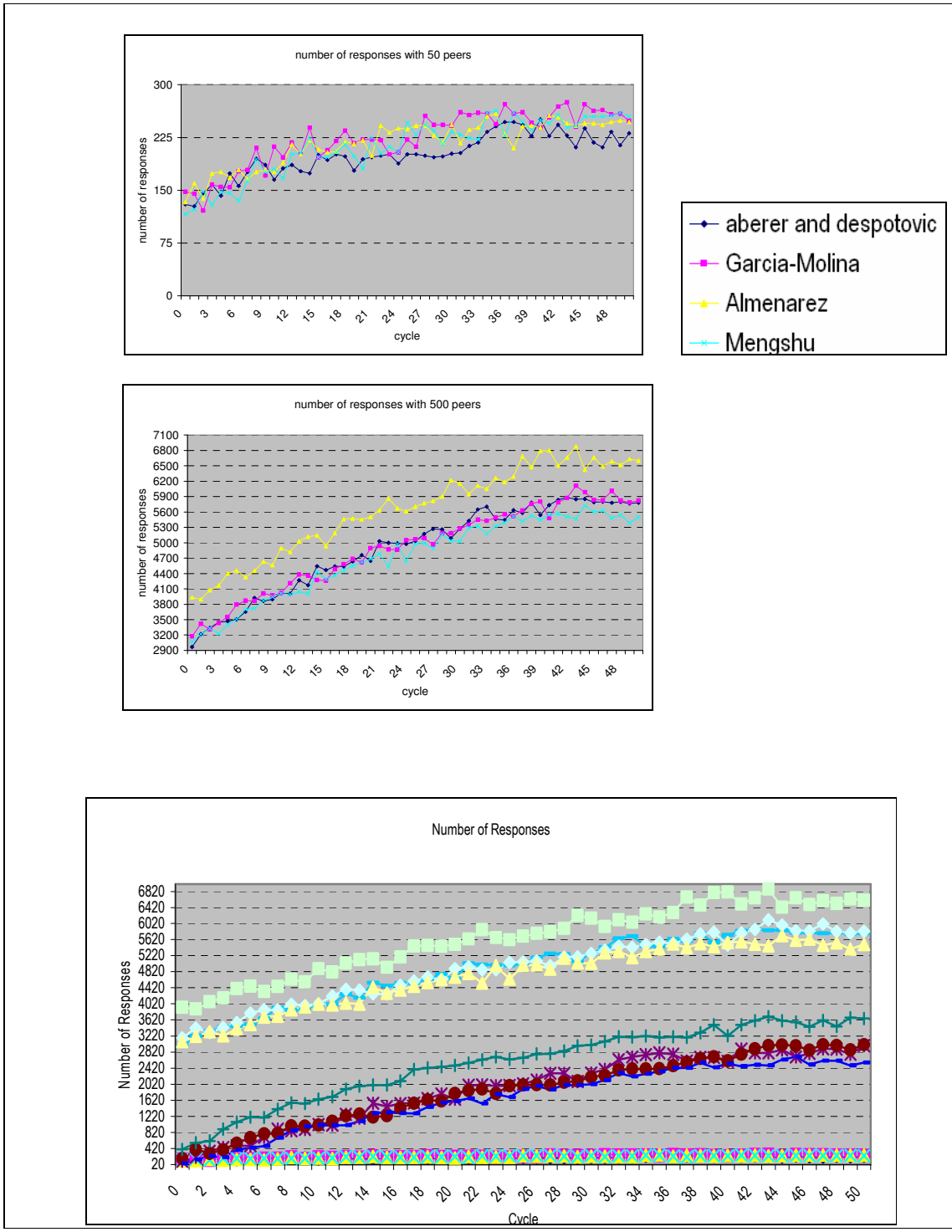
The testing process is started with the initialization phase by setting the data input, as shown in the previous section. Once this is set, the testing process can begin. The result will be presented in the next following section. This result will be compared with the previous information from chapter three to see whether the result from the simulation confirms the information.

4.5.1 Test Case 1: Reflexivity

The reflexivity requirement can not be seen in the formula, but it needs to be stated in the beginning of the trust value calculation process in order to meet the requirement. There are only two algorithms that meet this requirement, namely the algorithms from Garcia-Molina and Almenarez. Two other algorithms do not state this requirement. However, even though

these last two algorithms do not state their reflexivity in the algorithm, it does not mean that they do not apply the requirement in their algorithm. Because the subject of this test case is outside the scope of this research, we will only consider that an algorithm will satisfy this requirement only if it states clearly in their algorithm. Therefore, only algorithms from Garcia-Molina and Almenarez that meet this requirement based on this definition.

4.5.2 Result for test case 2: Scalability



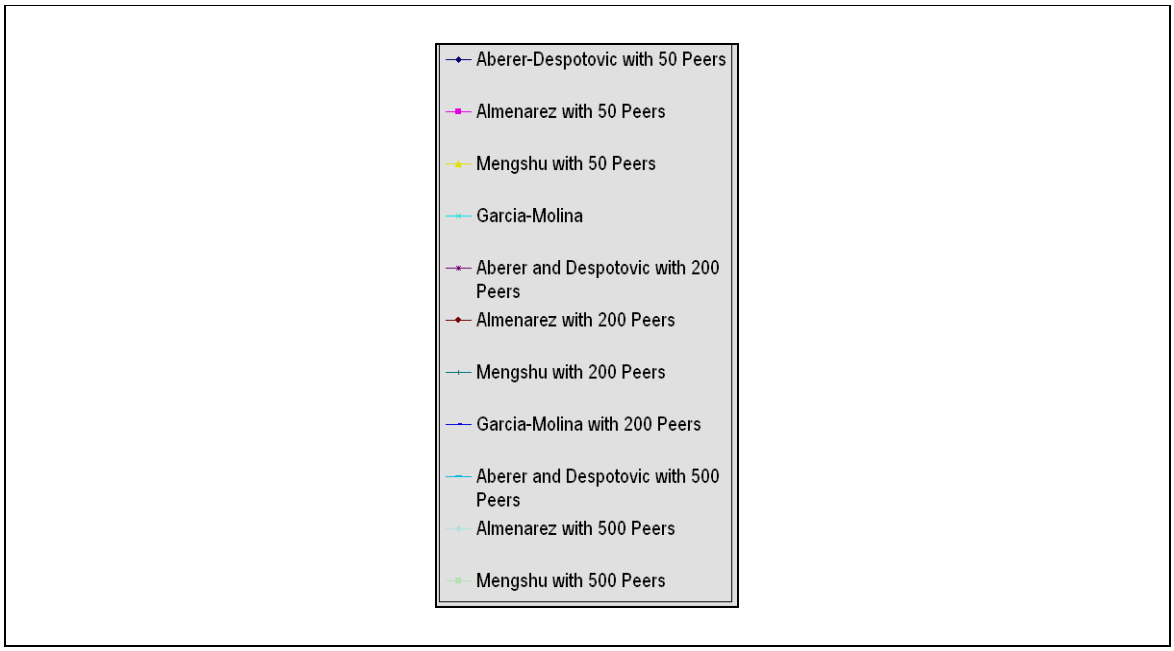


Figure 4.8 Test Result Numbers of Responses

General Result:
Based on the result of the simulation, it can be seen that the algorithms performs differently for network with small number of peer and network large number of peer. The first figure of 4.8 shows tendency of a graphic with low slope or gradient. Round cycle 30, the graphic is starting to get horizontal, where it indicates that the network is reaching a mature stadium. On the other hand, in the second figure, the graphic shows more steep gradient, and becoming horizontal in later cycle. It shows that the network with large number of peers needs more time to reach mature stadium than a small network as shown in the last figure of 4.8 where the number of responses are combined together (with additional information of network with 200 peers).

Algorithm Aberer: this algorithm has a bad result especially for the large network. The result for the small network is similar with the result from other algorithms because the number of the peers is still limited. On the other hand, for large network, this algorithm is performing far under Mengshu algorithm. This is caused by how the queries propagate within the network as previously mentioned in chapter three. The query is sent randomly to the network until the query TTL is up. The random behaviour leads to limited number of responses as seen in the figure because the query is widely spread within the network. This confirms the result of previous theoretical evaluation.

Algorithm Almenarez: this algorithm performs well for small network but less for large network because of how the query is submitted to the network. This algorithm is basically flooding the network the peer query, which does not give significant influence in small network. On the other hand, with a large number of peers in the network, this system will cause heavy traffic that leads to less number of responses as shown in the figure above.

Algorithm Garcia-Molina: this algorithm has similar performance with that of Almenarez for different reason. This algorithm requires quite complex and longer procedure which has influence to the number of the query that can be processed. This algorithm is suitable for

network with low number of peers (as result is shown in the figure where this algorithm still performs well for small network) but it has difficulty to cope with large network scale which indicated with lower number of responses.

Algorithm Mengshu: this algorithm has the best result for this category, because of the query propagation system which relies on the help of the direct neighbours to help spreading the query throughout the network. Based on this system, the scale of the network will not create a problem. The advance of Mengshu's system can be seen in the result of the simulation where the number of responses is a lot higher than that of other algorithms for the network with 500 peers. Thus, for the scalability, this system is the most suitable for p2p network environment.

4.5.3 Result for test case 3: Context / Classification

This test case needs simulation test to evaluate the influence of context on each of the algorithm. But due to the limited feature of the simulation tool, it is not possible to conduct a simulation test for this requirement. Therefore, the evaluation process will be done based on the analyses on the formula. There are only two algorithms that implement context into their calculation system. The rest of the algorithms do not mention this requirement in their trust value calculation process. However, the two algorithms that mention this requirement do not really implement it in the algorithm, they only mention it as a basic design idea. Therefore, the evaluation process will be perform based on the query propagation system since this information is available for all four algorithms and it will define the number of responses files that a peer will receive.

Aberer and Despotovic: This algorithm is based on the random query propagation system. If the query are based on the file's context (file's type), it will certainly reduce the number of responses file that a peer will receive. This is caused by the the query propagation itself. The random system has no pattern and coordination within the network and among the peers.

Almenarez: This algorithm is based on the flooding query propagation system. If the query is based on the context of the files, there will be more queries that are flooding the network. This will cause a network traffic and can affect to the number of responses files that a peer will receive.

Mengshu: This algorithm has a coordinated propagation system that will support the implementation of context-based files query within the network. Even though the number of queries within the network will increase but it will not have the same impact as that of Almenarez because this algorithm implements coordinated propagation system among the peers in the network.

Garcia-Molina: This algorithm has a similar propagation system with that of Mengshu but with extra additional propagation for the pre-trusted peer. The algorithm provides a supportive system for implementing context-based files query which can specify each query to the pre-trusted peer and to the network.

4.5.4 Result for test case 4: Dynamic

In this test case, we will conduct the evaluation by comparing the dynamic of the trust value among the four algorithms. This simulation takes place in a p2p network with only two good peers and the rest of the peer are malicious peers. The objective is to evaluate whether the

trust value of a certain peer will correspond with its behaviour. Because of the hostile environment (high number of malicious peers), a good peers will not have other options but to do the transactions with other good peers within the network. If the malicious peers give false recommendations or spread malicious files, it will affect the good peer's trust value. The simulation will be run for 10 cycles to give the peers in the network to have several transactions with other peers in the network. The following figure shows the trust value evolution of peer 1 from the first cycle until cycle 10.

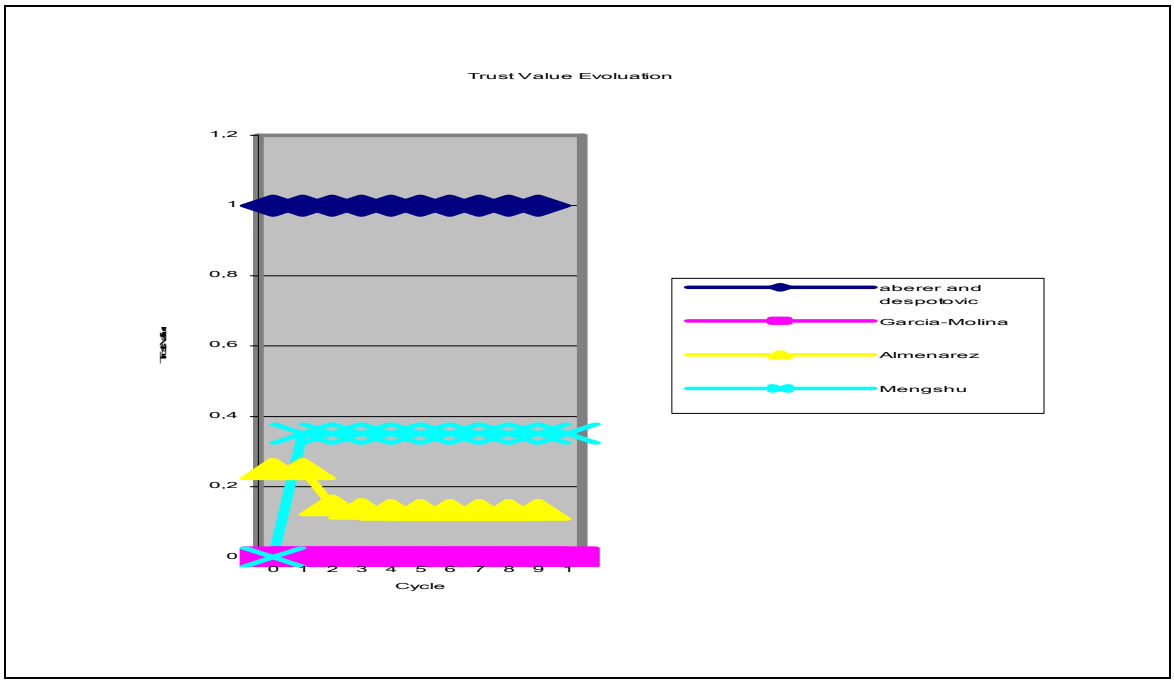


Figure 4.9 Test Result Trust Value Evolution

Algorithm Aberer and Despotovic: Algorithm such as Aberer, which is a binary algorithm, only provides two possible outcomes 1 or 0 which makes it very difficult to evaluate the result. As seen in figure 4.10, there is no changing value for this algorithm. Despite the hostile environment, the good peer is still able to maintain its high trust value. The problem is that it is difficult to see whether this trust value corresponds with the peer behaviour. Because of the binary outcomes it is difficult to make distinction between successful transaction (which lead to high trust value) and zero transaction (the trust value remains the same). Thus, in this algorithm the dynamic of the trust value is hardly visible.

Algorithm Mengshu: Mengshu algorithm shows visible trust value transformation from the first cycle to the second one. For the rest of the cycle, the trust value remains the same because the peer is not having any transactions with other peers in the network. This is caused by the hostile environment where there a lot of malicious peers in the network which prohibits the good peer to download the requesting file.

Algorithm Garcia-Molina: There is no trust value transformation for this peer. This is because the requesting file is always in the possession of malicious peer which prohibits the peer to have transaction or download the file, hence the static trust value.

Algorithm Almenarez: this algorithm has similar system with that of Mengshu. As previously described, this algorithm introduces the first user intervention variable, security level, which

can be defined by the peer itself. For malicious peer who abuses the security level, the punishment will be the radically changing of trust value. In figure 4.11, peer 1 does not have any transactions anymore after the second cycle. This is caused by the hostile environment where most of the peers are malicious and usually having a low security level, hence the static trust value.

4.5.5 Result for test case 5: Conditional Transitivity

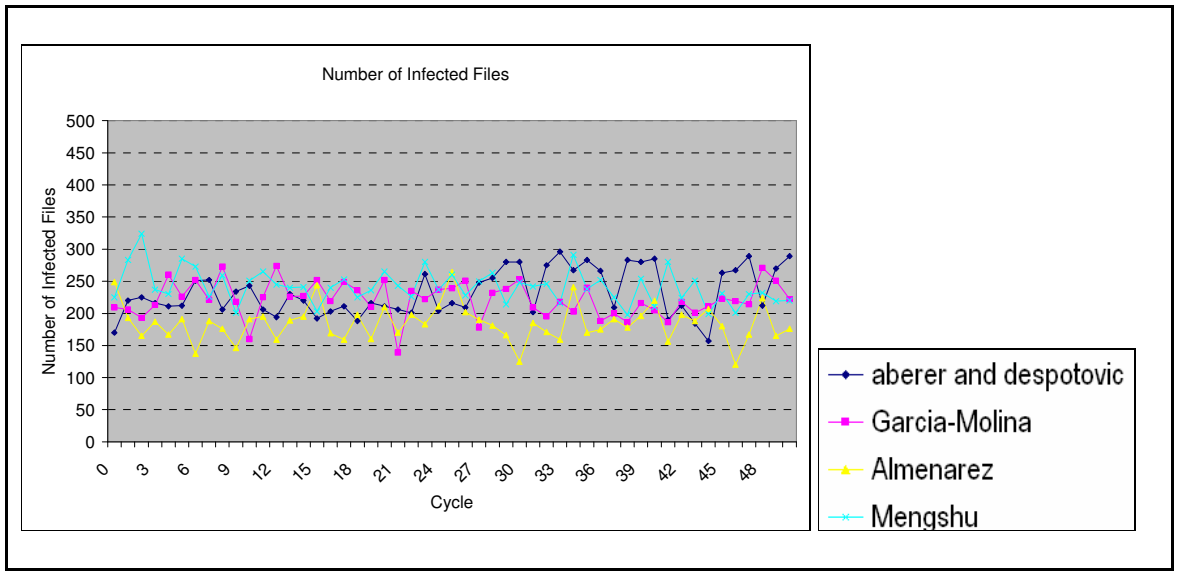


Figure 4.10 Test Result Number of Infected Files

The graphic is simulated among 500 peers who have a very high number of malicious peers. The graphic shows almost horizontal line (zero gradient) which means the result remains almost the same through the cycles. The figure shows that the average number of infected files from Garcia-Molina is the highest, followed by Almenarez and Aberer-Despotovic which has almost the same average number of infected files, and last in the line is Mengshu which has the lowest average number of infected files. Algorithm Garcia-Molina and Aberer-Despotovic which have the highest number of infected files are the algorithms that do not make distinction between recommendations and own experience, while the other two algorithms make this distinction. This will result in low accuracy of trust value calculation because of the false recommendation. The algorithms which implement this distinction are able to calculate the trust value more accurately because it handles own experience and recommendation differently. This is an important feature for calculating trust value because not every recommendation is trustworthy even though it carefully acquired from reliable peers. In the case where there are limited recommendations available in the network, each recommendation has larger influence in defining the outcome i.e. trust value. If a peer receives false recommendations it will also result in false trust value calculation, hence the high number of infected files. But in the case where there are large number of peers in the network, there will be likely larger number of transactions and recommendations available, therefore each recommendation has smaller influence in defining the trust value.

4.5.6 Result for test case 6: Peer Anonymity

Due to the limited capacity of the simulation it is not possible to conduct a test on this test case. Therefore, the test will be carried out based on the theoretical evaluation on the algorithm formula. The evaluation will be performed on each algorithm separately.

- Aberer and Despotovic:

This algorithm is operating based on the complaints from peers that are stored according to P-Grid system where it uses binary tree to store and access the data stored in various peers in the network. One drawback in this binary tree method is that each root-peer always has several leaf-peers in which the complaint from root-peer is stored. But each root-peer always knows which peers are their leaf-peers. This makes peer anonymity for this algorithm can not be guaranteed. If a particular peer is malicious peer and intends to impair other peer's reputation, it can be simply done by giving false complaints.

- Mengshu:

This algorithm is based on confirmation theory which seeking certainty factor from random peers in the network that gives recommendations for trust value calculation. Because of the randomness of the process, the peer anonymity is more assured than that of Aberer. For example, if peer A wants to download or share file from peer B, it will need to calculate peer B's trust value first. In order to calculate the trust value, peer A needs other peers recommendations regarding peer B's reputation. This process is completely unknown to peer B. The only thing that peer B will find out is whether peer A will agree to download file from peer B or not. Thus, in this case the peer anonymity can be guaranteed.

- Garcia-Molina:

This is the algorithm that requires the most complicated process from other three algorithms. It is complicated because this algorithm takes into account a lot of issues in p2p environment, including peer anonymity. The architects of this algorithm never ignores maintaining peer anonymity from the starting moment when a peer joins the network. Because this is a global trust algorithm, there is going to be only a single global trust value for each peer for the entire network. The process starts when the peers join the network, they will be automatically divided into small clusters. Every cluster has a so-called cluster manager (normally the pre-trusted peers are the managers). The cluster manager will distribute task to the a member of the cluster to assess the other member of the cluster. Each peer in the cluster will not know which peers have been assessed them. The assessment would be reported to the cluster manager and the global trust value will be calculated based on this information. The global trust value will be stored in several peer managers in the network. So through the whole process, the anonymity of each peer is assured.

- Almenarez:

This algorithm has basic similar calculation system with Mengshu, where a peer will need other peer recommendation that is obtained from random peer in the network. Thus, if peer A needs to calculate peer B's trust value, it will ask some random peers in the network for recommendations. But these recommenders will remain anonymous to peer B until the trust value of peer B is acquired and peer A can decide whether it will download file from peer B or not. Therefore, for this test case, Almenarez has the same score with Mengshu.

4.5.7 Result for test case: Non-symmetric

This test case also does not need a simulation test. It can be done by analyzing the formula of the algorithm. All of the four algorithms meet this requirement since all the trust value calculation only applies to one specific peer. Thus the result of trust value calculation of peer B by peer A is not the same with the result of trust value calculation from peer A by peer B. For this test case, all of the four algorithms have the same score.

4.5.8 Result for test case: Performance on Computation Process, Data Storage and Message Complexity

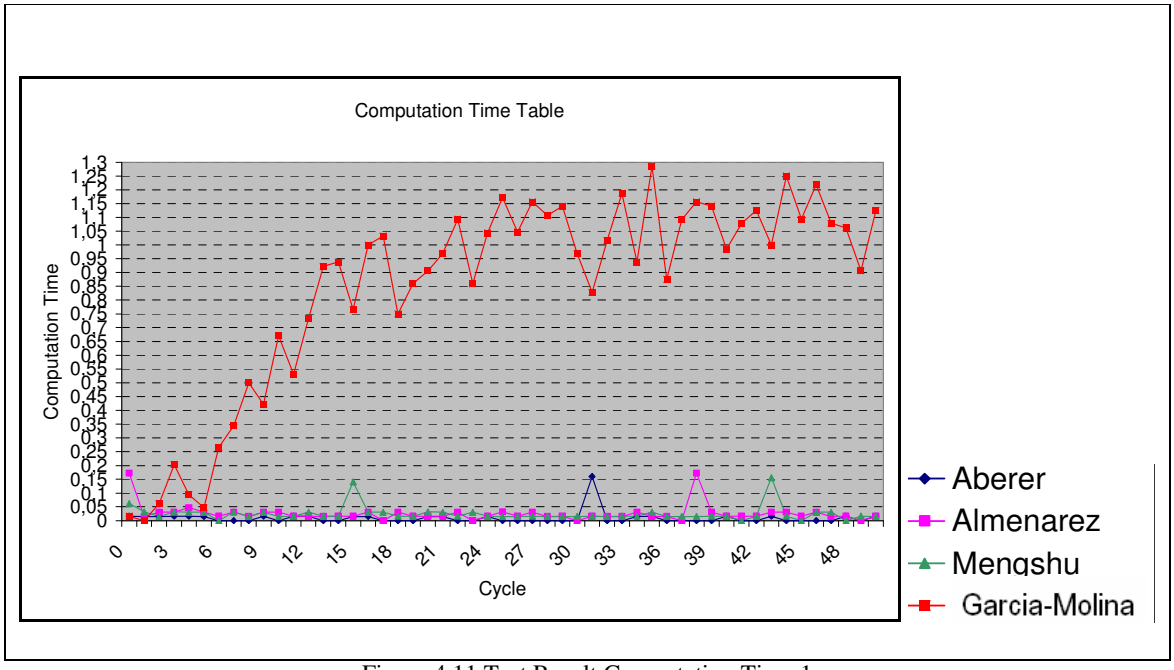


Figure 4.11 Test Result Computation Time 1

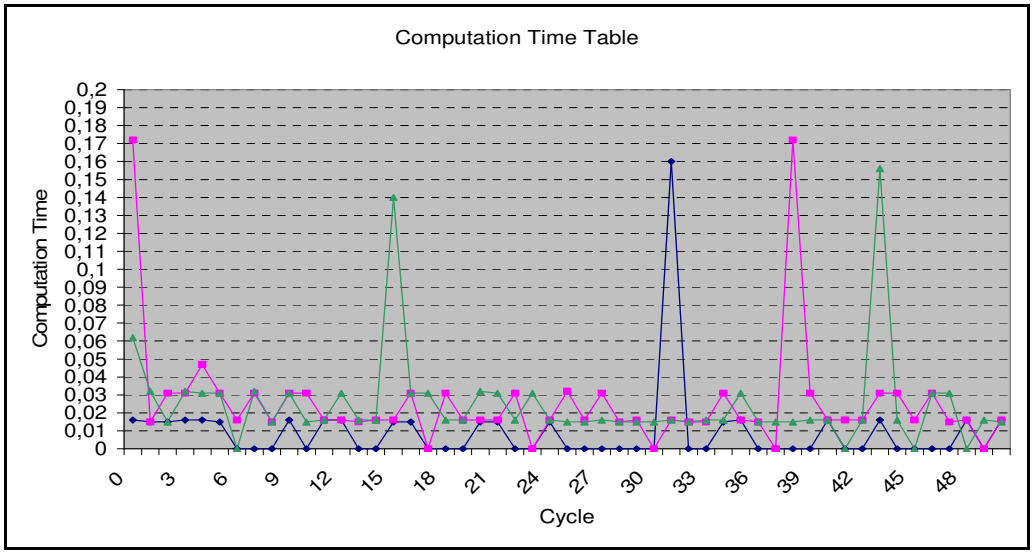


Figure 4.12 Test Result Computation Time 2

From the test it is proven that algorithm Garcia-Molina has the average highest computation time. This is because of the global trust algorithm where every peer has to compute another peer's global trust value which makes the computation time is longer. Aberer has the lowest average computation time, thanks to the P-Grid system which allows fast access to the data storage. The problem is that if peer A holds data of a popular file q, peer A will receive a lot of request from other peers regarding the file. If two of more peers requires data to peer A, the second peer will have to wait until the first peer finishes the transaction with peer A. this will result if long process time for the second peer and even longer for third peer. The extreme value from several cycles (such as cycle 16, 32, 38 or 44) is the result from this occurrence.
Mengshu has slightly higher average computation time than that of Aberer because there are several iterations in the calculation process which causes longer computation time. The last algorithm from Almenarez requires slightly longer calculation and messaging process than Mengshu which of course results in even longer computation time.

4.5.9 Result for test case: Implementing a user intervention system

This ninth test case is dealing with implementing a user intervention system in the algorithm process in order to give the peer the suitable preference. The test is conducted by analysing the algorithm process. Algorithm from Almenarez is the only one that satisfies this requirement by giving the possibility to the peers to give input for security level that matches with their preferences. The rest of the algorithms do not have this feature because they are fully automated by the system which are not allowed the peers (users) to have an active part in the trust value calculation system.

4.5.10 Result of test case: Take into account the role of pre-trusted peer

This last test case has the similar procedure the previous test case. The test will be conducted through an analyses of the formula. The only algorithm that meets this requirement is only algorithm from Garcia-Molina. The rest of the algorithm does not take into account the role of pre-trusted peer into their trust value calculation process.

4.6 Summary

Based on the simulation result, it can be summarized as follows:

1. For test case: reflexivity

Based on the evaluation process, only two algorithms (Almenarez and Garcia-Molina) that satisfy the requirement, the other two algorithms fail to meet the requirement.

2. For test case: scalability.

The system from Mengshu where it relies on its neighbors to spread the query is shown to be suitable for p2p network environment because it can cope with both small and large network scale. This system can be implemented in the design of Hybrid algorithm.

3. For test case: context/classification

From the previous discussion, it is concluded that two of the algorithms (Almenarez and Aberer & Despotovic) are not suited for implementation of context in their algorithms. While

other two algorithms that have coordinated query propagation can provide a supportive system for context-based file query implementation.

4. For test case: dynamic

The dynamic of the trust value is defined by the trust value evolution during the cycles. In this test case, only algorithm from Aberer and Despotovic that barely has a visible trust value evolution because of its binary output system. The other three algorithm can show more changes during the early cycle, but due to the hostile envoriment, the trust value remains static because no transaction takes place among the peers in the network.

5. For test case: conditional transitivity

The result shows that algorithm which satisfy the requirement is more sensitive to the behavior of the peer. It is shown with the lower number of infected files that are found within the network. Therefore, the designing process of the Hybrid has to take into account this requirement to obtain accurate trust value.

6. For test case: peer anonymity

Based on the theoretical evaluation, algorithm from Garcia-Molina has the best system on the maintaining peer anonymity. However, this algorithm is very complex to implement and needs high coordination among the peer and with the network itself. Thus, for this requirement, there should be combination of system from Garcia-Molina and another algorithm to maintain the peer anonymity without having complex process.

7. For test case: non symmetric

Based on the evaluation, all of the algorithms satisfy the requirement. Therefore, all of the algorithms have the same score for this test case.

8. For test case: performance on computation process, data storage and message complexity.

The result from the simulation shows that algorithm with complex calculation requires longer computation time which could lead to add an extra load to the network. The designing of the Hybrid algorithm should keep the computation as simple as possible without ignoring other requirement

9. For test case: implementing a user intervention system

Based on the result of the evaluation, only one algorithm (Almenarez) that meets the requirement for this test case. The rest of the algorithm fail to satisfy the requirement for the test case. This user intervention system will be used in the next chapter for designing the Hybrid algorithm.

10. For test case: taking into account the role of pre-trusted peer.

For this test case, only algorithm from Garcia-Molina that satisfies the requirement; other does not prevail. The role of pre-trusted peer will also be utilized for designing the Hybrid algorithm in the following chapter.

CHAPTER 5

Designing The Hybrid Algorithm

From the previous chapter, we have obtained the result and evaluation from the current four algorithms. This result will be implemented in the designing process of the Hybrid algorithm that will take place in the next following section. The objective of this chapter is to create a design concept of the Hybrid algorithm that by the end of this chapter can be simulated.

5.1 Design Requirements

Based on the previous result, the design requirement for the Hybrid algorithm can be stated as follows:

- Reflexive: As previously described, this element needs to be stated and included in the design requirement to declare that a peer trusts its own system before starting to calculate other peer's trust value.
- Scalability: From the previous simulation test, the results showed that algorithm Mengshu has the best result to cope with the scalability of the network by utilizing the neighbour to propagate the query throughout the network. The Hybrid algorithm will implement the similar system in the designing process.
- Context/Classification: Based on the various types of files, the trust value calculation should be associated with each type of files that are being shared in the network. Because of the limitation of the simulation tool, the context will not be able to be simulated but it would be implemented in the design.
- Dynamic: From the result of the previous simulation it can be seen that algorithm which expresses trust value in continuous interval between 0-1 is more able to show the changes in trust value through the cycle during the simulation. Algorithm such as Aberer (with binary trust value) has limited space to express the changes in the trust value. Moreover, the design of the Hybrid algorithm should take into account the fact that trust value should be built gradually through time, not based on one single action only.
- Conditional Transitive: From the previous simulation test, the results showed that algorithms that implement distinction between own experience and recommendation were able to perform better than the algorithm which do not implement this requirement.
- Non-Symmetric: Based on the previous description, this requirement is needed to enhance the accuracy of the trust value calculation. All of the previous four algorithms are proven to be able to satisfy this requirement. The Hybrid algorithm will also apply non-symmetric into its trust value calculation process.
- Peer Anonymity: The testing process is conducted through theoretical evaluation of the algorithm because of the limited capability of the simulation. Algorithm Garcia-Molina is considered as the algorithm that is able to assured the peer anonymity better than the others. Unfortunately this algorithm requires complicated coordination between the network and the peers, which makes the implementation quite challenging. Algorithm Almenarez and Mengshu on the other hand require much less complicated implementation. These two algorithms can be combined together to maximizing the peer anonymity.

- Performance on computation process, data storage and message complexity: The result from previous simulation shows that three algorithms has similar result except the result from Garcia-Molina. This is caused by longer process of trust value calculation and complicated implementation of Garcia-Molina algorithm. The rest of the algorithms have similar performance because their algorithms require less complicated implementation and more simple process. The Hybrid algorithm design has to take into account this simplicity factor to improve the performance of the algorithm.
- Combining the user intervention system and pre-trusted peer role: the Hybrid algorithm will use one more combination element from Garcia-Molina and Almenarez to improve the performance of the algorithm. In the real p2p network, pre-trusted peer(s) always exists. Because they have long history, a lot of experience and have high trust value. These assets can be used to improve the robustness of the trust value calculation. Only in this design, the dependency of having pre-trusted peer in the network will be reduced so that it will not create a network failure when the pre-trusted can not be found or temporarily not active. Beside using the pre-trusted peer, the Hybrid algorithm design also implements the security level as introduced by Almenarez. This security level is considered an important part of trust value calculation because it allows user to set the accuracy of the trust value calculation based on their preferences.

5.2 Concept Design

To design the concept of the Hybrid algorithm, a step by step description from the moment a peer joins a p2p network is provided as follows:

- When peer A joins a network, it should assign the security level first although it does not have any knowledge about other peers in the network. It means that it can not trust anyone. The best choice is to connect with the pre-trusted peer or high trust peer (such as in Garcia-Molina). If this peer does not possess the requested file, it can give recommendation to which peer it should connect which has the requested file.
- There is also probability that the pre-trusted peer or high trust peer does not exist or temporarily non-active. In this case, peer A should use its own judgement by incorporating part of Almenarez algorithm. Peer A will send to the network a query regarding the requesting file. Any peers that possess the file welcome to respond the query. Peer A will have to choose from the incoming responses which peer is trustworthy. Because peer A has not had own experience with other peer yet, it has to ask other peer in the network that has information about that particular peer, let say peer B. Peer A does this by sending a second query, this time a query which ask information to other peer in the network about peer B. If there are some peers answering the query, peer A will have to select these peers as well because malicious peers can give misleading information. Peer A selects the peers by looking at their security level. Any information that comes from a peer that has lower security level than peer *i* will be discarded. Based on this information, peer *i* will be able to calculate peer *j*'s trust value. If peer *j* is not trustworthy, then peer *i* will have to do all the procedure from the beginning to find other peer.
- Once a peer gains experience about other peer in the network, it can use part of algorithm from Mengshu to calculate other peer's trust value by using the certainty factor.

To get a clear view about the process, a flow chart is constructed below:

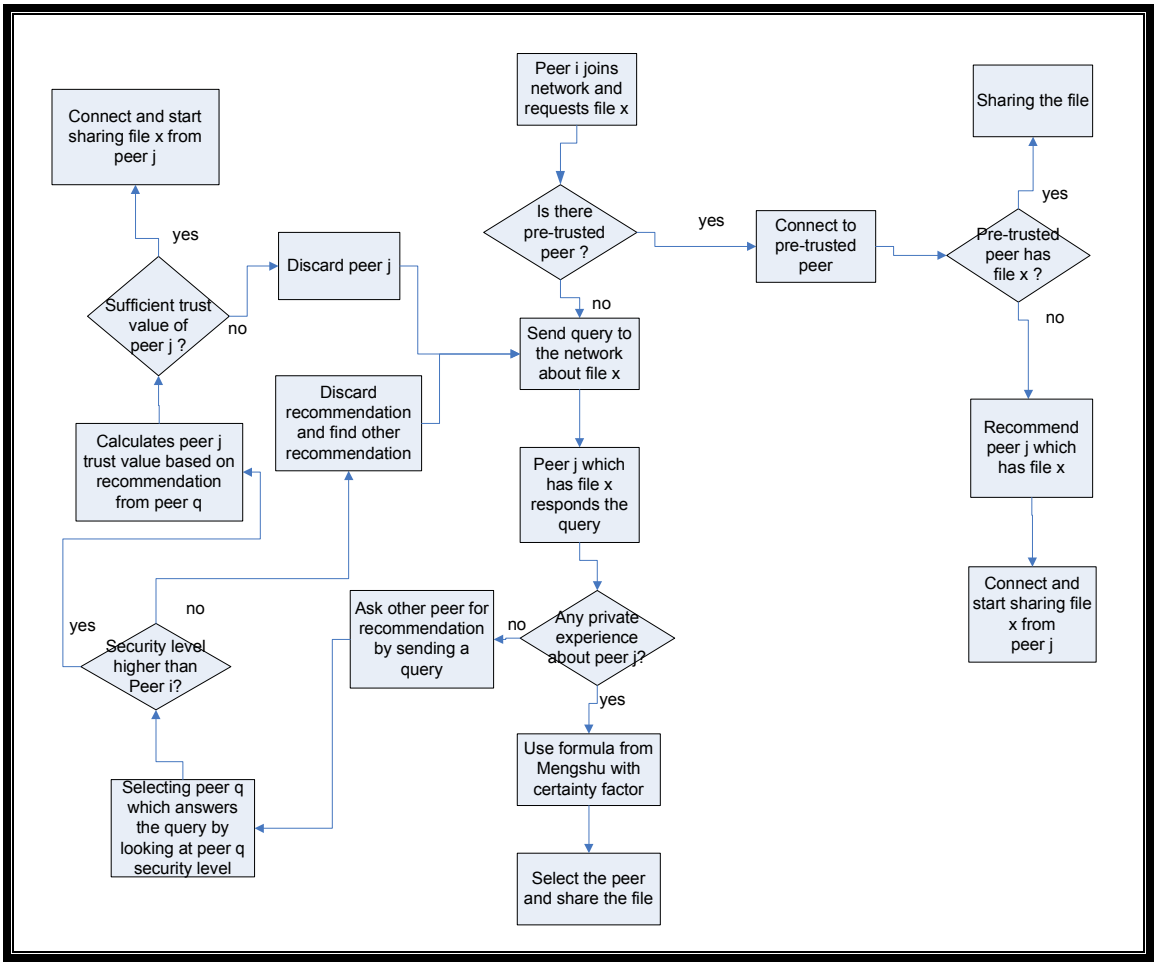


Figure 5.1 Flow Chart Hybrid Algorithm

5.3 Concept Design Formalisation

1. Peer A joins network (it means also that peer A declares its reflexivity when it joins the network): assign security level β_A

   $\beta_A \in$ N: N= {1, ..., 9}

2. Peer A submits a query for file x, which will be firstly responded by pre-trusted peer F if peer A does not know or trust anyone.

   $T_{AF}^{c} = T_{F}^{c}$ (peer A will trust pre-trusted peer F for a specific trust value context c without having to calculate peer F trust value first).

3. In the case where peer F does not have the requested file x, it will give a recommendation to peer A which other good peer in the network (let say peer B) has the requested file.

   $T_{AB}^{c} = T_{FB}^{c}$ (trust value of peer B by peer A is equal to trust value of peer B that calculated by peer F)

4. In the case of no pre-trusted peer, peer A will send a query to the network through its neighbours (see query propagation from Mengshu), called request query, R_Query which contains identification of the requested file (Request_ID), identification of the requestor (Requestor_ID) which is peer A in this case and time to live of the query (TTL).

   Req_Query = (Request_ID, Requestor_ID, Context_ID, TTL)

5. Peer C which has the requested file x will respond the query and peer A will check first whether security level of peer C is at least equal or higher that peer A. If this condition is not satisfied, peer A will discard peer C, and choose another responding peer. If the condition is satisfied, peer A will check whether it has private experience with peer C with checking the local trust value C_{AC}^{c} which is the result of total successful transaction between peer A and peer C (S_{AC}^{c}) and unsuccessful transaction between peer A and peer C (F_{AC}^{c}):

$$C_{AC}^{c} = \begin{cases} \dfrac{S_{AC}^{c} - F_{AC}^{c}}{S_{AC}^{c} + F_{AC}^{c}} & \text{if } \beta_{\mathrm{C}} \geq \beta_{A} \\ \text{discard peer C} \quad \text{otherwise} \end{cases}$$

6. If the value C_{AC}^{c} is obtained, peer A needs a recommendation from other peer to calculate peer C trust value. This is obtained by sending a second query to the network through the neighbours, regarding recommendation of peer C. If for example peer D responds the query, peer A will have to check peer D security level, just as the previous procedure before receiving the recommendation:

$$T_{AC}^{c} = \begin{cases} (1-\alpha)C_{AC}^{c} + \alpha T_{DC}^{c} & \text{if } \beta_{\mathrm{D}} \geq \beta_{\mathrm{C}} \\ \text{discard peer D} & \text{otherwise} \end{cases}$$

7. After the transaction between per A and peer C is completed, peer A will update its trust value calculation of peer C based on the result of the transaction by using updating formula from Almenarez:

$$V^{c} = (1 - \frac{F_{AC}^{c}}{S_{AC}^{c} + F_{AC}^{c}}).W^{\beta}$$

$$T_{AC}^{c,new} = V^{c}.\alpha + T_{AC}^{c}.(1-\alpha)$$

8. If the transaction is unsuccessful (peer D is malicious), peer A will send a message to the network to warn other peers regarding the malicious peer D

Warning_Query = (Sender_ID, Malicious_ID, Context_ID, TTL)

Based on the described process above, the query propagation will be as follows:

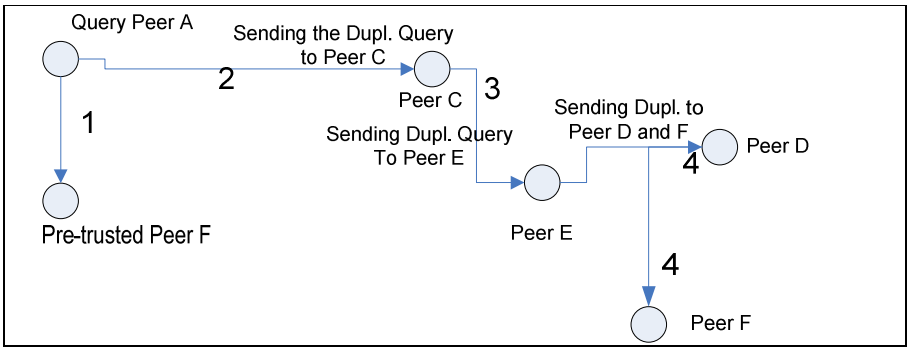


Figure 5.2 Query Propagation Hybrid Algorithm

The propagation of the peer is similar with that of Garcia-Molina where the query is firstly propagated to the pre-trusted peer F in the network, if this peer exists. If not, the query will be propagated to the neighbours of the peer A who sent the query.

Legend	
Variable	**Description**
A	peer in the network who wants to calculate other peer trust value (requestor peer)
B	peer whose trust value is being assessed (requested peer)
C	peer whose trust value is being assessed (requested peer)
T_F	Trust value of pre-trusted peer F
T_{AB}	Trust value of peer B by peer A
T_{FB}	Trust value of peer F by peer B
V	Action Value
S_{AC}	Number of successful transaction between peer A and peer C
F_{AC}	Number of unsuccessful transaction between peer A and peer C
$F_{AC} + S_{AC}$	Number of total transaction between peer A and peer B
W	Action Weight; defined by transaction that just completed. With successful transaction = 1and by unsuccessful transaction = 0.
β_A β_C β_D	Security level of peer A, peer C and peer D respectively. This variable is defined by peer itself.
α	Constant between 0-1. This variable is defined by the system to find tune (give balance) between old trust value and the new one.
T_{AC}^{new}	The up-dated trust value calculation of peer B by peer A

Table 5.1 Legend

Based on this information, the query result depends on the security level of peer A. If peer A has high security level, it will deliver less result than if peer A has lower security level. The following figure shows example of two search result of query that is submitted by peer A. The first figure shows search result based on the query that is lower than the second figure.

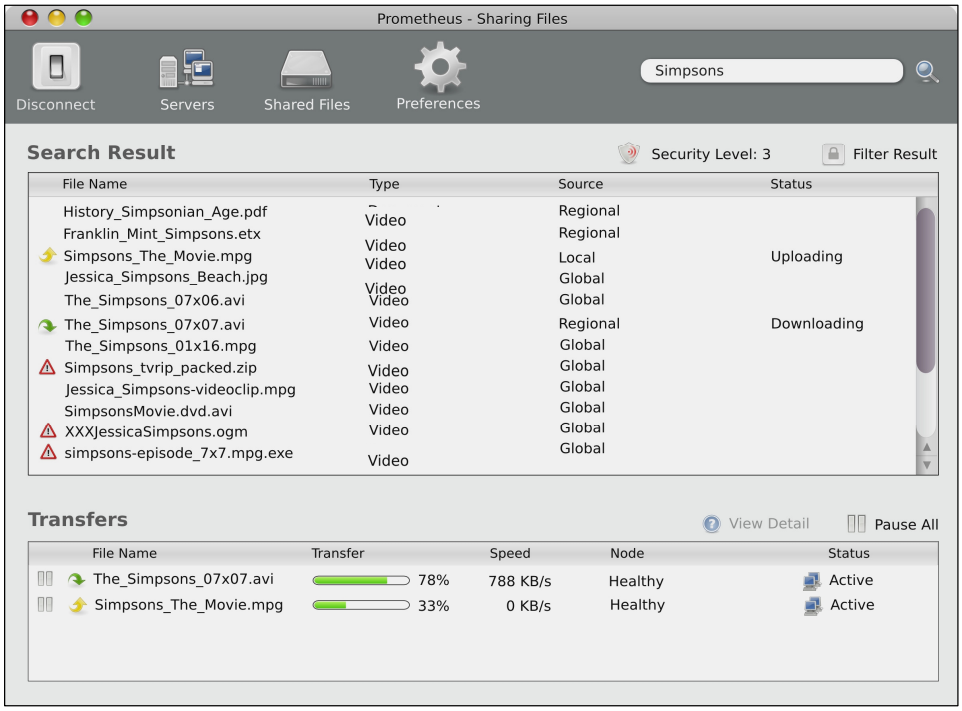


Figure 5.3 Search Result Hybrid Algorithm with lower security level

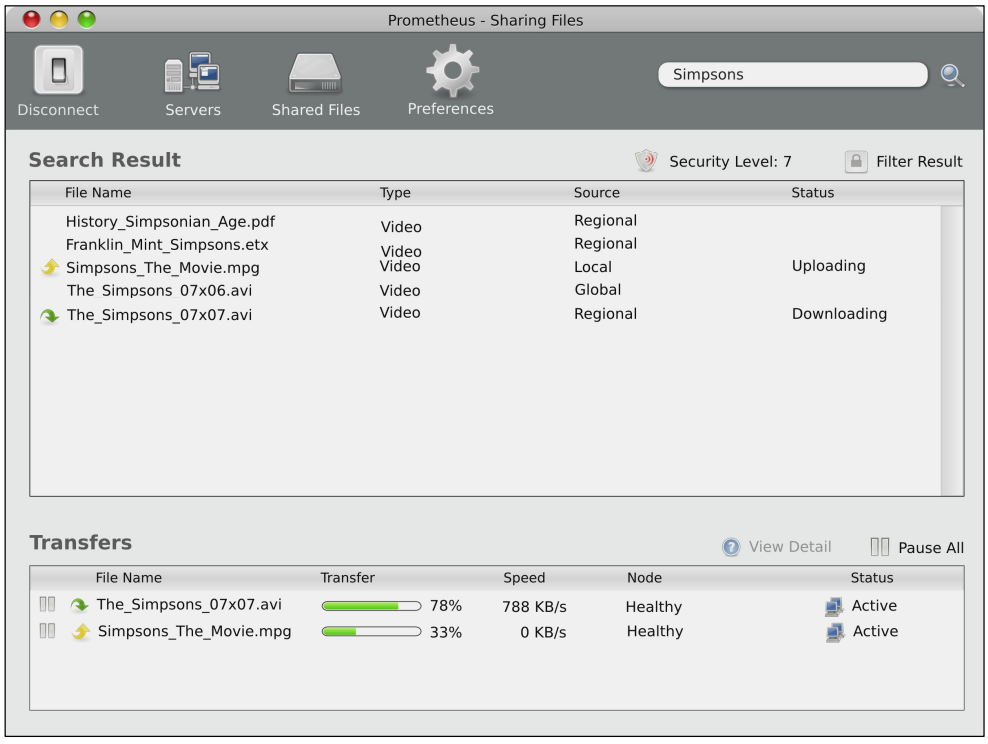


Figure 5.4 Search Result Hybrid Algorithm with higher security level

This concept design will be validated with the help of simulation tool which will be conducted in the following chapter.

5.4 Validation Process of the Simulated Hybrid Algorithm

Three important notes for this validation process:

- For the sake of simplicity, during this validation, the constant α has been set to 0.2, 0.5 and 0.8 instead of random number.
- The observation of the simulation is performed during cycle 1 of the simulation.
- Because there are few number of peers in the network, the peer that is being assessed in the validation is always peer 1 whether it is good or malicious peer.

- ***Mathematical formula based on setting B:***

Based on the formula, if the there is pre-trusted peer, peer A will automatically connect to the pre-trusted peer. In this case, peer A will have other option because there are only two peers in the network, peer A and pre-trusted peer F.
Thus based on this information, $T_{AF} = 1$ during cycle 1 of the simulation.

- ***Result from the simulation***

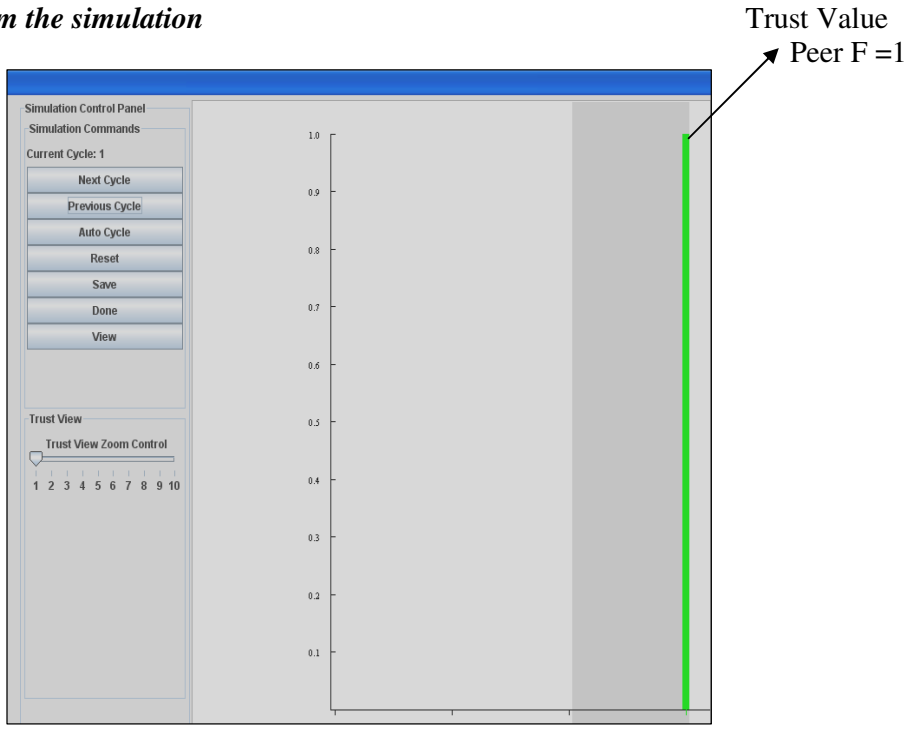


Figure 5.5 Result Simulation Hybrid Algorithm

α	β	Result	Setting					
			A Total peer: 2 Good peer: 1 Malicious: 1 Neighbour: 1	B Total peer: 2 Good peer: 1 Pre-trusted:1 Malicious: 0 Neighbour:	C Total peer: 2 Good peer: 0 Malicious: 2 Neighbour: 1	D Total peer: 3 Good peer: 2 Malicious: 1 Neighbour: 1	E Total peer: 3 Good peer: 3 Malicious: 0 Neighbour: 1	F Total peer: 3 Good peer:0 Malicious: 3 Neighbour: 1

				1				
0.2	5	Mathematical calculation	0	0.2	-	0	0.2	-
		Simulation	0	0.2	-	0	0.2	-
0.5	5	Mathematical calculation	0	0.5	-	0	0.5	-
		Simulation	0	0.5	-	0	0.5	-
0.8	5	Mathematical calculation	0	0.8	-	0	0.5	-
		Simulation	0	0.8	-	0	0.5	-

Table 5.2 Validation Result Hybrid Algorithm

From the result comparison, it can be seen that two types of results are the same, which proof that the simulated Hybrid algorithm is valid.

5.5 Result and Evaluation

Once the design is proven to be valid, the simulation process can be proceed. The test will use the same test case the previous one.
The result from the test is presented as follows:

Result from test case 1: Reflexivity
The Hybrid algorithm meets this requirement by stating the reflexive requirement under the design requirement section.

Result from test case 2: Scalability

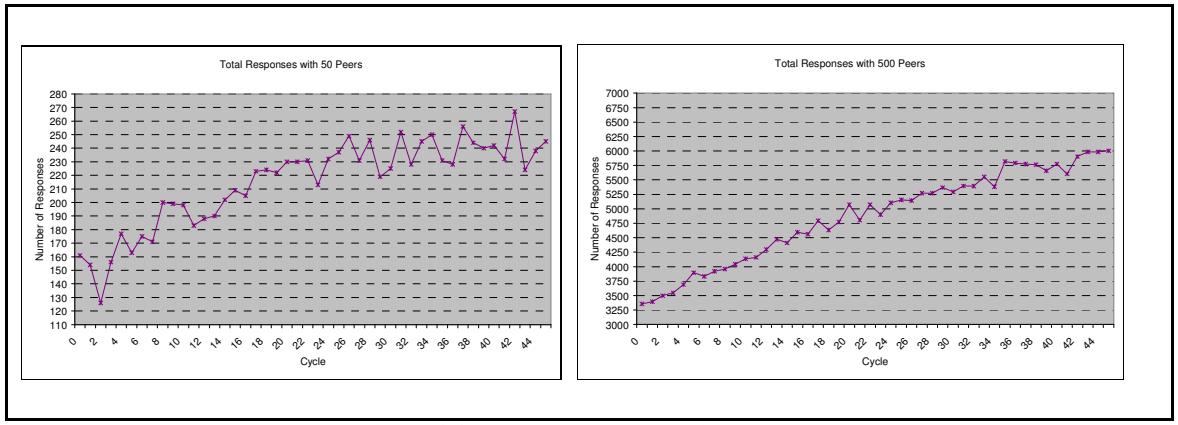


Figure 5.6 Test Result Number of Responses

The result of the simulation above shows that the algorithm performs quite well for both limited number of peers (50 peers) and large number of peers (500 peers) in the network. The graphic line shows both linear trend which indicates that the number of responses increases along with the number of the cycle. Thus, this algorithm can cope with the scalability of the distributed environment such as p2p network.

Result for test case 3: Context/Classification

The Hybrid algorithm clearly meets this requirement because it implement the context factor I the trust value calculation. But the limited feature from the simulation tool prohibits the test case to be simulated in order to evaluate its influence on the algorithm process.

Result for test case 4: dynamic

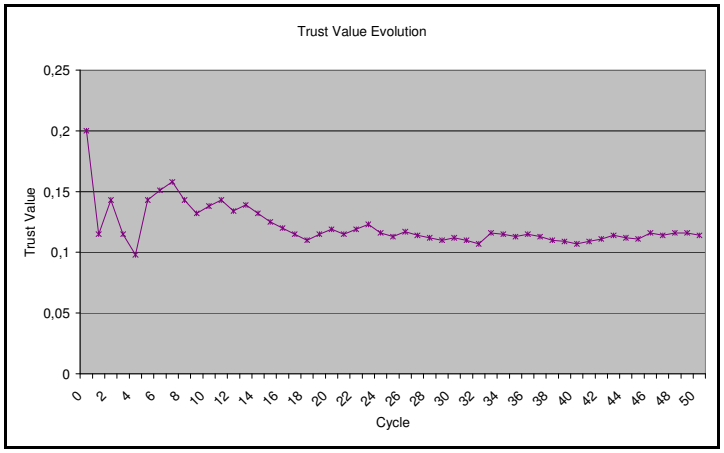


Figure 5.7 Test Result Trust Value Evolution

This algorithm expresses the trust value in real number in interval between 0-1. the result of the trust value during the simulation can be seen in the illustration above, where the changing in trust value can be recognized. It can be seen that the changing is visible but not radical which means that a single transaction does not have large influence on trust value calculation.

Result for test case 5: Conditional Transitivity

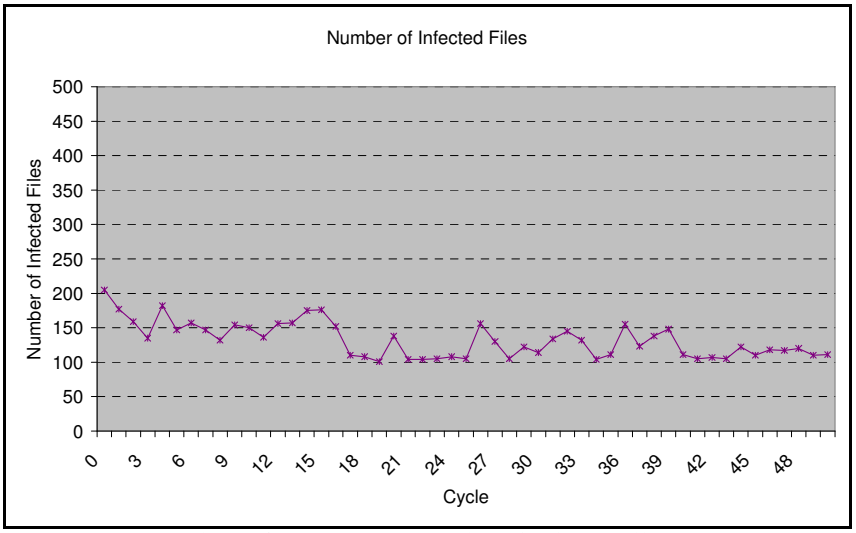


Figure 5.8 Test Result Number of Infected Files

This algorithm, as previously mentioned, implements conditional transitive trust calculation which means that the algorithm recognizes the difference between own experience and recommendation from other peers. It can be seen from two illustrations above that the number of infected files is actually getting lower in relation with the cycle of the simulation. The first graphic is based on 50 peers and the right one is based on 500 peers. It means the algorithm process is able to minimize the number of malicious activity based on the given information, both peer's own experience or recommendation from other peers.

Result for test case 6: Peer Anonymity

This algorithm is basically combination between the 4 previous algorithms, where the beneficial features of the algorithm are combined together and trying to minimize the weakness as much as possible. The basic formula of this algorithm is similar with that of Mengshu and Almenarez where if a peer A wants to calculates peer B's trust value, peer B does not know where peer A obtained the information from. Based on this information, this algorithm has the similar score as Mengshu and Almenarez for this test case.

Result for test case 7: Non-Symmetric

The Hybrid algorithm meets this requirement because the trust value calculation only applies to one specific peer. As previously discussed, if peer A calculate peer B's trust value, the result only applies for peer B, peer A's trust value needs to be separately calculated if it is needed.

Result for test case 8: Performance On Computation Process, Data Storage and Message Complexity

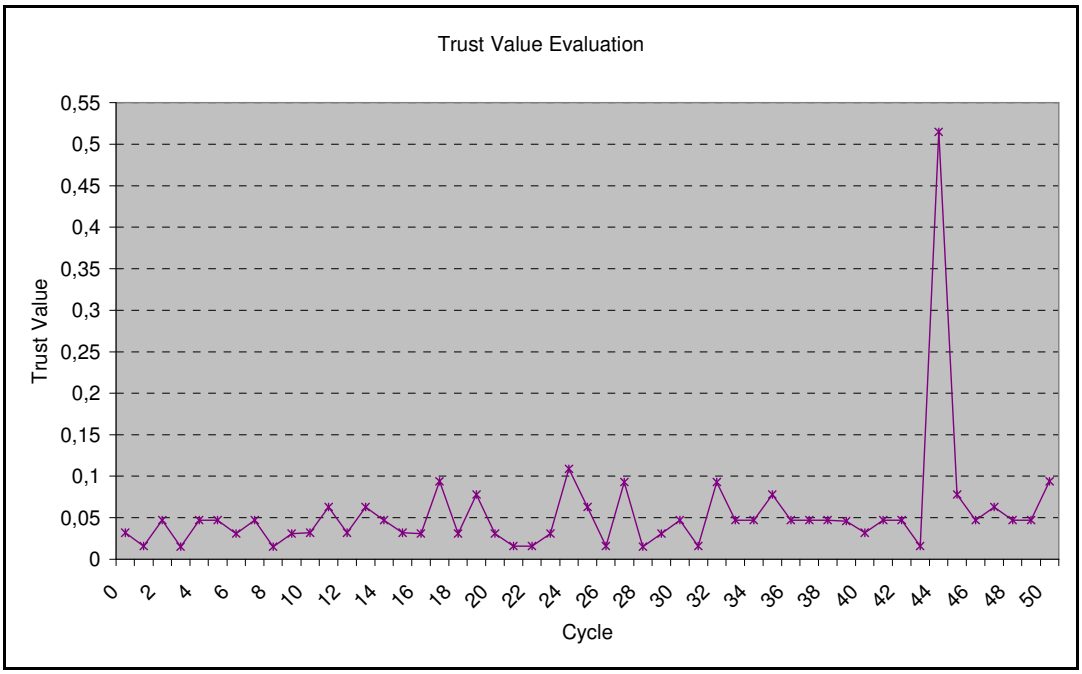


Figure 5.9 Test Result Computation Time

The average computation time of the algorithm compared to other algorithms in the previous simulation is lower, which means that the algorithm is able to process and calculate trust

value faster than the 4 previous algorithms. The extreme value from cycle nine, as previously mentioned, is caused by the waiting time which occurs when a peer with a popular file has multiple requests from several number of peers in the network.

This result from Hybrid algorithm will be compared the result from other four algorithms in the next chapter and the result will be evaluated.

Result for test case 9: Implementing a user intervention system

The Hybrid algorithm implements a user intervention system by borrowing the security level system from Almenarez to match the algorithm with the peer's preference. Therefore, this algorithm satisfies this requirement.

Result for test case 10: Take into account the role of pre-trusted peer

Similar with the previous test case, the basic idea is borrowed from the algorithm of Garcia-Molina. Only the implementation is modified so that the algorithm does not utterly depend on the pre-trusted peer. By implementing this system, the Hybrid algorithm satisfies this requirement.

5.6 Summary

The Hybrid algorithm has been designed, validated and tested based on the requirements. According to the result, this Hybrid algorithm satisfies all of the requirements. In the next chapter, this algorithm will be compared with the results from the other four current algorithms to find which algorithm has the best overall results.

Chapter 6

Evaluation On Overall Results

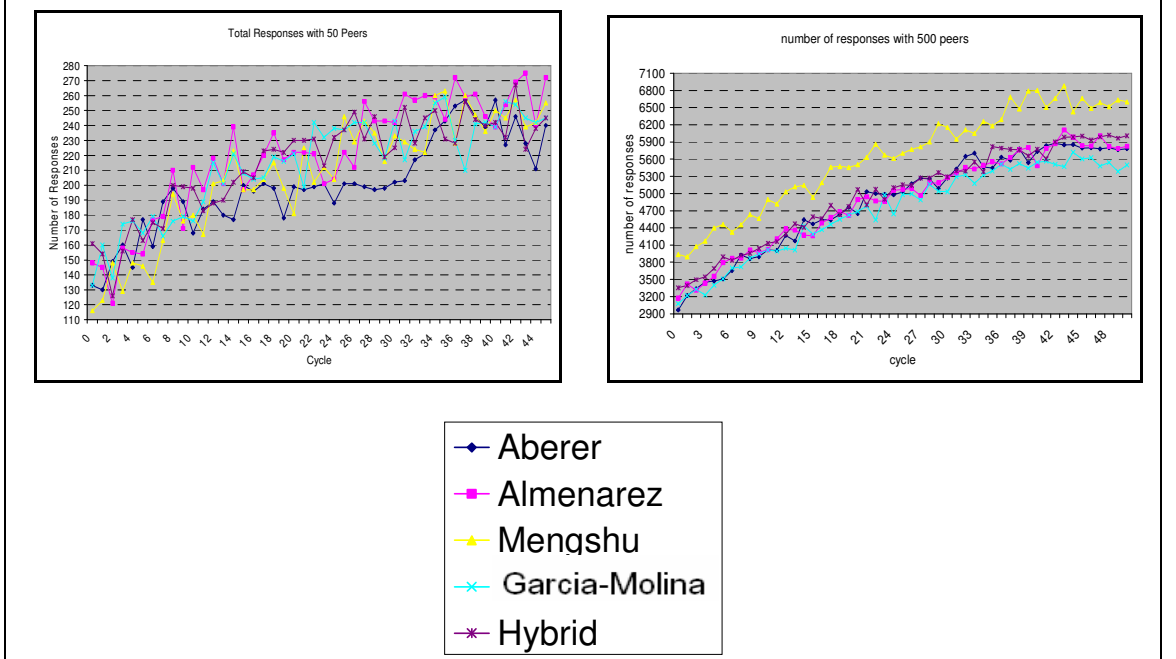
In this section, the result of the test will be evaluated and compared with the result from the previous test. The evaluation process will be perform separately for each test case.

6.1 Evaluation Test Case 1: Reflexivity

In this test case, there is no result to be compared because to satisfy this requirement, the algorithm only needs to state it in the beginning of the process. The Hybrid algorithm, the Almenarez and Garcia-Molina algorithms meet this requirement while the rest does not clearly states it in their algorithms.

6.2 Evaluation Test Case 2: Scalability

In this test case, we evaluate the performance of the algorithms in coping with small and large number peers in the network. The next illustrations show the result of fives algorithm in number of responses for various number of peers in the network. We can see that the for small network and large network Hybrid algorithm does not have highest number of total responses but it has quite high performance in the both figures.



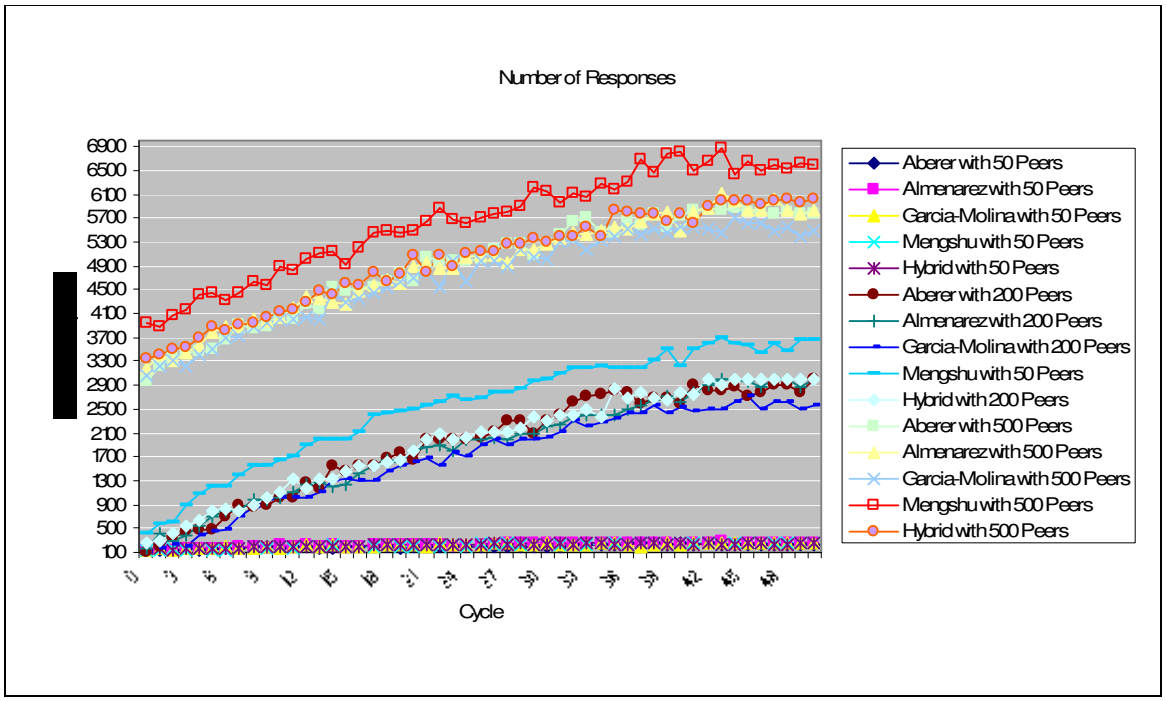


Figure 6.1 Test Result Numbers of Responses

6.3 Evaluation Test Case 3: Context/Classification

This Hybrid algorithm has an advance in this test case, because the Hybrid algorithm really implements the context in the trust value calculation process, while the other two algorithms (Almenarez and Garcia-Molina) only acknowledge the context requirement but they do not really implement it in the formula.

6.4 Evaluation Test Case 4: Dynamic

In this test case we can see that the Hybrid algorithm does not have trust value transformation after the second cycle. This is a similar behaviour with that of Mengshu algorithm which is caused by the hostile environment (high number of malicious peer). However, even though there is no transformation after the second cycle, it is still more visible than that of Aberer and Despotovic algorithm which only has binary result.

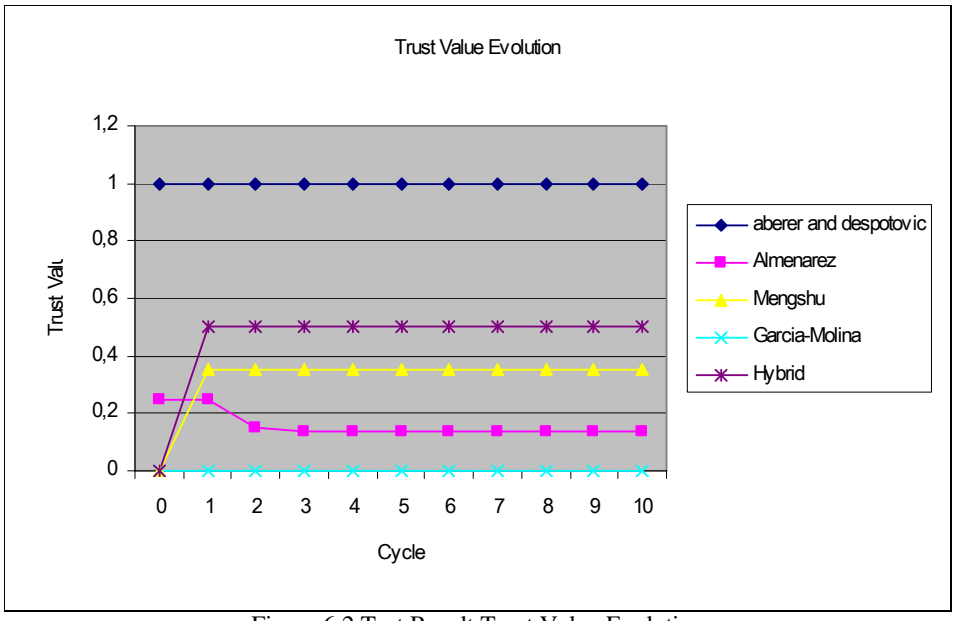


Figure 6.2 Test Result Trust Value Evolution

6.5 Evaluation Test Case 5: Conditional Transitivity

In this test case, we evaluate the number of infected files of each algorithm. Hybrid Algorithm has the best performance in this test case where the numbers of the infected files are reducing along the cycle. Algorithm Garcia-Molina and Aberer perform less and the other three algorithms as the result of their algorithms process where they do not implement the conditional transitive element.

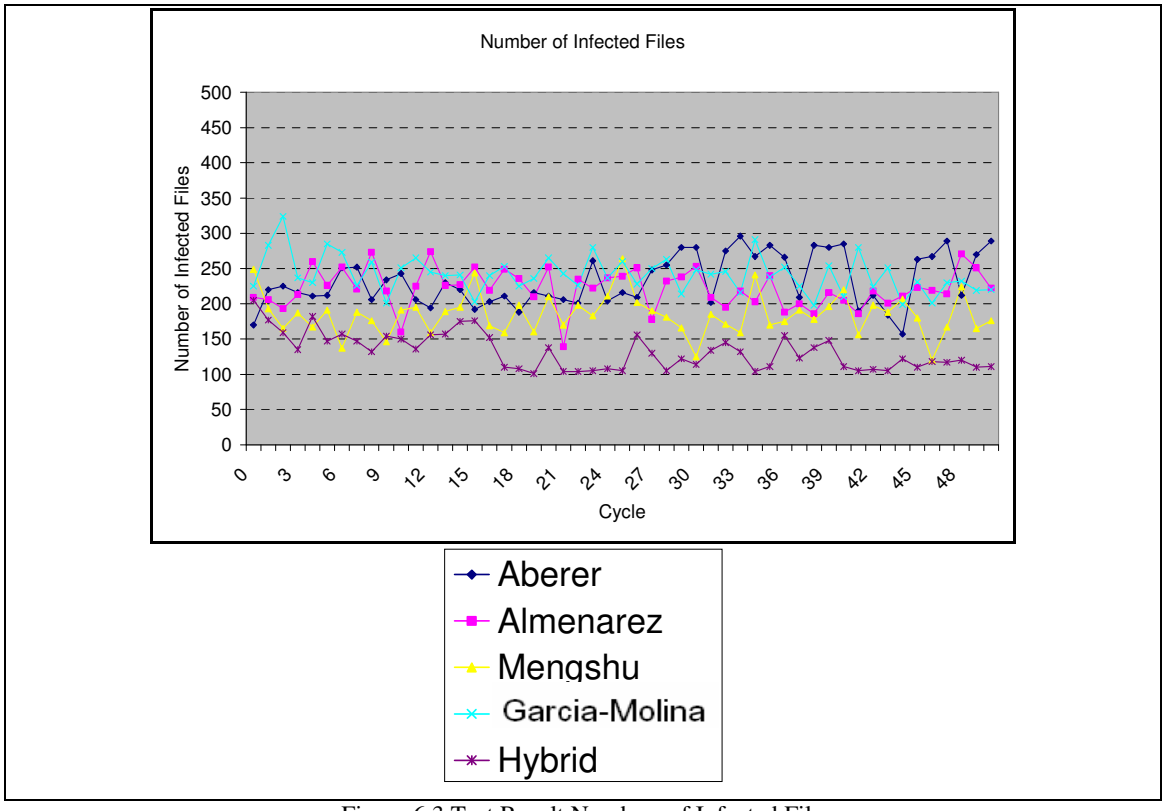


Figure 6.3 Test Result Numbers of Infected Files

6.6 Evaluation Test Case 6: Peer Anonymity

Based on the anonymity issue, the performance of the Garcia-Molina algorithm is the best, compared with other four algorithms because the peer anonymity can be assured through the whole computation process. Algorithm Hybrid, Mengshu and Almenarez are approximately at the same level for maintaining the anonymity level because in general the algorithms operate based on the similar procedure. The algorithm Aberer is at the last of the lists. As previously said in Aberer Algorithm, the flaw lies on the storage system of the data and for the Hybrid algorithm it lies on the pre-trusted peer which is not always anonymous for other peers.

6.7 Evaluation Test Case 7: Non-Symmetric

For non symmetric requirement, all of the algorithms satisfy the requirement, because the trust value calculation applies only to one certain peer. Therefore, all of the algorithms score the same for this test case.

6.8 Evaluation Test Case 5: Performance On Computation Process, Data Storage and Message Complexity

The computation time of the Hybrid algorithm, as seen in the illustration, has the lowest average number which indicates that the Hybrid algorithm does not have a heavy or complicated computation process.

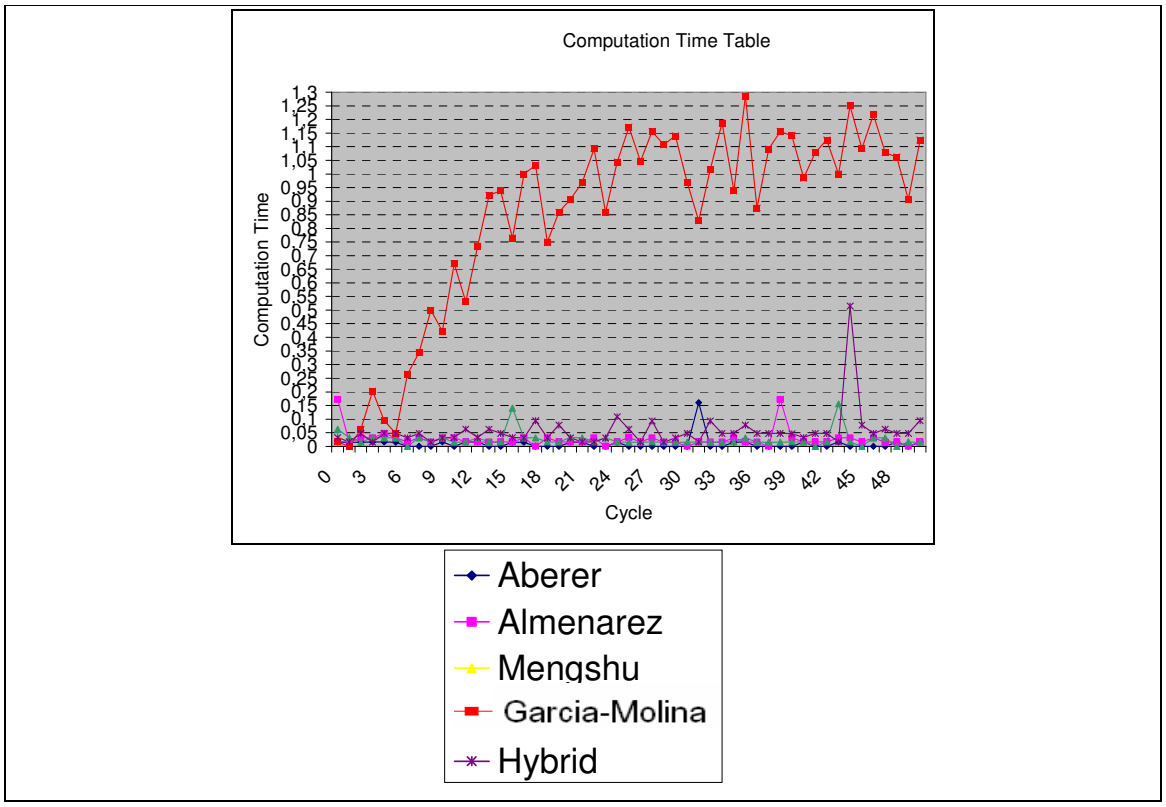


Figure 6.4 Test Result Computation Time

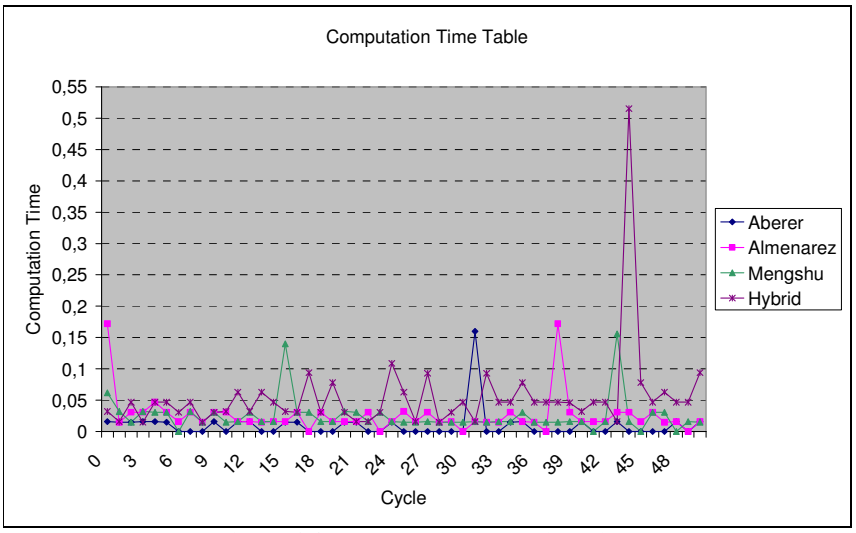


Figure 6.5 Test Result Computation Time

6.9 Evaluation Test Case 9: Implementing A User Intervention System

For this test case only two algorithms that satisfy the requirement, namely the algorithm Almenarez and Hybrid. Both implement the security system in order to give the peers the possibility to set their preferences. The rest of the algorithms do not recognize this feature in their systems.

6.10 Evaluation Test Case 10: Take Into Account The Role of Pre-Trusted Peer

For the last test case, there are also only two algorithms that satisfy the requirement, namely the algorithm from Garcia-Molina and Hybrid. Both implement the pre-trusted role in the algorithm process but with different ways. The algorithm from Garcia-Molina fully depends on the pre-trusted peer while the Hybrid algorithm has less portion of pre-trusted in the system so that this will become a single point of failure.

6.11 Summary

The result from the Hybrid algorithm has been compared with the results from the other algorithms. The Hybrid algorithm has the best result in several numbers of the requirements. Only in peer anonymity and scalability test cases that this algorithm performs less than that of algorithm Mengshu and Garcia-Molina.

CHAPTER 7

Conclusion and Recommendation

7.1 Answering The Research Questions

We have designed some research questions in chapter one in order to guide the research process. In this final chapter, we will see whether these questions can be answered.
There are several sub-questions that are designed as steps towards the answer of the main research questions:
1. What are the shortcomings of the existing algorithms to measure trust in p2p networks?
2. Which beneficial factors can be employed in the construction process of the Hybrid algorithm based on the result of the evaluation of the current algorithms?
3. To what extent does the Hybrid trust algorithm meet the requirement criteria?

1. What are the shortcomings of the existing algorithms to measure trust in p2p networks?
The shortcomings of the existing algorithms to measure trust in p2p networks lay on different parts on each algorithm. Aberer's drawback lies on the binary system which limits the dynamic of trust value. Algorithm Garcia-Molina has a flaw on their trust value calculation where it does not distinguish between private experience and recommendations from other peers which result in higher number of infected files and moreover this algorithm has complicated implementation. Algorithm Mengshu and Almenarez have flaws on the computation process that requires longer process time which leads to lower number of responses and shared files when a query is submitted to the network.

2. Which beneficial factors can be employed in the construction process of the Hybrid algorithm based on the result of the evaluation of the current algorithms?
The beneficial factors that can be employed basically consist of several things:

- Using of system of query propagation from Mengshu that is proven to be suitable for p2p environment.
- Utilize the experience of pre-trusted peer in the network for improving the robustness of trust value calculation against malicious peer activity but reducing the dependency of the need for pre-trusted peer in the network.
- Employ the security level as a means for user intervention where the peers can set up their security according to their preferences

3. To what extent does the new trust algorithm meet the requirement criteria?

Based on the evaluation process, the Hybrid algorithm has the best result in several requirements except for peer anonymity and scalability. For the reflexivity and non symmetric requirements, the Hybrid algorithm has similar performance with other algorithms.
The contribution of this algorithm lies on the combination between user intervention and usage of pre-trusted peer. Some other algorithm depends solely on the pre-trusted peers while others ignore their presence. On the other hand, almost all of the algorithm is automatically set up by the network which does not allow its users to find the best tune. Hybrid algorithm balances this two feature by combining them together to improve the algorithm's performance.

7.2 General Conclusion

The following table shows the summary of the evaluation process of the algorithms. The red colour represents the algorithm that has the best result, while the green colour indicates that there is no best result but same score instead.

Requirements	Algorithms				
	Hybrid (2008)	**Mengshu et. al. (2002)**	**Garcia-Molina et. al. (2003)**	**Almenarez et. al. (2003)**	**Aberer and Despotovic (2001)**
The presence of non-symmetric factor.	✓	✓	✓	✓	✓
The presence of conditional transitivity factor	✓	✓	X	✓	X
The presence of context or classification.	✓	X	✓	✓	X
The presence reflexivity factor	✓	X	✓	✓	X
The presence of dynamic factor.	✓	✓	✓	✓	✓
The presence of scalability factor.	✓	✓	✓	✓	✓
Maintaining peer anonymity	X	X	✓	X	X
Performance on: -computation process, -data storage, -message complexity.	✓	✓	X	X	✓
Implementing a user intervention system	✓	X	X	✓	X
Take into account the role of pre-trusted peer	✓	X	✓	X	X

Table 7.1 Summary

From the result it can be seen that the Hybrid algorithm, has the best result in several of the requirements. Only in maintaining peer anonymity and scalability requirement, the algorithm from Garcia-Molina and Mengshu has a better result. However, this shortcoming gives a room for improvement because there is no such perfect trust algorithm whereas the p2p network system is still developing and getting more complex and advance which always creates new challenges and welcomes a new idea.

7.3 Future Recommendation

There is still a lot of room for future works in this area as this is one of the fastest growing technology in the internet world. As the p2p network is getting complex and advance, there will be a need to develop a simulation tool that has more feature, especially for developing peer feature in the network so that there are more capabilities in the simulation process. There will be also a need to develop new criteria that could improve the performance of the algorithm.

The major drawback of the trust algorithm is that there is still not possible to really apply on the real p2p network during the trial stage because of the large resources need. If in the future this could be made possible, it will make a huge difference because no simulation can beat the real p2p network.

One last thing that limits the performance of the algorithm is the pre-trusted peer. There will be need for a new system on choosing and maintaining anonymity of a pre-trusted peer in the future if a network depends on it; because if the pre-trusted peer turns malicious it could result in a network failure.

References

Aberer, K and Despotovic, Z. (2001). Managing Trust in A Peer to Peer Information System," *CIKM'01*. November 5-10. Atlanta, Georgia.

Aberer, K. and Cudre-Mauroux, P. (2005).Semantic Overlay Networks. *In Proceedings of the 31st VLDB Conference*. Trondheim, Norway.

Abdul-Rahman, A., Hailes, S.(2000).Supporting Trust in Virtual Communities, *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Honolulu, Hawaii.

Abrams, Z. McGrew, R. and Plotkin, S. (2005). A Non-Manipulable Trust System Based on EigenTrust. *in ACM*. Vol: 4. No.4. Page: 21-30.

Agostini, A. and Moro, G.(2004). Identification of Communities of Peers by Trust and Reputation. *in AIMSA 2004*. LNAI 3192. Page: 85-95.

Akerman, A. 2001. Privacy in Context, *HCI*, vol:16, no.2, pp.167-179

Almenarez, F. Marin, A. Campe, and Gracia, CR.(2003). PTM: A Pervasive Trust Management Model for Dynamic Open Environments, *Dept. Telematic Engineering, Carlos III University of Madrid*. Madrid, Spain. 2003.

Antifakos, S, Kern, N, Schiele, B, Schwaninger, A. (2005). Towards Improving Trust In Context-Aware Systems By Displaying System Confidence. *ACM International Conference Proceeding Series,* Vol.111 pp.9-14

Atif.Y, 2002. Building Trust in E-Commerce, *IEEE Internet Computing*, vol.6, no.1.

Bin. Y, Munindar. P. S. (2000). *A* Social Mechanism of Reputation Management in Electronic Communities Proc. of the 4th International Workshop on Cooperative Information Agents, *Lecture Notes in Computer Science*, Vol. 1860.

Capra, L. 2004. "Engineering Human Trust in Mobile System Collaborations", *Foundations of Software Engineering 2004,* pp. 107-116

Crespo, A. and Garcia-Molina. (2005). *Semantic Overlay Networks for P2P Networks.* Technical Report, Stanford University.

Chen. M, Singh. J.P. (2001). Computing and Using Reputations for Internet Ratings", *Proc. of 3rd ACM Conf. Electronic Commerce*.

Daskapan. S. (2005). Medusa: survivable information security in critical infrastructures (PhD thesis, Faculty of Technology, Policy and Management, Delft University of Technology, 2005).

Despotovic.Z, Aberer.K. (2004). Maximum Likelihood Estimation of Peers' Performance in P2P Networks, *Proc. of Workshop on Economics of Peer-to-Peer Systems*.

Despotovic, Z. and Aberer, K. (2005). Probabilistic prediction of peers performance in P2P networks," *School of Computer and Comunication Sciences. Lausanne, Switzerland.* page:771-783.

Dey.A, et al. (1999), Towards a Better Understanding of Context and Context-Awareness, *Technical Report, Georgia Institute of Technology*, pp.9-22.

Guo, et al. (2005). "Trust Model Based On Similarity Measure of Vectors in P2P Networks", *Department of Computer Science, University of Science and Technology of China, China, GCC,* Vol. 3795, pp. 836-847.

Jøsang, A., Ismail, R.2002. (2002). The Beta Reputation System. *Proceedings of the 15th Bled Conference on Electronic Commerce*.

Jøsang, A, Keser, C, Dimitrakos, T. (2005). Can We Manage Trust?, *Lectures Notes in Computer Science,*Vol.3477, pp.93-107

Jin, H. Tu, X. Han, Z and Liao, X. (2005). A Community-Based Trust Model for P2P Networks, *Cluster and Grid Computing Lab, University of Science and Technology.* Wuhan, China.

Kaasinen, E. (2003). "User Needs for Location-Aware Mobile Services", *Personal and Ubiquitous Computing,* Vol.7 No.1 pp.70-79

Kaasinen, E. (2005). "User Acceptance of Location-Aware Mobile Guides Based on Seven Field Studies", *Behaviour and Information Technology*, Vol.24, No.1 pp.37-49

Kamvar.S.D, Scholsser.M.T, Garcia-Molina.H. (2003). The EigenTrust Algorithm for Reputation Management in P2P Networks, *Proc. 12th Intl World Wide Web Conf.*, 2003.

Kamvar.S.D, Schlosser.M.T, Garcia-Molina.H. (2003). Incentives for Combatting Free Riding on P2P Networks", *Technical report*, Stanford University.

Kenneth. B, Clifford. S, Robert. L. D. (2006). *Discrete mathematics for computer science*. Emeryville: Key College Publishing.

Kinateder, M, Baschny, E, Rothermel, K. (2005). Towards a Generic Trust Model-Comparison of Various Update Algorithms, *Lecture Notes in Computer Science*, Vol. 3477. pp.177-192

Kinateder, M. Rothermel, K. (2003). Architecture and Algorithms for a Distributed Reputation System, *Proceedings of the First International Conference on Trust Management*. Vol. 2692 pp.1–16

Marsh, S.P. (1994). Formalising Trust as a Computational Concept. *PhD thesis, Department of Mathematics and Computer Science*, University of Stirling

Mengshu, H et. al. (2003). A Trust Model Of P2P System Based On Confirmation Theory, *College of Computer Science and Engineering of UEST of China*, Chengdu, China, pp.1-7.

Oram, A et al. (2001). *Peer to Peer: Harnessing the Benefits of a Disruptive Technology*. CA: O'Reilly & Associates. 2001

Pirzada, AA. And McDonald, C. "Establishing Trust In Pure Ad-Hoc Networks," in 27th Australian Computer Society, Inc. Conference in Research and Practice in Information Technology. Vol:26. The University of Otago, Dunedin, New Zealand. 2004.

Ratnasamy, P. Francis, M. Handley, R. Karp, Shenker.S, 2001. *A* Scalable *Content-Addressable Network* Proc. of the ACM SIGCOMM.

Resnick, P, Zeckhauser. R, Friedman.E, Kuwabara. K, (2000). "Reputation systems: Facilitating trust in Internet interactions", *Communications of the ACM*, 43(12): 45 - 48.

Ruohomaa, S, Kutvonen, L. (2005)."Trust Management Survey", *Lecture Notes in Computer Science*, Vol.3477, 2005 pp.77-92

Sabater, J.2003. Trust and Reputation for Agent Societies. *PhD thesis, Institut d'Investigaci en Intelligencia Articia*l, Bellaterra

Shuqin, Z. Dongxin, L. and Yongtian, Y. (2004). A Fuzzy Set Based Trust and Reputation Model in P2P Networks," *in IDEAL 2004*. pp 211-217.

Sierra, C. and Debenham, J. (2005). An Information-Based Model for Trust. July 25-29. Utrecht, Netherlands. 2005.

Stoica. R, Morris, D. Karger. F, Kaashoek, H, Balakrishnan. I. (2001) chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications Proc. of the ACM SIGCOMM, *ACM.*

Shi, J. Bochmann, G. and Adams, C. 2003). A Trust Model with Statistical Foundation. *System Science, School of Information Technology and Engineering*, University of Ottawa. Ottawa, Canada.

Wang, Y. and Varadharajan, V. (2004). Interaction Trust Evaluation in Decentralized Environments. *Department of Computing, Macquarie University*. Sidney, Australia.

Wang, Y and Vassileva, J. (2004). Bayesian Network Trust Model in Peer to Peer Networks. *Computer Science Department, University of Saskatchewan*. Saskatton, Canada. 2004.

Xiong, L and Liu, L. (2000). Building Trust in Decentralized Peer to Peer Electronic Communities. *in International Conference on Electronic Commerce Research.*

Yu, B. and Singh, MP, (2003). Detecting Deception in Reputation Management,". July 14-18. Melbourne, Australia.

Yu, B. and Singh, MP. (2002). An Evidential Model of Distributed Reputation Management," *in AAMAS'02*. July 15-19. Bologna.

Verschuren, P. and Doorewaard, H. (1999). *Designing a research project*. Utrecht: Lemma BV

APPENDIX 1
Current Trust Algorithms

This section will provide brief descriptions on the current algorithms that are not chosen to represent the four sub-categories. The description will start with the mathematical formula and move further with the propagation of the query from the requested peer in the network to obtain the responses from other peers in order to obtain the requested sharing file. However, most of the following algorithms lack on providing the necessary information regarding the query propagation which makes it difficult to evaluate the responding process. In this case, the evaluation will be provided based solely on the mathematical formula.

Categorization:	**Category 1: Partial Algorithm**	**Category 2: Global Algorithm**
Distinction between direct and indirect trust	- Trust algorithm based on Fuzzy Set (Shuqin et. al, 2004) - Trust algorithm with Confirmation Theory (Mengshu et. al, 2002) - Trust algorithm with Community Peers (Agostini and Moro, 2004) - Trust algorithm with Distributed System (Abdul-Rahman and Hailes, 1997)83 - - Trust algorithm with Evidential Model (Yu and Singh, 2002) - Trust algorithm with Statistical Foundation (Shi et. al, 2003)	- Trust algorithm with Bayesian network (Wang and Vassileva, 2004) - Trust algorithm with Beta Reputation System (Josang and Ismail, 2002) - Trust algorithm based on Similarity Measure of Vector (Guo et. al, 2005) - Trust algorithm with Ad-Hoc Environment (Almenarez et. al, 2003)
No distinction between direct and indirect trust	- Trust algorithm with Information System (Aberer and Despotovic, 2001) - Trust algorithm with Information Based Model (Sierra and Debenham, 2005)	- Trust algorithm in Decentralized Community (Xiong and Liu, 2004) - Trust algorithm in Decentralized Environment (Wang and Varadharajan, 2004) - Trust algorithm in Pure Ad-Hoc Network (Pirzada and Mc.Donald, 2004) - Trust algorithm with Community Based Model (Jin, 2005) - Trust algorithm with EigenTrust System (Garcia-Molina et. al, 2003)

Table 1 Algorithm Categorization

1. Category 1: Partial Trust Algorithms

1.1 Trust algorithm based on Fuzzy Set (Shuqin et. al, 2004)

In this algorithm the trust value is divided into two direct trust value and indirect trust value. The indirect trust value or often called reputation is obtained through aggregation of other peers' judgements (recommendations). These direct and indirect trust values are obtained through a fuzzy set based model. A fuzzy vector, T, is associated with a certain trust level which later defines the direct trust value. For the indirect trust value, some peers, called RMs,

are chosen as reputation managers in which other peers could fetch the recommendation values from. These RMs possess recommendation tables which contain recommendation of the other peers.

Trust Value Calculation:

Refer to the context scenario, if peer A wants calculate peer B's trust value and peer F is the reputation manager, trust value of peer A, T_{AFB}, can be calculated as follows:

$$T_{AFB} = \alpha T_{AB} + (1-\alpha) T_{FC} \qquad 0 < \alpha < 1$$

Variable	Description
A	The peer who wants to calculate other peer trust value
B	The peer whose trust value is being assessed
F	A peer who gives recommendation to other peer
T_{AFB}	trust value of peer B by peer A based on direct trust value and indirect value
α	proportion direct trust to indirect trust
T_{AB}	direct trust value of peer B by peer A based on private experience
T_{FC}	indirect trust value of peer B by recommendation from recommender peer F

Table 2 Legend

Context Scenario:

Even though the algorithm makes distinction between direct and indirect value, it is not described how these two values can be obtained and calculated. It is also not described how the peer manager is chosen and how they will aggregate the trust value. Moreover, this algorithm provides no further information on how the query from the peers in the network are being propagated throughout the network, therefore it is very difficult to evaluate how this algorithm with cope with the network scalability and its influence on the network (and peer) performance. Thus, if peer A in the context scenario submits a request query for a file to the network, it is not clear how this query will be propagated through the network and as the result the query responses is difficult to observe.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	X
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	X
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 3 Result of Shuqin Algorithm

1.2 Trust algorithm with Community Peers (Agostini and Moro, 2004)

This algorithm is basically an adaptive game on how to choose the most reliable route in terms of choosing the right peers in order to be able to receive the right response. In this algorithm, the trust value is also calculated based on the combination between the private experience of the interacting peers and the recommendations from other peers.

Trust Calculation:
If peer A wants to have a certain file from the network, it will send several request queries to other peers. The type of the file will define the context of the query that later will define the query propagation cluster (a cluster represents a group of peers which has similar type of files). Let say peer C and peer D respond to the query, peer A will calculate the trust value of peer C and peer D to find which peer is the most trustworthy. This is done first by calculating peer A private experience with peer C and peer D (in this algorithm defined as rate of success, σ) and combine this value with the recommendation about peer C and peer D that is obtained from a peer manager F within the network. The trust value symbol is denoted as follows:

$$T_{AC}^{Q} = \sum_{i=1}^{n} \sigma_i + T_{FC}^{Q} \quad \text{.....(I)}$$

$$T_{AD}^{Q} = \sum_{i=1}^{n} \sigma_i + T_{FD}^{Q} \quad \text{.....(II)}$$

Equation (I) and (II) show the trust value calculation of peer C and peer D respectively. Based on this value, peer A will be able to make decision which peer is the most trustworthy one.

Variable	Description
A	The peer who wants to calculate other peer trust value
C	The peer whose trust value is being assessed
D	The peer whose trust value is being assessed
F	The peer who acts as peer manager
Q	Query context
n	Number of transactions
σ	Rate of success
$\sum_{i=1}^{n} \sigma_i$	Local trust value of peer C or peer D by peer A
T_{ABQ}	Trust value of peer A by peer B regarding query context Q

Table 4 Legend

Context Scenario:
Referring to the context scenario, if peer A wants to have a certain file from the network, it must indicates first the context of the file query that it is about to be sent (such as: document file, music file, move file, etc). The context of the query will define the peer cluster to where the query will be propagated. Next, peer A will send the file query to any random peers within the defined cluster. This query will be sent couple of times to find several recommenders who have the requested file.

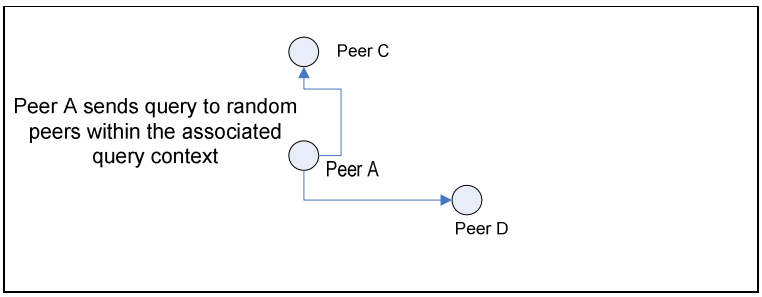


Figure 1 Query Propagation

If the TTL of the query is up, the peers in the cluster that have the requested file will send their response back to peer A. Next, peer A will starts to calculate the responding peer's trust value before displaying it under search result.

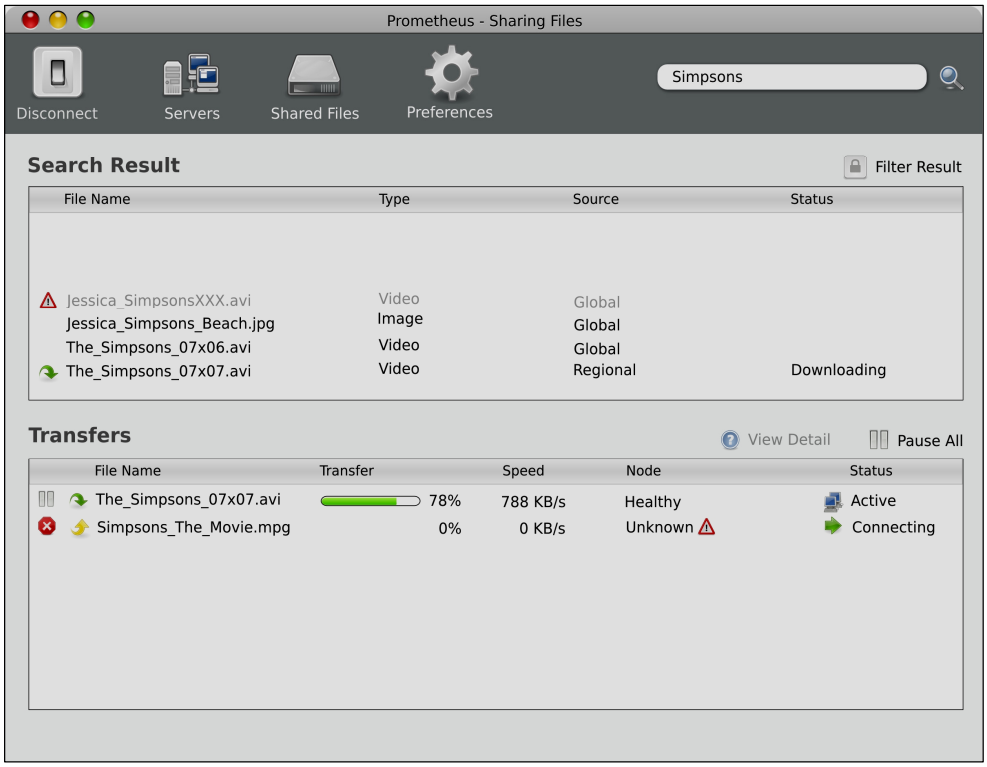


Figure 2 Query Result

Even though the process to query propagation and query response is described here but the detail mathematical formula on calculating the private experience (local trust value) and how to calculate recommendation is missing. Therefore, it is difficult to evaluate the performance of the algorithm and to implement this algorithm in the simulation tool.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 5 Requirement Agostini and Moro

1.3 Trust algorithm with Distributed System (Abdul-Rahman and Hailes, 1997)

This algorithm is one of the algorithms that frequently cited in various journals. Abdul-Rahman and Hailes algorithm is considered as the simplification from the Marsh algorithm (Marsh, 1994). The algorithm propose an algorithm that more simple than that of Marsh to be able to be implemented in real distributed system. This algorithm is the first one that introduced the notions general and situational trust which respectively means the global trust value and the local trust value of a specific peer. The trust value is computed in this algorithm based on these two notions. The first notion refers to other well known notion, recommendation, which is information that a peer receives from other peers. And the second one refers to the private experience between the connecting peers.

Trust Value Calculation:
Referring to the context scenario, if peer A wants to calculate peer B's trust value with the help recommendation from peer C, the trust value computation can be defined as follows:

$$T_{ACB} = \frac{T_{AC} \times T_{CB}}{4} \times T_{AB}$$

Variable	Description
T_{ACB}	Trust value of peer B by peer A with recommendation from peer C
T_{AC}	Trust value of peer C by peer A (how much peer A trusts peer C)
T_{CB}	Trust value of peer B by peer C (recommendation value from peer C to peer A)
T_{AB}	Trust value of peer B by peer A (private experience or local trust value)

Table 6 Legend

As seen in the equation, the algorithm is implementing the conditional transitivity factor but the detail mathematical formula is missing from this algorithm on how to calculate the local trust value of peer B by peer A. Abdul Rahman and Hailes (1997) provide an example with a local trust value already defined.

Context Scenario:
Furthermore, even though this algorithm is provided with the detail of the composition of the file query and recommendation query, the propagation process itself is missing. As a result, it is difficult to evaluate the scalability of the algorithm and its impact on the performance of the network and the peers. Therefore, if peer A submits a query for a certain file, it would be difficult to observe the result of the query. The complete result of the evaluation on the algorithm is provided in the next table.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	**X**
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 7 Requirements Abdul Rahman and Hailes

1.4 Trust algorithm with Evidential Model (Yu and Singh, 2002)

The algorithm uses this theory of evidence as foundation of the trust computation where the recommendations from other peers are considered as the evidences found, and combined together as belief functions. As there are two important factors that define the trust value (the private experience and recommendation from other peers), the final trust value are the combination of these two factors.

Trust Value Calculation:
Based on the context scenario, if peer A wants to have a transaction with other peer B, the process will be as follows:

1. The system will set up a lower threshold ω and upper threshold Ω for every calculated trust value of each peer in the network.
2. Peer A will have to check first if it has private experience with peer B (checking its local belief regarding peer B). The local belief is the basic elements of calculating peer B trust value. It is obtained by calculating a basic probability assignment bpa_{AB} of peer B by peer A. If $f(x_k)$ denotes the probability that a certain trust value of peer B happens between the lower and upper threshold, the bpa can be calculated as follows:
   $$bpa = \sum_{x_k=\Omega}^{1} f(x_k) + \sum_{0}^{x_k=\omega} f(x_k) + \sum_{x_k=\omega}^{x_k=\Omega} f(x_k) \quad \text{.....(I)}$$
3. If peer A does not have private experience with peer B it will ask other peer for recommendation by sending query to any random peers in the network. Let say if peer

C and peer D give recommendation regarding peer B (bpa_{CB} and bpa_{DB}), the final result will be the sum of the recommendations:

$bpa_{total} = bpa_{CB} \oplus bpa_{DB}$ .....(II)

Based on the result of the (total) bpa, peer A can decide whether it will proceed to have transaction with peer B.

Variable	Description
bpa	basic probability assignment which reflects the trust value of a certain peer
ω	lower threshold, defined by the system
Ω	upper threshold, defined by the system
$f(x_k)$	probability that a certain trust value of a peer happens between the lower and upper threshold.
bpa_{CB}	basic probability assignment of peer B by peer C
bpa_{DB}	basic probability assignment of peer B by peer C

Table 8 Legend

Context Scenario:

Referring to the context scenario, if peer A wants to have a certain file from the network, it will send the file query to any random peers within the network, let say to peer C and peer D, as shown in the following figure.

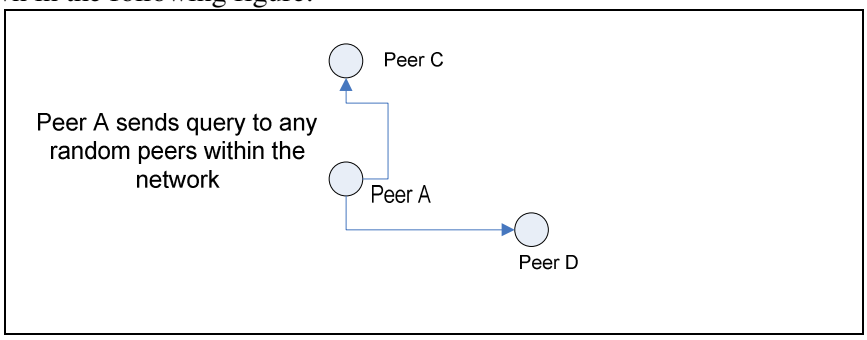


Figure 3 Query Propagation

The result of the query is displayed in the following figure. Because of the random propagation, the result will be less than an algorithm that uses more coordinated propagation.

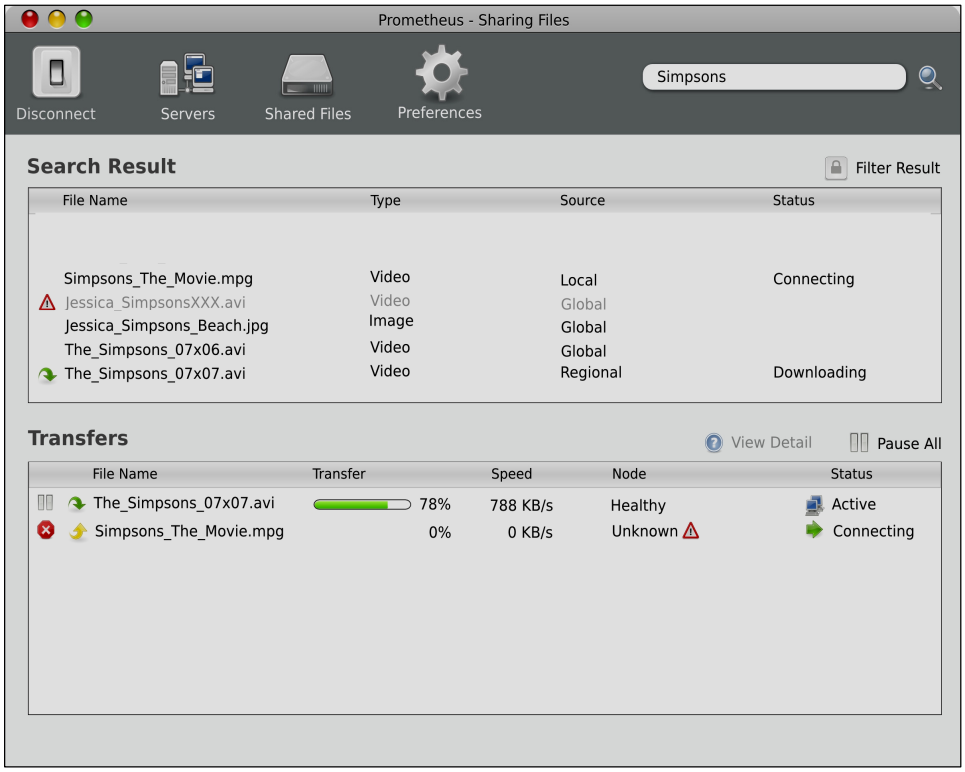


Figure 4 Query Result

The drawback of this algorithm is that it relies on honest feedback from other peers in order to sufficiently compute the trust value. If peer A finds it self surrounded by malicious peers it result in falsely trust value. Another drawback is that there is no distinction between the rating of a new peer and rating of malicious peer. A new peer can be wrongly identified by other peer as malicious one and therefore limits the new peer's activity with other peer in the network.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	**X**
The presence reflexivity factor	✓
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 9 Requirements Yu and Singh

1.5 Trust algorithm with Statistical Foundation (Shi et. al, 2003)

This algorithm is based on statistical analysis that intended to assist the decision making process on the interaction among peers in the network in order to make an optimal choice. The trust algorithm here based on the notion that trust behaves like stochastic process where the outcome can not be fully predicted.

Trust Value Calculation:
Based on the context scenario, if peer A wants to calculate peer B's trust value, peer A will have to first define the context of the trust value that is about to be calculated. Next, peer A can calculate the trust value based on probability of peer B's trust value distribution. If O is the space of possible outcome, c is the context of the trust value and N is the number of the possible outcome, the trust value of peer B by peer A (T_{AB}^{c}) can be described as follows:

$$T_{AB}^{c} = \frac{1}{N}\sum o_c$$

Variable	Description
T_{AB}^{c}	trust value of peer B by peer A regarding context c
N	number of possible outcomes
c	trust value context
o_c	possible outcomes within the context c

Table 10 Legend

Context Scenario:
The major drawback of this algorithm is the lack of information on how to get the information from other peers to compute the trust value (query propagation) also where and how the trust will be stored in the network. If peer A wants to submit a request query for a certain file, the propagation process will be missing which makes it difficult to study the query response. Therefore, it makes it difficult to assess the complexity of the computation process and the volume of data storage the system will be needed.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	X
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	X

Table 11 Requirements Shi

1.6 Trust algorithm with Information Based Model (Sierra and Debenham, 2005)

This algorithm is based on information theory with the goal to assist the decision making process of a peer. In this algorithm, trust is defined as *a measure of expected deviations of behaviour along a given dimension for a given value* (Sierra and Debenham, 2005).

Trust Value Calculation:
Thus, referring to the context scenario if peer A wants to share file with peer B it has to calculate peer B's trust value. In this algorithm it is done by calculating the conditional probability between peer A and peer B based on their previous transactions. Thus, the trust value of peer B by peer A measured at time t and within context c is defined as follows:

$$T_{AB}^{ct} = \frac{1}{\beta} \sum P_{AB}^{ct} \log P_{AB}^{ct}$$

Variable	Description
T_{AB}^{ct}	trust value of peer B by peer A, measured at time t within context c.
β	a constant, set by every peer which defines how much peer A trusts the conditional probability calculation
P_{AB}^{ct}	conditional probability of peer B's trust value by peer A
c	context of the trust value
t	specific time when trust value is being calculated.

Legend 12

The algorithm does not provide the detail mathematical formula on how the conditional probability is being measured which makes it complicated to be evaluated. However, the algorithm provides a brief description on the query propagation.

Context Scenario:
If peer A needs to obtain a certain file from other peer, it will send a query to any random peers in the network. The peers who have the requested file are welcome to respond. The illustration of the query propagation is provided the following figure and the result of the evaluation is shown in the following table.

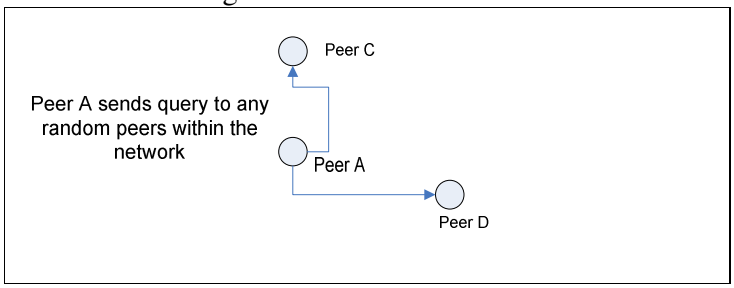


Figure 5 Query Propagation

The result of the query is displayed in the following figure. Because of the random propagation, the result will be less than an algorithm that uses more coordinated propagation.

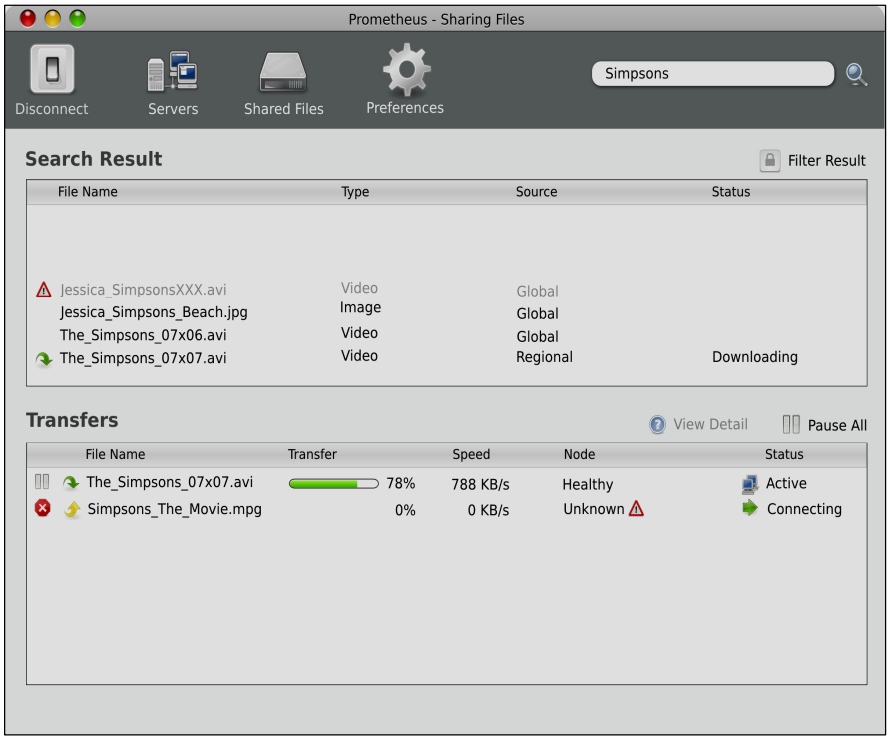


Figure 6 Query Result

Based on the provided information, the result of the evaluation process of this algorithm is presented in the following table.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	**X**
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 13 Requirements Sierra and Debenham

2. Category 2: Global Trust Algorithms

2.1 Trust algorithm with Bayesian network (Wang and Vassileva, 2004)

This algorithm is based on the theory of probability called Bayesian Rules. This theory describes the outcomes of probability proportion of a random event. If X and Y are two

random events, the Bayesian Rules defines the probability a new event will happen based on the events X and Y. The Bayesian probability can be written as follows:

$$P(X \mid Y) = \frac{P(Y \mid X)\,P(X)}{P(Y)}$$

Where:

$P(X)$ = the prior probability of event X happened

$P(X \mid Y)$ = the probability of event X will happen, given event Y

$P(Y \mid X)$ = the probability of event Y will happen, given event X

$P(Y)$ = the prior probability of event X happened

Trust Value Calculation:

In this algorithm, every peer within the network keeps a Bayesian computation for each peer with whom it has performed transaction with. The computation has two possibility outcomes: 0 = successful, 1= successful. The algorithm is implementing the context factor into the calculation process. Wang and Vassileva (2004) give example of five different contexts. The contexts are presented in the conditional probability table (CPT) as follows:

	T=1	T=0
Music	p = (FT="Music"| T=1)	p = (FT="Music"| T=0)
Movie	p = (FT="Movie"| T=1)	p = (FT="Movie"| T=0)
Document	p = (FT="Document"| T=1)	p = (FT="Document"| T=0)
Image	p = (FT="Image"| T=1)	p = (FT="Image"| T=0)
Software	p = (FT="Software"| T=1)	p = (FT="Software"| T=0)

Table 14 The CPT Table of peer A for peer B (Wang and Vassileva, 2004)

The table shows the probability of successful transaction (trust value, T=1) or unsuccessful transaction (trust value, T=0). FT stands for file type that defines the context of the trust value. Based on the Context Scenario, if peer A wants to calculate peer B's trust value, peer A will keep a CPT table as shown above for peer B (and every other peer it has transactions with). Thus, the probability that the transactions between peer A and peer B will be successful for sharing music files can be written as follows:

$$p(FT="Music" \mid T=1) = \frac{p(FT="Music",\ T=1)}{p(T=1)}$$

$$p(FT="Music",\ T=1) = \frac{S}{S+F} \quad \text{.....(I)}$$

Where:

S = number of successful transactions between peer A and peer B

F = number of unsuccessful transactions between peer A and peer B

$S + F$ = number of total transactions between peer A and peer B

The result of this calculation denotes the trust value of peer A by peer B based on the private experience of peer A with peer B.
If peer A is not sure about the result from equation (I), it can ask other peer in the network for recommendation. It is done by sending a recommendation query to any random peers in the network. If for example, peer C and peer D answer the query, the final recommendation value will be:

$$T_{ACDB} = W_C * \frac{T_{AC} * T_{CB}}{T_{AC}} + W_D * \frac{T_{AD} * T_{DB}}{T_{AD}}, \text{ where } W_C + W_D = 1 \quad \text{.....(II)}$$

Variable	Description
S	Number of successful transactions between peer A and peer B
F	Number of unsuccessful transactions between peer A and peer B
T_{ACDB}	Trust value of peer B by peer A based on the calculation from recommendation from peer C and peer D
W_C and W_D	Weights that indicates how the peer A values the importance of recommendation from peer C and peer D, where $W_C + W_D = 1$
T_{AC}	Trust value of peer C by peer A
T_{CB}	Trust value of peer B by peer C
T_{AD}	Trust value of peer D by peer A
T_{DB}	Trust value of peer B by peer D

Table 15 Legend

Context Scenario:
Referring back to the context scenario, if peer A wants to have a certain file from other peer in the network, it will send a file query to any random peers in the network. The peers which have the requested file can send a response back to peer A. The query propagation can be shown as follows:

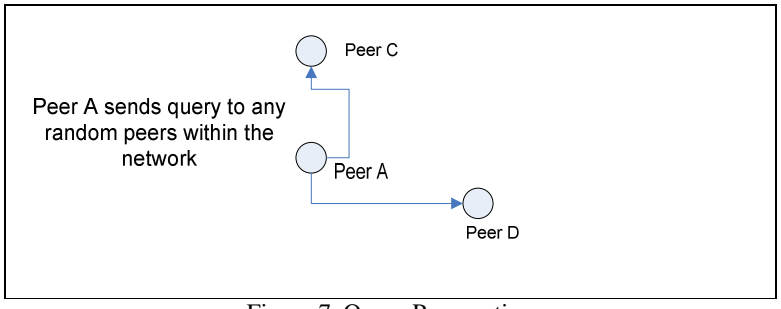


Figure 7 Query Propagation

As previously described, the query is equipped with TTL which defines the period of time of the query before it will be discarded. If the TTL is up, the other peers in the network can start to respond the query. Because of the random query, the number of the query response will always vary and the result will be less than that of other algorithm that is used more coordinated query propagation such as Mengshu algorithm.

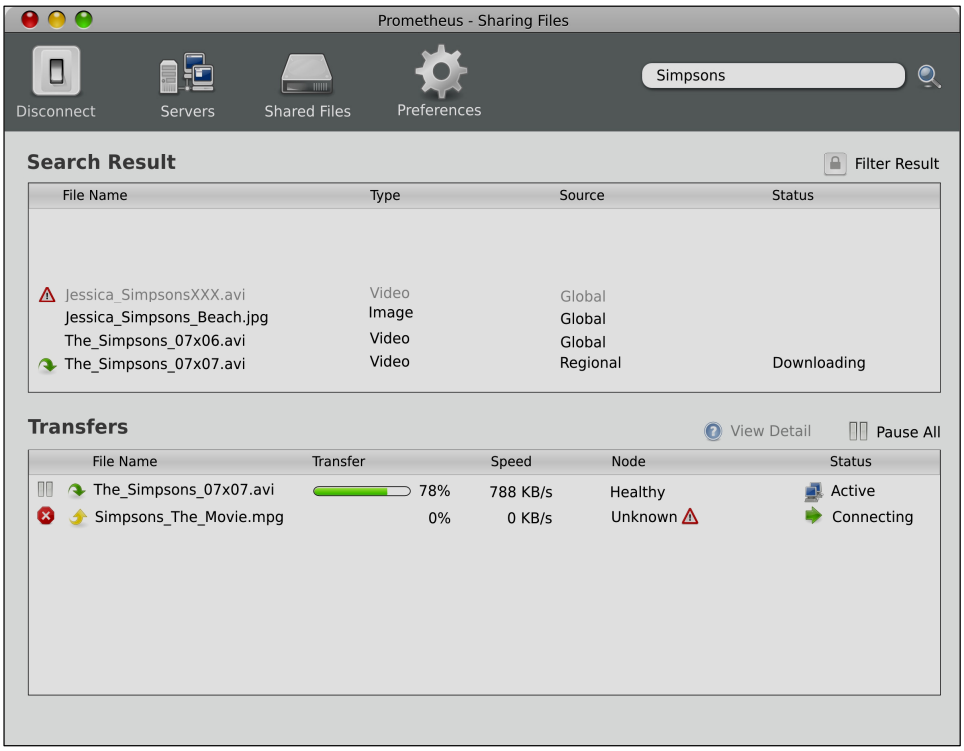


Figure 8 Query Result

The downside of this algorithm is that it works well in the small network where the peers have a high number of transactions with the same peers, but for a large network where the peers mostly have only one transaction with the same peer, the Bayesian probability theory is not effective. Other shortcoming in this algorithm is that in order to make to algorithm works well, all of the peers in the network must have matching element of the Bayesian network or in other words the peers must have similar preferences, which is very unlikely in the real p2p network. The following table provides the result of the evaluation process on the algorithm.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	X
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 16 Wang and Vassileva

2.2 Trust algorithm based on Beta Reputation System (Josang and Ismail, 2002)

This algorithm is based on two basic mechanisms, namely:

- A method to compute trust value, based on various inputs, including the peer's private experience and information from other peers.
- A propagation mechanism that allows peers to gain information about other peers reputation whenever it is needed.

Trust Value Calculation:

The basic theory of this algorithm is the beta probability density function that can be used to compute the probability distribution of a binary event.

The beta probability is represented by two parameters α and β. This probability function $f(p|\alpha, \beta)$ can be presented using gamma function Γ as follows:

$$f(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \text{ where } 0 \le p \le 1, \ \alpha>0, \beta>0 \quad \text{.....(I)}$$

And the probability expectation value E (p) can be written as:

$$E(p) = \alpha/(\alpha+\beta) \quad \text{.....(II)}$$

Suppose, with the possible outcome $\{x, \bar{x}\}$, where S as the outcome (successful transaction) of x and F as the outcome of $\bar{x}$ (unsuccessful transaction), we can derive the reputation function of peer B that is calculated by peer A as follows:

$$\varphi(p \mid S_{AB}, F_{AB}) = \frac{\Gamma(S_{AB}+F_{AB}+2)}{\Gamma(S_{AB}+1)\Gamma(F_{AB}+1)}, \text{ where } 0 \le p \le 1, 0 \le S_{AB}, 0 \le F_{AB} \quad \text{.....(III)}$$

Where:

Variable	Description
$\varphi(p \mid S_{AB}, F_{AB})$	probability of trust value of peer B by peer A
S_{AB}, F_{AB}	reputation parameter of peer B by peer A
S_{AB}	successful transaction between peer A and peer B
F_{AB}	successful transaction between peer A and peer B

Table 17 Legend

And the probability expectation that the result from equation (III) will happen can be written as:

$$E(\varphi(p \mid S_{AB}, F_{AB})) = \frac{(S_{AB}+1)}{(S_{AB}+F_{AB}+2)} \quad \text{.....(IV)}$$

Naturally, we can not believe all of the information that we receive. The same notion is incorporated in the algorithm where a peer should believe information from a peer with good reputation more than from a peer with a worse reputation. This is achieved by implemented weight that corresponds with the reputation rating of a peer.

In addition with the weight implementation method, the algorithm also proposes an updating method, where a peer can implement a forgetting factor λ, where if λ=1 means that all of the information will be kept, while λ=0 means that only the latest information will be kept.

All of the information of the peers' reputations are collected and stored by a central directory. All of the peers in the network which needs information regarding other peer reputation can send a query to fetch the information from the directory.

Context Scenario:

Based on the context scenario, if peer A wants to obtain a certain file from the network, it will send a query to the central server to find which peers that have the requested file.

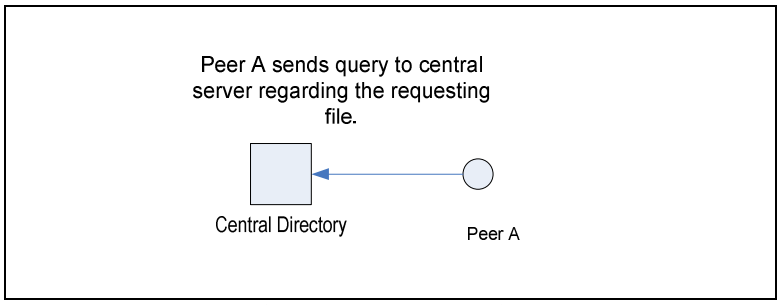


Figure 9 Query Propagation

The central directory will inform peer A which other peers in the network that have the requested file. The result is displayed in the figure below. Next, peer A will have to calculate the peers' trust values before choosing one of them.

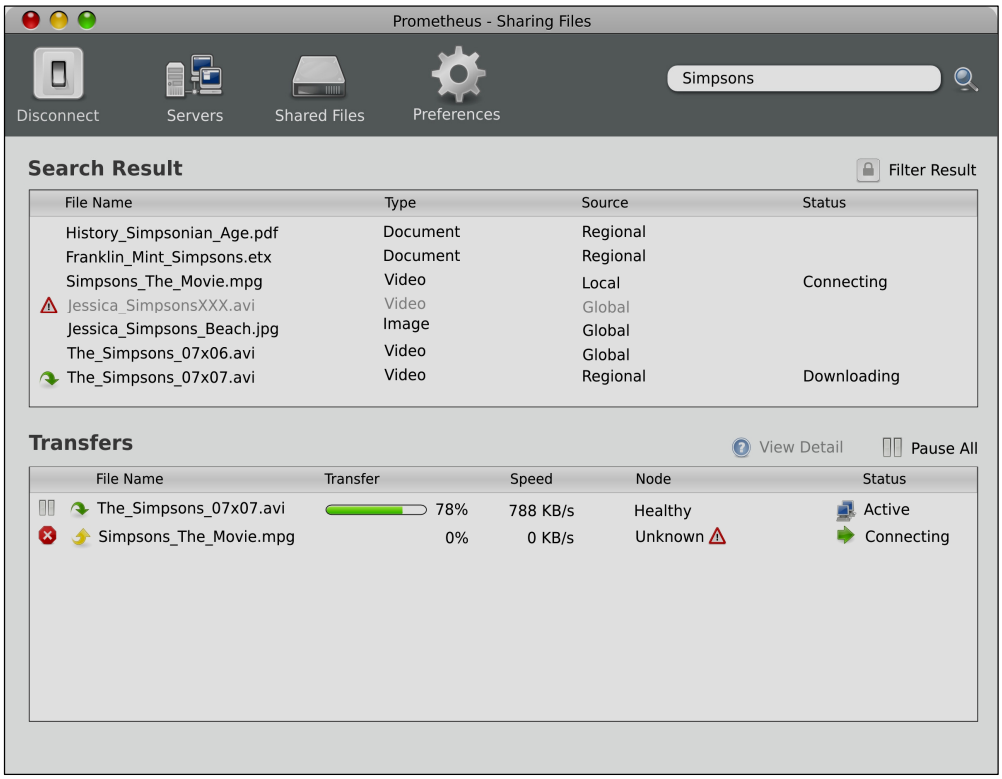


Figure 10 Query Result

Based on the provided information above, the algorithm is evaluated with the help of design requirements. It can be seen from the description that this algorithm does not meet several of the requirements, such as reflexivity factor and context/classification factor. The algorithm never states in the process regarding the reflexive factor and also the algorithm never mentions the role of context/classification during the calculation process. The overall result of the evaluation on this algorithm is provided in the following table:

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	X
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	X
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	✓
Performance on: - Computation process - Data storage - Message complexity.	X

Table 18 Josang and Ismail

2.3 Trust Algorithm in P2P Network with R-Chain (Guo et. al, 2005)

This trust algorithm is based on implementation of R-Chain system. This algorithm system is a self-maintained reputation management in p2p network (Liu et al. 2004), which features two important factors, namely time-sensitive factor and service-type factor (context/classification factor), to enhance the accuracy of trust evaluation within the network (Guo et. al, 2005).
In this algorithm, there are two different trusts that exist, namely direct and indirect trust value. Direct trust is trust that based on the transactions between two connecting peers, while indirect trust is based on the recommendation from other peers.

Trust Value Calculation:
The following figure illustrates the data structure and the implementation of R-Chain. If peer A wants to calculate peer B's trust value, peer A will have to calculate first the direct trust value which is based on its previous transactions (private experience) with peer B. If C_{AB} denotes the local trust value of peer B by peer A, X denotes the context of the trust value of peer B (defined by) and n denotes the number of transactions between A and B, the trust value T_{AB}^{X} can be calculated as follows:

$$T_{AB}^{X} = \frac{1}{n} * \sum (C_{AB} * X_B) \quad \text{.....(I)}$$

If this value is calculated, peer A will have to calculate the indirect trust value that is based on the recommendation from other peer. This is the time where the R-Chain system is used. Figure 11 shows the data structure of R-Chain where peer A keeps every previous transaction record with other peers including the peer ID, denoted with TR.

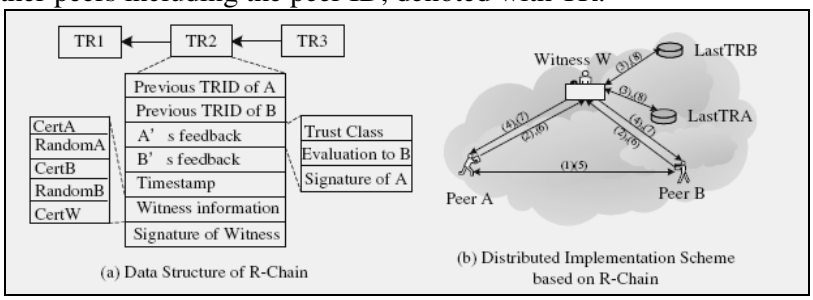


Figure 11 R-Chain Structure (Guo et. al, 2005)

Based on this system, peers A's recommendation information will be organized in a linked chain. Thus, to get a recommendation value from other peer, peer A will select a random recommender peer W. Peer B will send its last TR to peer W. Peer W will verify the information in the TR by contacting the peer with whom peer B has had the last transaction. If this is completed, peer W will send the result back to peer A. Next, peer A can calculate its indirect trust value based on the recommendation from peer W. If the other factors are defined: set of recommenders: $RSet$, peer W trust value by peer A within context X: T_{AW}^{X}, number of recommendation: m, and context of trust value: X; the indirect trust value of peer B by peer A with recommendation from peer W can be defined as:

$$T_{AWB}^{X} = \frac{1}{|RSet|} * \sum_{\forall R \in RSet} [T_{AW}^{X} * \frac{1}{m} * \sum (T_{WB}^{X} * X_B)] \quad \text{.....(II)}$$

If the equation (I) and (II) are combined, the final trust value of peer B by peer A can be written as:

$$T_{AWB}^{X} = \lambda * T_{AB}^{X} + (1-\lambda) * T_{AWB}^{X} \quad \text{.....(III)}$$

As previously mentioned, this algorithm implements time sensitive factor as part of an important variable in the trust value calculation. With this factor, the trust value can be updated every time a new transaction takes place. If δ denotes the proportion between the trust value that is calculated between t0 ($T_{AWB,t0}^{X}$) and t1($T_{AWB,t1}^{X}$), the new calculated trust value (T_{AWB}^{X}) can be obtained as follows:

$$T_{AWB}^{X} = \delta T_{AWB,t0}^{X} + (1-\delta) T_{AWB,t1}^{X} \quad \text{……..……(IV)}$$

Variable	Description
A	the peer who wants to calculate other peer trust value
B	the peer whose trust value is being assessed
W	the recommender peer
X_B	trust category which defines the quality of the sharing file within the same context. This is defined by peer B itself. All peers have to assign a certain number ($0 \le X_B \le 1$) to define the quality of their sharing files.
T_{AWB}^{X}	trust value of peer B by peer A with recommendation of peer W within context X
λ	self-confident factor to balance the proportion between local direct and indirect trust value calculation. this is defined by the system
T_{AWB}^{X}	indirect trust value of peer B by peer A with recommendation from peer W
T_{AB}^{X}	direct trust value of peer A by peer B within context X
C_{AB}	local trust value of peer B by peer A based on the number of transactions
n	number of transaction
$RSet$	sets of recommendation peers
T_{AW}^{X}	Trust value of peer W by peer A within the context X
δ	The proportion between trust value calculated at t0 and t1
$T_{AWB,t0}^{X}$	Trust value calculated at t0
$T_{AWB,t1}^{X}$	Trust value calculated at t1
m	Number of recommendation

Table 19 Legend

Context Scenario:
Based on the information about this algorithm and the context scenario, is peer A wants to have a certain file from the network, it will send a file query randomly to the network, as illustrated in the following figure.

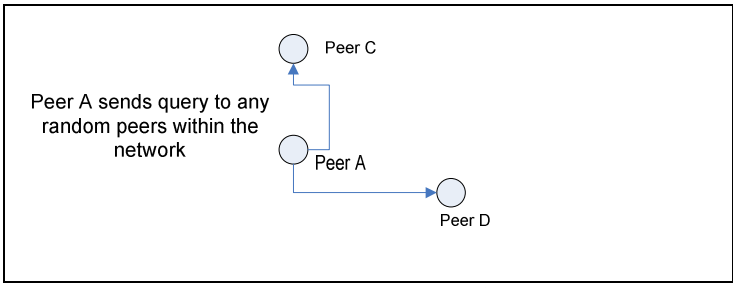


Figure 12 Query Propagation

The peers who have the requested file will answer the query. Peer A will calculate each of this peer's trust value before displaying it under the search result in the following figure.

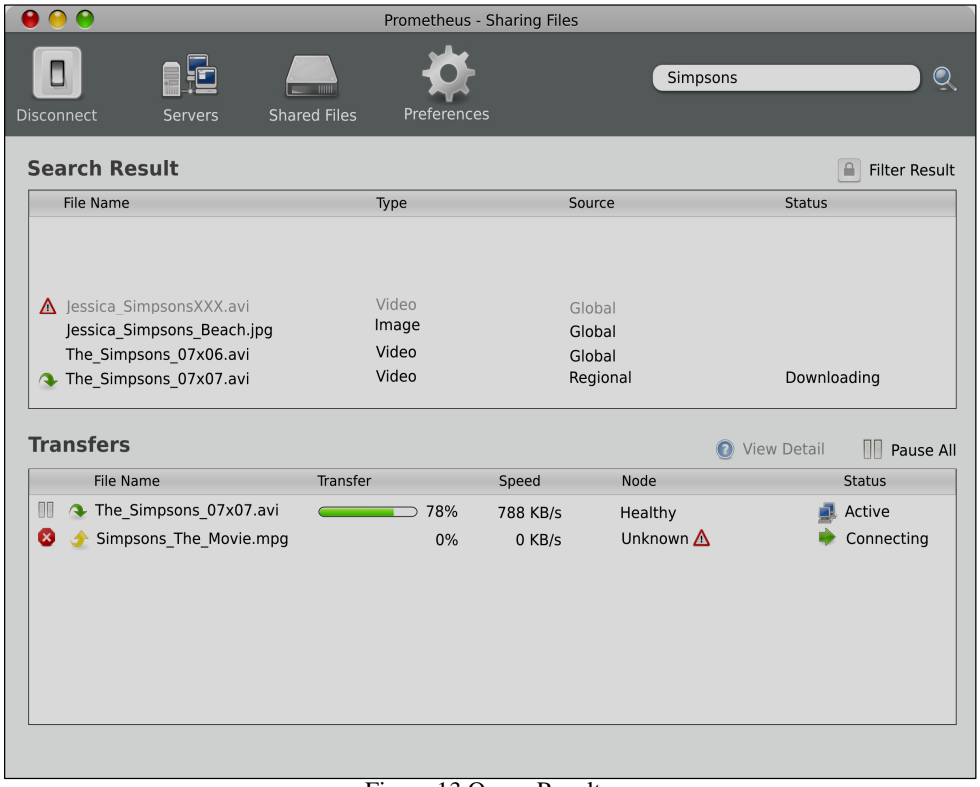


Figure 13 Query Result

As shown in the figure…..the search result of the random query is less than the result of the algorithm that which is used the more coordinated query propagation (such as Mengshu). This random system will influence the ability of the algorithm to maintaining its performance when the network scale is expanding. The other result of the evaluation process of the algorithm is presented in the next table.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	✓
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 20 Requirements Guo

2.5 Trust algorithm in Decentralized Community (Xiong and Liu, 2004)

This is an algorithm that based on reputation system. It means that the trustworthiness of a peer is defined by the evaluation of peer's reputation it receives from having transaction with other peers. Based on the algorithm system, the system will divide the peers in few clusters. Each cluster has a peer manager. The peer manager has a task of calculating a peer reputation when it is needed. The algorithm defines five important factors in trust calculation process.

Trust Value Calculation:
Thus, if peer A wants to calculate peer B trust value, the calculation process will depend on:
- the feedback from other peers, let say peer C, in connection with successful and unsuccessful transaction with peer B, T_{CB}.
- the reputation of peer recommender peer C, denoted as C_C, obtained from the peer manager (the network has several peer managers).
- the number of total transaction that peer A has executed with peer B, $S_{AB} + F_{AB}$.
- the transaction context factor, TF_{AB}, addressing the type of the transactions and the impact on the evaluation process (such as time of trust calculation), this is defined by the system.
- the community context factor, CF_{AB}, addressing the impact of community factor on the evaluation process (such as type of sharing file), this is defined by the system

Based on these four important factors and referring to the context scenario, if peer A wants to calculate peer B trust value, it can be defined as follows:

$$T_{AB} = \alpha * \frac{C_{AB}^{norm} * C_C * T_{CB} * TF_{AB}}{S_{AB} + F_{AB}} + (1-\alpha) * CF_{AB}$$

Variable	Description
$S_{AB} + F_{AB}$	Number of transactions between peer A and peer B during given period.
α	Proportion of trust value from local trust value Calculation and trust value based on community factor
C_{AB}^{norm}	the normalized local trust value of peer B by peer A.
C_C	Peer C reputation, obtained from the peer manager
T_{CB}	trust value of peer B by peer C.
TF_{AB}	adaptive transaction context factor of peer B during trust value calculation by peer A.
CF_A	adaptive community context factor of peer B during trust value calculation of peer A

Table 21 Legend

Context Scenario:

Based on the context scenario, if peer A submits a query to the network, it will send a random query to its neighbours. These neighbour peers will forward the query until the query TTL is up and the peers who have the requested file will start to respond.

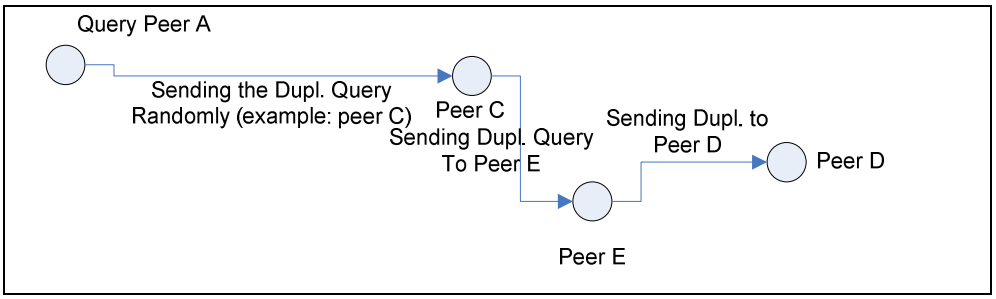


Figure 14 Query Propagation

If the responses are received by peer A, it will start to calculate the trust value of all responding peers. This algorithm uses P-Grid system for data routing and lookup. Thus, if peer C is one of the responding file, peer A will calculate the trust value of peer C by fetching a recommendation from other peer in the network. This is done by submitting a random recommendation query to peer A's neighbours. The neighbours will forward the query until the peers who store the recommendation of peer C is found (based on the P-Grid data storage system). The system is similar with that of Aberer and Despotovic Algorithm. If, let say, peer D stores the recommendation about peer C, peer A will contact the peer manager (the network has several peer managers) to ask about the reputation of peer D. If its reputation is good enough, peer A will receive the recommendation from peer D and calculate peer C's trust value. This process applies for every responding peer, and if peer is trustworthy enough, the file will be displayed in under the search result.

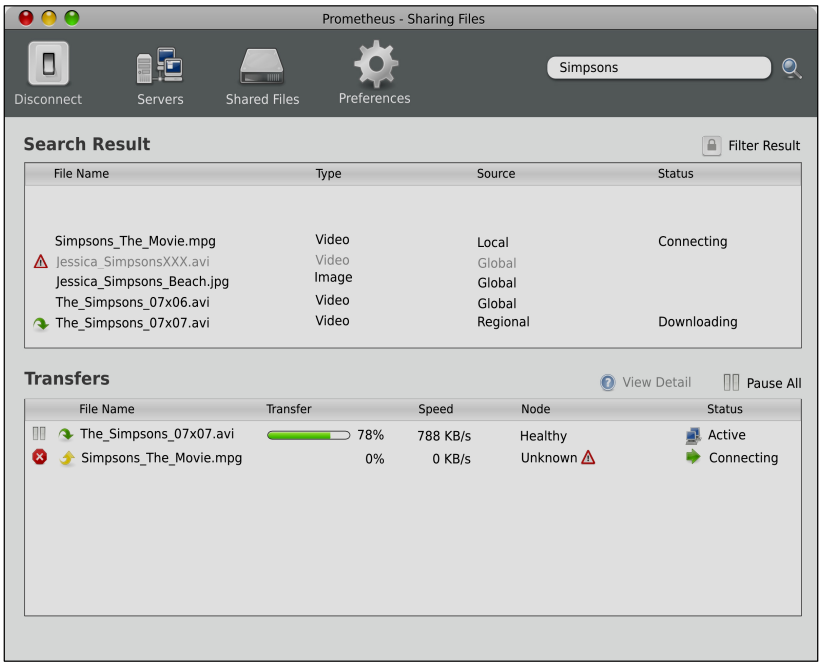


Figure 15 Query Result

The algorithm is providing the not only the trust calculation process but also the data storage and messaging system. But the algorithm lacks on the detail on the process of how the peer manager is chosen and its role detail. Moreover, the random query propagation makes it less flexible to cope with network scalability. The result of the algorithm evaluation is provided in the following table.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	X
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	✓
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 22 Requirements Xiong and Liu

2.6 Trust algorithm in Decentralized Environment (Wang and Varadharajan, 2004)

This algorithm is based on the peer evaluation of trust values based on recommendation from other peer. The evaluation process is crucial when a peer needs to get other peer reputation but they both never have performed any transactions before. In this analysis, a probability formula of trust value is calculated based on Gauss Distribution, where the probability of a peer's trust value distribution is compared to a threshold number. An addition to the evaluation process, there is also algorithm to up date the trust value, once the transaction is completed.

Trust Value Calculation:
Suppose, peer A needs to perform a transaction with peer B, with whom it never has prior experience. To compute the trust value of peer B, peer A needs to ask recommendation from other peers that have history about peer B. The outcome of the computation is a number between the interval 0 and 1. Consider the following case: suppose peer A has query to several recommender peers {peer C and peer D}. From these intermediate peers, peer A will get recommendation:

$\{T_{CB}, T_{DB}\}$

The trust value T_{ACDB} of peer B by peer A with the help of peer C and peer D can be calculated as:

$$T_{ACDB} = \tfrac{1}{2}(T_{CB} + T_{DB})$$

If there are k numbers of peers that give recommendation, the formula can be written as:

$T = \frac{1}{k}\sum_{i=1}^{k} T_{xy}$ where x is a random peer in the network that sends the query and y is a random peer that responds the query. This T variable is considered as mean trust value $\overline{T}$. And accordingly the sample variance S^2 will be:

$$S^2 = \frac{1}{k-1}\sum_{i=1}^{k}(T_{xy} - \overline{T})^2$$

If $\mu = \overline{T}$, $\sigma^2 = S^2$, $T \sim N(\mu, \sigma^2)$ for any random value T and v is a value between the interval [0,1] will result in distribution function as follows:

$$F(v) = P(T \le v) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\frac{v-\mu}{\sigma}} e^{-\frac{x^2}{2}} dx \text{ and accordingly } P(T > v) = \frac{1}{\sqrt{2\pi}\sigma}\int_{\frac{v-\mu}{\sigma}}^{-\infty} e^{-\frac{x^2}{2}} dx$$

Combining the formula above, if peer A wants to calculate peer B trust value, it can be done by calculating the probability that peer B's trust value is better than a given value $v \in [0,1]$:

$$P(T > v \mid T \in (0,1)) = \frac{P(v < T < 1)}{P(0 < T \le 1)} = \frac{\int_{\frac{v-\mu}{\sigma}}^{\frac{1-\mu}{\sigma}} e^{-\frac{x^2}{2}} dx}{\int_{-\frac{\mu}{\sigma}}^{\frac{1-\mu}{\sigma}} e^{-\frac{x^2}{2}} dx}$$, where v is a threshold value, set up by the system.

As previously mentioned, this algorithm also provides an up dating method for trust computation after the transaction has been performed.
Let:

$s_{t+1} \in [0,1]$ = satisfaction degree after transaction between peer A and peer B is completed

m = strictness factor (defines how much a peer defines the accuracy of the result of its trust value calculation , where $m \geq 1$)

θ_{t+1} = impact factor of recent change in trust value

The updating trust formula can be defined as:

$$T_{ACDB}^{t+1} = T_{ACDB} + \theta_{t+1}.(s_{t+1} - T_{ACDB}^{\frac{1}{m}})^m \text{ with } \theta_{t+1} = \frac{e^{1-T_{ACDB}} - 1}{e+1}$$

Based on the information that is provided above, it can be seen that this algorithm does take into account the distinction between private experience and recommendation.

Context Scenario:

There is also no information provided on how the query is spread through the network which makes it difficult to evaluate how the algorithm will cope with the network scale. Based on the context scenario, if peer A wants to submits a query to the network fro a certain file, it would difficult to predict the result of the query. There are also several other shortcomings from this algorithm that is provided in the following table:

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	X
The presence of context or classification.	X
The presence reflexivity factor	X
The presence of dynamic factor.	✓
The presence of scalability factor.	X
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	✓
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 23 Requirements Wang and Varadharajan

2.7 Trust algorithm in Pure Ad-Hoc Network (Pirzada and Mc.Donald, 2004)

This algorithm is based on ad hoc or temporarily network, where the peers within the network support one another in terms of passing data packet. It occurs often that the receiver of the data packet is beyond the range of the original sender. In this case, choosing the right intermediate peers are very crucial.

Pirzada and Mc.Donald (2004) proposed a trust algorithm in how choosing the right peers based on the DSR protocol (Johnson, Maltz and Hu, 2003). This is an on-demand routing protocol where the data packet does not depend on the intermediate peers in terms of choosing a route because each data packet carries a complete route as it leaves the original

sender. Based on this protocol, Pirzada and Mc.Donald (2004) divides trust in various categories, which represent different features in the DSR protocol.

Trust Value Calculation:
There are five categories that presented on the trust algorithm, namely: P_A (Passive Acknowledgement which is dealing with how the query is transmitting throughout the network), P_P (Packet Precision is dealing with the composition of the query), G_R (Gratuitous Route Replies which is dealing with the response query), B_L (Blacklists which is dealing with malicious peer) and S_G (Salvaging which is dealing with the case where the query propagation is not possible to be carried further to the next peer). These categories are assigned to different weights W that will be ultimately computed in the final trust calculation.

Thus, referring to the context scenario, if peer A wants to calculate peer B's trust value, the trust value calculation will be:

$$T_{AB} = \sum_{i=1} [W_{Ai} \times T_{Ai}]$$

Variable	Description
A	the peer who wants to calculate other peer's trust value (requestor peer)
B	the peer whose trust value is being assessed
T_{AB}	trust value of peer B by peer A
W	weight assigned to trust category i
i	trust value categorie

Table 23 Legend

Then the final computation of trust for peer B by peer A can be written as:

$$T_{AB} = W_A(P_A) \times T_A(P_A) + W_A(P_P) \times T_A(P_P) + W_A(G_R) \times T_A(G_R) + W_A(B_L) \times T_A(B_L) + W_A(S_G) \times T_A(S_G)$$

Context Scenario:
From the description above, it shows that this algorithm takes into account context or classification factor in the calculation process. However, even though the query propagation route, as already mentioned, does not depend on the peers' neighbours, its propagation still needs other peers as mediators to reach the destination peer. Thus, the scale of the network still gives significant influence to the propagation process. Another shortcoming from this algorithm is that because the route is known to every intermediary peer, the anonymity of the peer can not be guaranteed.

Moreover, this algorithm is still in the concept phase where there is no further validation process or simulation result provided, which makes it very difficult to evaluate its performance.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	**X**
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 25 Requirements Pirzada and Mc.Donalds

2.8 Trust algorithm with Community Based Model (Jin, 2005)

This algorithm is based on community trust where trust value is computed based on private experience and community recommendation. The community is created through signing a digital certificate such as X.509. This certificate is issued by Certificate Authority, CA. This is an example case where trust model is provided in hierarchical model. A peer can join the community by fetching a digital signature from a member of the community.

Trust Value Calculation:

Thus if a peer, let say, peer A wants to calculate peer B trust value it will calculate the first its local trust value which corresponds the private experience of peer A with peer B. To calculate the local trust value, peer A has to define the number of successful transaction S_{AB} and unsuccessful transaction F_{AB}. Next, the local trust value of peer B by peer A, T_{AB}, can be written as:

$$T_{AB} = \frac{S_{AB} - F_{AB}}{S_{AB} + F_{AB}} \quad \text{.....(I)}$$

If the local trust value is obtained, peer A can calculate the value of community recommendation by defining first the value of community trust value T_C which is calculated by:

$$T_C = \frac{\sum_{k=1}^{C} T_k}{C} \quad \text{.....(II)}$$

$\sum_{k=1}^{C} T_k$ denotes the sum of trust value from every peer in the network regarding peer B while C denotes the number of peers in the community. Based on calculated value of T_C, the final trust value can be calculated as follows:

$$T_{ACB} = \alpha T_{AB} + \beta T_C \qquad \text{where } \alpha + \beta = 1 \quad \text{.....(IV)}$$

α and β denote the weight between local trust value and community recommendation and defined by the system.
This algorithm also provides an updating method to manage the community trust value. The computation can be presented as follows:

$$T_{ACB} = \begin{cases} T_{ACB,old} & : t \le Inter \\ T_{ACB,old} e^{(Inter-t)} + (1 - e^{(Inter-t)}) T_{ACB,new} & : t > Inter \end{cases} \quad \text{.....(V)}$$

Inter denotes a specified time interval which is defined by the system and t is the time where the trust value is last calculated.

Variable	Description
S_{AB}	Number of successful transaction between peer A and peer B
F_{AB}	Number of unsuccessful transaction between peer A and peer B
T_{AB}	Trust value of peer B by peer A
T_C	Trust value of peer B by community
$\sum_{k=1}^{TotalPeerNum} T_k$	The sum of trust value by each peer in the community
C	Number of peers in the community
α and β	The weight between local trust value and recommendation trust value
T_{ACB}	Final trust value of peer B by peer A

Table 26 Legend

Context Scenario:
This algorithm does not provide the information on the query propagation if a peer submits it to the network. Thus, based on the context scenario, the process and of the query of peer A starting from when it is submitted to the network until it is responded will be unclear. Therefore, it is difficult to evaluate how the search result will look like for the submitted file query. However, based on the provided information, the algorithm has been evaluated, and the result is provided in the following table.

Requirements:	Availability
The presence of non-symmetric factor.	✓
The presence of conditional transitivity factor	✓
The presence of context or classification.	✓
The presence reflexivity factor	**X**
The presence of dynamic factor.	✓
The presence of scalability factor.	**X**
Implementing a user intervention system	X
Take into account the role of pre-trusted peer	X
Maintaining peer anonymity	**X**
Performance on: - Computation process - Data storage - Message complexity.	**X**

Table 27 Requirements Jin