RECOMMENDATION BASED P2P FILE SHARING ON DISCONNECTED MANET

Teegala Spandana*[1], Sesha Bhrgavi[2]

[1]M. Tech Student (Computer Networks and Information Technology), G. Narayanamma Institute Of Technology and Science, Hyderabad, Telangana, India.

Asst. Prof, G. Narayanamma Institute Of Technology and Science, Hyderabad, Telangana, India.

ABSTRACT

Our ultimate aim is to improve the file sharing system with reduced file searching cost and delay. In past few years, personal mobile devices such as laptops and smart phones have been more and more popular. MANETs consisting of digital devices, nodes are constantly changing the location, forming disconnected MANETs with opportunistic device encountering. In this paper, we propose a P2P Information-based secured file sharing system, namely RFS, for disconnected MANETs. In this system, we enhance our base work by using recommendation to select the peers. The system uses an interest extraction algorithm to derive a node's interests for Information-based file searching without involving untrustworthy node. Each group has one Global leader for each known foreign group, which serves as the bridge to the group.

Keywords: P2P, security, file sharing, load balancing.

1. INTRODUCTION

P2P is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged participants in the application. Each device (laptop, smart phones) in the network is referred to as a node. The owner of each node on a P2P network would set aside a portion of its resources - such as processing power, disk storage, or network bandwidth - to be made directly available to other network participant, without the need for central coordination by servers or stable hosts. With this model, peers are both suppliers and consumers of resources, in contrast to the traditional client to server model where only the server supply (send), and clients consume (receive). Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual group thereby empowering it to engage in greater tasks beyond that can be accomplished by individual peers, yet are beneficial to all the peers.

Peer-2-peer systems often implement an abstract overlay network, built at Application Layer, on top of the native or physical network topology. Such overlays are used for indexing and peer discovery and make the P2P system independent from the physical network topology. Information is typically exchanged directly over the underlying Internet Protocol (IP) network. Anonymous peer-2-peer systems are an exception, and implement extra routing layers to obscure the identity of the source or destination user/node. A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client to server model where communication is usually to and from a central server. In structured P2P networks, peers are organized following specific criteria and algorithms, which lead to overlays with specific topologies and properties.

*Corresponding Author

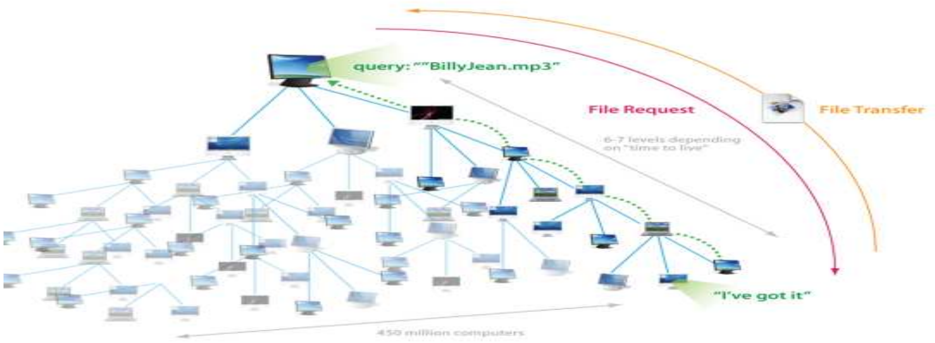


Fig. 1: peer to peer mode

Unstructured P2P networks do not impose any structure on the overlay networks. Peers in these networks connect in an ad-hoc fashion based on some loose set of rules. Ideally, unstructured P2P systems would have absolutely no centralized elements/nodes, but in practice there are several types of unstructured systems with various degrees of centralization. Three categories can easily be seen:

In pure peer-2-peer systems the entire network consists solely of equipotent peers. There is only one routing layer, as there are no preferred nodes with any special infrastructure function. In centralized peer-2-peer systems, a central server is used for indexing functions and to bootstrap the entire system. Although this has similarities with a structured architecture, the connections between peers are not determined by any algorithm.

Hybrid peer-2-peer systems allow such infrastructure nodes to exist often called super nodes.

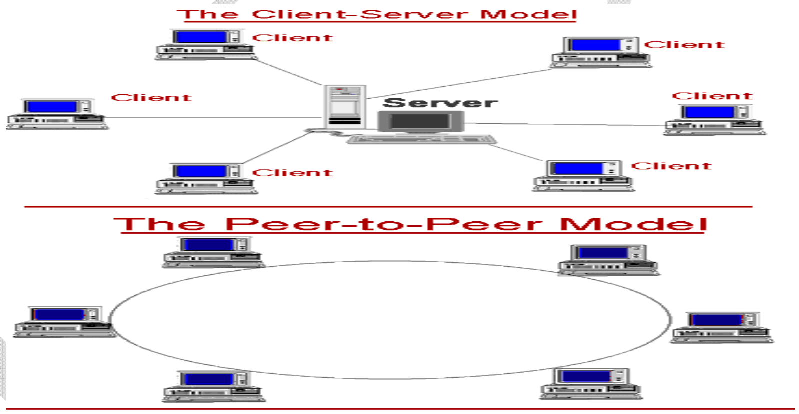


Fig. 2: Peer to peer and client/server model

Wired peer to peer file sharing systems have already become a popular and successful paradigm for file sharing among millions of users. As the mobile digital devices are carried by people that usually belong to certain social relationships, in our base paper, researchers focused on the P2P file sharing in a disconnected MANET group consisting of mobile users with social network properties. In this paper, we focused on the security in file searching system.

2. RELATED WORKS

In this paper [1], author proposed an approach that uses combined reputations of servants and resources, providing more informative polling's and overcoming the limitations of servant based only solutions. Servant reputations are associated with the servant identifier, which has to be tamper resistant.

In this paper [2], author proposed a Bayesian network-based trust model and a method for building reputation based on recommendations in peer-2-peer networks. Since trust is multi-faceted, peers need to develop differentiated trust in different aspects of other peers' capability. The peer's needs are different in different situations. Depending on the situation, a peer may need to consider its trust in a specific aspect of another peer's capability or in multiple aspects.

In this paper [3], author motivated the importance of anonymity, especially in such trust based systems. Author presented 'Trust Me a secure and anonymous underlying protocol' for trust management. The protocol provides mutual anonymity for both the trust host and the trust querying peer.

In this paper [4], author presented an approach that addresses the problem of reputation based trust management and the semantic level. Author employs at both levels scalable data structures and allow to access trust by computing an agents reputation from its former interaction with other agent

2.1. Summery of existing system

In previous model, there is no trusted server to validate the peer. At same time trust mechanism is needed to punish peers that exhibit malicious behavior and furthermore, an access control mechanism is developed to secure the files sharing p2p network. In that model, each system stores the experience of file sharing in its own memory for future use. In this type, peer know about upload peer whether good or bad, which already downloaded file from that peer, another peers only considering the reputation in that uploading peer, it may chances to hack by malicious node **.** And there is no time specification, so it'll make some problem in transaction.

2.2. Previous work summery

In flooding-based methods exploits the mobility of nodes within a geographic area to disseminate web Information among neighbors. It uses local broadcasting for Information searching and sets up Information indexes on nodes along the reply path to guide subsequent searching. In another method each file holder regularly broadcasts an advertisement message to inform surrounding nodes about its files. These flooding-based methods produce high overhead due to broadcasting. Though the advertisement-based methods reduce the overhead of flooding-based methods, but they still generate high overhead for advertising and cannot guarantee the success of file searching due to node mobility.

3. PROPOSED WORK

In our proposed method, the file is grouped based on the frequent searching processes regarding the files. In our proposed technique, we are considering the disconnected MANET as group. In our proposed system, we take advantage of different types of node mobility for file sharing. We define group Local leader and Global leader nodes in the view of a social network. A group Local leader is an important and popular node in the group. In our method, we are enhancing the existing model with some modification. In our proposed model we are introducing following things P2P rep model, Trusting peer, Evaluate peer, Dictionary access control. More secure than existing model and it satisfies the requirements of access control for p2pfile sharing system. In our work, peers send reputation queries to peers interacted in the past, which reduces network traffic comparing to flooding-based approaches. Further more, each peer expands its trust network with time and can obtain more credible recommendations from acquaintances.

3.1 Modules

We have divided our proposed technique into small modules, they are given below, Network design (Global leader Node, Local leader Node, Member node), Group Formation (File type, File searching), Own risk model, P2P rep model, Trusting peer, Evaluate peer, Volunteer recommendation.

3.1.1. Network Design

Each node can act with any one of the three different properties according to situation. 1) Global leader Node (The node which capable to collect the neighbor foreign group information. This node can connect the different groups to share the file). 2) Local leader Node (The node which is stable in the group, and contacting to the group node frequently. These nodes which are capable to collect the information of file availability in own group.) 4) Normal node (The node which maintaining only the own information)

3.1.2. Group formation

In this module, we planned to group the nodes, based on file Information. The group formation depends on the file information; group of members contains the different type of files. So the group will be formed based on the file availability and searching process to improve the file searching system.

In this module, we planned to divide the file searching scheme into two sub-modules, the file search will be done by the interest oriented file searching algorithm. In this module, the Local leader collects the information of file availability in the group. So if any member needs the file files then the nodes can ask to the Local leader. If searching file information is not available in Local leader node, the file may available in other group. That file information will be collected by using Global leader node from other group

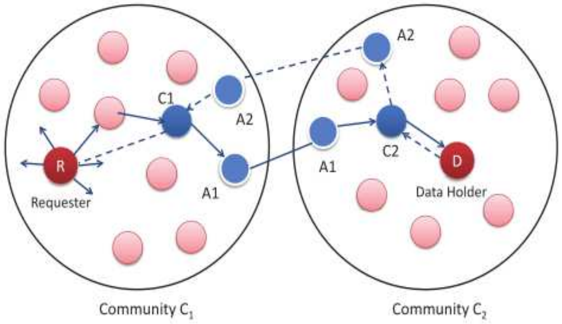


Fig. 3: Example model of different properties of network devices

3.1.3. Own risk model

All of the nodes in network not having any other node information at initial time. Therefore node can’t believe wither node is good or bad The requesting peer will select the uploading peer based on the downloading agreement (file size, packet size, bandwidth allocation, total duration to upload the file).

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

3.1.4. P2P rep model

The reputation metric measures a stranger’s trustworthiness based on recommendations. If node finds number of peer width indented file then its need to confirm wither peer is good or bad, so it will request to all peer about indented peer. By receiving recommendation from other peers, node can calculate the reputation value. The requesting peer will select the best peer based on higher reputation value (if own downloading history is low)

$$r_{lm} = \frac{[\mu_{sh}]}{sh_{max}}\left(ecb_{lm} - {}^{eib_{lm}}/_{2}\right) + \left(1 - \frac{[\mu_{sh}]}{sh_{max}}\right) er_{lm}$$

3.1.5. Trusting peer

In this model, we are introducing the method to accept the trust based peer selection. If requesting peer already done more transaction then it can believe the peer with less number of recommendation. Based on own history value, the node will select the best peer to download the file

$$rt_{ln} = \frac{rh_{lp}}{rh_{max}}\left(rcb_{lp} - {}^{rib_{lp}}/_{2}\right) + \frac{rh_{max} - rh_{lp}}{rh_{max}} r_{lp} \cdot$$

3.1.6. Enhanced volunteer recommendation

In our base model, they have considered the limiting the number of downloader's to maintain the own trustworthiness in other peers. In base model, if limit is crossed then the uploading peer will ignore the req. To find the good peer, requester needs to spend most of the time on recommendation checking. We have enhanced base model to resolve the problem of delay. In this module, if limit is crossed then the node will check the highly trusted peer with requested file. If peer found then node will generates volunteer recommendation

If any volunteer recommendation is received, then the node will check recommender is highly trusted node or not. If yes then the node won't make **Recom_req**, directly it will download the file from recommended node. In this module, the node evaluates the peer in two ways, 1) Service based, 2) Recommendation based. After each download, the peer will verify the agreement with final download level. Based on the performance, service info will be updated. After receiving recommendation from number of peer, the node will verify the peer's bad recommendation by comparing all recommendation. Then new value will be updated. By checking service and recommendation, the best peer will be considered for file download and recommendation.

Selection of best service provider may overload few peers while other peers having same resources are idle. A load balancing mechanism is implemented in this work, to utilize the resources of eligible good peers. In this method, each peer's simultaneous operations are limited to a maximum. If a peer reaches its maximum number of simultaneous operations, instead of simply rejecting the incoming requests, it suggests another good peer having the same resource to the service requester. Hence, this method avoids time required for the service requester to find another good service provider and also reduces the network traffic.

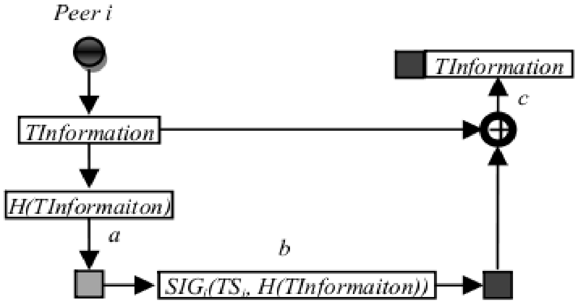


Fig. 4: File sharing request

Algorithm:

```
ALGORITHM
            $U_c$ is Uploading count,
$N_{L_{Th}}$ is Max up limit threshold
$W_p$ is working period
$NW_p$ is non working period
$B$ bandwidth, $B_{U_R}$ bandwidth upload reserved,
 $B_{us}$ bandwith used
STEP 1
 If (peer has to download file)
1.          Generate file request and send
2.          Wait for reply and good per selection for a time
 STEP 2
     If (any peer received the request)
1.          Check for upload limit (ENHANCEMENT)
```

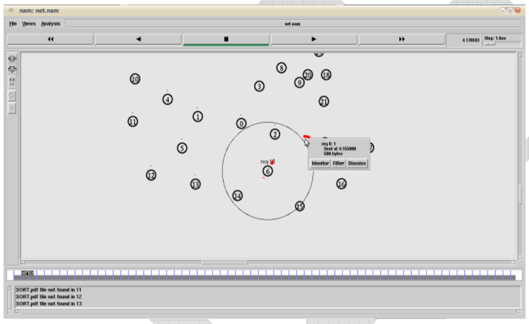
```
$U_c = \begin{cases} 0, & At\ initial \\ \sum_{l=1}^{n} xl, x = 1, n = 1,2,3..n \end{cases}$
If (limit is not crossed) $U_c \leq N_{L_{Th}}$
If (file found)
Set the bandwidth possibilities
$B = B_m - (B_{U_R} * Rand) - B_{us}$
                    Give reply
            1.  else
                    Ignore
If (limit is crossed) $U_c > N_{L_{Th}}$
        1.  check for good peer $st_{lm} = \frac{sh_{lm}}{sh_{max}}\left(cb_{lm} - {}^{ib_{lm}}/_{2}\right)$
            For each   $j \epsilon Ai$
                If $St_{lm} > S_{Th}$ found
                 Recom $\boldsymbol{m}$
            else
                    Ignore **m**
STEP 3
    If (request received)
        1.  add peer in to lis
                1.  $P_l \cup P_{List}$
        2.  send recommendation request to all other peers
STEP 4
    If (recommendation request received)
    1.  Checks the history
        1.  For-each $H_l \epsilon SH_{List_m}$
                a. if (info found)
                Send recommendation info
              b. else
                    Ignore
STEP 5
      If (recommendation received)
        1.  add the recommendation info in to a list $R_{m} \cup R_{List}$
STEP 6
      If (best peer recommendation received)
```

```
        1.  send data request
        2.  collect the data
        3.  set the satisfaction = $W_p/(W_p + NW_p)$
STEP 7
    If (data request)
        1.  check crossed the limit $\mathrm{U_c} > N_{LTh}$
            I.  If crossed limit
                a.  send objection message
STEP 8
    Time out for check best peer
        1.  filter the recommendation by SORT algorithm
        2.  Select best peer
```

4. RESULTS

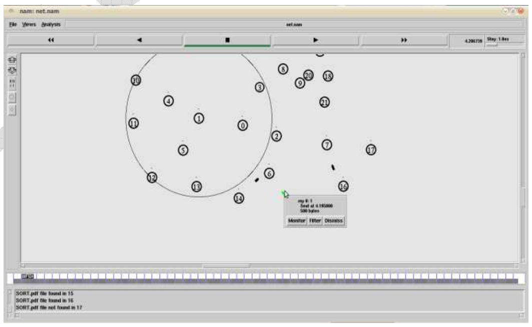
We have tested our proposed network with popular simulation tool called NS2. We have used the Single PC with configuration of 20 GB Hard disc space, 1 GB RAM, software's Linux OS (Ubuntu 10.04) and NS2.34. We have written the program by TCL (**Front End language**). We simulated our proposed system with two types of results. One is Nam and Xgraph.

In this section, we presented main result steps in fig 5-9, which shows the different packets used in the trust management process.



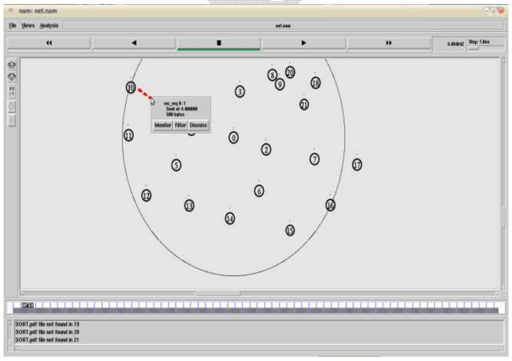
```
SORT.pdf file not found in 11
SORT.pdf file not found in 12
SORT.pdf file not found in 13
```

Fig. 5: File searching request



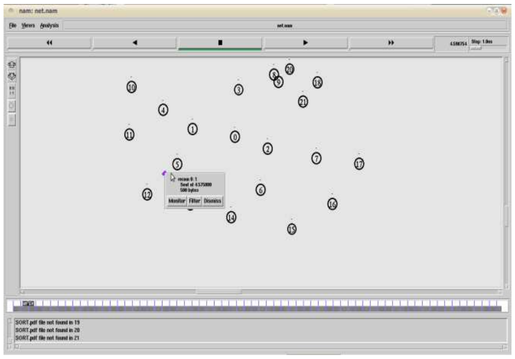
```
SORT.pdf file found in 15
SORT.pdf file found in 16
SORT.pdf file not found in 17
```

Fig. 6: File available reply



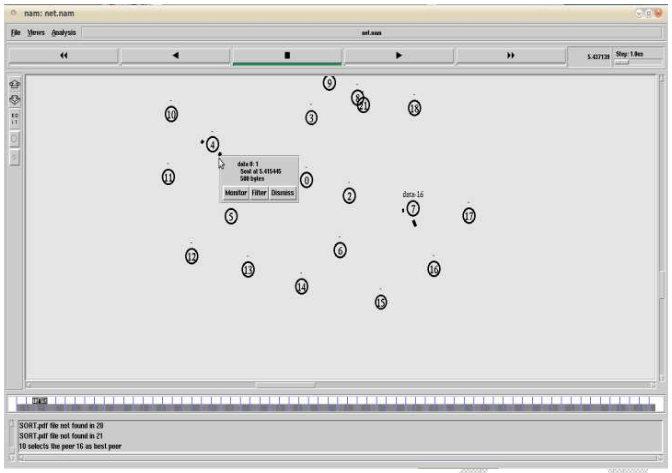
```
SORT.pdf file not found in 19
SORT.pdf file not found in 20
SORT.pdf file not found in 21
```

Fig.7: Requesting for best peer recommendation



```
SORT.pdf file not found in 19
SORT.pdf file not found in 20
SORT.pdf file not found in 21
```

Fig. 8: Recommendation ion about best peer



```
SORT.pdf file not found in 20
SORT.pdf file not found in 21
10 selects the peer 16 as best peer
```

Fig. 9: Download the file from best peer

The graph (10 & 11) shows packet delivery and overhead comparison. From graph we can know our proposed system works well in un-trusted environment

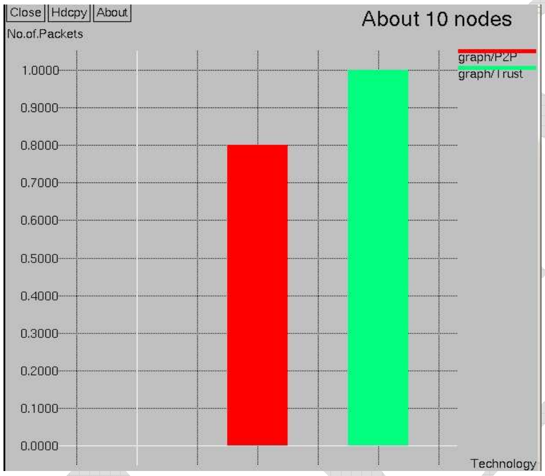


Fig. 10: Pkt delivery graph with basic p2p and trust based P2P

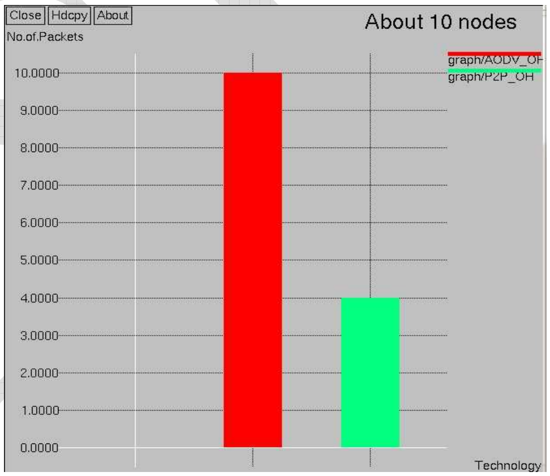


Fig.11. overhead comparison AODV and P2P method

5. CONCLUSION AND FUTURE WORK

In our proposed method, the file is Information based on the frequent query processes regarding the files. In our proposed technique, we are considered the disconnected MANET as group groups. And we have implemented less overhead file searching system and we have tested successfully. The problem of identifying wrong recommendations is reduced in this work. It reduces the service based attacks and it also reduces the recommendation based attacks if there are not more than 50% malicious nodes in the P2P network. It uses three types of metrics, service, and reputation and recommendation trust metrics to create a trust network in a peer's proximity. This work also implements the load balancing mechanism to utilize the network resources effectively. When the best service provider in the network reaches its maximum number of simultaneous operations, it suggests another good peer having the same service to the service requester. Hence, the time required for service requester to choose a different peer is reduced.

It helps reducing large amounts of attacks but, this work does not solve all the security issues of a P2P network. This issue should be focused in future work to use the trust model in various applications. Future work may consist of using different load balancing algorithms to reduce the delay of getting a resource.

REFERENCES

[1] Selcuk AA, Uzun E, Pariente MR. A Reputation-Based Trust Management System for P2P Networks. Proc. IEEE/ACM Fourth Int'l Symp. Cluster Computing and the Grid (CCGRID), 2004.

[2] Hoffman K, Zage D, Nita-Rotaru C. A Survey of Attack and Defense Techniques for Reputation Systems. ACM Computing Surveys 2009; 42(1): 1:1-1:31.

[3] Zhou R, Hwang K. Power trust: A Robust and Scalable Reputation System for Trusted Peer-2-Peer Computing. IEEE Trans. Parallel and Distributed Systems 2007; 18(4): 460-473.

[4] Burak Can A, Bhargava B. SORT : A Self-Organizing Trust Model for Peer-2-Peer Systems. IEEE Trans. Dependable and Secure Computing 2013; 10(1).

[5] Marsh S. Formalising Trust as a Computational Concept, PhD thesis, Dept. of Math. and Computer Science, Univ. of Stirling, 1994.

[6] Abdul-Rahman A, Hailes S. Supporting Trust in Virtual Communities. Proc. 33rd Hawaii Int'l Conf. System Sciences (HICSS), 2000.

[7] Yu B, Singh M. A Social Mechanism of Reputation Management in Electronic Communities. Proc. Cooperative Information Agents (CIA), 2000.

[8] Despotovic Z, Aberer K. Trust-Aware Delivery of Composite Goods. Proc. First Int'l Conf. Agents and Peer-2-Peer Computing, 2002.

[9] Zhou R, Hwang K, Cai M. Gossiptrust for Fast Reputation Aggregation in Peer-2-Peer Networks. IEEE Trans. Knowledge and Data Eng. 2008; 20(9): 1282-1295.

[10] Ooi B, Liau C, Tan K. Managing Trust in Peer-2-Peer Systems Using Reputation-Based Techniques. Proc. Fourth Int'l Conf. Web Age Information Management, 2003.