Secured and Trusted Routing based on Self Organizing Trust Model for Peer to Peer Network

K.Sundaramoorthy
Research Scholar
St.Peters University

Thirukkumaran
Infotechkumaran16@gmail.com
Agni College of Technology

Kumaran
Infotechkumaran16@gmail.com
Agni College of Technology

Dr.S.Srinivasa Rao Madhane
Adhiparasakthi College of Engineering, Kalavai

Abstract—In this paper, we present building trust relationships among peers can mitigate attacks of malicious peers. This paper 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 and do not try to learn global trust information. Two contexts of trust, service, and recommendation contexts are defined to measure trustworthiness in providing services and giving recommendations. Bidirectional Interactions and recommendations are evaluated based on importance, recentness, and peer satisfaction parameters. Additionally, recommender's trustworthiness and confidence about a recommendation are considered while evaluating recommendations. Simulation experiments on a file sharing application show that the proposed model can mitigate attacks on 16 different malicious behavior models. In the experiments, good peers were able to form trust relationships in their proximity and isolate malicious peers.

Index Terms—Peer-to-peer systems, trustworthiness, service, and recommendation, security

I. INTRODUCTION

PEER-TO-PEER (P2P) systems rely on collaboration of peers to accomplish tasks. Ease of performing malicious activity is a threat for security of P2P systems. Creating long-term trust relationships among peers can provide a more secure environment by reducing risk and uncertainty in future P2P interactions. However, establishing trust in an unknown entity is difficult in such a malicious environment. Furthermore, trust is a social concept and hard to measure with numerical values. Metrics are needed to represent trust in computational models. Classifying peers as either trustworthy
or untrustworthy is not sufficient in most cases. Metrics should have precision so peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trust- worthiness a challenge.Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other .
Management of trust information is dependent to the structure of P2P network. In distributed hash table (DHT)-based approaches, each peer becomes a trust holder by storing feedbacks about other peers [1], [3], [4]. Global trust. Information stored by trust holders can be accessed through DHT efficiently. In unstructured networks, each peer

stores trust information about peers in its neighbor- hood or peers interacted in the past [2], [5], [6]. A peer sends trust queries to learn trust information of other peers. A trust query is either flooded to the network or sent to neighborhood of the query initiator. Generally, calculated trust information is not global and does not reflect opinions of all peers.

We propose a TRUST ROUTING IN PEER-TO-PEER SYSTEMS USING SELF-ORGANIZINGTRUST MODEL that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity. No a priori information or a trusted peer is used to leverage trust establishment. Peers do not try to collect trust information from all peers. Each peer develops its own local view of trust about the peers interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers. Since peers generally tend to interact with a small set of peers [7], forming trust relations in proximity of peers helps to mitigate attacks in a P2P system.

In trust routing in peer-to-peer systems using self-organizing trust model, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester. An acquaintance's feedback about a peer, recommendation, is evaluated based on recommender's trust- worthiness. It contains the recommender's own experience about the peer, information collected from the recommender's acquaintances, and the recommender's level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender.

II. THE COMPUTATIONAL MODEL OF SORT

We make the following assumptions. Peers are equal in computational power and responsibility. There are no privileged, centralized, or trusted peers to manage trust relationships. Peers occasionally leave and join the network. A peer provides services and uses services of others. For simplicity of discussion, one type of interaction is considered in the service context, i.e., file download.

Modules:

- Creating Network Scenario
- Node Creation and Configuration
- Computational Trust Model
- Calculation of Metrics
- Performance Evaluation

Creating the Network Scenario:

In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines. There are two ways of defining network geometry: the physical topology and the logical (or signal) topology. The physical topology of a network is the actual geometric layout of workstations. Logical (or signal) topology refers to the nature of the paths the signals follow from node to node. The number nodes is going to participate in the simulation is decided. Here we conduct experiments to a group of wireless nodes in a network that operates on the Dynamic Source Routing (DSR) protocol. We hence use only a logical topology as it is wireless environment.

Node Creation and Configuration:

Node creation is nothing but the creation of the wireless nodes in the network scenario that is decided. Node configuration essentially consists of defining the different node characteristics before creating them. They may consist of the type of addressing structure used in the simulation, defining the network components for mobile nodes, turning on or off the trace options at Agent/Router/MAC levels, selecting the type of adhoc routing protocol for wireless nodes or defining their energy model. Simulator::node-config accommodates flexible and modular construction of different node definitions within the same base Node class. For instance, to create a mobile node capable of wireless communication, one no longer needs a specialized node creation command.

Computational Trust Model:

A P2P system consists of peers parity in terms of responsibility and computational power. There are no privileged, centralized, or trusted peers to manage trust relationships among peers. Peers are indistinguishable in

computational power, network bandwidth and storage space. Although a small fraction of peers may behave maliciously, the majority of them are expected to behave honest. Peers occasionally leave and join the network and provide services to others and use services from others. For simplicity in discussion, one service operation, e.g., file request/download is considered. Depending on the value of trust obtained from the various computations the next forwarder node in a normal communication is obtained.

Calculation of Metrics:

SORT defines several trust metrics. Three of them are important: reputation, service trust, and recommendation trust. Reputation metric is a value resulting from the evaluation of recommendations of acquaintances. Service trust metric represents a peer's trust in an acquaintance in service context based on its past service interactions and reputation. Reputation and service trust values of pi about pj are denoted by 0 5 ri.7; stij 5 1 respectively. Service trust value is the primary metric to make trust decisions when making decisions about a service provider. Recommendation trust metric is analogous to service trust metric in recommendation context and is used when selecting acquaintances for reputation queries and evaluating recommendations. Its value is calculated based on past recommendation interactions and reputation. Recommendation trust value of pi about pj is denoted by 0 <= rtij <= 1 respectively. These metrics are dynamically evaluated inside the network.

Performance Evaluation:

During simulation time the events are traced by using the trace files. The performance of the network is evaluated by executing the trace files. The events are recorded into trace files while executing record procedure. In this procedure, we trace the events like packet received, Packets lost, Last packet received time etc. These trace values are write into the trace files. This procedure is recursively called for every 0.05ms. So, trace values recorded for every 0.05ms. The simulated results support the performance of the system described theoretically.

Selecting Service Providers

When pi searches for a particular service, it gets a list of service providers. Considering a file sharing application, pi may download a file from either one or multiple uploaders. With multiple uploaders, checking integrity is a problem since any file part downloaded from an uploader might be inauthentic. Some complex methods utilizing Merkel hashes, secure hashes, and cryptography [47], [48] can be used to do online integrity checking with multiple uploaders. Since this issue is beyond the scope of this paper, the next sections assume one uploader scenario. Service provider selection is done based on service trust metric, service history size, competence belief, and integrity belief values. When pi wants to download a file, it selects an uploader with the highest service trust value. If service trust values are equal, the peer with a larger service history size (SH) is selected to prioritize the one with more direct experience.

.

III. EXPERIMENTS AND ANALYSIS

A file sharing simulation program is implemented in Java to observe results of using SORT in a P2P environment. Some questions studied in the experiments are as follows: how SORT handles attacks, how much attacks can be mitigated, how much recommendations are (not) helpful in correctly identifying malicious peers, and what type of attackers are the most harmful.

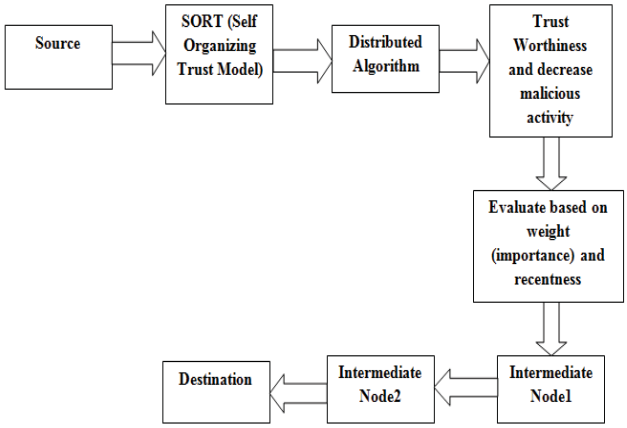

Fig.1 Block Diagram of the Proposed System

Method

The simulation runs as cycles. Each cycle represents a period of time. Downloading a file is an interaction. A peer sharing files is called an uploader. A peer downloading a file is called a downloader. The set of peers who downloaded a file from a peer are called downloaders of the peer. An ongoing down- load/upload operation is called a session. Simulation parameters are generated based on results of several empirical studies [8], [9], [10] to make observations realistic. A file search request reaches up to 40 percent of the network and returns online uploaders only. A file is downloaded from one uploader to simplify integrity checking. All peers are assumed to have antivirus software so they can detect infected files. Four different cases are studied to understand effects of trust calculation methods under attack conditions.

No trust:

Trust information is not used for uploader selection. An uploader is selected according to its bandwidth. This method is the base case to under- stand if trust is helpful to mitigate attacks. .

No reputation query:

An uploader is selected based on trust information but peers do not request recommendations from other peers. Trust calculation is done based on SORT equations but reputation (r) value is always zero for a peer. This method will help us to assess if recommendations are helpful. All SORT equations are used as in Section 3. .

Flood reputation query:

SORT equations are used but a reputation query is flooded to the whole network. This method will help us to understand if getting more recommendations is helpful to mitigate at- tacks. A peer may request a recommendation from strangers. In this case, a peer assigns a recommendation trust value to the stranger as rt stranger ¼ where rt and rt are the mean and standard deviation of recommendation trust values of all acquaintances. If a peer does not have any acquaintances, rt stranger= 0:1.

Before starting a session, a downloader makes a bandwidth agreement with the uploader, which declares the amount of bandwidth it can devote. Parameters of each finished session are recorded as an interaction. The satisfaction parameter is calculated based on following variables Tables Data charts which are typically black and white, but sometimes include color.

Bandwidth:

The ratio of average bandwidth (Ave Bw) and agreed bandwidth (Agr Bw) is a measure of reliability of an uploader in terms of bandwidth. .

Online Period:

The ratio of online (On P) and offline (Off P) periods represents availability of an uploader.

The weight of an interaction is calculated based on two variables:

File size:

A large file is more important than a small one due to bandwidth consumption. However, importance is not directly related to the file size. We assume that files over 100 MB have the same importance.

Popularity:

Popular files are more important than unpopular ones. We assume that number of uploaders of a file is an indication of its popularity. File Formats For Graphics

Attacker Model

Across multiple platforms. When submitting your final paper, your graphics should all be submitted individually in one of these formats along with the manuscript.

Unfairly high. Giving a positively-biased trust value about a peer where r ¼=cb = ib ¼=1 and sh =shmax. .

Unfairly low. Giving a negatively-biased trust value about a peer where r =cb = ib = 0 and sh =sh max author chooses, however it is recommended that figures are not sized less than column width unless when necessary.

Setting sh = sh max maximizes the effect of a misleading recommendation. A fair recommendation is the recommender's unbiased trust information about a peer. A service- based attack can be detected immediately since a virus infected or an inauthentic file can be recognized after the download. However, it is hard for a peer to determine a recommendation-based attack if the peer's own experience conflicts with a recommendation. Since a recommender might be cheated by attackers, there is no evidence to prove that a recommendation is intentionally given as misleading.

A good peer uploads authentic files and gives fair recommendations. A malicious peer (attacker) performs both service and recommendation-based attacks. Four different attack behaviors are studied for malicious peers: naive, discriminatory, hypocritical, and oscillatory behaviors. A non-malicious network consists of only good peers. A malicious network contains both good and malicious peers. If malicious peers do not know about each other and perform attacks independently, they are called as individual attackers. Individual attackers may attack each other. For individual attackers, attack behaviors are as follows:

Naive:

The attacker always uploads infected/ inauthentic files and gives unfairly low recommendations about others .

Discriminatory:

The attacker selects a group of victims and always uploads infected/inauthentic files to them,[5]. It gives unfairly low recommendations about victims. For other peers, it behaves as a good peer.

Hypocritical:

The attacker uploads infected/authentic files and gives unfairly low recommendations with x percent probability [3], [5]. In the other times, it behaves as a good peer.

Oscillatory:

The attacker builds a high reputation by being good for a long time period. Then, it behaves as a naive attacker for a short period of time. After the malicious period, it becomes a good peer again.

Analysis on Individual Attackers

This section explains the results of experiments on individual attackers. For each type of individual attacker, two separate network topologies are created: one with 10 percent malicious and one with 50 percent malicious. Each network topology is tested with four trust calculation methods. In the experiments, a hypocritical attacker behaves malicious in 20 percent of all interactions. A discriminatory attacker selects 10 percent of all peers as victims. An oscillatory attacker behaves good for 1,000 cycles and malicious for 100 cycles.

Service-based attacks:

Table 2 shows the percentage of service-based attacks prevented by each trust calculation method. When a malicious peer uploads an infected/ inauthentic file, it is recorded as a service-based attack. Number of attacks in No Trust method is considered as the base case to understand how many attacks can happen without using trust information. Then, number of attacks observed for each trust calculation method is compared with the base case to determine the percentage of attacks prevented. In the table, No RQ and Flood RQ denote "No reputation query" and "Flood reputation query" methods, respectively.

In a 10 percent malicious network, all methods can prevent more than 60 percent of attacks of naive attackers. No

RQ method's performance is close to other methods since a good peer identifies a naive attacker after having the first interaction. Thus, recommendations are not very helpful in the naive case. For discriminatory attackers, the situation is similar since their naive attacks easily reveal their identity to victims. For the hypocritical and oscillatory attackers, a good peer may not identify an attacker in the first interaction. Therefore, recommendations in SORT and Flood RQ methods can be helpful to identify some attackers before an attack happens.

In a 50 percent malicious network, the prevention ratio can be maintained over 60 percent for naive and discriminatory behaviors since attackers are identified quickly. No RQ method can perform close to SORT and Flood RQ methods. In hypocritical and oscillatory behaviors ,SORT can prevent nearly 40 percent of all attacks, which is still good considering the extreme number of attackers. Although attack prevention ratio is higher in the naive behavior, number of attacks is 4-8 times higher than other attacker types. Thus, naive attackers can be considered more successful than other type of individual attackers.

In SORT, a peer interacts less with strangers as its set of acquaintances grows. Therefore, rate of service-based attacks decreases with time. In all cases, SORT's prevention ratio for service-based attacks is close to Flood RQ method. However, Flood RQ method causes 7-10 times more recommendation traffic than SORT. The difference in misleading recommendations is much higher as explained below. Thus, SORT has a better performance tradeoff than Flood RQ method.

Sample Screen Shots:

Trust routing in p2p system using sort can be implemented using ns2 simulator with help of nam . Based on the experiment result during the simulation the result will be listed below . her totally 30 nodes are created.

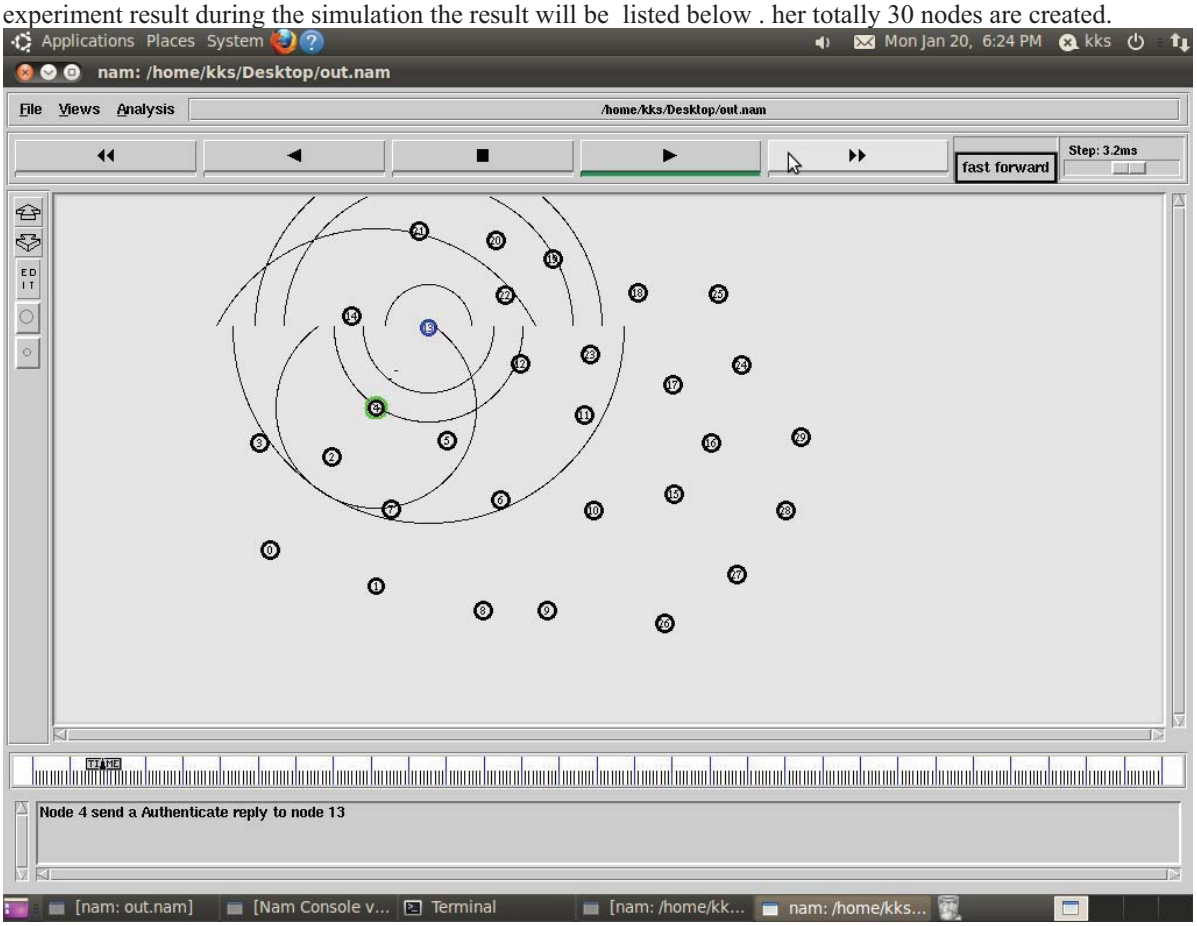

Fig 2: authentication reply nodes

The above fig 2 will define the authenticate nodes reply between node 4 to node 10

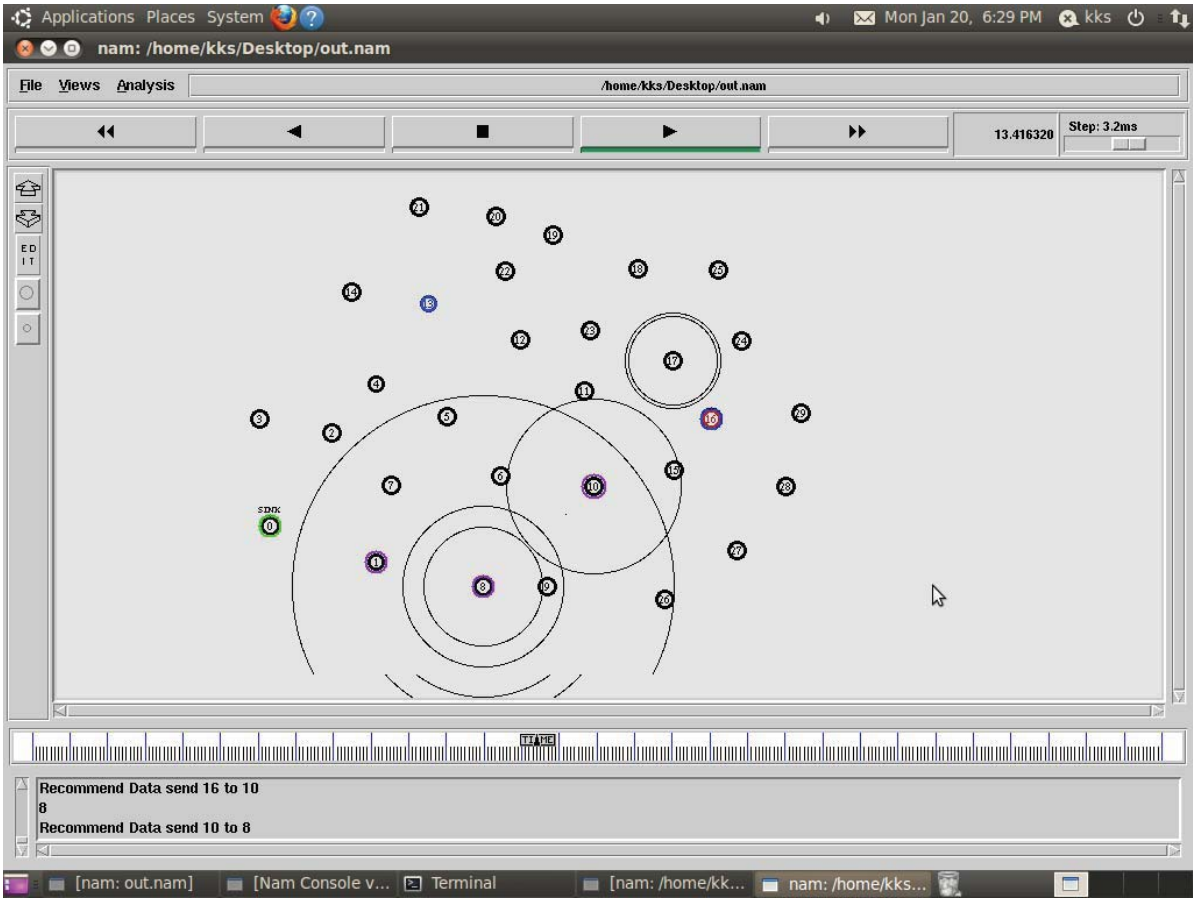

Fig 3: recommend data send

Recommendation-based attacks:

In the simulations, when a malicious peer gives a misleading recommendation, it is recorded as are commendation based attack. When SORT is used, peers form their own trust network with time and do not request recommendations from untrustworthy peers.

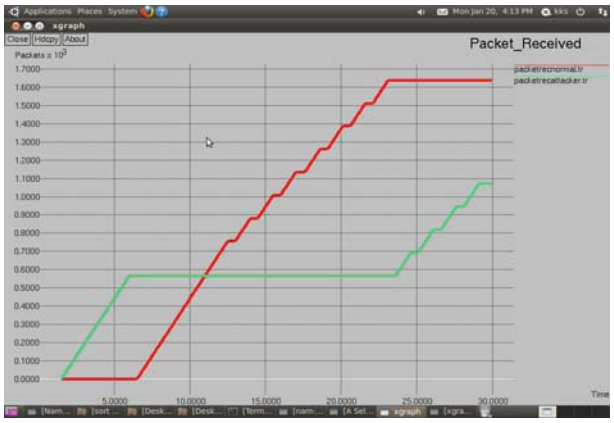

Fig 2: Packet received

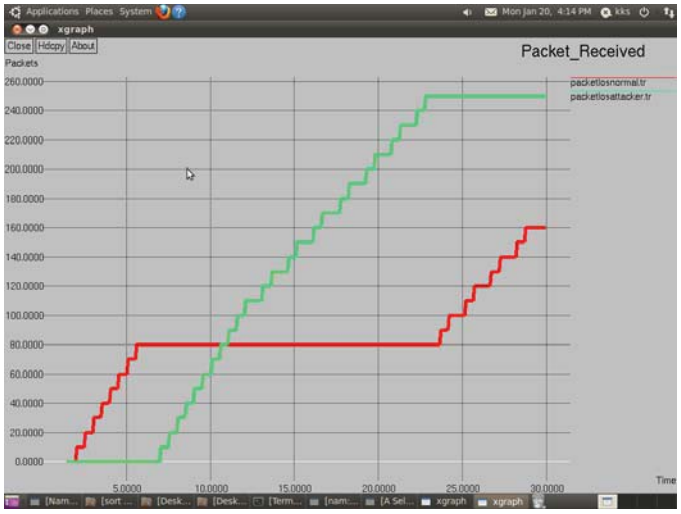

Fig 3: packet loss

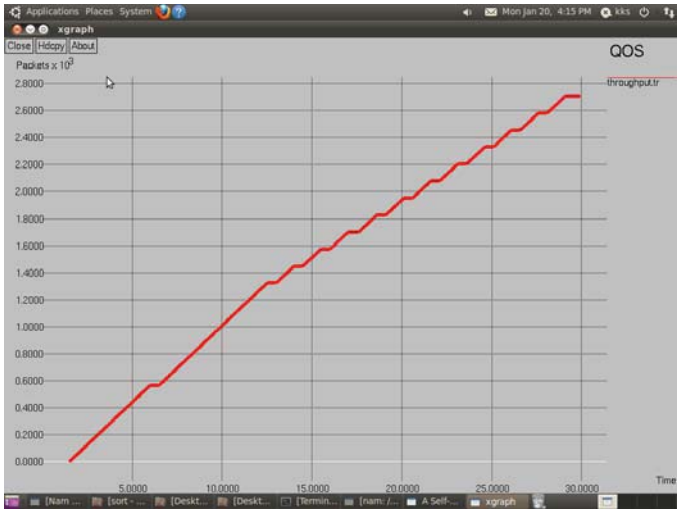

Fig 4: Throughput

Therefore, SORT can effectively mitigate recommendation-based attacks with time. In Flood RQ method, peers collect more recommendations from both acquaintances and strangers. Therefore, attackers find opportunity to disseminate more misleading recommendations as strangers. In Flood RQ method, numbers of recommendation based attacks are roughly 10 to 30 times more than SORT in discriminatory, hypocritical, and oscillatory behaviors. Naive attackers cannot disseminate misleading recommendations with SORT since they are identified after the first interaction. In Flood RQ method, if a peer is not interacted with a naive attacker before, it can request recommendations from the attacker as a stranger. Therefore, naive attackers can disseminate more misleading recommendations than other attacker types in Flood RQ method. This observation shows that instead of considering public opinion, collecting recommendations from acquaintances provides more reliable information.

In 50 percent malicious network, recommendation trust values of attackers are close to good peers so attackers can disseminate more misleading recommendations. However, SORT still mitigates misleading recommendations 5-10 times more than Flood RQ method.

Distribution of trust metrics:

Peers with higher capabilities (network bandwidth, online period, and number of shared files) can finish more interactions successfully. Thus, they generally have better reputation and service trust values. Recommendation trust

values are not directly related to peer capabilities since giving a recommendation does not require high capabilities.

IV. CONCLUSION

A trust model for P2P networks is presented, in which a peer can develop a trust network in its proximity. A peer can isolate malicious peers around itself as it develops trust relationships with good peers. Two context of trust, service and recommendation contexts are defined to measure capabilities of peers in providing services and giving recommendations. Interactions and recommendations are considered with satisfaction, weight, and fading effect parameters. own experience, information from its acquaintances, and level of confidence in the recommendation. These parameters provided us a better assessment of trustworthiness. Individual, collaborative, and pseudonym changing attackers are studied in the experiments. Damage of collaboration and pseudo spoofing is dependent to attack behavior. Although recommendations are important in hypocritical and oscillatory attackers, pseudo spoofers, and collaborators, they are less useful in naive and discriminatory attackers. SORT mitigated both service and recommendation-based attacks in most experiments. However, in extremely malicious environments such as a 50 percent malicious network, collaborators can continue to disseminate large amount of misleading recommendations. Another issue about SORT is maintaining trust all over the network. If a peer changes its point of attachment to the network, it might lose a part of its trust network. This issue might be studied as a future work to extend the trust model.

Using trust information does not solve all security problems in P2P systems but can enhance security and effectiveness of systems. If interactions are modeled correctly, SORT can be adapted to various P2P applications, e.g., CPU sharing, storage networks, and P2P gaming. Defining application specific context of trust and related metrics can help to assess trustworthiness in various tasks.

REFERENCES

[1] Habib, D. Xu, M. Atallah, B. Bhargava, and J. Chuang, "A Tree-Based Forward Digest Protocol to Verify Data Integrity in Distributed Media Streaming," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 1010-1014, July 2005

[2] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9, pp. 1282-1295, Sept. 2008.

[3] L. Xiong and L. Liu, "Peer trust: Supporting Reputation-Based Trust for Peer-to-Peer Ecommerce Communities," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized Gossip Algorithms," IEEE/ACM Trans. Networking, vol. 52, no. 6, pp. 2508-2530, June 2006.

[5] R. Zhou and K. Hwang, "Power trust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 4, pp. 460-473, Apr. 2007