Equilibrium-based voting

A strategy for electing service providers in P2P E-learning

Eid, S.; Kismihók, G.; Mol, S.T.

DOI
10.1109/ITHET46829.2019.8937337

Publication date
2019

Document Version
Final published version

Published in
ITHET 2019: 2019 18th International Conference on Information Technology Based Higher Education & Training

Equilibrium-Based Voting : A Strategy for Electing Service Providers in P2P E-Learning

Shereif Eid
Learning and Skills Analytic Group
Leibniz Center for Science and Research (TIB)
Hannover, Germany
shereif.eid@tib.eu

Gábor Kismihók
Learning and Skills Analytic Group
Leibniz Center for Science and Research (TIB)
Hannover, Germany
gabor.kismihok@tib.eu

Stefan T. Mol
Amsterdam Business School
University of Amsterdam
Amsterdam,The Netherlands
S.T.Mol@uva.nl

***Abstract*—Filtering trusted learning resources is paramount for a successful P2P e-learning future. Unlike centralized systems which feature a centralized authority that maintains reputation values of all users in the system, in decentralized P2P networks, the popular approach is using a decentralized mechanism to achieve global estimates of peers' reputations. While decentralized reputation systems solve part of the problem, there is still the well-known problem of manipulating reputation systems and voting mechanisms to achieve selfish objectives or satisfy malicious intent. In the context of P2P e-learning, interacting with untrustworthy service providers, i.e. manipulative or cheating nodes, may suppress the e-learning process. We approach this problem by presenting an upgrade of reputation-based voting mechanisms, called equilibrium-based voting (EquiVote).We show by empirical analysis that our approach achieves verification and correctness of erroneous reputation values, caused by pre-election manipulation activities, at the time of electing learning service providers in P2P e-learning environments.**

Keywords— P2P E-Learning, Reputation and Trust, Consensus, Multi-Agent Systems

I. 1 INTRODUCTION

'E-learning', 'online learning', 'online collaborative learning' are terms which are interchangeably used nowadays to refer to learning activities which take place through web-enabled technology such as social networking sites, online learning environments, P2P networks, or even mobile-based applications [1]. The rise in popularity of these tools among digital age learners, and the pressure they impose on the educational community is creating an unprecedented transmission from bureaucratic into loose, democratic, and hybrid educational paradigms which integrate formal and informal learning methods [2-4]. Therefore, e-learning tools now form the backbone of new dynamic learning models which foster collaboration and student engagement, and enhance knowledge acquisition through online channels .In fact, many nonconventional models of learning that are based on enhanced technology have lately been developed which all revolve around encouraging students to adopt a proactive behavior, i.e. to be active participants rather than merely being passive recipients [4]. Some models in practice have introduced significant changes to the educational process, such as the flipped classroom model, wherein tutoring takes place outside classroom hours through social media [5].Other models have extended traditional personal learning environments, used by many formal institutions nowadays such as blackboards, to integrate informal social learning spaces for students to collaborate with online peers and aggregate information across social networks [2]. These modern, decentralized modes of learning encourage students to seek collaborative environments to leverage collective intelligence and to use the experience of peers. Moreover, modern approaches for e-learning employ autonomy and artificial intelligence by integrating software agents with intelligent tutoring systems. Software agents can act as personalized assistances and social companions, and they are able to collaborate to solve complex problems such as filtering learning resources according to learners' preferences. In P2P e-learning, software agents act on behalf of human users and they learn about their interests and preferences [6-10]. More importantly, they act as a mediator between different users to preserve users' privacy and protect their sensitive information from being revealed [11].

The architectural basis of agent-based distributed e-learning in general is a P2P networking architecture because P2P networks are collaborative in their nature. In P2P e-learning, different educational stakeholders collaborate, and they interact in an asynchronous manner. One of the most challenging problems in P2P e-learning is filtering trusted content and choosing among learning service providers. In P2P e-learning, communication is unstructured, asynchronous, and massive, and moreover, agents deal with 'total strangers' [12] on regular basis, i.e. those that they have never met in the past. Moreover, P2P networks are usually open; new peers usually enter or leave the network, and while agents share a common collaborative goal there is uncertainty about their intent. For example in P2P e-learning societies, peers may register as either service providers or service clients depending on their competencies [11]. Therefore, some learning service providers may seek publicity by gaining more transactions while not holding the required qualifications to provide the service. This is generally a problem in agent-based P2P networks, for example in e-commerce getting more transactions means engaging in more trade deals and making more profit [12]. Therefore, myriads of works in collaborative multi-agent systems (MAS) and P2P networks present various models and implementations of voting mechanisms that are coupled with decentralized reputation systems to address two main problems: 1.The heterogeneity of resources which are difficult to map to users' interests and preferences, given the large number of participants 2.The great uncertainty involved with respect to interactions in such massive and fully distributed virtual environments where agents cannot fully confirm the intent of their counterparts or validate the quality of the services they provide.

978-1-7281-2464-3/19/$31.00 ©2019 IEEE

Unfortunately, studies have shown that building trust, through decentralized modes, is still not sufficient to treat the aforementioned problems [12-15]. The intelligence embedded in software agents motivates those with bad intentions to perform strategic attacks to manipulate the outcomes of reputation systems and voting mechanisms. Cheating agents usually seek to boost their reputations or the reputation of their friends (in case of coalitions).This problem has been rising lately in many online reputation systems where reputation values are used as means of voting for quality service providers. In P2P e-learning, fraudulent manipulation of reputation values and/or user votes means dealing with untrustworthy learning resources which may suppress the e-learning process. Indeed, this paper presents a novel reputation-based voting mechanism based on two equilibrium reaching mathematical manipulation strategies for conducting positive and negative votes. Our mechanism is supported by consensus reaching protocols in the context of service selection in P2P e-learning networks. We show by empirical analysis that our approach achieves the following:

1. Verification and correctness of cumulative reputation values gathered based on services provisioned in the past.
2. Strategy validation by demonstrating how equilibrium is achieved when positive and negative votes are fairly distributed among voters.

In the next section we provide an overview of the technical background and related work. Section III provides a case study which explains the basis of our work. Sections IV and V describe our approach, our main arguments, and reliability requirements. Sections VI and VII present our simulation results, analysis, and evaluation. Finally, section VIII concludes the paper.

II. 2 Background and Related Work

Voting mechanisms are used to facilitate collective decision making in online environments and to filter resources based on their qualities. Voting is based on the reputation of resources or service providers with respect to services provisioned in the past. In online environments they are often integrated with reputation systems such that entities with the highest number of ratings are the most reputable, thus, they are the most recommended. Filtering resources based on reputation values assigned in the form of user ratings is common in online systems and social networking sites such as Amazon, EBay, Facebook, and YouTube. In such systems, users provide star ratings, likes, or share contents based on their personal experiences, and these ratings are aggregated into cumulative index of its reputation. This is also the case in some e-learning environments such as the PLEM architecture [17] where total votes are simply the total number of positive user actions associated with a certain learning resource. For example, in PLEM, such actions may include commenting, liking, rating, and sharing. [17]. In the end, total votes are counted and learning resources are ranked based on the cumulative number of positive actions [17]. What the aforementioned examples have in common is that reputation values are stored in a centralized manner, and moreover, trust is based on a subjective decision making such that users simply choose resources with the highest number of votes.

However, in P2P and distributed systems, there is no centralized authority that could store reputation values of all users in the network or, for example, manage a centralized voting mechanism. Moreover, agent nodes are bounded in the number of direct interactions they can have with other nodes in the network. When it comes to agent-based e-learning, some work in the field (see for example [9] and [18]) relies on the intelligence of autonomous agents to learn about their human owners' preferences and interests to predict future decisions. In the work of [18] for example, in the initial phases of their presented voting mechanisms, students manually vote for courses while software agents learn their selection preferences and use these in later stages to predict future course selections or course cancellation. Other researchers have used a topology-based approach to filter learning resources based on user preferences, for instance in the proposal of [11], software agents form an overlay network of users with similar properties, i.e. similar interest. Note that in the latter example no reputation mechanism was used, therefore, the authors based their work on the general assumption that all nodes in the network exhibit 'honest behavior' [11].

The goal of decentralized reputation systems therefore is gathering objective measures of the trustworthiness of participants in distributed networks .Reputation systems exploit the property of 'conditional transitivity' of trust [19], which implies that the more aggregated peer opinions about the trustworthiness of an agent, the higher the probability that an agent is trustworthy in future interactions. 'Trust aggregation' is another term that is commonly used to refer to the dissemination of reputations, witness information, or recommendations. In P2P and multi-agent societies, trust can be defined as the evaluation of an agent with respect to a service provisioned in the past [20]. These evaluations help predict agents' future behavior, thus, they help agents decide who to choose for future interactions [20]. Reputation systems aggregate evaluations and feedback information from participants to facilitate the sharing of experiences and create reciprocal relationships between agents. Consequently, reputation is an important source of information for agents. Indeed, where direct experience has not yet been acquired, trust is primary based on others' experiences, in other words, referrals from others [21].In this fashion, reputation systems overcome the dependability on direct interactions.

Decentralized reputation systems are particularly intended for P2P and distributed networks wherein social interactions among large numbers of agents feature great uncertainty. Due to the large numbers of agents in P2P networks, it is more likely that agents will interact with those that they have never met in the past. While agents collaborate to reach a common goal, such as a particular learning objective, they usually have their own selfish intent. In the best scenarios, agents do perform honestly within P2P networks yet merely cheat on occasional situations. In the

worst cases, agents only join P2P networks to execute a malicious plan. In reputation systems, dishonest agents aim to increase their reputation values or the reputation of their friends (for example, in the case of coalitions), or degrade the reputation of others. In voting mechanisms the same behavior applies and dishonest agents seek to manipulate the outcomes of the election process for their own benefit. The popular approach in decentralized reputation systems is maintaining a local value which reflects agent j's trustworthiness in the eyes of agent i , this could be based on any trust model. Then agent i attempts to achieve a global reputation value of agent j which reflects the best estimate of the average reputation of j among all participants. This is the case in EigenTrust [15] for example, and its improved versions [16], and in consensus reaching reputation systems such as gossip-based trust presented in [12].EigenTrust relies on a probabilistic model where agent i iterates the network with a probability r_{ij} for the reputation of agent j, and after a number of n iterations, it will most likely arrive at a reputable node.

Consensus reaching algorithms, in particular gossip-based trust has proved to be more robust to faults, with faster (exponential) conversion rates, thus, more efficient in aggregating trust values from all nodes in the network. This means that gossip-based trust is faster and more accurate in achieving global reputation values. Both of these approaches, EigenTrust and gossip-based trust normalize local reputation values to the range [0,1] to limit manipulating reputation values to very high or very low figures. However, [12] have shown through their test attack strategies that there exist mathematical-based manipulation strategies, which can possibly inflate reputation values yet keep them in the legal range [0, 1].

III. The Concept of Fluctuating Reputation – A Case Study

Equilibrium-based voting is based on the idea that , given an online social community ,at the time of running elections to select among service providers , or specific resources , the publicity of these resources change , and therefore there reputation values fluctuate. For instance, consider any digital learning culture which provides educational or tutoring videos such as YouTube Learning or TED, we assume that users vote by liking the content. Let's say that at any time t , the reputation ,r, of any content c is the difference between the number of likes and the number of dislikes. Moreover, we assume that we run an election process, for electing quality service providers in a certain educational context such as Modern History. If the election process starts at time t whereas it ends at time $t+1$, then we say that at time $t+1$ the new reputation of content c is r', and it could be the case that $r > r'$ or the vice versa for many reasons. We argue that r' is the corrected value of r, i.e. r' is the more accurate value for the reputation of content c , at least for the sole purpose of the elections. The case described in the previous example is simpler than the problem we intend to address because there is a central authority which stores reputation values and user votes. In our context , this is not the case in P2P e-learning environments , but in the next section we will describe our equilibrium-based voting mechanism which achieves two main properties : 1.Use the consensus phenomenon through running the Push-Sum gossip protocol to agree on a global reputation value for content c (or the service provider which provides the content).2.The aggregated reputation value is manipulated by two fair , equilibrium reaching mathematical strategies for conducting positive and negative votes.

IV. 3 Equilibrium-Based Voting

The goal of gossip-based algorithms is to aggregate values such as sums and averages and to compute functions out of those values in a decentralized way. The approach stemmed from the work of Frieze and Grimmet [22] where they demonstrated the telephone call problem. The problem is inspired by human social interactions when a group of people spread rumours across the same place. If we assume a uniform gossip, in each time step, a person who knows the rumour spreads it to another person chosen at random. The upper bound on the gossiping process is identified by the number of time steps required such that everyone knows the rumour.

The push-sum protocol, due to Kempe et al [23], inherits all the properties of such a uniform gossip, and is a favourable alternative to Laplacian-based consensus algorithms; in addition it has proved to be efficient. Its central notion is the aggregation of gossips from nodes in a network. As illustrated in algorithm (1), nodes aggregate pairs of sums and weights at every step time t, those pairs are used to compute a close estimate of the average of all the values stored at all nodes. Therefore, at any time t, a pair of a *sum* $s_{t,i}$ and a *weight* $w_{t,i}$ are buffered at each node i. The sum is initialized to $s_{0,i} := x_i$, and the weight is initialized to $w_{0,i} := 1$. At time 0, it sends the pair $(s_{t,i}, w_{t,i})$ to itself, and in each subsequent time $t > 0$, each node i follows the protocol given in Algorithm (1).

Algorithm 1 Protocol Push-Sum [23]
1: Let $\{(\hat{s}_r, \hat{w}_r)\}$ be all pairs sent to i in round $t-1$
2: Let $s_{t,i} := \sum_r \hat{s}_r$, $w_{t,i} := \sum_r \hat{w}_r$
3: Choose a target $f_t(i)$ uniformly at random
4: Send the pair ($\frac{1}{2}\, s_{t,i}, \frac{1}{2} w_{t,i}$) to $f_t(i)$ and i (myself)
5: $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step t

Prior to any interaction, agent i can run the push-sum algorithm for as many rounds as it wishes, and as it takes to achieve a certain level of accuracy. The level of accuracy is controlled through identifying a threshold for the aggregation error. The more accuracy claimed, the smaller should be the identified threshold, and the more steps required. The aggregation error is the distance between the local value and the global value computed at each round of running the protocol. Therefore, it is defined as the absolute difference between r_i and $\overline{r_i}$ as follows [12]:

$$\left| r_{ij} - \frac{1}{n} \sum_k r_k \right| \leq \varepsilon$$

Bachrach et al. [22] have built on the advantages of the push-sum to distribute reputation in decentralized reputation systems. In this case, the state value x_i can represent the local reputation value of any agent j. So for example, we can assume that $r_{ij}= x_i$ is the reputation of agent j stored by agent i.In other words, it is agent j's trustworthiness as inferred by agent i. In our context, regardless of the trust model used to infer trust, or the data management scheme that is used for achieving global reputation values, we use the push-sum gossip protocol as a temporary mechanism at the time of elections to distribute votes across the network. However, our mechanism is not decoupled from local reputation values whether they are acquired by each node based on direct experiences or recommendations from other agents in the network. Rather we use it to launch temporary elections at the time of selecting among learning service providers.

The main objective of our voting mechanism is to reach an agreement between all nodes i about the reputation of each node j at the time of running the elections. This is not different from traditional one dimensional voting systems discussed earlier where user ratings determine the ranking of products and services. Yet we modify the voting mechanism to suite the fully distributed architecture of P2P e-learning environments. In traditional systems, users know which service has gathered the most votes but in P2P networks, indeed, we need a reliable mechanism to allow any agent i to get the best estimate of the reputation of any agent j, or the best estimate of its ranking among other peers at the time of the election. In the same time, we keep in mind that this ranking should be linked with a high probability to the reputation of agent j with respect to services provisioned in the past. Therefore, any agent i who needs to identify the best learning service providers in the system launches the election process through running the push-sum gossip protocol.

V. Reliability Requirement :Modelling Consensus

In every step of the gossip aggregation, the state of the process, let it be k, is updated to be $k + l$,where l is the number of new agents who know the rumour, and there is an upper bound on n identified by :

$$S_n = \min\{i: Y_i = n\}$$

Where Y_i is the state of the process after step i such that $Y_0 = 1$, and S_n is the upper bound on the number of steps required until every agent knows the rumour. There are two important behaviours in this process [22]; an asymptotic behaviour, and a deviation estimate at the final step $Y_i = n$ .The former qualifies this process to be a consensus process. The latter can be used to identify the acceptable accuracy required to end the process. In this sense, gossip-based algorithms are robust to faults [12, 23] because they self-stabilize their aggregated values. In other words, the effect of faulty nodes and erroneous propagations are masked by the interactions of valid ones. They are scalable because they need not be managed by a central authority. They are stable against stress and disruption, and they are simple to implement [23].

In agent society, consensus is defined as reaching an agreement about a quantity that interests all agents and which has a dependency relation to the state of each agent [24]. A consensus protocol defines the rule of interaction [24], and is required to fairly reach the state of consensus. Consensus is highly important in the coordination of cooperative work in MAS [25]. As the aforementioned definition indicates, this general agreement is required to be reached particularly on the coordination data, which means that the coordination data must eventually asymptotically converge to an agreed on value [25]. To add more highlight, according to Ren et al [25], researchers have summarized consensus protocols in discrete time. What we focus on here is that the general agreement is said to be achieved between a group of agents if $||x_i - x_j|| \to 0$ as $t \to \infty, \forall i \neq j$[31]. Where x_i and x_j are the states of agent i and agent j.We refer to this as the consensus requirement. The state of each agent is represented by the data to be coordinated between agents.

Assuming that it is efficiently implemented, and under normal conditions, the push-sum protocol guarantees the efficient aggregation of values from all nodes in a network. This has been demonstrated in previous experiments. Analysis has also shown that the push-sum has an exponential conversion rate [12, 23]. However, the error threshold plays a crucial role in identifying the rate of conversion to the global reputation value. The rate of conversion in turn depends on the speed of diffusion of r_{ij} through the network.Therefore, the more accuracy requested, the more push-sum steps are required, therefore $U(n, \varepsilon, \delta)$ represents also an upper bound on consensus time; the number of time steps required for all nodes to reach a consensus agreement.

A. Voting Strategy

A number of strategies have been introduced in the literature for manipulating reputation or voting systems. These strategies were used to test reputation systems robustness to attacks or to identify vulnerabilities. It is known that normalizing local values to a narrow range limits the effect of manipulation attacks because they add a constraint on the outcome of reputation systems [15], for example, if the range is [0,1], as in our case , any manipulation attempt that pushes the final aggregate (global reputation value) beyond the range is easily detected [12,15]. Out of the strategies in the literature, Bachrach et al [12] have introduced a strategy which they call *StrategyII* .This strategy, *StrategyII*, is an elegant mathematical strategy because regardless of the inflation ratio the final aggregate is always in the range [0, 1]. We were interested to use *StrategyII* as a strategy for conducting positive votes for a specific learning service provider, given that we use a reputation-based voting mechanism. Many voting mechanisms only count or aggregate positive votes and discard negative votes from the final count. Other approaches consider no vote as a negative vote and for example assign a value of 1 to a positive vote

and a value of 0 to a negative vote such that no voting does not really affect the outcome of elections. While reproducing the simulation of *StrategyII*, we noticed that as the numbers of positive voters increase, the reputation of agent j is inflated to a large value. While it always holds that the reputation of agent j is still in the legal range [0, 1], the great increase in the reputation of agent j at the time of running the election is unrealistic from a probabilistic point of view. Therefore, we were interested to develop a strategy, which we call *StrategyIII* for reversing the inflations made by *StrategyII*, and to bring r_{ij} back to its normal value. The argument this paper makes is; within a decentralized reputation system, when positive and negative votes are fairly distributed among voters, if two manipulation strategies ,when simultaneously executed, feature the capability of achieving equilibrium, then they qualify to construct a seamless reputation-based voting mechanism with verification and correctness property such that ; after completing the election process, the new global reputation values are the best corrected estimates of reputations with respect to services provisioned in the past, and the effect of pre-election manipulation is reversed. Next, we describe both strategies, *StrategyII* and *StrategyIII* as follows

- *StrategyII* [12]: i^m ensures that r_{ij}, for any i, converges to a value that is at most 1. For specific proportion of time, the manipulator i^m sets the evaluation of r_{ij} to be 1. So, for each round $t = 1, \dots \dots . T$,i^m sets $s_{t,i^m} := w_{t,i^m}$.
- *StrategyIII*: for a specific proportion of time $t \leq T$, i^m updates $s_{t,i}$ to be $\frac{\sum_r s_r}{2^n}$, and n>0.In other words, at time t , i^monly aggregates a proportion of the sums buffered in the previous time $t-1$. Using this strategy, it always holds that $r_{ij} \in [0,1]$. More specifically, it always holds that $r_{ij} > 0$.

In our context , any node i that is conducting positive votes for agent j runs *StrategyII* for 100 % of the time , and on the contrary, if node i is negative votes for agent j , then it runs *StrategyIII* for 100% of the time, i.e. during the full duration of running the election process. Moreover, the algorithm stops at time t when the consensus requirement is satisfied and it holds that at any time $t' > t$,$\left|r_{ij} - \frac{1}{n}\sum_k r_k\right| \leq$ ε. Note that, in *StrategyIII*, the exponent n controls the intensity of this strategy. However, we prefer to leave it unidentified for freely controlling the intensity of *StrategyIII* for simulation purposes.

VI. Analysis

The first step towards our analysis of EquiVote is to validate the reliability of the simulation framework through measuring the performance of our implementation of the push-sum protocol. Our implementation of the push-sum shows very good results with respect to consensus and gossip requirements. Meaning that we satisfy the basic consensus requirement $||x_i - x_j|| \to 0$ as $t \to \infty, \forall i \neq j$. Figure (1) shows gossip stages required for conversion under different error thresholds. With respect to reliability, our results demonstrate faster conversion rates for some groups than some results presented in the literature [26]. Moreover, our results demonstrate the ideal scalability expected when executing a uniform gossip. In particular, our program is characterized by 0% conversion overhead for any number of participants. Indeed, in a real-life situation there could be communication overheads and the possibility of message loss. But for simulation purposes, the general performance and scalability results of the framework is ideal for experimentation, and is satisfactory to build on and proceed to simulate and analyze our voting strategies.

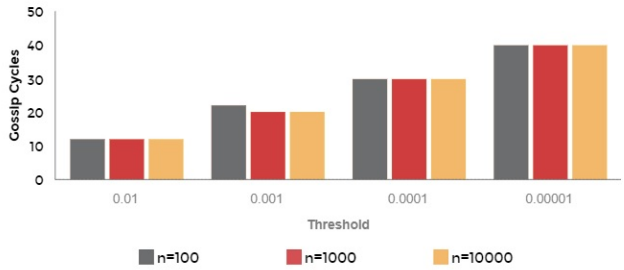

Fig. 1. Gossip cycles required for different error thresholds.

A. Manipulation Test

Both strategies, *StrategyII* and *StrategyIII,* have vigorous behaviours when used as manipulation strategies, i.e. when either one of them is disabled. However, this also depends on the number of voters (manipulators in this case). We tested both strategies in different settings; with different sizes of agent networks, variable error thresholds, and many different values for the number of positive votes. For example, for a network of size 100 agents , when 10% of the nodes adopted either *StrategyII* or *StrategyIII*, the aggregation error was approximately in the range $[0.23 - 0.29]$.A sample of the result of this experiment, in addition to the program interface, are illustrated in Figures (2) and (3). In the previous example, *StrategyII* caused an inflation with an aggregation error $+0.29$, while *StrategyIII* caused a deflation with an aggregation error -0.23.Note that in our simulation we chose n=2 , but higher values could increase the deflation effect such that the absolute difference between aggregation errors caused by both strategies diminishes.

However, the joint effect of both strategies nearly illustrates a mirror reflection if analysed with the signed value of the aggregation error as shown in Figure (4).

B. Strategy Validation – Simulation of EquiVote

For the purpose of our analysis, we launched many active sessions of the program. All sessions used a fixed number of agent nodes (100 agents for each). We grouped our program sessions into three groups, in the first group, voters only conducted positive votes using *StrategyII* , while in the second group, voters only conducted negative votes using *StrategyIII*. In the third group, positive and negative votes

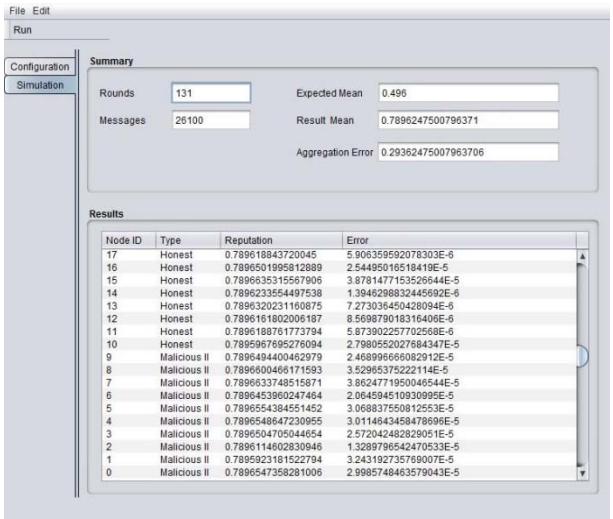

Fig. 2. StrategyII when adopted by 10% of the nodes, network of size 100 agents, threshold=1E-5.

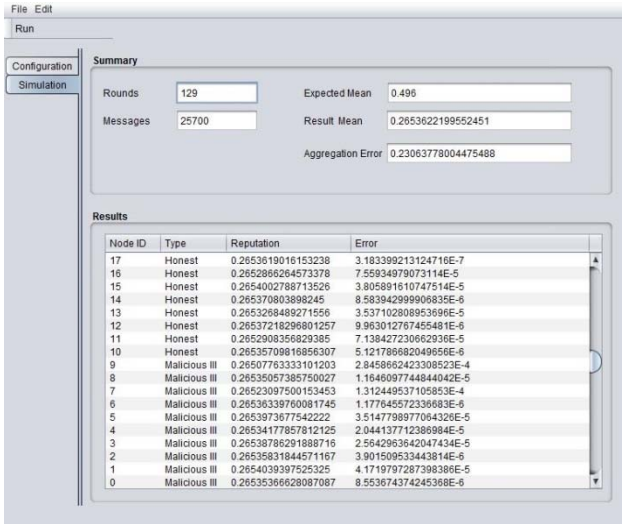

Fig. 3. StrategyIII when adopted by 10% of the nodes, network of size 100 agents, threshold=1E-5.

were fairly distributed among voters. In all three groups we used various numbers of voters and different error thresholds. However, at fixed interval of time steps we halted the voting protocol and recorded the aggregation error against the number of time steps during which voting was active. Finally, we measured the average aggregation error versus the number of time steps for all groups. A sample of the results is illustrated in Figure (5).Our experiment showed that, regardless of the error threshold or the number of rounds, when the number of positive votes is approximately equal to the number of negative votes there is a state of equilibrium, such that the effect on the reputation value r_{ij} is very slight.

VII. Evaluation

The argument this paper makes is that if two manipulation strategies are able to achieve equilibrium under a fair setting

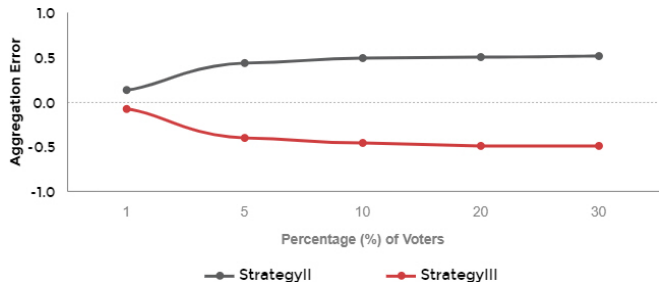

Fig. 4. StrategyII and StrategyIII when adopted by 10% of the nodes, network of size 100 agents, threshold=1E-5.

wherein positive and negative votes are equally distributed among voters , then both strategies can constitute a reputation-based voting mechanism with verification and correctness properties such that ; after completing the election process, the new global reputation values are the best corrected estimate of reputations with respect to services provisioned in the past, and the effect of pre-election manipulation is reversed .We have seen in our demonstration how manipulations can exhibit vigorous behaviours and cause great inflation or deflation of global reputation values. Yet we have also demonstrated the capability of EquiVote to reverse manipulation effects and bring global reputation values back to approximately a best estimate of their original values, given pre-election manipulation activities in the system.

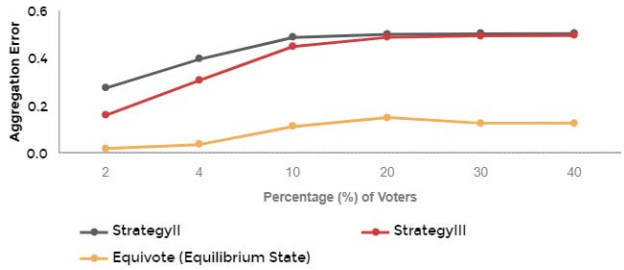

Fig. 5. Equilibrium-Based Voting in a network of 100 agents.

VIII. Conclusion

Achieving 'trust' among participants is not a matter of choice in a P2P e-learning environment. Yet it is a crucial requirement in a digital learning society which intends to empower learners and influence their future perspectives. This task in P2P networks is not a simple task due to the heterogeneity of resources and the large numbers of participants in such a huge virtual community. Despite the efficiency of some decentralized reputation systems in achieving global objective measures of peers' trust, the intelligence embedded in software agents enhance manipulation attacks performed by selfish or malicious nodes. Therefore, this paper has presented an upgrade of reputation-based voting mechanisms called equilibrium-based voting (EquiVote) which is based on two equilibrium reaching mathematical strategies. We have proved by

empirical analysis that our approach exhibits strong verification and correctness properties at the time of electing learning service providers. In particular, we have demonstrated how EquiVote exploits the consensus phenomenon to correct erroneous reputation values caused by pre-election manipulation activities. In future research, we plan to search for more strategies to explore the trusted boundaries of EquiVote within more vigorous manipulation activities.

ACKNOWLEDGMENT

The software simulation presented in this paper was developed at the School of Computing, Newcastle University, United Kingdom. The authors would like to thank Steve Riddle and Aad van Moorsel at Newcastle University for their input and discussions which helped in the creation of this paper.

REFERENCES

[1] J. L. Moore, C. Dickson-Deane, and K. Galyen, "e-Learning, online learning, and distance learning environments: Are they the same?," *The Internet and Higher Education*, vol. 14, no. 2, pp. 129–135, Mar. 2011.

[2] N. Dabbagh and A. Kitsantas, "Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," *The Internet and Higher Education*, vol. 15, no. 1, pp. 3–8, Jan. 2012.

[3] C. Greenhow and C. Lewin, "Social media and education: reconceptualizing the boundaries of formal and informal learning," *Learning, Media and Technology*, vol. 41, no. 1, pp. 6–30, Jan. 2016.

[4] D. Casanova, A. Moreira, and N. Costa, "Technology Enhanced Learning in Higher Education: results from the design of a quality evaluation framework," *Procedia - Social and Behavioral Sciences*, vol. 29, pp. 893–902, 2011.

[5] A. Wallace, "Social learning platforms and the flipped classroom," in *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, 2013, pp. 198–200.

[6] S. Hammami, H. Mathkour, and E. A. Al-Mosallam, "A multi-agent architecture for adaptive E-learning systems using a blackboard agent," in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 184–188.

[7] G. J. Gordon, "Agendas for multi-agent learning," *Artificial Intelligence*, vol. 171, no. 7, pp. 392–401, May 2007.

[8] A. Kerry, R. Ellis, and S. Bull, "Conversational Agents in E-Learning," in *Applications and Innovations in Intelligent Systems XVI*, T. Allen, R. Ellis, and M. Petridis, Eds. London: Springer London, 2009, pp. 169–182.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[9] S. Schiaffino, P. Garcia, and A. Amandi, "eTeacher: Providing personalized assistance to e-learning students," *Computers & Education*, vol. 51, no. 4, pp. 1744–1754, Dec. 2008.

[10] Y. Kim, A. L. Baylor, and E. Shen, "Pedagogical agents as learning companions: the impact of agent emotion and gender: Learning companions," *Journal of Computer Assisted Learning*, vol. 23, no. 3, pp. 220–234, Jan. 2007.

[11] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A Distributed Agent-Based Approach for Supporting Group Formation in P2P e-Learning," in *AI*IA 2013: Advances in Artificial Intelligence*, vol. 8249, M. Baldoni, C. Baroglio, G. Boella, and R. Micalizio, Eds. Cham: Springer International Publishing, 2013, pp. 312–323.

[12] Y. Bachrach, A. Parnes, A. D. Procaccia, and J. S. Rosenschein, "Gossip-based aggregation of trust in decentralized reputation systems," *Auton Agent Multi-Agent Syst*, vol. 19, no. 2, pp. 153–172, Oct. 2009.

[13] Q. Feng, Y. L. Sun, L. Liu, Y. Yang, and Y. Dai, "Voting Systems with Trust Mechanisms in Cyberspace: Vulnerabilities and Defenses," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1766–1780, Dec. 2010.

[14] Aameek Singh and Ling Liu, "TrustMe: anonymous management of trust relationships in decentralized P2P systems," in *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*, 2003, pp. 142–149.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," p. 12.

[16] S. Rao, Y. Wang, and X. Tao, "The Comprehensive Trust Model in P2P Based on Improved EigenTrust Algorithm," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2010, vol. 3, pp. 822–825.

[17] M. Amine Chatti, Anggraeni, M. Jarke, M. Specht, and K. Maillet, "PLEM: a Web 2.0 driven Long Tail aggregator and filter for e learning," *Int J of Web Info Systems*, vol. 6, no. 1, pp. 5–23, Apr. 2010.

[18] A. M. Aseere, E. H. Gerding, and D. E. Millard, "A Voting-Based Agent System for Course Selection in E-Learning," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Toronto, AB, Canada, 2010, pp. 303–310.

[19] A. Abdui-Rahman, S. Hailes, and S. Hailes, "A Distributed Trust Model," p. 13.

[20] Li Xiong and Ling Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[21] I. Pinyol and J. Sabater-Mir, "Computational trust and reputation models for open multi-agent systems: a review," *Artif Intell Rev*, vol. 40, no. 1, pp. 1–25, Jun. 2013.

[22] A. M. Frieze and G. R. Grimmett, "The shortest-path problem for graphs with random arc-lengths," *Discrete Applied Mathematics*, vol. 10, no. 1, pp. 57–77, Jan. 1985.

[23] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, Cambridge, MA, USA, 2003, pp. 482–491.

[24] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[25] Wei Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the 2005, American Control Conference, 2005.*, Portland, OR, USA, 2005, pp. 1859–1864.

[26] R. Zhou, "Gossip-Based Reputation Management for Unstructured Peer-to-Peer Networks," p. 34.