IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-1 E-ISSN: 2347-2693

Secure Trust Model for Peer to Peer Network

Roshan Bhandure[1], Siddharth Gujrathi[2*], Jangila Basumatary[3], Pooja Padekar[4] , Rakesh Shirsath[5]

[1]Computer Engineering, Pune University,India, bhandureroshan@gmail.com
[2*] Computer Engineering, Pune University,India, sidh.gujrathi@gmail.com
[3] Computer Engineering, Pune University,India, jangbasu@gmail.com
[4]Computer Engineering, Pune University,India, poojapadekarp@gmail.com
[5]Computer Engineering, Pune University,India, rakesh.shirsath@gmail.com

www.ijcaonline.org

Received: 07January2015 Revised: 15 January2015 Accepted: 26 January2015 Published: 31 January2015

***Abstract*—** The peer to peer network has open nature which leads to exposure of various malicious activities. To eradicate these malicious activities, peers can build trust relationships among them. This paper presents distributed algorithm that enable a peer to reason about trustworthiness of other peers based on past interaction and recommendation. Using this algorithm peers will have their own adjacency trust network. Two modules of trust, service and recommendation are proposed to measure trustworthiness for providing services and giving recommendation. Interactions and recommendations are computed based on importance, recentness and peer satisfaction parameter. Also, while computing recommendation, recommender's trustworthiness and confidence will be considered.

***Keywords*—** Peer to Peer Systems; Trust Models; Trust Relationships; Reputation; Networking; Network Security

I. INTRODUCTION

PEER-TO-PEER (P2P) network is created when two or more PCs are connected and share resources without going through a separate server computer. While continues evolving technologies around networking, P2P also has been improved than ever and thus most of infrastructures based on Client Server architecture are now moving towards P2P architecture for their applications. P2P systems are rely on collaboration of peers to accomplish any task. While evolving with technologies the P2P architecture itself grown over Internet too, but having open nature of peer to peer system exposes them to malicious activities. So, to prevent P2P from those malicious activities is creating trust relationship among peers using trust models, which provides a more secure environment by reducing risk in future P2P interaction. However, classifying peers as either trustworthy or untrustworthy is not sufficient in many cases. For measure trust among peers we can use interaction and feedback of peers, in such case also interaction provides certain information about peer but feedback might contain deceptive information.

We propose a Self Organizing Trust Model (STM) a distributed algorithm that aims to mitigate malicious activities and establish trust relationship among peers in P2P systems. In presence, a central server is a preferred way to store and manage trust information. Also collect trust information from all the peers in system, in proposed system no trusted peer / central server is use to leverage trust establishment. Peers do not try to collect trust information from all peers. Rather, each peer develops its own local view of trust about other peers interacted in past. In this way, good peers form their own dynamic trust group, as generally peers tend to interact with small set of peers frequently [3].

In STM, P2P system initialize with null and peers are assumed to be strangers to each other. If any peer interact with other by having service or providing service then they become known to each other. Using a service of peer is an *interaction value*, which is evaluated based on importance of interaction and satisfaction of requester. *Recommendation value* is what requester provides feedback about peer, which evaluated based on recommender's trustworthiness. This contains recommender's own experience about service provider , information collected from it's own trust network/ group and level of confidence in recommendation. If this confidence is low then provided recommendation has low value in evaluation.

There might be situation where peer may be good service provider but bad recommender or vice versa. Thus, STM consider both services provided and giving recommendation as different task in two contexts of trust *service and recommendation context* .

STM defines three trust metrics to build trust relationship between peers, *Reputation* metric which is calculated based on recommendations. *Service trust* metric and *Recommendation trust* metric which are used to measure trustworthiness in the service and recommendation context, respectively. Service trust metric is used when selecting service providers and the Recommendation trust metric is used requesting recommendation.

To understand and show impact of STM to mitigate attacks we are going to implement P2P file sharing

Corresponding Author: *Siddharth Gujrathi, sidh.gujrathi@gmail.com*

application. Parameters related to peer capabilities (bandwidth, number of shared files), peer behavior (online/ offline periods, waiting time for sessions), and resource distribution (file sizes, popularity of files) are approximated to several empirical results [4], [5], [6]. This enabled us to make more realistic observations on evolution of trust relationships.

Paper Statement: This Paper provides the detail analysis and study of distributed algorithm called Self Organizing Trust Model For Peer To peer System.

Purpose and Motivation: The peer-to-peer architecture has come to prominence in recent years on the back of the many file-sharing systems that use it as their model for resource location and sharing. Although file-distribution isn't the only application for peer-to-peer technology, it is the primary concern of this report, and the purpose for which we research the model associated with it STM. Within this we describe our research into the area of peer-to-peer networking, and it includes topics such as a general overview of the field, a review of some of the existing systems currently in operation and a description of some of the unique problems that must be addressed when designing trust building in peer-to-peer systems. The second half describes the working of STM, the features it implements and the approaches we took to address some of the problems unique to the peer-to-peer trust models.

II. Related Works

2.1 Aberer and Despotovic's Trust Model

Over the last years, mainly due to advancements in technologies there are more possible ways to do business electronic or over the Internet, by which people are started recognize importance of trust management in e-businesses. Visitors at e-commerce sites like 'amazon.com', 'flipkart.com' usually look for reviews provided by other customers before buy any product from there site. So, in both the systems they use completely centralize mechanism for storing and exploring reputation data.

Likewise Peer-To-Peer systems which are particularly driving major part in era of distributed computing. But, managing trust in P2P environment is quite difficult where one frequently encounters with unknown peer (agent). Existing methods for trust management which are based on reputation. They do not scale as they either rely on a central database or require maintaining global knowledge at each peer to provide data on earlier interactions.

So, the approach to trust management that Aberer and Despotovic proposed is based on analysis earlier transaction of agents and deriving from that the reputation of a peer. The reputation probably can be cheat easily by other peer. Thus, the method can be interpreted as simple method of data mining using statistical data analysis of former transaction. The analysis is performed by decentralize method P-Grid [1].

This trust model has architecture for trust management which relies on all system layers like network, storage and trust management. These are all different system levels of P2P computing as shown in fig 2.1. In such architecture a mechanism implemented at higher level in P2P manner has always to take into account the properties, in particular the quality of service, of the mechanism of the underlying layers.

III. PROBLEM STATEMENT

Peer-to-Peer network is used by many real time applications like Bittorrent (File Sharing) application with private or global network, but this network can lead to exploits by many malicious attacks which leads to integrity and peer failure problem in P2P network. So, the proposed system presents distributed algorithms that enable a peer to reason about trustworthiness of other peers based on past interactions and recommendations. Peers create their own trust network in their proximity by using local information available.

IV. HYPOTHESIS

Research hypotheses are the specific testable predictions made about the independent and dependent variables in the study. Usually the literature review has given background material that justifies the particular hypotheses that are to be tested. Hypotheses are couched in terms of the particular independent and dependent variables that are going to be used in the study.

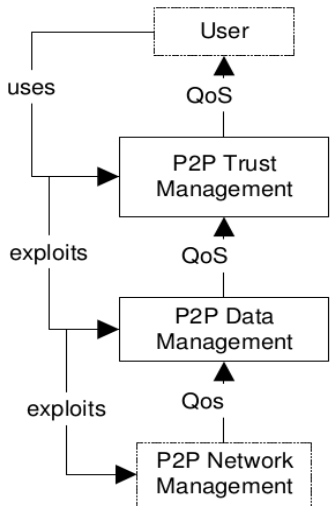


Fig. 2.1 Different system levels of P2P computing

Managing Trust in a Decentralized System

Formally following problem of reputation-based trust management will be taken into account. Let P denotes the set of all agents. The behavioral data B are observations *t(q,p)* an agent $q \in P$ makes when he interacts with an agent $p \in P$. *Based on these* observations one can assess the behavior of *p* based on the set.

$$B(p) = \{ t(p, q) \text{ or } t(q, p) \mid q \in P \} \subseteq B$$

That means when data is available globally, the reputation of a peer *p* can be derived from B(p). Thus formulate the problem of managing trust in decentralized information system we can partition it now, more precisely, into three sub problems that need to be studied :

1. Global trust model: What us the model that describes whether an peer is trustworthy? This model could be based on simple statistical methods. On experiences gain in economic and sociological sciences or on theocratic foundation.
2. Local algorithm to determine trust: What is the computational procedure that peer can apply in order to determine trust under the limitations discussed above, which unreliability of the agents providing trust data both with respect to their trustworthiness themselves as well as their reach ability over the network.

Decentralized data Management

In order to store data in a P2P network in scalable way this method uses a method that we have proposed earlier, namely P-Grid [1]. As shown in Fig 2.2 a simple example of P-Grid

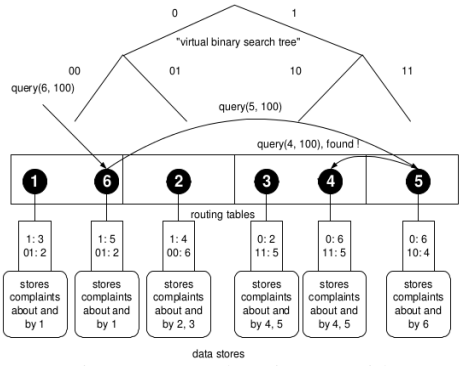


Fig 2.2 Example using P-Grid

6 peers support three together a virtual binary search tree of depth 2. Each peer is associated with one path of the search tree. It stores data items for which the associated path is a prefix of the data key. For the trust management application this are the complaints indexed by the agent number. Each agent can serve any search path associated with the agent processing the request, or the agent can use its routing table for complementary part of the search tree. In fig the processing of one sample *query* (6,100) using search structure. As agent 6 is not associated with keys starting with 0 its looks up in its routing tale agent 5, to whom it can foreword the query. Agent 5 in turn cannot process a query starting with 10 and before looks up in its routing table peer 4, who can finally answer the query, as it stores all data with keys, that start with 10.

At the leaf level the agents store complaints about the agents, whose identifier corresponds to search key, using the encoding 1= 001, 2= 010, . . . 6= 110. once can see that multiple agents can be responsible for the complaints on specific agent. Thus, the same data can be stored at multiple agents and we have replicas of this data. Replicas make the access structure robust against failure in network.

2.1.2 Bayesian network-based trust model

Trust and reputation mechanism

In our model a peer builds two kinds of trust in another peer, say peer A and peer B respectively. The first one is the trust that peer A has in peer B's *capability in providing services*. The other is the trust that peer A has in peer B's Reliability *in providing recommendations* about other peers. Here the reliability includes two aspects:

- *Truthfulness* – whether peer B is truthful in telling its Information
- *Similarity* – whether peer B is similar to peer A in preferences and ways of judging issues.

Reliability = Truthfulness? Similarity, i.e. a peer B's reliability as a referee depends on both being truthful and Similar in its preferences to the peer requesting the recommendation. Since peers are heterogeneous, they may have different preferences and judge issues by different criteria. For example, some peers may consider a movie Provider good because it provides movies with high quality, while others may consider the movie provider bad because the speed of download from it is very slow. If two peers A and B are similar in their evaluation criteria, peer A can trust peer B's recommendations, if it knows that peer B is truthful. However, if the peers have different evaluation criteria, peer A cannot trust peer B's recommendations even when peer B tells the truth.

A search request in file sharing peer-to-peer applications usually results in a long list of providers for an identical file. If a peer happens to select a provider of files with bad quality or slow download speed, the peer will waste time and effort, which may lead to user frustration

and abandoning the system. In order to solve the problem, we use the mechanism of trust and reputation as shown in Fig 2.3.

Once a peer receives a list of file providers for a given search, it can arrange the list according to its trust in these file providers. Then the peer chooses one of the file providers on top of the list. If the file provider is trustworthy according to the peer's previous experiences, the peer will interact with the file provider (download files). If the file provider is not trustworthy, the peer will select another file provider to interact with. If the peer is not sure about the trustworthiness of the file provider, for example, the peer has no interactions or only a few interactions with the file provider, it can ask other peers to make recommendations for it.

How the peer uses the reputation and its own trust to make a decision with which file provider to interact is an open question. Some peers may prefer to trust their own experience and rely on their trust even if they had very few interactions with the service provider. Others may be more cautious and rely on the reputation of the service provider. After each interaction, the peer updates its trust in the file provider according to its evaluation of the interaction. If the interaction is satisfying, it will increase its trust in the file provider; if the interaction is not satisfying, it will decrease its trust in the file provider. If the decision of interaction is based on other peers' recommendations, the peer will also update its trust in each of the peers that give recommendations (we call these peers "referees"). If the referee's recommendation is consistent with the peer's evaluation of the interaction, the peer will increase its trust in the referee; otherwise, it will decrease its trust.

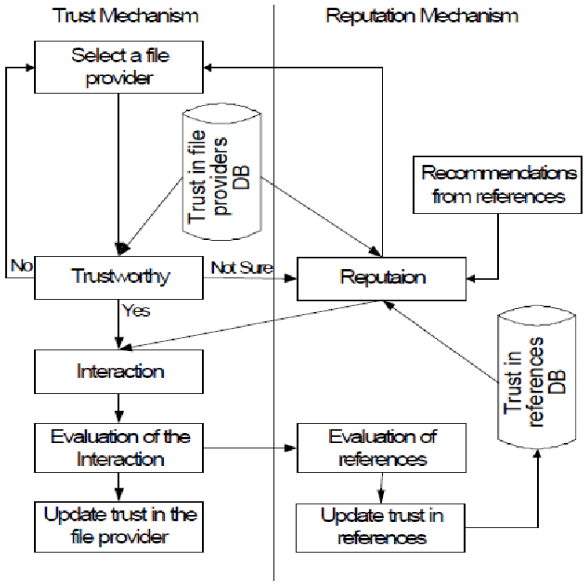


Fig 2.3 Functionality of trust and reputation

A Bayesian network model

A Bayesian network provides a flexible method. It is a relationship network that uses statistic methods to represent probability relationships between different elements [6]. We use a nave Bayesian network to represent the trust of a peer in a file provider. Every peer develops a naive Bayesian network for each file provider that it has interacted with. Each Bayesian network (see Figure 2) has a root node *T* that represents the peer's trust in the file provider's capability in providing files. It is the percentage of interactions that are satisfying. The leaf nodes under the root node represent the file provider's capability in different aspects. The node, denoted by *FT*, represents the set of file types. Suppose it includes five values, *"Music"*, *"Movie"*, *"Document"*, *"Image"* and *"Software"*. The node *"DS"* denotes the set of downloads speeds. It has three values, *"Fast"*, *"Medium"* and *"Slow"*, each of which covers a range of download speeds. The node *"FQ"* denotes the set of file qualities. It also has three values, *"High"*, *"Medium"* and *"Low"*.

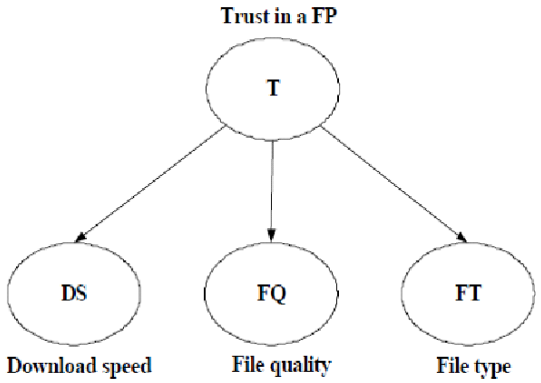


Fig 2.4 Bayesian Network model

Handling recommendations

When a peer is not sure about the trustworthiness of a file provider, it can ask other peers for recommendations. The recommendation requests can vary according to the peer's needs. For example, if the peer is going to download a movie, it may care about the movie's quality. Another peer may care about the download speed. So the request can be "Does the file provider have movies with good quality?" If the peer cares both about the quality and the download speed, the request will be something like "Does the file provider offer files with good quality and fast download speed?" When other peers receive these requests, they will check their trust representations, i.e. their Bayesian networks, to see if they can answer such questions. If a peer has downloaded movies form the file provider before, it will

answer the first question with its trust in the file provider under the condition that the file provider providers files with good quality and the second question with its trust under the condition that the file provider provides files with good quality and fast download speed according to its Bayesian network.

If the peer interacts with the file provider, it will not only update its trust in the file provider, i.e. its corresponding Bayesian network, but also its trust in the referee-peers that provide recommendations by the following reinforcement learning formula:

$$tr_{ij}^{n} = \alpha * tr_{ij}^{o} + (1 - \alpha) * e_{a}$$

trij denotes the new trust value that the *ith* peer has in the *jth* referee after the update; *trij* denotes the old trust value. *a* is the learning rate – a real number in the interval [0,1]. *ea* is the new evidence value, which can be -1 or 1. If the value of recommendation is greater than *q* and the interaction with the file provider afterwards is successful, *ea* is equal to 1. If there is a mismatch between the recommendation and the actual experience with the file provider, the evidence is negative, so *ea* is -1. Another way to find if a peer is reliable in making recommendations is the comparison between two peers' Bayesian networks relevant to an identical file provider. When peers are idle, they can "gossip" with each other periodically, exchange and compare their Bayesian networks. This can help them find other peers who share similar preferences more accurately and faster. After each comparison, the peers will update their trusts in each other according the formula:

$$tr_{ij}^{n} = \beta * tr_{ij}^{o} + (1 - \beta) * e_{\beta}$$

The result of the comparison *eb* is a number in the interval [-1, 1]. *b* is the learning rate – a real number in the interval [0,1], which follows the constraint $b > a$. This is because the Bayesian network collectively reflects a peer's preferences and viewpoints based on all its past interactions with a specific file provider. Comparing the two peers' Bayesian networks is tantamount to comparing all the past interactions of the two peers. The evidence *ea* in formula is only based on one interaction. The evidence *eb* should affect the peer's trust in another peer more than *ea* .

2.3 Facilitating trust in Internet interactions

Working of model in Internet

In eBay, the largest person-to-person on line auction site, with more than 4 million auctions open at a time. eBay offers no warranty for its auctions; it only serves as a listing service while the buyers and the sellers assume all the risks associated with transactions. There are fraudulent transactions to be sure. Nonetheless, the overall rate of successful transactions remains astonishingly high for a market as "ripe with the possibility of large-scale fraud and deceit" as is eBay.

EBay attributes its high rate of successful transactions to its reputation system, the Feedback Forum. After a transaction is completed, the buyer and seller have the opportunity to rate each other (1, 0, or -1) and leave comments ("*Good transaction. Nice person to do business with! Would highly recommend*."). Each participant has his running total of feedback points attached visibly to his screen name, possibly a pseudonym. Yahoo! Auction, Amazon and other auction sites feature reputation systems like eBay's, with variations such as a rating scale from 1-5, or using several measures (friendliness, prompt response, quality product, etc), or averaging rather than totaling feedback scores.

Approach for the model

Reputation systems seek to restore the shadow of the future to each transaction by creating an expectation that other people will look back upon it. The connections of such people to each other may be significantly less than is the case with transactions on a town's Main Street, but their numbers are vast in comparison. At eBay, for example, a stream of buyers interacts with the
same seller. They may never buy an item from the seller again, but if they share their opinions about this seller on the Feedback Forum, a meaningful history of the seller will be constructed. Future buyers, having no personal history, may still base their buying decisions on a sufficiently extensive public history. If buyers do behave this way, the seller's reputation will affect her future sales. Hence, she will seek to accumulate as many positive points and comments as possible, and avoid negative feedback. Through the mediation of a reputation system, assuming buyers provide and rely upon feedback, isolated interactions take on attributes of a long-term relationship. In terms of building trust, a vast boost in the quantity of information compensates for a significant reduction in its quality.

Drawbacks

- Eliciting feedback encounters three related problems. The first is that people may not bother to provide feedback at all. For example, when a trade is completed successfully at eBay, there is little incentive to spend another few minutes filling out a form. That many people do so is a testament to their community-mindedness, or perhaps their gratitude or desire to exact revenge.
- It is especially difficult to elicit negative feedback. For example, at eBay it is common practice to negotiate

first before reSTMing to negative feedback. Therefore, only really bad performances are reported.

- One party could blackmail another—that is, threaten to post negative feedback unrelated to actual performance. At the other extreme, in order to accumulate positive feedback a group of people might collaborate and rate each other positively, artificially inflating their reputations.
- Finally, there is also a potential difficulty in aggregating and displaying feedback so that it is truly useful in influencing future decisions about who to trust. eBay displays the net feedback (positives minus negatives). Other sites such as Amazon display an average. We believe that these simple numerical ratings fail to convey important subtleties of online interactions.

V. METHODOLOGY

Service Trust Metric (st_{ij})

Service history is used to determine competence belief and integrity belief. Competence belief is related to past interactions[8][10][11]. Needs of past interactions are satisfied or not is determine by competence belief. Whereas predictability about future interaction is given by Integrity belief. Competence take into consideration by weight and recentness. Consistency is in competence is important term. Hence Competence belief can be calculated as,

$$cb_{ij} = \frac{1}{\beta_{cb}} \sum_{k=1}^{shij} \left(s_{ij}^{k} . w_{ij}^{k} . f_{ij}^{k} \right)$$

Here normalization coefficient is,

$\boldsymbol{\beta}_{ab}$ = value of cb_{ij} varies between 0 and 1.Integrity belief ib_{ij} is deviation from average behavior can be calculated as,

$$\bar{\imath}b_{ij} = \sqrt{\frac{1}{sh_{ij}} \sum_{k=1}^{sh_{ij}} \left(s_{ij}^{k} . w_{ij}^{\mu} . f_{ij}^{\mu} - cb_{ij} \right)^{2}}$$

Small value of ib_{ij} indicates more predictable behavior in future interaction. After competence belief and integrity belief level of satisfaction is considered for the expectations by peer. If Normal distribution is followed by satisfaction parameters then mean(μ) and standard deviation (σ) are considered as satisfaction parameters for cb_{ij} and ib_{ij} respectively. Weight and fading effect parameters are not considered as they are independent of satisfaction parameters.

$$f_{ij}^{\mu} = \frac{1}{sh_{ij}} \sum_{k=1}^{sn_{ij}} f_{ij}^{k} = \frac{sh_{ij} + 1}{2sh_{ij}} \approx \frac{1}{2}$$

If st_{ij}=cb_{ij} then satisfaction value will be less than cb_{ij} of half of the integrity belief.

$$st_{ij} = cb_{ij} - ib_{ij}/2$$

For total value of Service trust metric reputation value is needed. So here above equation is not complete. Trust relationship is very important in stages while building network. So each network is completely rely on reputation of each peer available in network. Peer with high reputation value is always recommended first. St_{ij} can be calculated as,

$$st_{ij} = \frac{sh_{ij}}{sh_{max}} \left(cb_{ij} - ib_{ij}/2 \right) + \left(1 - \frac{sh_{ij}}{sh_{max}} \right) r_{ij}$$

Reputation Metric (r_{ij})

In network reputation built as per recommendation given by other peers. Whenever any peer needs a service in network first step is to take recommendations from other peers. Let us consider pi wants service then all other peers in network will give the recommendation to pi so that most recommended peer will be selected for taking service. Among recommended peers most reputed peer is considered. So after recommendations , Reputation of metric will check. The metric which is most reputed will be given a chance to provide services to peer pi.

Recommendation can be given by those peers who has interacted previously with that peer. Service history size is one of the parameter to be considered in reputation. History size as well as good recommendations will built confidence about peer.

After collecting all recommendations peer p_i will calculate estimated competence belief and estimated integrity belief.

$$ecb_{ij} = \frac{1}{\beta_{ecb}} \sum_{p_k \varepsilon T_i} \left(rt_{ik} \right.$$

$$e\bar{\imath}b_{ij} = \frac{1}{\beta_{ecb}} \sum_{p_k \varepsilon T_i} \left(rt_{i} \right.$$

So reputation r_{ij} can be calculated as,

$$r_{ij} = \frac{|\mu_{sh}|}{sh_{max}} \left(ecb_{ij} - e\bar{\imath}b_{ij}/2 \right) + \left(1 - \frac{|\mu_{sh}|}{sh_{max}} \right.$$

Recommendation Trust Metric(rt_{ij})

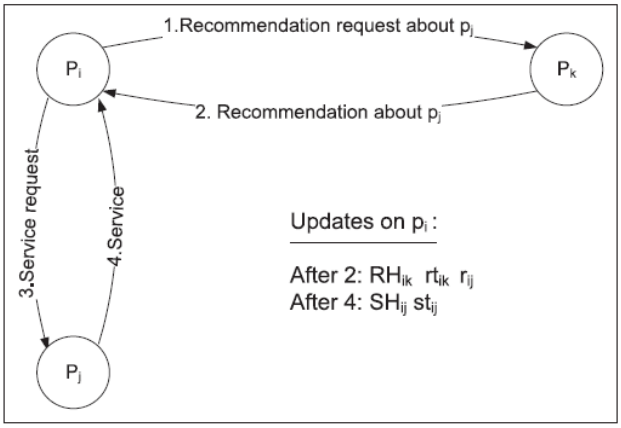
In this metric recommendations are calculated as per accuracy of recommendations. Satisfaction ,weight and

fading effect these three parameters are considered when calculating recommendations about peer.

$$(rs_{ik}^{z}, rw_{ik}^{z})$$

A tuple above denotes information about recommendation. RH_{ik} is recommendation history. Satisfaction parameter with respect to r_{kj},i_{kj},cb_{kj} and er_{ij}, ecb_{ij},eib_{ij} are compared. And calculated as,

$$rs_{ik}^{z} = \left(\left(1 - \frac{|r_{kj} - er_{ij}|}{er_{ij}}\right) + \left(1 - \frac{|cb_{kj} - ecb_{ij}|}{ecb_{ij}}\right) + \left(1 - \frac{|ib_{kj} - eib_{ij}|}{eib_{ij}}\right)\right)$$



Here p_i wants services from p_j. First pi wants to know reputation of p_j, if p_j is reputed then pi will take services from p_j. To calculate reputation of p_j, p_i collects recommendations from all other peers in network. After request of recommendations reputation of p_j will be calculated. Result will be stored in recommendation history and reputation metric will be updated. Finally service will be provided to pi.

VI. EXPERIMENT AND ANALYSIS

There are some questions which needs to be analyzed such as, how to handle attack by using STM, how recommendations help in identifying the malicious peers, how many attacks can be mitigated.

In file sharing application two most important actions are uploading and downloading of file. when peer shares a file is known as uploader while a peers downloading a files known as downloaders important term related to these two operations is session. Ongoing download or upload operation is nothing but a session. During attacking conditions what exactly the trust calculation affects can be determine by using following three conditions:

No trust: In this case trust information of uploader is not considered. Uploaders are selected according to it's bandwidth. So it will tell us how much calculation of trust is necessary.

No reputation query: In this case using trust information uploader is selected. Recommendation of uploader will not be considered. So reputation value of peer is zero and this case will indicate how recommendations helpful.

Flood reputation query: Reputation query is flooded to whole network as well as STM equations are used. This will tell us that dealing with more recommendation helps to mitigate attacks and determine malicious peers. Satisfaction and weight are two important parameters for analysis of peers using STM.

Weight can be calculated by two variables file size and popularity. Satisfaction can be calculated using bandwidth and online period.

Attacker Model

Attacker model introduces two types of attacks service based attacks and recommendation based attacks. service based attack happens when virus infected file is uploaded and recommendation based attacks are nothing but giving misleading recommendations.[10]. Here recommendation based attacks are not easy to recognize as compare to service based attack. Service based attacks can be detected after downloading file but it is always be hard to recognize about misleading recommendations. Malicious peers behaves differently as naïve, discriminatory, hypocritical and oscillatory.

Naïve: It includes intentionally giving low recommendations and uploading virus infected file.
Discriminatory: It is acting as good peer but always uploads virus infected files for particular group of victims[2].
Hypocritical: In this type peer always give low recommendations by acting as good peer. Also it uploads inauthentic files.

Oscillatory: Being good peer for long time it becomes reputed. Then after it behaves as naïve attacker for malicious period of time and again it behaves as good peer.
In a network there are some good peers and some malicious peer. But malicious peers do not know each other so that they attack independently are known as individual attackers. Collaborators are when malicious peers knows each other. Collaborators and individual attacker both follows above four types of behavior.

VII. CONCLUSION

Peer to peer network can easily develop trust in their proximity by using STM. Malicious activities can be handle

by detecting malicious peers in a network and developing trustworthy environment. For developing trust relationship among good peers three metrics are used service trust metric, recommendation trust metric and reputation metric. It defines capability of peers for providing services and giving recommendation. Also it helps to choose most reputed peer. Three parameters are considered satisfaction , weight and fading effect. Recommendations are given by own experience of peer stored in its local storage. For trustworthiness all above parameters are considered.
There is an experimental study about individual and collaborator attackers. These attackers affects the trustworthiness. So task of STM for peer to peer network is to detect malicious activities and malicious peers. By doing this inauthentic file sharing of virus infected file sharing can be avoided.

Peer to peer does not solve all security problems. If 50 percent of peers available in network then handling malicious activities is difficult task. But it will definitely enhance security and to meet the level of satisfaction. Some of the applications of peer to peer network using STM are e.g. CPU sharing ,storage network, P2P gaming, File sharing.

VIII. SCOPE FOR FUTURE RESEARCH

For further research there are mainly four issues to be discussed. First one is load balancing. Choosing the most reputed and recommender peer as service provider leads to increase load on particular peer. That peer considers all its resources while providing services. One point occurs at which it reaches to maximum number of uploads and start rejecting all upcoming requests. That upcoming request are given to another peer to provide service. This simple load balancing needs to be manage, so further research is needed for load balancing.
Another scope for research includes misleading recommendations. In many malicious activities and attacks (recommendation based attacks) there are intentionally misleading recommendations available in a network. Malicious peers always leads to provide low recommendations intentionally, so peer cannot be depend on such recommendations. Such attacks are difficult to recognize. Determining misleading recommendations needs further research.
Maintaining trust in network is another issue. Each peer in a network is attached to that network. Sometimes it might happen , the peer need to change the network. At such a moment point of attachment is changes. As peer is part of trust network in STM , changing the point of attachment leads to lose part of trust network. All over the network maintenance of trust is scope for further research.
Stranger is one of the scope of further research. If suppose stranger is available in a network and there is a time at which stranger gives recommendations then at what extend it should be considered needs more research. Stranger may or may not interacted with each peer available in network, it might be malicious , it can share inauthentic files. So recognizing nature of stranger with respect to all other parameters is important issue.

ACKNOWLEDGMENT

We are profoundly grateful to Prof.R.S.Shirsath for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express our deepest appreciation towards Prof.D.D.Shinde, Principal, SIEM, Nashik, Prof.U.D.Pawar, HOD Computer Engineering Department. We are particularly grateful to Prof.Kamini Nalavade who allow us to work on ["STM: A Self-Organizing Trust Model for Peer-to-Peer systems"]. At last we must express our sincere heartfelt gratitude to all staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

REFERENCES

[1] F. Cornelli, E. Damiani, S .D. C. di Vimercati, S. Paraboschi, and P. Samarati," Choosing Reputable Servents in P2P Network," Proc. 11th World Wide Web Conf. (WWW), 2002
[2] A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks, " Proc. IEEE/ ACM Forth Int'I Symp. Cluster Computing and the Grid (CCGRID) , 2004.
[3] J. Kleinberg, " The Small-World Phenomenon: An Algorithmic Perspective," Proc. 32nd ACM Symp. Theory of Computing, 2000
[4] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems, "Proc. Multimedia Computing and Networking, 2002.
[5] M. Ripeanu, I. Foster, and A. Lamnitchi, "Mapping The Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design, "IEEE Internet Computing, Vol.6,no. 1,pp. 50-57, Jan. 2002
[6] S. Saroiu, K. Gnummadi, R. Dunn, S.D. Gribble, and H.M. Levy, "An Analysis of Internet Content Delivery System, " Proc. Fifth USENIX Symp. Operating System Design and Implementation (OSDI), 2002.
[7] Y. Zhong, "Formation of Dynamic Trust and Uncertain Evidence for User Authorization," PhD thesis, Dept. of Computer Science, Purdue Univ., 2004.

[8] D.H. McKnight, "Conceptualizing Trust: A Typology and E-Commerce Customer relationships Model," Proc. 34th Ann. Hawaii Int'I conf. System Science (HICSS) , 2001.

[9] C. Dellarocas, "Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior," Proc. Second ACM Conf. Electronic Commerce (EC), 2000.

[10] S. Xiao and I. Benbasat," The Formation of Trust and Distrust in Recommendation agents in Repeated Interactions: A process-tracing Analysis," Proc. Fifth ACM Conf. Electronic Commerce (EC), 2003.

