Construct a Simulator of a Proposed Trust and Reputation Model in (PD) P2P File Sharing Systems

Dujan B. Taha **Abdullah M. Salih**

dujan_taha@uomosul.edu.iq

College of Computer Sciences and Mathematics
University of Mosul

Received on: 07/09/2011 **Accepted on: 02/11/2011**

ABSTRACT

Trust concept is an important requirement for sustained interactions between peers, and to deal with malicious peers in P2P file sharing systems. Traditional security mechanisms and services are unable to protect against malicious behaviors, therefore trust and reputation management is considered an appropriate solution that can provide a protection against such threat. In this paper, we focused on the advantage of some existing trust models to formulate a *new model* that solves problems raised in the previous models. We also considered the *partially decentralized* (*PD*) peer-2-peer (P2P) architecture to execute the proposed model. *Finally*, we construct a *C# based simulator* to test proposed model on the partially decentralized P2P file sharing network. Simulation results show that the model is able to identify malicious peers effectively and isolate them from the system (sharing files), hence reducing the amount of inauthentic uploads and increasing peers' satisfaction.

Keywords: Trust and Reputation, (PD) P2P, File Sharing Systems.

بناء محاكي لنموذج ثقة وسمعة مقترح في أنظمة مشاركة ملفات (PD)P2P

عبد الله محمد صالح **دجان بشير طه**

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ قبول البحث: 2011/11/2 **تاريخ استلام البحث:2011/9/7**

الملخص

أن مفهوم الثقة يعتبر متطلب مهم للتفاعلات المتواصلة بين الأقران وللتعامل مع الأقران الخبيثة في أنظمة مشاركة الملفات (P2P Systems). أن آليات وخدمات الأمنية التقليدية غير قادرة للحماية ضد السلوكيات الخبيثة، لذلك تعتبر أدارة الثقة والسمعة حلاً مناسباً للحماية ضد هذا النوع من التهديدات. في هذا البحث تم التركيز على نقاط القوة في بعض نماذج الثقة الموجودة لتكوين نموذج جديد يحل المشاكل (نقاط الضعف) في النماذج السابقة, بالإضافة إلى ذلك تم الأخذ بنظر الاعتبار هيكلية الأنظمة اللامركزية جزئياً لتنفيذ النموذج المقترح. أخيراً، قمنا ببناء برنامج محاكي باستخدام لغة C# 2008 الموجودة ضمن بيئة (Visual Studio .Net) لاختبار النموذج المقترح، بعد تنفيذه على شبكة افتراضية لامركزية جزئياً لمشاركة الملفات. بينت نتائج المحاكاة بأن النموذج المقترح له القدرة على تحديد الأقران الخبيثة بكفاءة ومنعها من مشاركة الملفات، بالإضافة إلى ذلك تقليل عدد الملفات الغير موثوقة (التي يتم رفعها من قبل الأقران الخبيثة) وزيادة الملفات الموثوقة (التي يتم تنزيلها) التي تمثل رضا الأقران (Peers' satisfaction).

الكلمات المفتاحية: نموذج ثقة وسمعة، (PD) P2P، انظمة مشاركة الملفات.

1. Introduction

Peer-to-peer file-sharing networks are currently receiving much attention as a means of sharing and distributing information, these networks become an essential part of the Internet and many successful P2P applications have been developed and widely used, such that grid computing, semantic web, web services and file sharing applications. P2P file sharing systems provide a large collection of files available for download. In traditional systems, little information is given to the user to help in the peer-selection and/or file-selection processes.

For example, if a user wants to download a file, the user is given a list of peers that has the requested file. The process of selecting the right peer without a prior information is frustrating and risky [8].

To foster positive interactions and reduce the risk involved in P2P file sharing systems, peers need to reason about trust, and reputation systems. Reputation systems are based on collecting information about peers' past transactions and computing a reputation value for these peers. The reputation values will be the basis for identifying trustworthy peers.

P2P systems can be divided into several categories (Figure 1): centralized, completely decentralized or partially decentralized systems [10]. Centralized P2P file sharing systems uses a centralized directory for searching files, while downloading a file is achieved directly between peers. In completely decentralized P2P systems, all peers have equal role and responsibilities. Partially decentralized P2P systems occupy the middle ground between centralized and completely decentralized systems. In these systems, supernodes or superpeers are peers that have extra capabilities and assume more responsibilities than regular peers. A supernode acts as a centralized server for the peers connected to it.

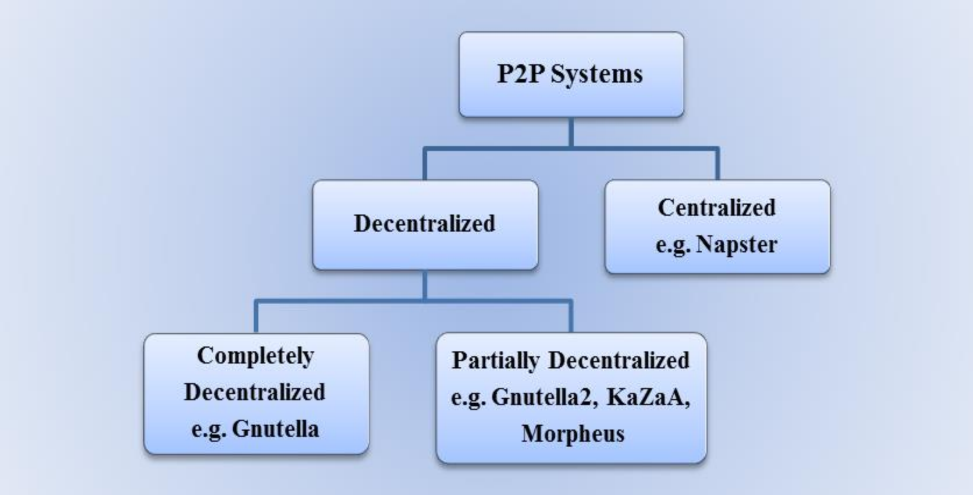

P2P Systems
Decentralized
Centralized e.g. Napster
Completely Decentralized e.g. Gnutella
Partially Decentralized e.g. Gnutella2, KaZaA, Morpheus

Figure (1). P2P systems [10].

While centralized P2P systems suffer from the single point of failure, the major challenge in completely decentralized systems is how to collect feedbacks and perform reputation computation efficiently. Several reputation-based systems have been proposed for completely decentralized systems. However, all proposed research works in this field have completely focused on these systems. Almost no attention was directed toward partially decentralized systems [9, 4].

2. Aim of the Work

The research objectives can be explained as follows:

1-Study the concept of trust and reputation management and the existing models related to the subject.

2-Propose a Trust and Reputation model that combines good points of some existing schemes and solves problems raised in those schemes. The proposed model works on the partially decentralized file sharing systems.

3- Construct a simulator for testing the proposed model on a partially decentralized file sharing system.

3. Related Works

Trust and reputation systems have been proposed recently, the most popular reputation system is the feedback scheme used by the eBay [2]. In 2003, Kamvar et al, proposed **EigenTrust** (inspired by PageRank) global trust ranking system [6]. Also, in 2003, Lee, Sherwood and Bhattacharjee have developed **NICE** distributed trust inference [7]. In 2004, Xiong and Liu have developed **PeerTrust**, a reputation-based trust supporting framework [13]. In 2005, Song et al. built the **FuzzyTrust** System (They proposed a P2P reputation system based on fuzzy logic inferences) [15]. Also, in 2005, Liang and Shi proposed **PET**, a **P**ersonalized **E**conomic-based **T**rust model for the P2P resource sharing [14]. In 2007, Zhou, Hwang and Cai. have developed the **PowerTrust** system for DHT-based P2P networks [17], Their group has also built the **GossipTrust** system in network by its gossip-based aggregation scheme [15]. In 2008, Zhao and Li have proposed selective aggregation schemes **H-Trust** [15]. In 2009, also Zhao and Li proposed vector based trust management scheme (**VectorTrust**) for aggregation of distributed trust scores [16].

4. Trust and Reputation Concept

4.1 Trust Definition

Trust is as old as the existence of human beings on this earth. People were grouped in tribes and within the same tribe, they trusted each other. The concept of trust has a significant role in the surviving of human beings; we experience and rely on trust on daily basis. However, trust is difficult to define clearly and precisely [8].

Researchers from different fields such as psychology, sociology, philosophy, history, law, business and economics have tackled the concept of trust from different views. According to the Oxford Dictionary, trust is a firm belief in the reliability, truth, ability, or strength of someone or something [8]. In 1973, Deutsch [5] has specified that trust is the confidence that an individual will find what is desired from another, rather than what is feared. In this work, we considered the first definition presented by Deutsch [5].

4.2 Reputation Definition

Reputation has been widely used in different disciplines such as psychology, sociology, business and economics. From the Oxford Dictionary, reputation is the beliefs or opinions that are generally held about someone or something. Abdul Rahman and Stephen. [1] define reputation as “an expectation about an agent’s behavior based on information about its past behavior”. Sabater et al. define it as an “opinion or view of one about something”. Mui et al. define it as “the perception that an agent creates through past actions about its intentions and norms” [8].

4.3 Trust and Reputation Values Representation

Trust levels have been represented differently by researchers from different domains based on the adopted trust definition and/or the applications or environments in which it is implemented. In general trustworthiness value which represents the outcome of the interaction, can be represented as follows [8, 12]:

1) ***Binary value***: The trustworthiness value can be 0 or 1 (e.g., XREP, Travos, CredibilityRecords) which means that a requester peer is satisfied from the transaction or not, the provider peer is trusted or not trusted.
2) ***Discrete value:*** The trustworthiness value can also be represented as discrete value (e.g., Excellent, Good, Fair and Poor) as in Amazon and DistributedTrust.
3) ***Real value:*** It can also be represented on a continuous scale from [0,1] (e.g., Nice, PeerTrust).
4) ***Probability value:*** It can also be the result of some probability measurements in the range of (0, 1), (e.g., EigenTrust and PowerTrust).

4.4 Traditional Systems versus Reputation-Based Systems

In traditional P2P systems (i.e., without any reputation mechanism), a user is given a list of peers that can provide the requested file. The user has then to choose one peer from which the download will be performed. This process is frustrating to the user because this latter struggles to choose the most trustworthy peer.

Reputation-based P2P systems were introduced to solve this problem. These systems try to provide a reputation management system that will evaluate the transactions performed by peers and associated a reputation value to these peers, The reputation values will be used as selection criteria among peers. The following figure (2) explains the difference (life cycle) between traditional and reputation-based systems [10].

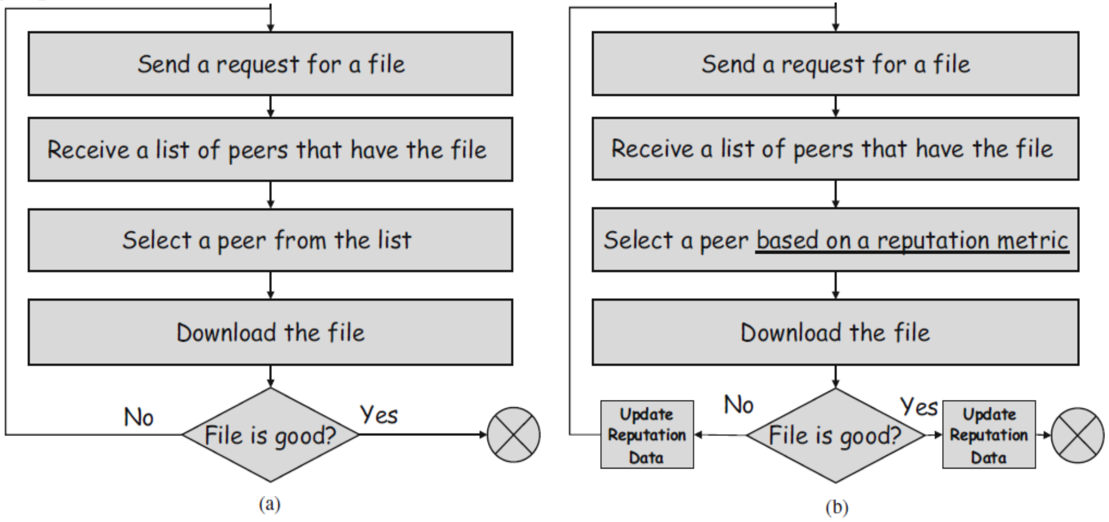

Send a request for a file
Receive a list of peers that have the file
Select a peer from the list
Download the file
No
File is good?
Yes
(a)
Send a request for a file
Receive a list of peers that have the file
Select a peer based on a reputation metric
Download the file
Update Reputation Data
No
File is good?
Yes
Update Reputation Data
(b)

Figure (2). (a) Life cycle in a traditional P2P system, (**b**) Life cycle in a reputation-based P2P system.

5. Problems of Existing Models

In this section, we'll identify problems raised in the previous works and present solutions for them. *First*, some reputation management schemes use the number of negative and positive downloads e.g., EigenTrust, other schemes use the negative downloads only. In some schemes, the size of the download is more important than the number of uploads. In EigenTrust model, the local trust value S_{ij} is defined as the sum of the ratings of the individual transactions that peer i has downloaded from peer j, in this model local trust values are normalized in the following equation.

$$C_{ij}=\max(S_{ij}, 0)/\sum j(S_{ij}, 0) \qquad \ldots(1)$$

This normalization form ensures that all values will be between 0 and 1, but there are some drawbacks of normalizing in this manner. The first drawback is that the normalized trust values do not distinguish between a peer with whom peer i did not interact and a peer with whom peer i has had a poor experience.

In this model, local trust value will be computed based on$\{\mathrm{sat}(i, j) - \mathrm{unsat}(i, j)\}$, only the difference between satisfied and unsatisfied upload is considered (which is called difference based algorithm (DB)). This may not give a real idea about the behavior of the peers, therefore, we considered the (Inauthentic Detector scheme IDA) that presented in [10] to solve EigenTrust problem.

Secondly, some of the proposed feedback-based reputation schemes (e.g., EigenTrust, fuzzyTrust) rely on peers' reputation for their peer-selection process. In this case, the most reputable peer usually has been selected. Other peers that are still in the process of building their reputation will not be selected to perform the upload. They will not be able to increase their reputation values. This may lead to peers' *starvation*. We proposed a solution to this problem, first we'll define *Threshold* value and choose a set of peers *Pj* such that $Behavior_{Pj} \geq Threshold$, this set is a candidate peers to be a provider peer.

Now, the following question may be asked. *Where to store trust/reputation information?* To answer this question, we should take into consideration the three types of P2P systems. In centralized P2P systems, the central entity will be used to store trust information (e.g., eBay, amazon). In completely decentralized P2P systems, the trust information regarding a trusted peer can be stored at each peer's level the storage cost is increased linearly as the number of peers increases (e.g., H-Trust, VectorTrust). In partially decentralized P2P systems, the reputation values stored at the superpeer level. In this research, we used the idea of H-Trust model, in H-Trust each peer has three local tables; they are (*local trust rating table, local service history table, and local credibility table*).

6. The Proposed Model

In this model, we combined multiple existing trust schemes to build a new model that solve some problems raised in the previous works. First, we'll explain these schemes in those models and their advantage, and then we'll depict, our designed *partially decentralized* architecture of p2p file sharing systems. The goal of this model is to detect malicious peers that are sending inauthentic files (e.g., corrupted files, misleading file names,..., etc.) and isolates them from the system (Prevents them from sharing files (uploading)).

In this model, we used a simple scheme [9] (called number based appreciation) to update reputation information (satisfied download SD, unsatisfied download UD, satisfied upload SU, unsatisfied upload UU) of both requestor i and provider j after each transaction. This scheme is illustrated as:

```
IF $AF_{ij}$=1     $SD_i = SD_i+1$
ELSE          $UD_i = UD_i+1$
```

Where AF_{ij} is the the appreciation of peer *Pi* for downloading the file *F* from *Pj*. This scheme can be replaced by another one (called size based appreciation), as follows:

```
IF $AF_{ij}$=1     $SD_i = SD_i$ + size(F)
ELSE          $UD_i = UD_i$+ size(F)
```

In this model, we used a simple IDA (Inauthentic detector Algorithm) which takes into consideration not only the difference between satisfied and unsatisfied upload (SU_j and UU_j), but also the sum of these values [10].

In the following scheme, we compute the real behavior (AB_j authentic behavior) of a peer *Pj* as:

$$AB_j = (SU_j - UU_j) / (SU_j + UU_j) = (SU_j - UU_j) / U_j \quad \text{IF } U_j \neq 0$$

$$AB_j = 0 \qquad \text{Otherwise}$$

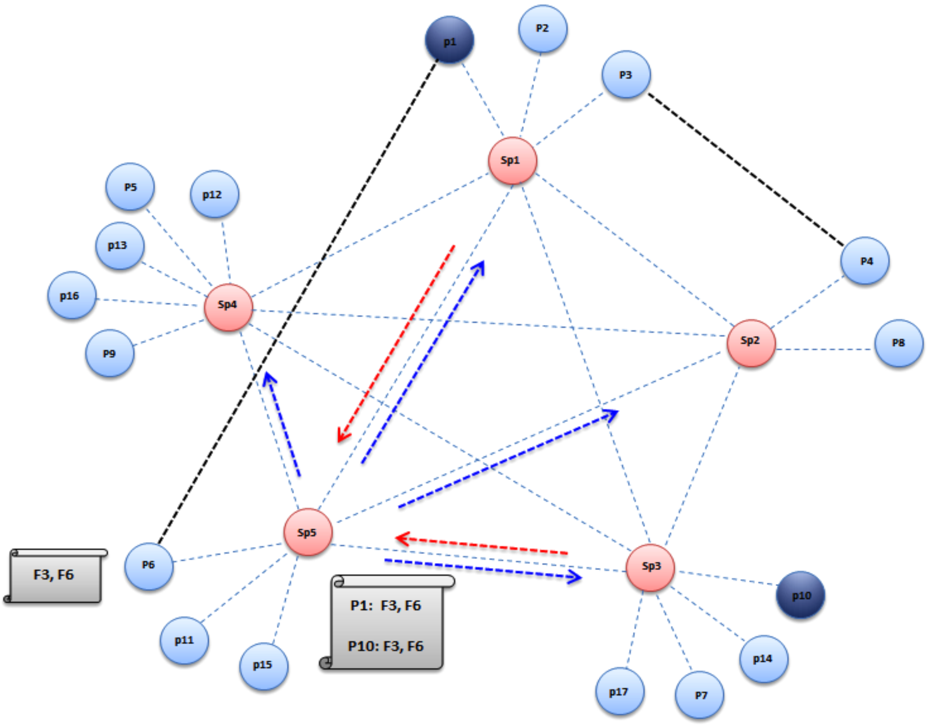
Note that the reputation *(AB)* as defined in the previous equation is a real number between -1 (IF SU_j=0) and 1(IF UU_j=0).

Also, we proposed a new approach to solve peer selection problem, In this approach, first we'll define *Threshold* value and choose a set of peers *Pj* such that $AB_j \geq$ *Threshold* the result is a set of the most reputable peers. Then select one peer randomly from these peers. This process will give those peers similar opportunity to be selected as a provider peer.

Finally, in this proposed model, we used new approach that is derived from (H-Trust and DHT in the PeerTrust) model but instead of using three tables (*local trust rating table, local service history table, and local credibility table*), we used two tables (Rep_ info table, Rep_values table). These two tables are stored at the superpeer level. After each transaction, Reputation information and reputation values are updated in those tables.

6.1 Partially Decentralized P2P Architecture

As we said in the previous sections, we considered the (*PD*) *partially decentralized* P2P architecture to solve problems of the distributed and fully decentralized systems. Now, we present a simple architecture of these systems, which is used in the simulator. **Figure (3)** illustrates this (PD) architecture. All the schemes that we discussed in the previous sections have been combined to formulate the proposed Algorithm, which was applied on the designed (PD) architecture, as described in figure (3).



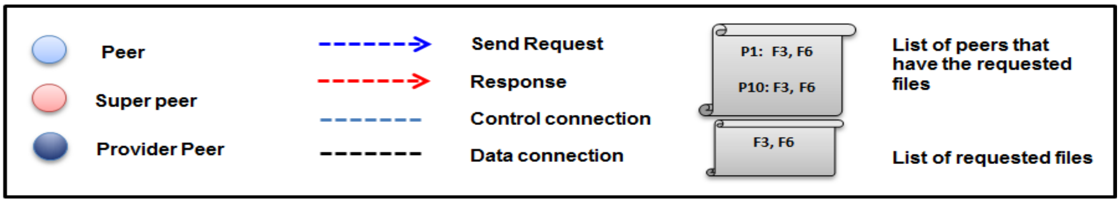


Figure (3). Partially Decentralized P2P Architecture

6.2 The proposed Algorithm

Assumptions:

***Pi* :** denotes the $peer_i$

***SD_{ij}* :** denotes the number of satisfied downloads performed from peer *Pj* by peer *Pi.*

***UD_{ij}* :** denotes the number of unsatisfied downloads performed from peer *Pj* by peer *Pi.*

***SU_{ij}* :** denotes the number of satisfied uploads by *Pj.*

***UU_{ij}* :** denotes the number of unsatisfied uploads by *Pj.*

***AF_{ij}* :** denotes the appreciation of peer *Pi* for downloading the file *F* from *Pj* .

***AB_j* :** denotes the authentic behavior of $peer_j$, ***Sup(i)* :** denotes the superpeer of $peer_i$.

Begin

Step1: *Pi* Sends a request Req_i^F for a file *F* to the superpeer *Sup*(*i*).

Step2: *Sup(i)* forwards the request to other superpeers.

Step3: *Sup(i)* Receives a list of candidate peers that have the requested file Based on the *AB, Sup(i)* select a set of the most reputable peers *Pj* such that $AB_j \geq$ *Threshold, Threshold value* is a parameter set by the system.
And then *Sup(i)* randomly selects one peer *Pj* from the candidate peers to be *(provider peer)*, and sends the response to *Pi (requestor peer)*, Res_i^F.

Step4: *Pi* is connected directly with the *Pj*, and sends the download request Req_{ij}^F

Step5: *Pj* responses with the requested file *F* (*Pi* download the requested file).

Step6: When the requested file has been downloaded, *Pi* assesses transaction process And sends the feedback (AF_{ij}) to its *Sup(i).*

Step7: *Sup(i)* updates reputation information of *Pi* in the following scheme

IF $AF_{ij}=1$ $SD_i = SD_i+1$
ELSE $UD_i = UD_i+1$

And then *Sup(i)* send the appreciation value (AF_{ij}) to the *Sup(j)*

Step8: *Sup(j)* updates reputation information of *Pj* in the following scheme

IF $AF_{ij}=1$ $SU_j = SU_j+1$
ELSE $UU_j = UU_j+1$

Once reputation information of *Pj* has been updated, AB_j will be updated in following scheme:

$AB_j = (SU_j - UU_j) / (SU_j + UU_j) = (SU_j - UU_j) / U_j$ IF $U_j \neq 0$
$AB_j = 0$ Otherwise

Step9: Initialize a new request, and go to step1

End

The sequence of these steps can be explained in figure (4).

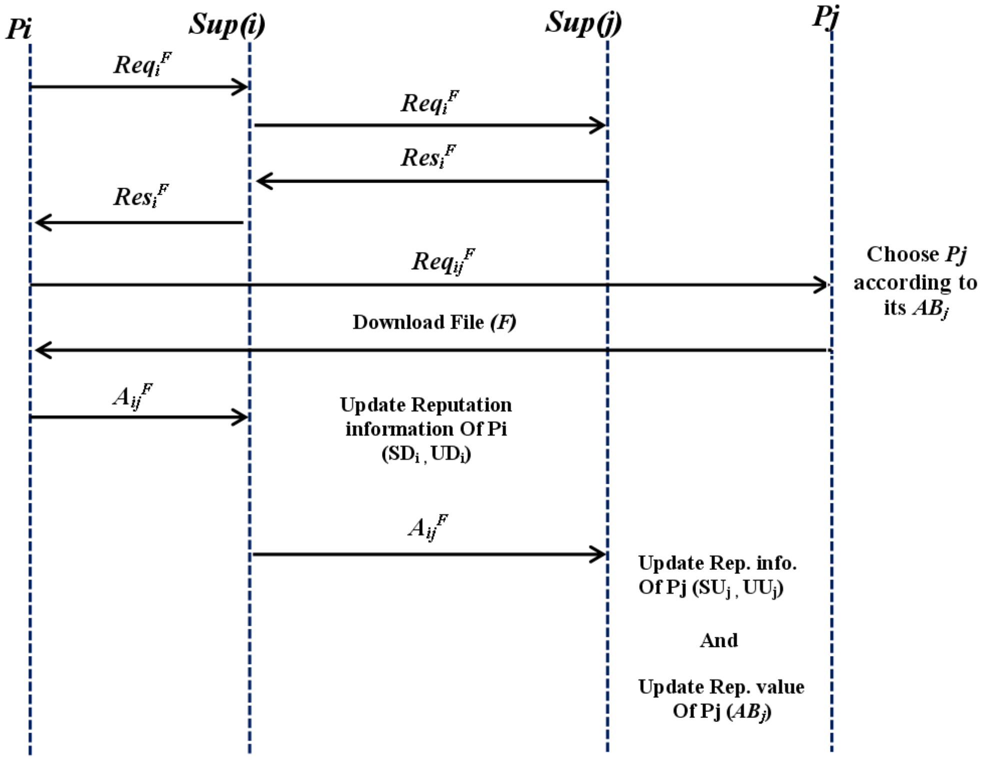

Pi
Sup(i)
Sup(j)
Pj
Req_i^F
Req_i^F
Res_i^F
Res_i^F
Req_{ij}^F
Choose *Pj* according to its AB_j
Download File *(F)*
A_{ij}^F
Update Reputation information Of Pi (SD_i , UD_i)
A_{ij}^F
Update Rep. info. Of Pj (SU_j , UU_j)
And
Update Rep. value Of Pj (AB_j)

Figure (4). Algorithm Steps

7. Practical Implementation of the Simulator

In this section, we will formally present and describe our proposed simulator. We implement this simulator using C# 2008 (In .NET Framework). This language has been chosen due to the facilities that it provides for drawing graphics in effective manner, as well as it allows creating an implicit database in collaborative with SQL Server (2005/2008). Finally, it provides an effective way for database connection and data access.

The main purpose of designing this simulator, is to understand how trust and reputation management applied in partially decentralized p2p systems. Trust and reputation managements are *complex, difficult, and much cost* to be applied in the real world applications; this is another reason for designing this simulator. The simulator can be considered as a guideline for researchers to work in this subject, and to apply trust concept in the real world applications.

Figure (5) explains the simulator interface, which contains two white areas, the top area will be used for displaying *(PD) network topology* (that is designed in the development process), and the bottom area will be used for displaying *Control messages.* Each superpeer in the designed *network topology* has two reputation tables illustrated in figure (6) and figure (7). (Sup_TDTable, and Superpeer_Rep Table). Sup_TDTable contents are (Local Peers ID, Rep_Info(s)), and Superpeer_RepTable contents are (Local Peers ID, Rep_value). These tables contain initial reputation (information/values) of the peers.

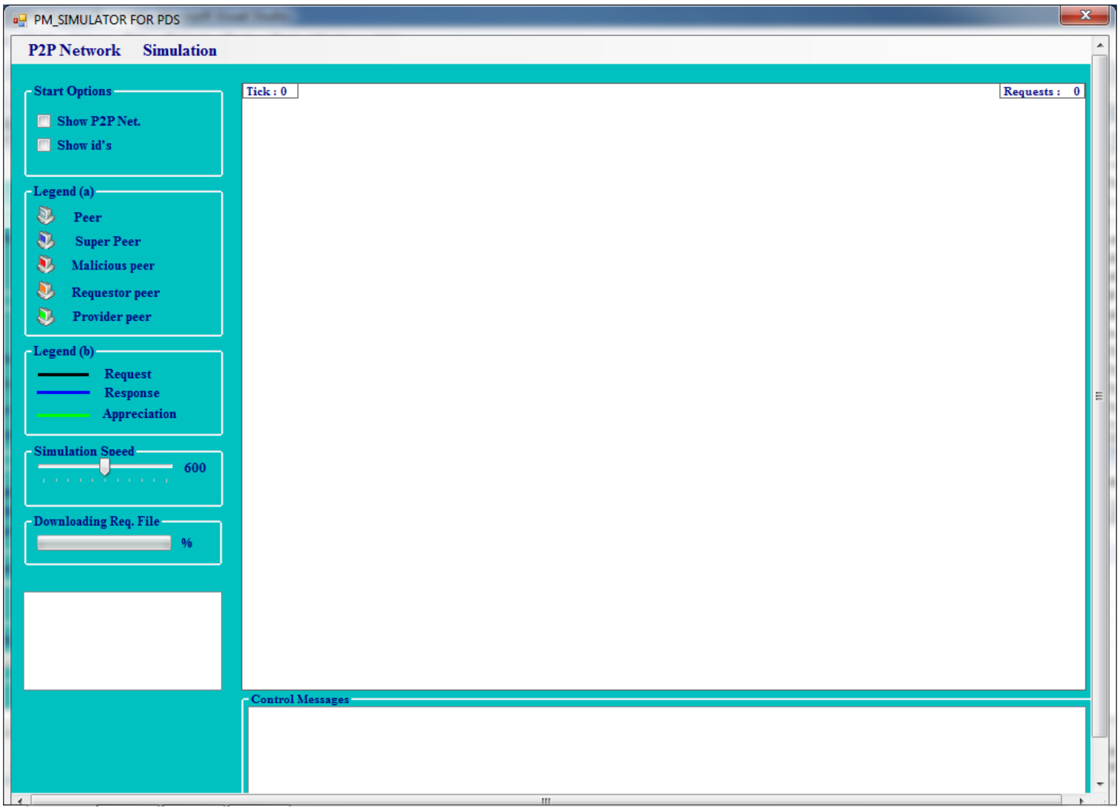
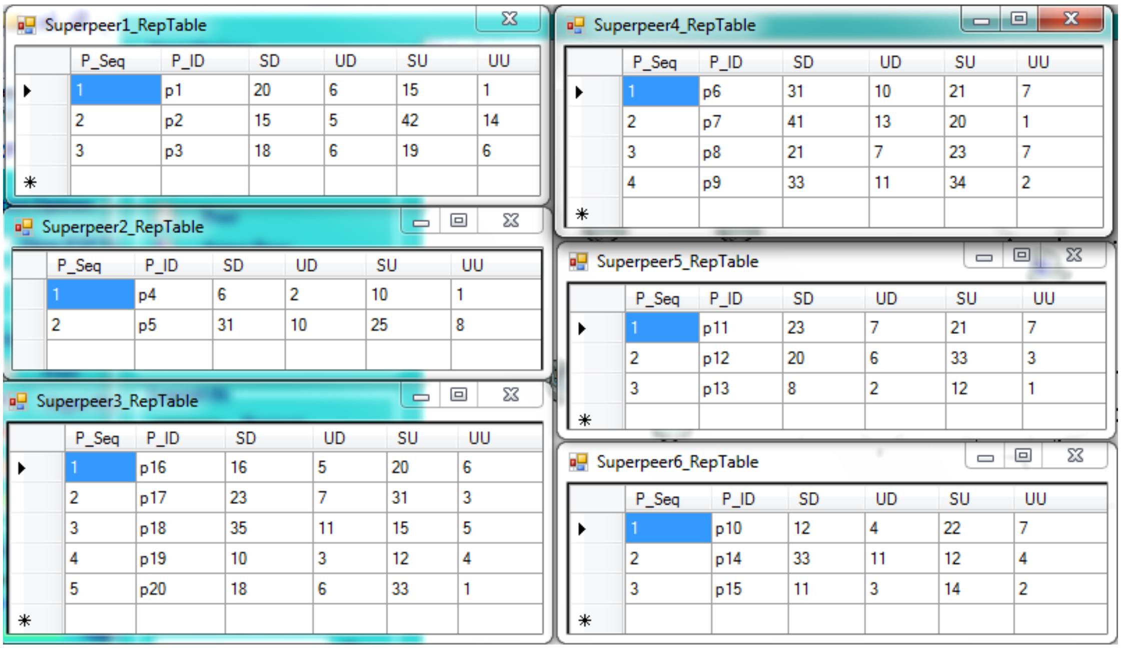


Figure (5). Simulator Interface



Superpeer1_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p1	20	6	15	1
2	p2	15	5	42	14
3	p3	18	6	19	6

Superpeer2_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p4	6	2	10	1
2	p5	31	10	25	8

Superpeer3_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p16	16	5	20	6
2	p17	23	7	31	3
3	p18	35	11	15	5
4	p19	10	3	12	4
5	p20	18	6	33	1

Superpeer4_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p6	31	10	21	7
2	p7	41	13	20	1
3	p8	21	7	23	7
4	p9	33	11	34	2

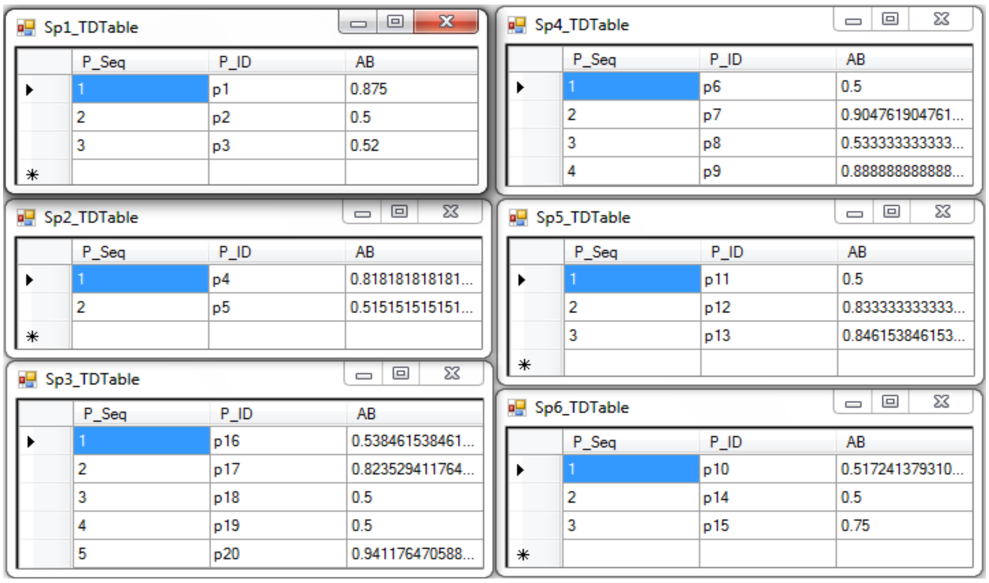
Superpeer5_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p11	23	7	21	7
2	p12	20	6	33	3
3	p13	8	2	12	1

Superpeer6_RepTable

P_Seq	P_ID	SD	UD	SU	UU
1	p10	12	4	22	7
2	p14	33	11	12	4
3	p15	11	3	14	2

Figure (6). Tables that contains Reputation Information



Sp1_TDTable

P_Seq	P_ID	AB
1	p1	0.875
2	p2	0.5
3	p3	0.52

Sp2_TDTable

P_Seq	P_ID	AB
1	p4	0.818181818181...
2	p5	0.515151515151...

Sp3_TDTable

P_Seq	P_ID	AB
1	p16	0.538461538461...
2	p17	0.823529411764...
3	p18	0.5
4	p19	0.5
5	p20	0.941176470588...

Sp4_TDTable

P_Seq	P_ID	AB
1	p6	0.5
2	p7	0.904761904761...
3	p8	0.533333333333...
4	p9	0.888888888888...

Sp5_TDTable

P_Seq	P_ID	AB
1	p11	0.5
2	p12	0.833333333333...
3	p13	0.846153846153...

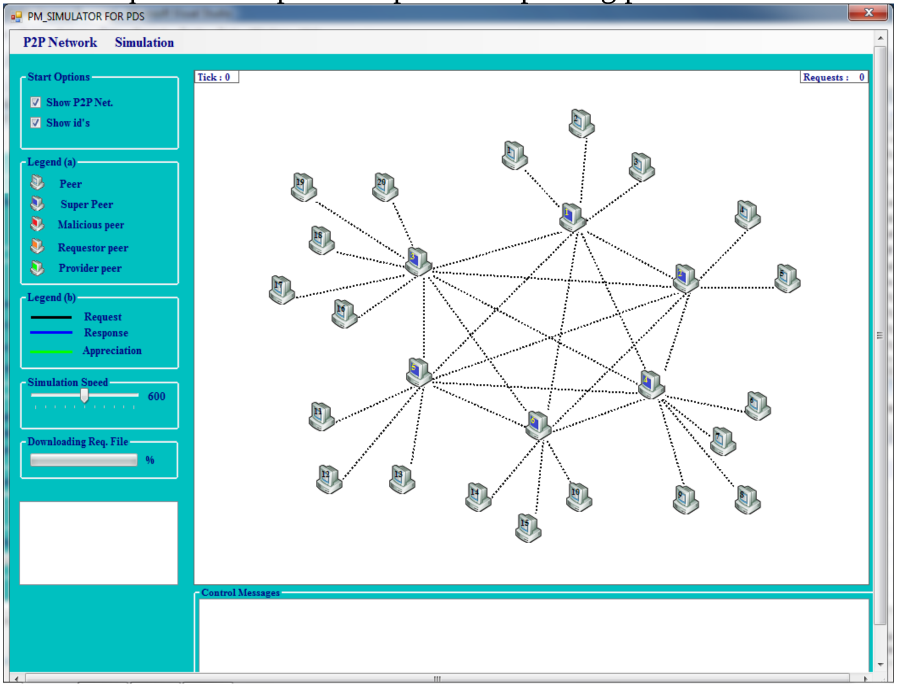
Sp6_TDTable

P_Seq	P_ID	AB
1	p10	0.517241379310...
2	p14	0.5
3	p15	0.75

Figure (7) – Tables that Contain Reputation Value
(*AB- Authentic Behavior*)

To start simulation, the start option (checkboxes) should be pressed to show the (PD) network topology that consists of *sixth superpeers* and *twenty peers*. As illustrated in figure (8) the next step is to press Start from the simulation menu. When the simulation starts, the proposed algorithm will be executed. In each iteration, one of the peers is selected to be requestor peer. This peer sends request to its superpeer which is forward the request to others.

Once the requestor peer receives ID of the provider peer, requestor peer is connected directly to the provider peer on the behind of its superpeer, and then download the requested file. After downloading process has been achieved, requestor peer sends the feedback (appreciation) to its superpeer. Based on the feedback value, the superpeer of the requestor and provider perform updating process.



PM_SIMULATOR FOR PDS

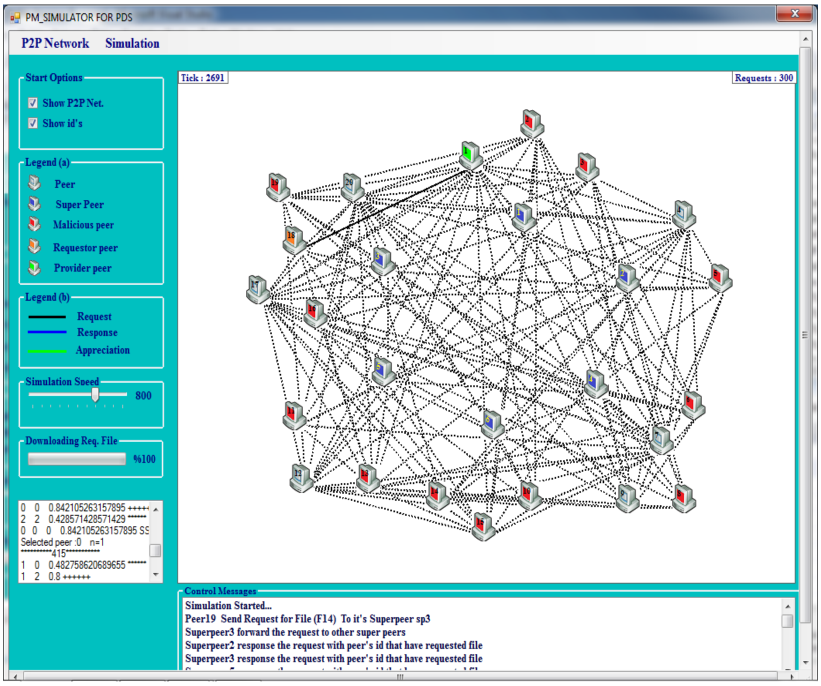
Start Options
☑ Show P2P Net.
☑ Show id's

Tick : 0

Requests : 0

Figure (8). The Designed P2P Network

Peers with low reputation value, won’t be selected as provider peer (green peer). Those peers can only download (files) from the other peers. Those peers also have a few opportunities to build its reputation. In each iteration, the selected available peers (that have the requested file), and their reputation value (*AB*) is very low (low reputable peers) will be identified as malicious peers (Red peers), and prevented from the contribution to the system. Figure (9) illustrates the identification of malicious peers (after 300 requests).



☑ Show P2P Net.
☑ Show id's

Tick : 2691

Requests : 300

Simulation Speed: 800

Downloading Req. File: %100

```
0  0  0.842105263157895 +++++
2  2  0.428571428571429 ******
0  0  0  0.842105263157895 SS
Selected peer :0   n=1
**********415************
1  0  0.482758620689655 ******
1  2  0.8 ++++++
```

```
Simulation Started...
Peer19  Send Request for File (F14)  To it's Superpeer sp3
Superpeer3 forward the request to other super peers
Superpeer2 response the request with peer's id that have requested file
Superpeer3 response the request with peer's id that have requested file
```

Figure (9). Malicious Peers’ Identification

8. Results

We use the following initial parameters to execute the proposed algorithm:

- Simulate a system with 6 superpeers, and 20 peers.
- The number of files is 510 (distributed among peers).
- 60 % of the peers are malicious.
- Initial reputation values illustrated in figure (6).
- The number of requests is distributed randomly among peers.

Based on the above parameters, the proposed algorithm has been executed. After 300 requests, the reputation values were taken from the tables (illustrated in figure (6) and figure (7)), and then viewed these values on charts. The authentic behavior of peers is illustrated in figure (9). In this figure two selection mechanisms have been considered (reputation based selection and random based selection). In reputation based selection, the most reputable peers selected to be file providers, whereas peers with low reputation values are not selected as provider peers. In random based selection mechanism, all peers have the same opportunities to be selected as provider peers.

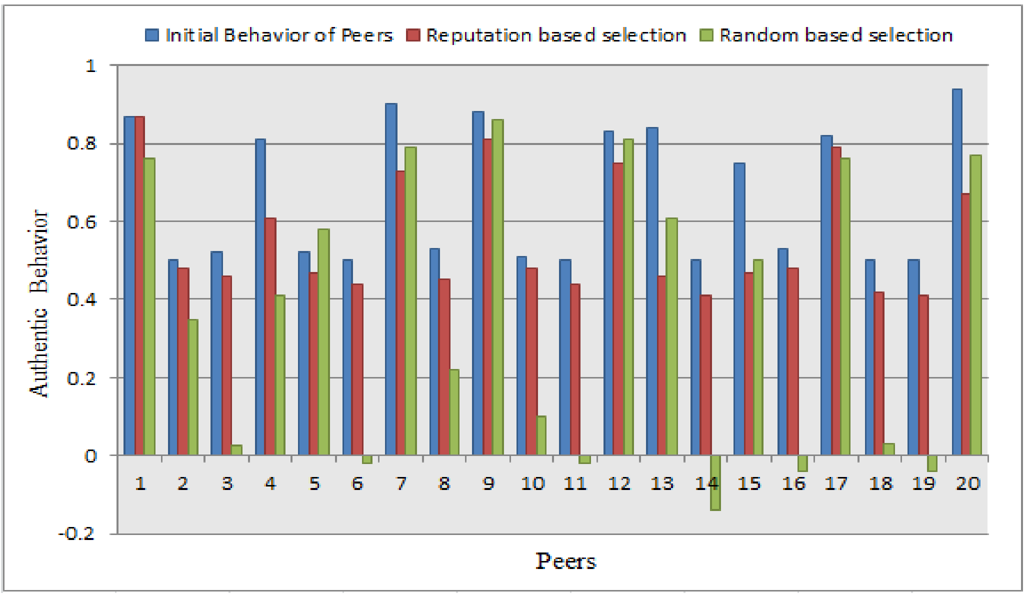


Figure (10). Authentic Behavior of Peers

As we can see in figure (10), in random based selection, the malicious peers' behaviors are decreased continuously while the simulator is running, whereas in reputation based selection the malicious peer behavior has been identified (not to be selected as a provider peer), therefore its behavior has not decreased continuously.

Figure (11) illustrates malicious uploads. In reputation based selection malicious uploads are decreased (prevents malicious peers' uploading and allows good peers' uploading), while in the random based selection malicious uploads increased (allows good/malicious peers' uploading).

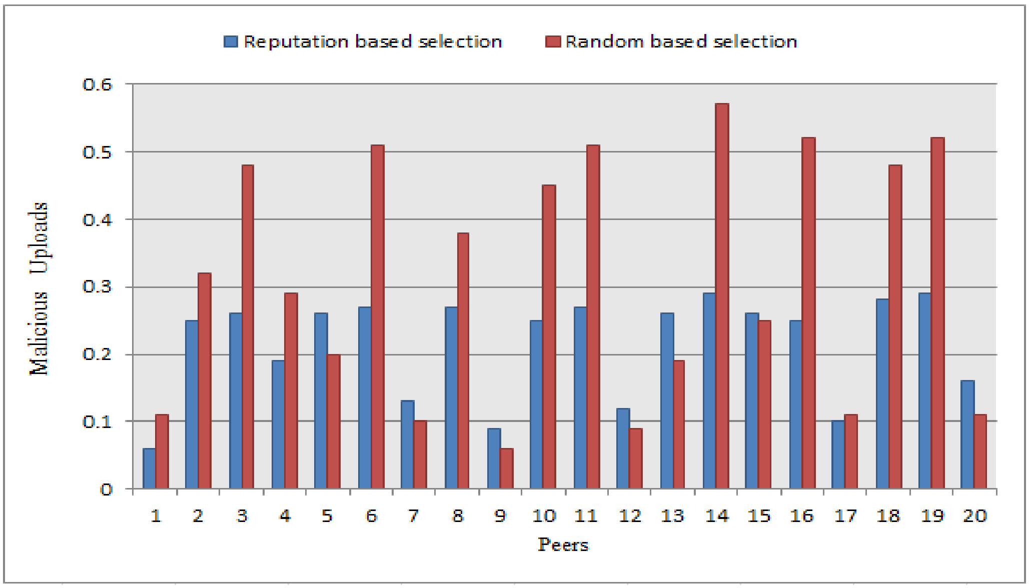


Figure (11). Malicious Uploads

Figure (12) illustrates the load shared among peers. In the reputation based selection, the most reputable peers (p1, p4, p7… etc.) have a high load share value, whereas malicious peers have very low load share value (this means that malicious peers do not perform any uploading process). In the random based selection mechanism, all peers have a convergent load share values (all peers have the same opportunities for uploading files).

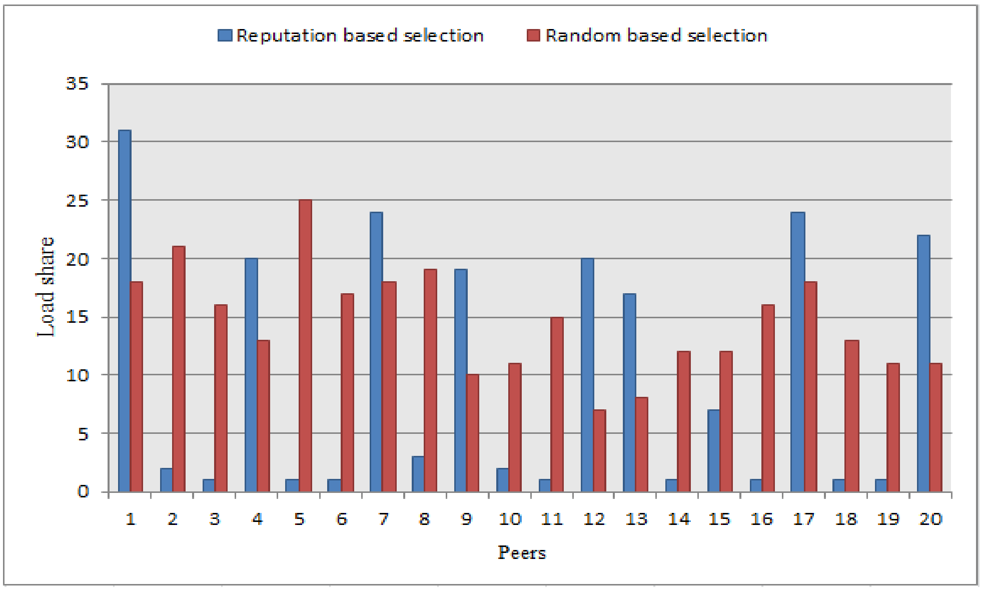


Figure (12). Load Share

Finally, figure (13) illustrates peer satisfaction. In reputation based selection, peers satisfaction is increased (only authentic files have been uploaded). In random based selection, peers satisfaction is decreased (malicious files also have been uploaded).

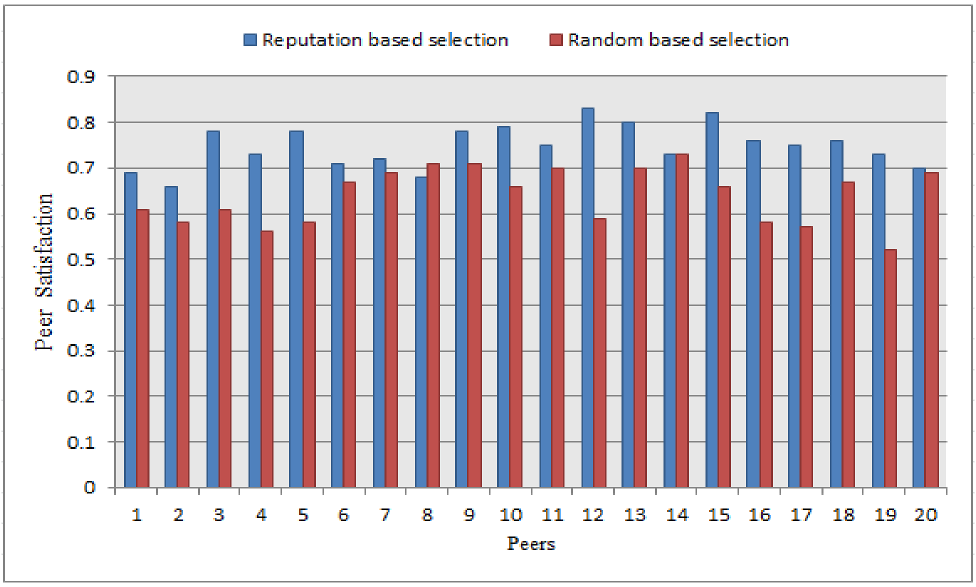


Figure (13). Peers' Satisfaction

9. Conclusions and Future work

A number of network simulators can be found nowadays, allowing us to test low level communication protocols. But, there is a lack of a simulator aimed to apply trust and reputation concept on the partially decentralized P2P file sharing. In this paper, we proposed a new *trust model*; also we build a new *simulator* for partially decentralized P2P file sharing systems. This simulator is one of the first simulators in these characteristics for (PD) P2P file sharing systems. In this work, we designed and developed a virtual (PD) P2P network topology, and explained how a trust and reputation concept can easily be applied to this kind of systems.

Practical results show that the proposed algorithm able to identify malicious peers effectively, increase peers' satisfaction; decrease inauthentic uploads and distributes the load share between the most reputable peers. Also, we made a distinction between reputation based and random based selection. The constructed simulator executes the proposed algorithm effectively and displays the designed P2P network in efficient manner.

For future work, some improvements and enhancements could be applied to this simulator. For instance, we are planning to add the ability to join (new peer) and leave (existing peer). Also, other trust components can be added to the proposed algorithm (such as credibility factor and context based factor), credibility factor helps to identify liar peers (that return wrong feedback), and context based factor allows us to compute trust value based on the multiple context.

REFERENCES

[1] Abdul-Rahman, A., Hailes, S., 2000: “Supporting Trust in Virtual Communities”. IEEE Computer Society.

[2] EBay, www.ebay.com.

[3] Gambetta D., 1990. “Trust: Making and Breaking Cooperative Relations”, Gambetta, D (ed.). Basil Blackwell. Oxford.

[4] Garbacki P., 2003, “Applying the Super-Peer Concept to Existing Peer-to-Peer Networks,” Tech. Rep. PDS-2003-010, Delft University of Technology.

[5] Hei, X., Liang, C., Liang, J., Liu, Y., Ross, K.W. 2006: A measurement study of a large-scale p2p system”. International World Wide Web Conference.

[6] Kamvar S. D., M. T. Schlosser, and H. G.-Molina, May 2003, “The Eigentrust algorithm for reputation management in p2p networks,” in Proceedings of the 12th International Conference on World Wide Web (WWW).

[7] Lee S., Sherwood R., and Bhattacharjee B., 2003, “Cooperative peer groups in nice,” in Proc. IEEE Conf.

[8] Mekouar L., Iraqi Y., and Boutaba R., 2009, “Handbook of Peer-to-Peer Networking”, chapter Reputation Management in Peer-to-Peer Systems: Taxonomy and Anatomy, Springer.

[9] Mekouar L., 2010, ”Reputation-based Trust Management in Peer-to-Peer File Sharing Systems” PhD Thesis, Canada.

[10] Mekouar L. , Iraqi Y., and Boutaba R., 2006, “Peer-to-Peer Most Wanted: Malicious Peers,” Computer Networks Journal, vol. 50, no. 4, pp. 545–562.

[11] Mekouar L., Iraqi Y., Boutaba R. , November 2004: “A Reputation Management and Selection Advisor Schemes for Peer-to-Peer Systems”, in The 15th IEEE International, USA.

[12] Momani. M. PhD thesis, July, 2008, ” Bayesian Methods for Modeling and Management of Trust in Wireless Sensor Networks”, Univ. in Sydney.

[13] Xiong L. and Liu L., July 2004, “Peertrust: Supporting reputation-based trust in peer-to-peer communities,” IEEE, no. 7, pp. 843–857.

[14] Zhengqiang Liang and Weisong Shi. , 2005. “Pet: A personalized trust model with reputation and risk evaluation for p2p resource sharing”, in System Sciences.

[15] Zhao H., and Li X., 2008, “H-Trust: A robust and lightweight group reputation system for p2p desktop grid,” IEEE ICDCS, Beijing, China.

[16] Zhao H., and Li X., 2009, “Vectortrust: The trust vector aggregation scheme for trust management in peer-to-peer networks”. In (accepted) The 18th International Conference.

[17] Zhou R. and Hwang K., May 2007. “Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing”. IEEE Transactions on Parallel and Distributed Systems, 18(4):460–473.