ISSN:0975-9646

A Fuzzy Approach for Reputation Management using Voting Scheme in Bittorrent P2P Network

Ansuman Mahapatra , Nachiketa Tarasia

School of Computer Engineering
KIIT University, Bhubaneswar, Orissa, India

***Abstract*— Reputation management for peers in Bittorrent P2P network is very much important due to its open and anonymous nature. Some malicious peers distribute low quality or fake files in the network and pollute the environment. In this paper, we propose a fuzzy inference system to design P2P reputation management system which generates the reputation values for each torrent files by interacting with other peers. The inference system takes the fuzzy voted values from the peers and calculates the reputation for torrent file. Fuzzy logic is effective for calculation of reputation as many factors affecting the reputation are uncertain and has fuzziness. We have taken three factors such as: quality, popularity and size of the file share by a peer and simulate the resulting reputation management system.**

Keywords— ***Fuzzy Reputation Management, Bittorrent P2P Network, Voting Scheme***

I. INTRODUCTION

The popularity of the Bittorrent file sharing protocol is growing extensively as it can distribute large files by efficiently utilizing the peers upload bandwidth. But due to open and anonymous nature of P2P network some malicious peers distribute low quality or fake files in the network and pollute the environment. So there should be some techniques present which will let the peers know about the trustworthiness of the shared files and the peer who is sharing the file. So a reputation management system is required, which will store the reputation based on some input factors given to the fuzzy inference system and the output will be the reputation of the shared file by that peer.

There are many reputation schemes are proposes in P2P networks. But there are very few number of reputation system exits for Bittorrent P2P network. Lan Yu has proposed a reputation management system X^{2BT}Rep [1]. In this scheme after download complete the peers will vote for the torrent file as well as for the peer who share that file. The votes are stored in three different repositories- Offerer repository, Torrent Repository and Credibility Repository. The votes are encrypted and sent to the central tracker. The tracker manages the vote submitted by the pooler & rejects the votes which will look suspicious.

In this paper, we will manage the reputation of the peers joining the Bittorrent P2P system. To calculate the reputation of the peer there are six steps: fuzzification of crisp input variables, application of fuzzy operator in the antecedent, fuzzy rule base creation, rule evaluation, aggregation of consequent across the rules, defuzzyfication to get the crisp output. The process is shown in the fig-1.

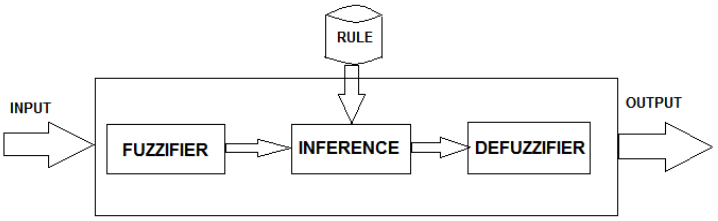

Fig 1: Basic components of a sensor node

II. RELATED WORK

Several works has been done related to trust, reputation and credibility measurement in P2P. However, all can't cover all the factors that affect the reputation of peer which will give robust trustworthiness. So we are using fuzzy model as in P2P network everything changes dynamically. Peers come and leave the system so frequently that it is not possible to measure exact values for all the factors. So we are taking fuzzy logic to analyze the reputation.

Damiani et al. propose XRep [8], a protocol for trust management in decentralized P2P systems. In XRep peers query about the reputation of resources or peers by using network broadcasts to retrieve votes from neighboring peers. Cornelli et al. present a trust system, called P2Prep [3], where peers poll the network for reputation opinions on service providers. Both of these approaches are designed for Gnutella, a first generation P2P system which allows message broadcasting to discover information sharing peers. Both models do not account for the credibility of information sharing peers in their trust calculations.

X^2Rep [4] proposed by Curtis, Safavi-Naini, Susilo, extends XRep and is designed to protect against the weaknesses of XRep. As in XRep, a distributed polling algorithm is employed in X^2Rep to manage the reputations based on resources and servants. However, it is an

improvement on XRep and has the ability to compute the weight of a peer based on past voting experiences.

Fajiang Yu, Huanguo Zhang, Fei Yan [2] presented a trust model of choosing trusted source peers in P2P system based on the fuzzy relation theory in fuzzy mathematics. The authors defined four types of fuzzy trust relations, and gave the ways of fuzzy global trust relation matrix computing, which used as the base of judging which source peers were trusted peers. They consider three types of relations between peers fuzzy direct trust relation, fuzzy recommending relation and fuzzy indirect trust relation. Then the reputations are combined to form the fuzzy global trust relation.

Shanshan, Hwang, Runfang, Kwok [5] proposed a fuzzy reputation aggregation system which can better handle uncertainty, fuzziness, and incomplete information in peer trust reports. This system aggregates peer reputations with affordable message overhead. By testing the system using eBay transaction data in the public domain, the authors seek to demonstrate the efficacy and robustness of two P2P reputation systems: Fuzzy Trust and Eigen Trust at establishing trust among the peers in P2P applications. Peer anonymity and secure storage of global reputations are not considered.

In our previous work, we have developed a fuzzy inference system which generates reputation for peers automatically by taking three fuzzy inputs number of files, seeding time and upload bandwidth of peer which reduces the communication overhead between peers of the Bittorrent system. [7]

From the above study we came to the conclusion that the reputation or trust management system works on the interaction between peers i.e. the reputation is given by one peer to the other peer directly or by recommendation basis and also the reputation is only calculated for the peers only, not for the files they shared. So we have taken three factors such as: quality, popularity and size of the file share by a peer & calculate the reputation for torrent file shared by that peer.

III. PROPOSED WORK

In this section, we introduce our proposed fuzzy model for the calculation of reputation of peer participating in the Bittorrent network.

A. The Concept

The reputation of a file is decided by three factors taken by us: quality, popularity and size of the file share by a peer. Quality of the file shared affects the reputation of the peer majorly, as it is the important factor which makes the user to decide whether to download the file or not. Good quality files got good reputation and bad or fake files got very low or no reputation. We take this factor to recognize whether a file is fake file. Second factor, popularity of the file plays an important role in deciding reputation of a file. If a file is more popular, the demand of the file increases in the internet. Many people will download that file if the quality of the file is good. As a file is more popular it is widely available in the internet. So the third factor size of the shared file decides which file to download. If the file quality is good and the size of one file is less than the size of other file. Then peers download the less size file to save their download bandwidth and time. So all the factors depend upon each other and combinedly affect the reputation of the peer and torrent file.

B. Fuzzy Interface Model

As seen from fig-1 there are four basic parts of a fuzzy inference model: fuzzifier, Inference system, Rule base, defuzzifier. Fuzzification is the first step in which the inputs are taken and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. As the peers stays in the system for very short period of time, it is much difficult to measure exact values for the input variables. We have three input variables to the fuzzy inference system: Quality, Popularity and Size of the shared file. It is difficult to derive a mathematical formula involving these three inputs to calculate reputation. So we are using fuzzy logic to calculate our reputation for peers associated with the system. The input variables are first fuzzified by using membership functions. We determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions.

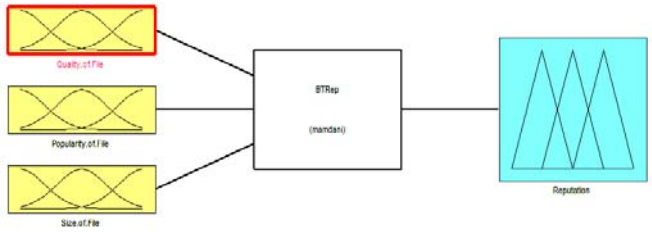

Fig 2: P2P Reputation Model based on Mamdani-type Fuzzy Inference System

A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. There are many Membership Functions such as triangular, Gaussian, Bell, Z, PI, Trapezoidal, Sigmoid and so on.

Fig-3, 4, 5 shows the membership function for quality of the file, popularity of shared files (in percentage) and size of the file (in Gb) respectively which has three linguistic values *low, medium, and high* for quality & popularity of file and less, medium, large for size of file.

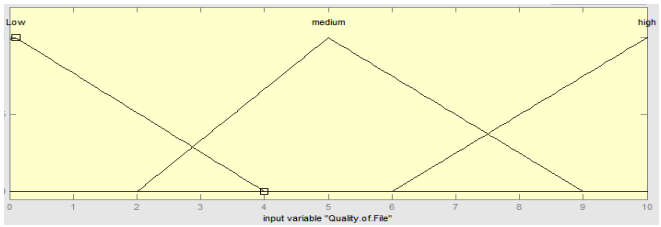

Fig 3: Membership function for Quality of share files

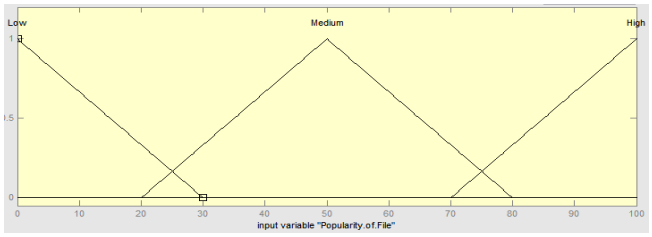

Fig 4: Membership function for Popularity of shared file

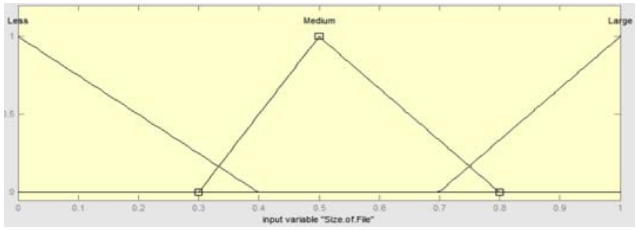

Fig 5: Membership function for Size of the shared file

Once the inputs are fuzzified, we apply the fuzzy rule base to arrive at the fuzzy output. Fuzzy Inference Rule Base comprises many Fuzzy Rules. Fig-6 shows the fuzzy rule base for the inference system.

Quality of the file

Popularity	LOW	MEDIUM	HIGH
LOW	VERY LOW	VERY LOW	LOW
MEDIUM	VERY LOW	LOW	LOW
HIGH	LOW	MEDIUM	HIGH

Less size of file

(A)

Quality of the file

Popularity	LOW	MEDIUM	HIGH
LOW	VERY LOW	LOW	MEDIUM
MEDIUM	LOW	MEDIUM	HIGH
HIGH	MEDIUM	HIGH	VERY HIGH

Medium size of file

(B)

Quality of the file

Popularity	LOW	MEDIUM	HIGH
LOW	LOW	MEDIUM	HIGH
MEDIUM	MEDIUM	HIGH	VERY HIGH
HIGH	HIGH	VERY HIGH	VERY HIGH

Large size of file

(C)

Fig 6: Fuzzy rule base for reputation management system

Fuzzy Inference Engine is Mamdani-type fuzzy inference system. This method is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. Mamdani-type inference, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. Mamdani is a Min-Min-Max fuzzy inference method [9].

The input for the defuzzification process is an aggregate output fuzzy set and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. The result of defuzzication U is:

$$U = i\mu(Ui)*Uii\mu(Ui) \qquad (1)$$

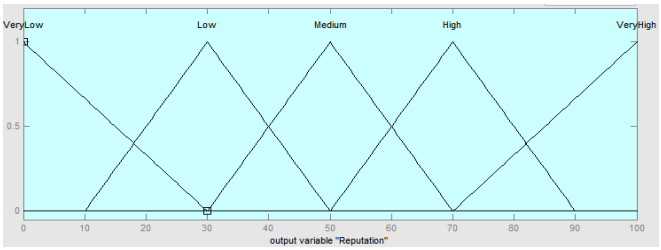

Fig 7: Membership functions for calculation of reputation

From the fuzzy output, we use center of gravity defuzzyfication to arrive at the crisp output which is our required reputation value. The membership function for calculating reputation is shown in the fig-7, which has five linguistic values *very low*, *low medium*, *high*, *very high*.

We than store this reputation value in two types of reputation repository for future use by other peers for decision making. If any peer wants to download any data from a peer it first requests for the reputation of the torrent file and then decides if it download the data. We have proposed an algorithm for this decision making process in section *F*.

C. Distributed Reputation Repository

We make assumptions that each peer i keep a consistent pair of public (PK_i) and private keys (SK_i) and a peer_id (P_i) associated with the peer, which is the digest of the public key, obtained using a secure hash function. Each torrent file is associated with an identifier (info_hash), which is the SHA1 hash of the "info" section in the torrent file.

We require a tracker that plays a significant role in the verification of torrent files and the distribution of reputation to the peers. The trusted tracker T is associated with a consistent pair of public (PK_T) and private keys (SK_T) for verification and security purpose. [1]

Each peer records its own experiences on torrent files in its local torrent repository and the credits of pollers who provide votes on torrent files is stored in credibility repository. Each peer is required to store data about its experiences in two local repositories:

- A *Torrent Repository*, which stores a pair (info_hash, value), where a value describes whether an info_hash that the peer has experienced is good or poor.

- A *Credibility Repository*, which is a pair (peer_id, value). The credibility value describes the reliability of peers giving votes and measures the trustworthiness of votes from them in the voting system. It is real value, with an interval from 0 to 1. Its value is set to zero during the initialization for a new poller. Whenever the pooler gives a valid vote, after validation of its vote, the value of its credibility repository increases. The credibility value will also increase if the peer will continue uploading after download the whole file.

D. Voting

There are two types of votes: the torrent vote and the credibility vote. In order to generate a vote on a torrent file and on offerer, each peer checks its local torrent and credibility repositories.

The vote for a torrent file is a value 0 or 1 associated with Info_hash, which is denoted as V_T= 0/1. We can get this value from torrent repository.

If a downloader receives a vote and several pieces from the client and finds that the file is genuine after downloading is complete. The downloader will then give some bonus credibility values according to the client's uploading contribution.

To protect authenticity and integrity, each poller generates a self-signed certificate signed by their secrete key called a vote certificate, which contains an identifier of a torrent, an offerer's_id, a poller's_id, vote on the torrent file and a timestamp, and then combines its public key for verification before submitting it to a tracker. [1]

E. Central Reputation Repository

A trusted tracker manages the reputation values from different peers. After receiving reputation related information from a peer, the tracker verifies it using the public key of the peer stores it in the central repository and removes suspicious ones. At the initial stage when a peer joins the system newly, it's all field is initialized to zero. When the time goes on these values are updated in the central repository automatically.

F. Search Algorithm

In this section we propose a search algorithm for peer who wants to download data from other peers. Peer want to download something from the Bittorrent system, it will send a *PollRequest () message* to the central tracker with its peer_id. When the tracker receives a download request from a peer this algorithm is invoked. Verification of the peer_id (P_i) of the requesting peer is done by its public key (PK_i). If the peer is verified then the reputation, IP address and port number of the peers, who has the data and is uploading, is sent to the requesting peer by *PollReply() message* and it is encrypted by trackers private key(SK_T). If the peer is not verified by the tracker the request is neglected and only IP address and port number is sent. [7]

PollRequest (P_i)

If P_i is verified
    Queries the central repository for reputation of peers;
    PollReply (Rep);
Else
    Return (Null);

Peer after receiving the reputation value (Rep) checks the authenticity of the *PollReply ()* message by verifying the trackers public key (PK_T). Then it extracts the reputation value and compares with the threshold reputation. If the reputation of the torrent file is greater than the threshold reputation value, then it connects to that peer and the piece exchange takes place.

Reputation updating is done in a regular timely manner or on demand by other peers by following *UpdateRepository ()* algorithm by the tracker. Tracker takes three inputs from peers and the reputation is calculated by the fuzzy inference system, and then stored in the repository. Votes of those peers are only calculated which are authentic to the system and the peer_id is verified by the tracker. [7]

UpdateRepository (n, t, b)
If (P_i is verified)
        Extract the three inputs Quality,
        Popularity and Size of the file;
        Input them to fuzzy inference system;
        Collect the crisp output;
        Store in the torrent repository;
Else
        Update failure;
Retry ();

To update the credibility repository *Credibility_Algorithm ()* is invoked. Before downloading a file peer must require the reputation information of that file. So other peers vote for that file. If the peer completed downloading that file and sees that the file is genuine then it

will update the credibility repository according to the amount of data he has downloaded from an offerer.

Credibility_Algorithm ()

If (the downloaded file is genuine)
Calculate the credibility;
Credibility award =No. of pieces downloaded from the seeder
Total no. of pieces in the file ;
Update the Credibility Repository;
Else
Give negative credit to the seeder/voter;

.

IV. IMPLEMENTATION

We have implemented our fuzzy inference system in Fuzzy Logic Toolbox Graphical User Interface Tool of Matlab. Fig-8 and 9 is a sketch map of Mamdani-type Fuzzy Inference. The Rule Viewer displays a roadmap of the whole fuzzy inference process. It is based on the fuzzy inference diagram described in the previous section.

Fig-10 and 11 gives a three-dimensional simulative sketch map of P2P Fuzzy Inference reputation. By inputting arbitrary Quality, Popularity and Size of the shared file, the figure wills maps to a fuzzy output value Reputation. Because this curve represents a two-input one-output case, you can see the entire mapping in one plot. When we move beyond three dimensions, we start to encounter trouble displaying the results. That's why we have represented two figures related to reputation output.

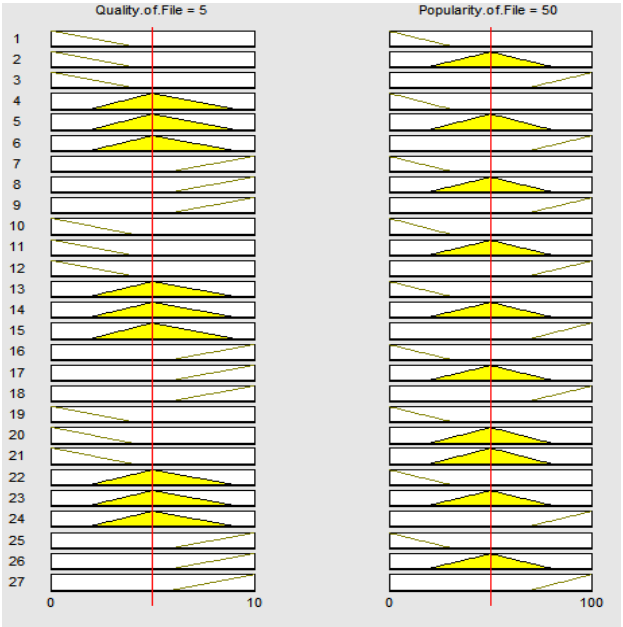

Fig 8: A sketch map of Mamdani-type Fuzzy Inference

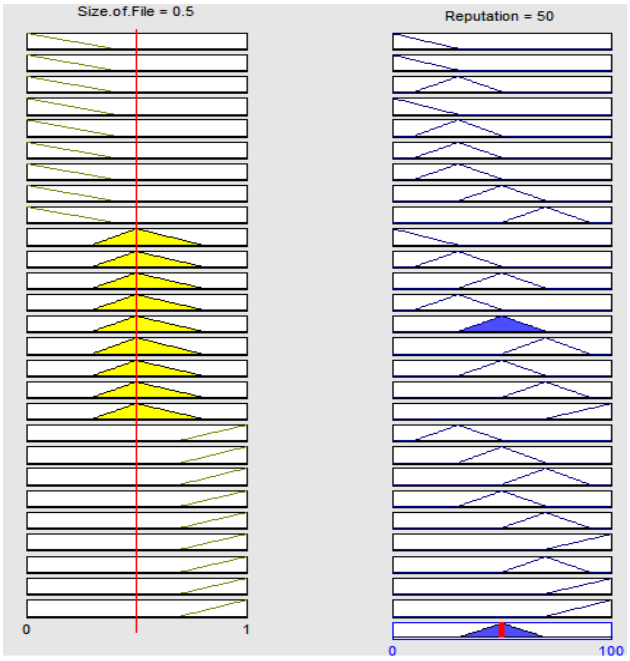

Fig 9: A sketch map of Mamdani-type Fuzzy Inference

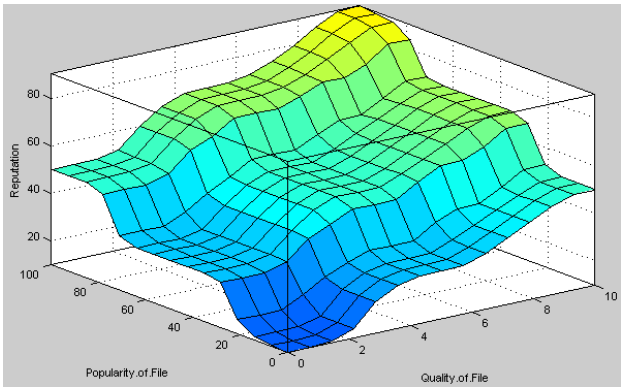

Fig 10: A three-dimensional simulative sketch map of P2P Fuzzy Inference Reputation plotting Upload bandwidth Vs. No. of files shared

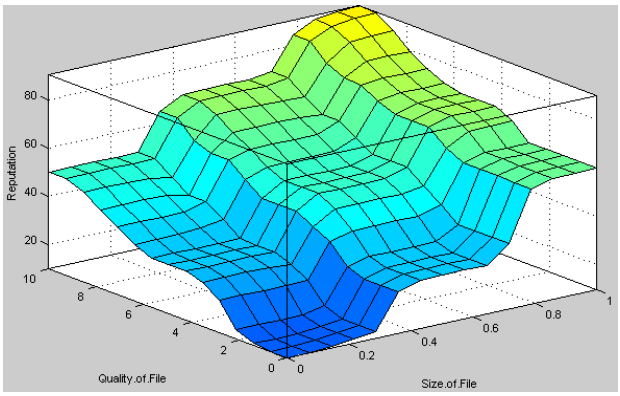

Fig 11: A three-dimensional simulative sketch map of P2P Fuzzy Inference Reputation plotting Seeding time Vs. No. of files shared

V. CONCLUSION

The evaluation of trust and reputation is a very important issue in an open, dynamic and distributed P2P environment. As all the factors are changing so dynamically it is good to take fuzzy logic to calculate reputation. In this paper, we present a new fuzzy inference trust model in Bittorrent P2P network environment. Our fuzzy model takes three inputs from the peers automatically and computes the reputation of the peer without any extra communication overhead. Collecting votes from other peers directly and collecting votes from recommending peers are not required. So this system is much more efficient than other existing reputation management systems. The central tracker automatically generates and stores the reputation securely in the global repository and provides it whenever any peer request for it.

REFERENCES

[1] Lan Yu, Willy Susilo and Rei Safavi-Naini, X^2BT Trusted Reputation System: A Robust Mechanism for P2P Networks, Cryptology and Network Security, Lecture Notes in Computer Science, 2006, Volume 4301/2006, 364-380, DOI: 10.1007/11935070_26.

[2] Fajiang Yu, Huanguo Zhang, Fei Yan, School of Computer, Wuhan University, A Fuzzy Relation Trust Model in P2P System, Computational Intelligence and Security, 2006 International Conference, 3-6 Nov. 2006, 1497 – 1502, 1-4244-0605-6.

[3] F. Cornelli, E. Damiani, S.D. di Vimercati, S. Paraboschi, P. Samarati, Choosing reputable servents in a P2P network, in: Proceedings of the 11th International Conference on World Wide Web, Honolulu, 2002.

[4] Nathan Curtis, R. Safavi-Naini, and W. Susilo, X^2Rep: Enhanced trust semantics for the XREP protocol. Applied Cryptography and Network Security. Second International Conference, ACNS2004. Proceedings. LNCS3089, pages 205–219, 2004.

[5] Shanshan Song, Kai Hwang, Runfang Zhou, Yu-Kwong Kwok, Trusted P2P Transactions with Fuzzy Reputation Aggregation, IEEE Internet Computing, vol. 9, no. 6, pp. 24-34, Nov./Dec. 2005, doi:10.1109/MIC.2005.136.

[6] Chunqi Tian, Shihong Zou, Lingwei Chu, Shiduan Cheng, A New Trust Framework Based on Reputation for Unstructured P2P Networks, Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International, 7-9 Dec. 2008.

[7] Ansuman Mahapatra, Nachiketa Tarasia, Anuja Ajay, Soumya Ray, A Fuzzy Approach for Reputation Management in Bittorrent P2P Network, 3rd International Conference on Electronics Computer Technology, ICECT 2011,IEEE, 8-10 Apr 2011.

[8] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati and F. Violante, A reputation-based approach for choosing reliable resources in peer-to-peer networks. Proceedings of the 9th ACM conference on Computer and Communications Security, pages 207–216, November 2002.

[9] Hongwei Chen, Zhiwei Ye, wei Liu, Chunzhi Wang, Fuzzy Inference Trust in P2P Network Environment, IEEE, 2009.