AD-SHARE: AN ADVERTISING METHOD IN P2P SYSTEMS BASED ON REPUTATION MANAGEMENT

Nikos Salamanos, Evi Alexogianni, Michalis Vazirgiannis
Department of Informatics, Athens University of Economics and Business
salaman@aueb.gr, alexogianni@yahoo.com, mvazirg@aueb.gr

Abstract--We present Ad-Share, a distributed method for mutual advertisement hosting among a group of participating entities in P2P architecture. In such architectures the issue of free riding is well known. A multitude of reputation and incentive-based methods have been proposed to improve the system performance. Ad-share is based on two reputation schemes, the reputation algorithm EigenTrust and a reputation-based incentive model. The Ad-Share constitutes a novel approach towards an online distributed advertising method lacking any payment scheme. We evaluate Ad-Share extensively in environment in which a group of participating organizations is heterogeneous with regard to their quality and services and we show that our method can effectively provide fair and robust advertisements assignments.

Index Terms-- Peer-to-Peer, Online advertising, Incentives, Reputation Management.

I. Introduction

Peer-to-peer (P2P) networks are distributed systems that allow the direct communication of participating peers without the necessary mediation of a server. An extensive survey on P2P networks can be found in [1]. The main characteristic of P2P networks is the absence of central control over the users. The users are autonomous, having control of their files and connect/disconnect to the network spontaneously. In P2P networks emerge the problem of free riding. The users mainly act as consumers and lack motivation to contribute. Extensive research has been conducted in this area and many models have been proposed to motivate the users' contributions. The research is focused on applications of reputation and incentive-based mechanisms in P2P networks. A comprehensive survey on online reputation mechanisms is [4] and on P2P reputation systems is [5]. Recalling from [5] "the core of any P2P reputation system is in the answer to the following question: how can a given peer use the experiences between the peers, that it can retrieve from the network, to evaluate the trustworthiness of any other peer?" In P2P systems there is no central authority to distribute the reputation values of the users. Thus there is the necessity for a distributed mechanism to spread the reputation ratings for the entities. In [8] the authors propose a distributed reputation algorithm that produces global reputation ratings for the evaluation of peers, robust to malicious peers.

The reputation problem in P2P has been studied extensively using game theoretic framework and many models have been proposed. Game theory is an appropriate tool for modeling systems with selfish and autonomous participating entities. The notion of Nash Equilibrium is a tool for prescribing of peers' behavior. The main underline assumption on the behavior of the participating entities is that are rational i.e. they act as to maximize their profit. Proposed approaches are the micro- payment model [9], differential service-based incentive scheme [2] and reputation-based incentive model [10].

The recent growth of online markets is impressive and has become the key point for the companies in the web. A significant number of firms, from small businesses to multinational corporations, incorporate online advertising into their marketing strategy. A large number of online advertising methods have been developed, some of the most popular follow: banner ads, email marketing and search engine marketing.

The online methods use different types of payment schemes, the most common being the CPA (Cost Per Action), the CPC (Cost Per Click) and the CPV (Cost Per Visitor). In online advertising the scheme is centralized, i.e. there is a service provider who defines places on the relevant web server pages to be occupied by the advertisees, ie. the entities that want to be advertised.

The P2P systems have be become very popular mainly as file sharing systems. In this paper, we deal with the problem of designing a distributed advertising method for P2P systems lacking any payment scheme. The methods have to be robust and fair with regard to advertisements assignments. We propose an online distributed advertising method which alleviates the need for a payment mechanism. Our method is decentralized in the sense that there is not central authority responsible for the advertisements (ads) assignments but the companies themselves. Our method is a first approach for the designing a P2P incentive-based advertising mechanism. We apply two reputation mechanisms, an incentive-based scheme [10] and the EigenTrust algorithm [11]. We consider a setting where the participating organizations (peers) differ with regard to their quality and services. We have developed two variations of the EigenTrust, the first, group the peers into homogenous clusters according to their reputation in real market and the second ensure the homogeneity of the clusters. Our proposed method efficiently achieves the desired goals of fair and robust advertisements assignments.

The paper is organized as follows. Section II covers the background of the present method. In Section III, we present the advertising method. In Section IV we present the evaluation of the method. Finally, in Section V we present the conclusions of the paper.

II. Background

In this section we review reputation models the methods we exploit in our advertising method. The authors in [11] present EigenTrust a distributed reputation algorithm. EigenTrust evaluates the peers' transactions and is based on the notion of transitive trust. Moreover, the algorithm aggregates the reputation values based on a distributed method robust to malicious peers. In [10] a reputation model is defined based on incentives in order to address the problem of free riding. In the following we present Eigentrust and the incentive-based model in detail as they are used as an integral part of our approach.

A. The EigenTrust Algorithm

The Eigentrust algorithm [11] is based on the notion of *transitive trust*. Recalling from [11], every peer assigns local trust values to the peers that have committed at least one transaction in the past. If peer i wants to rate unknown peers then it asks its acquaintances about their opinion about those peers. Acquaintances of peer i are peers which peer i has rated with high trust value in the past. . When peer i rates a transaction with peer j as positive, then $tr_{ij} = 1$ (if negative then $tr_{ij} = -1$). The local trust value s_{ij} that peer i assigns to peer j is the sum of the respective ratings, $s_{ij} = \sum tr_{ij}$. The normalized local trust value c_{ij} is defined as: $c_{ij} = \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)}$. When peer i rates an unknown peer k, it asks its acquaintances about their opinion about k. Peer i weights their opinion (c_{jk}) by the trust values (c_{ij}) it has assigned in them, thus: $t_{ik} = \sum_j c_{ij} \cdot c_{jk}$. If C is the matrix $[c_{ij}]$ and $\vec{c}_i$ a vector containing the trust values which i assigns to his acquaintances then $\vec{t}_i = C^T \cdot \vec{c}_i$. If peer i aims at improving its knowledge about a larger part of the network it may ask the friends of its friend's friends and so on. Then, after n steps we have $\vec{t}_i = \left(C^T\right)^n \cdot \vec{c}_i$. As demonstrated in [11] when n is large the trust vector $\vec{t}_i$ converges to the vector $\vec{t}^{(k+1)} = C^T \cdot \vec{t}^k$.

In Eigentrust the notion of *pre-trusted* peers is defined as a tool to avoid manipulation from malicious peers. The set P of pre-trusted peers is a subset of the set of peers and they are considered as honest from the beginning of the procedure. The notion of pre-trusted peer is important for the convergence of the algorithm. The pre-trust value for peer i is defined as: $p_i = \begin{cases} 1/|P| & \text{if } i \in P \\ 0 & \text{otherwise} \end{cases}$ (1). Finally, we have $\vec{t}^{(k+1)} = (1-b)C^T \cdot \vec{t}^k + b\vec{p}, \ \ 0 < b < 1$, where the parameter b is a constant. In fact each peer computes its global trust value as follows: $t_i^{k+1} = (1-b)\cdot(c_{1i}t_1^k + c_{2i}t_2^k + \ldots. + c_{ni}t_n^k) + b \cdot p_i$ (2).

The distributed EigenTrust algorithm is defined as follows:

For each peer i two sets of peers are defined,

A_i : The set of peers that have at least one transaction with i.

B_i : The set of peers with which peer i has at least one transaction.

Then the EigenTrust algorithm appears in Table I.

TABLE I
THE EIGENTRUST ALGORITHM

Each peer i do {
Query all peers $j \in A_i$ for $t_j^0 = p_j$;
repeat
Compute $t_i^{k+1} = (1-b)\cdot(c_{1i}t_1^k + \ldots. + c_{ni}t_n^k) + b \cdot p_i$;
Send $c_{ij}t_i^{k+1}$ to all peers $j \in B_i$;
Compute $\delta = \left| t_i^{k+1} - t_i^k \right|$;
Wait for all peers $j \in A_i$ to return $c_{ij}t_j^{k+1}$;
until $\delta < \varepsilon$
}

B. A Reputation based Incentive Model

The authors in [10] present an incentive-based scheme in P2P networks to improve the system performance. They apply a game theoretic model, an infinite repeated game, analyzing the incentive scheme and identifying the pure and mixed Nash Equilibria. The peer interactions are modeled as an infinite repeated game, the *Service Game* G^{∞}. Time is divided in infinite time periods. In each of them each peer can has two activities; to serve others, to obtain service for itself. Peers are motivated to serve others by imposing that every peer will receive service with probability equal to its current reputation. Peers gain reputation only by serving others. The reputation of each peer is measured each time period using a recursive function R. The value R_t^i is the reputation value of a peer i in the time period t. Function R_t^i is a linear function of the reputation value R_{t-1}^i and the reputation value that peer i gains from its actions in period t. In every time period each peer will increase (decrease) its reputation value R depending on its actions. The request from a peer i, in a time period t, will be served from a peer j with probability R_t^i. In each time period, each peer receives one request for service while it could receive service one time per time period.

The incentive scheme is evaluated using a game theoretic framework. A service game G^{∞} is initiated as an infinite repeated game and is defined as follows:

- Time t is divided into infinite time-periods, $t = 0,1,2,\ldots,\infty$.
- Each of the N peers is considered as a player.
- The set of possible actions for each peer is {Serve, Don't serve}

Function R is defined as: $R_t^i = \left\{(1-\alpha)\cdot R_{t-1}^i + \alpha\cdot\omega \quad t \geq 2\right\}$ (3).

While: $R_0^i = 0, \ \forall i$ and $R_1^i = \omega, \ \forall i$.

Parameter ω is set to 1 if the peer chose the action "serve", 0 otherwise. Parameter α, ($0 \leq \alpha \leq 1$) defines the portion of the reputation a peer maintains from its past performance vs. its current action. Low values for α imply small changes to a player's reputation if it does not serve. Basic assumption is that the population of peers is homogeneous i.e. have equivalent capabilities and then is difficult for them to coordinate.

The authors in [10] estimate the pure Nash equilibrium and the symmetric mixed Nash equilibrium of the service game. They prove that the pure Nash equilibrium is the action "don't serve" and a symmetric mixed Nash Equilibrium is the mixed strategy (p, 1-p), same for all peers. Briefly, the proof is as follows. According to Nash Folk theorem for the infinite repeated games [12], G and the infinite repeated game G^{∞} have the same Nash Equilibria. Therefore, we have only to detect the Nash Equilibria of the G. The pure Nash equilibrium of the advertising game is the action "don't serve". We notice that if a player chooses the action "serve" its payoff is -C instead of zero in the case it chose the "not advertise" action . The payoff of a player i who chooses to advertise is $-C$ because all the other players will chose the equilibrium action and will not advertise. This implies that each player's request will be rejected. This pure equilibrium is unstable, because if the players stay at the equilibrium then the system collapses.

For the mixed strategy Nash equilibrium, we notice the game G exhibits a symmetric Nash equilibrium because of the homogeneity of the peers. The peers are indifferent about the peer that they will request for service. We recall the following statements from [12]:

- (Mixed strategy equilibrium existence) "Every finite strategic game has a mixed strategy Nash equilibrium".
- "Every action in the support of any player's equilibrium mixed strategy yields that player the same payoff".

The proof is based on these statements and the existence of a symmetric mixed Nash Equilibrium. Assume that the symmetric equilibrium is {p, 1-p} where p is the probability that the player choose the action "serve". Then:

$$\text{payoff}_{\text{Serve}} = \text{payoff}_{\text{Don't}} \quad \Rightarrow$$

$$p(-C + {}^{\text{serve}}R_t^i \cdot U) = (1-p)({}^{\text{don't}}R_t^i \cdot U) \;\Rightarrow \ldots\; p = \frac{{}^{\text{don't}}R_t^i}{\frac{-C}{U} + {}^{\text{serve}}R_t^i + {}^{\text{don't}}R_t^i} \quad (4)$$

If we assume that $\frac{C}{U} \ll 1$, then $p = \frac{{}^{\text{don't}}R_t^i}{{}^{\text{serve}}R_t^i + {}^{\text{don't}}R_t^i}$.

If the function R is positive and ${}^{\text{serve}}R_t^i > {}^{\text{don't}}R_t^i$, then regardless of the relation ${}^{\text{serve}}R_t^i$ and ${}^{\text{don't}}R_t^i$, p is always less than 0,5. Finally, from (3) and (4) follows:

$$p = \frac{(1-\alpha)R_{t-1}^i}{2(1-\alpha)R_{t-1}^i + \alpha - \frac{C}{U}} \quad (5).$$

This is the symmetric Nash equilibrium for the game G^{∞}. Also, this is a stable Nash equilibrium. An important observation is that even we assume $\frac{C}{U} \ll 1$ the parameter α must be small in order to the probability p to converge to 0,5.

III. The Advertising Method

We develop an advertising method for mutual advertisement hosting in a group of participating companies, where each company owns a web site. We assume that the firms have been clustered in semantically coherent groups according to their services and products classification. These semantically related peers-companies are grouped into a semantic overlay network (SON). The authors in [3] and [6] show that files shared in P2P networks can be clustered efficiently by content categories. The exact method for the generation of the semantic overlay network is out of the scope of this paper. The semantic overlay network contains M semantic categories and each of them L subcategories. In each category peers are evaluated with regard to the quality of their services and products and are grouped into relevant quality categories. The advertising process starts at each semantic category of SON and at each quality category independently. The basic assumption is that if the peers in a quality category are relevant with regard to reputation level in the market, then they don't have the motivation to cooperate. This assumption is important for the game theoretic analysis of Ad-share.

The Ad-method is based on the combination of two reputation methods: the Eigentrust algorithm [11], and an incentive-based scheme [10], aiming to motivate the peers to accept advertisements at their web sites. Each peer has a dual role, act as advertiser (i.e. host advertisements on its web site) and as advertisee ("pay" for having an advertisement hosted). We have developed two versions of the EigenTrust algorithm, the Eigen-Clustering and Eigen-Test. The first version computes the reputation of the companies in the market. Eigen-Clustering clusters the peers in quality categories according to their reputation. The underlying assumption is that Eigen-Clustering values reflect the reputation of a company in the market. We cluster the peers in quality categories in order to ensure that each category has a homogenous population of companies. The second version of EigenTrust, the Eigen-Test, is used periodically, evaluating the homogeneity of the categories. The algorithm ensures that the peers participate in a group of companies witch is homogenous with regard to reputation level in the market.

Briefly the steps of the method are as follows: After the end of Eigen-Clustering the main advertising process starts in every quality category independently. The time is modeled by an infinite number of advertising periods (Ad-periods). In each Ad-period, every peer has a reputation value Φ. The reputation values are measured by a recursive function Φ_t. The reputation values are increased each time the peer acts as advertiser. We apply an incentive in order to motivate the peers to act as advertiser. The incentive is that the request of peer i as advertisee (i.e. asking for some other peer to host its advertisement) will be served with a probability equal to its reputation value. In the next sections we describe analytically the stages of the advertising method.

A. The Advertising Game

Similarly to the service game G^{∞} presented in [10] we define the *advertising game Ad-Game.* We assume infinite consecutive advertising periods. In every period each peer has to decide if it will make available for advertisement K slots (at its website) or not. The action "open K slots" implies that the peer is willing to accept to host exactly K-ads in its web site. We also assume that the peers are honest with their choices, i.e. if they have opened k slots they would not reject any request for advertising if they have at least one slot vacant. The underlying assumption is that we identify the action of "open K-slots" with the potential acceptance at the future K requests for ads hosting. We compute the reputation values of a peer with a recursive function Φ, whose values depend on the actions of a peer. In our model the action "open k slots" is equivalent to the action, "advertise K peers". The function Φ_t is the function R (3) using different initialization strategy. In the same manner as in the service game in section II, by default, the probability for a peer to be advertised in a web site is equal to its reputation. Under this incentive scheme, the peers have the motivation to be advertisers in order to increase their reputation and thus to increase their chances of having theirs ads accepted in other peers.

We apply a game theoretic approach analyzing the incentive scheme and identifying the pure and mixed Nash Equilibria of the game.

We define the Ad-Game an infinite repeated game as follows:

1. The time t is divided in infinite ad-periods, $t = 1, 2, 3.....,\infty$.
2. We define the game G.
3. The Ad-Game based on the infinite repetitions of the G.
4. The set of N players is the set of N peers.
5. The set of actions $A = \{a_1, a_2\}$, for each peer is:

{Open K slots, Don't open K slots} ≡ {Advertise K peers, Don't advertise}.

6. We define the function Φ as follows:

$$\Phi_t^i = \begin{cases} (1-\alpha)\cdot\Phi_{t-1}^i + \alpha\cdot\omega & t \geq 1 \\ 1 & t=0 \end{cases} \quad (6).$$

The N players have to decide independently if they are willing to accept to host K ads in their website. Each player/peer represents a company with a web server. During the first period all peers have a reputation value 1. This is a different initialization strategy to the one presented in [10]. We give a bonus of reputation to every peer at the beginning of the game in order to have a successful Ad-Hosting from the first period. Parameter ω is set to 1 when the player opens K slots and 0 otherwise.

Following an analysis similar to [10] we can detect the Nash Equilibria of the Ad-Game. The pure Nash equilibrium of the advertising game is the action {Don't open k slots}. The proof for the symmetric mixed Nash equilibrium is the same with the proof for the service game G^∞ in section II, since we assume that the group of companies is homogenous. It is easy to prove that the symmetric mixed Nash equilibrium for the Ad-Game is the strategy (p, 1-p) where:

$$p = \frac{(1-\alpha)\Phi_{t-1}^i}{2(1-\alpha)\Phi_{t-1}^i + \alpha - \frac{C}{U}} \quad (7).$$

B. The Eigen-Clustering and Eigen –Test Algorithms

The Ad-method uses two versions of EigenTrust algorithm. The first version, Eigen-Clustering, clusters peers in quality categories *Low, Medium* and *High*. Each peer is a company with a web site. We assume the peers' market reputation as proportional to the size of the company, and the quality of its products. Thus the peers participating in the advertising already know some of the other peers and have an opinion for them. We define the sets A_i and B_i of peer i in the same fashion as in the original EigenTrust. Set A_i is the set of peers who have an opinion about i. Set B_i is the set of peers that peer i knows from the market and it can evaluate them (i.e. the opinions of peer i for the peers it is aware of). We define the local trust values c_{ij} as the evaluation of peer i to peer j, $0 \leq c_{ij} \leq 1$ with $\sum_{j\in B_i} c_{ij} = 1,\ \forall i, j$. The Eigen-Clustering follows the same approach as EigenTrust appearing in Table-I.

We consider all the peers as pre-trusted, |P|=N, meaning that in the first repetition of the algorithm each peer has $t^0 = 1/|P|$. The output of the algorithm is the t_{market} values used for the subsequent clustering. The experimental results show that this simple measure is efficient for the clustering process.

The second version, the Eigen-Test, evaluates the homogeneity of the quality categories. It runs periodically at the end of a randomly chosen ad-period. The algorithm runs simultaneously in all quality categories of the semantic subcategory. We assume that the peers cannot predict the exact ad-period during which Eigen-Test is executed. We choose the ad-period randomly in order to retain the game theoretic settings of the infinite repeated game and to ensure that the equilibrium remains the same. Also, the test can run only after the ad-period r, where r is the number of periods needed for the recursive function Φ to converges, and has been defined experimentally.

We define sets A_i and B_i and the local trust values c_{ij} as follows:

Assume the Eigen-Test runs at the end of the ad-period μ, then:

A^i_μ : The set of peers that peer i has advertised during the μ Ad-periods.

B^i_μ : The set of advertisers of peer i during the μ Ad-periods.

Then, we define: $c_{ij} = \dfrac{v^i_{j,\mu}}{\sum_{k \in B^i_\mu} v^i_{k,\mu}}$ (8)

where:

- $j \in B^i_\mu$
- $v^i_{j,\mu}$: The sum of user clicks on the Ads of peer i , with j as advertiser, during the μ Ad-periods.
- $\sum_{k \in B^i_\mu} v^i_{k,\mu}$: The sum of user clicks on the Ads of peer i.

The entity c_{ij} is the evaluation of peer i to its advertiser peer j, with regards to the overall advertising profit that peer i received during the μ Ad-periods. The distribution of the global Eigen-Test values in a quality category reflects the reputation level of peers as advertisers. If the Eigentrust values did not exhibit significant differences then we can assume that the quality category is homogeneous, i.e. peers are similar with respect to their reputation/performance as advertisers. If not, then the peers must be clustered anew, based on the Eigen-Test values in all quality categories.

C. The Ad-Share Algorithm

Summarily we present the overall steps of the Ad-Share method as execute in each semantic category of a SON.

Semantic Overlay Generation Algorithm

1. Semantically related peers are grouped into SON.
2. {< peer id, name of the company, url, Category, SubCategory >$_1$,..,< >$_N$} denotes the set of peers in a Semantic Category.
3. In each Semantic Category the Eigen-Clustering run.

Eigen-Clustering

1. The peers are clustered in quality categories Low, Medium, High
2. In every quality category the Ad-hosting process run.

Ad-Hosting

In every ad-period:

1. Each peer i do
2. Compute probability p, using (7)
3. Decide, with probability p, “opens K- slots” or “Don’t open K-slots”

4. Compute Φ_i.
5. Propagate L requests for ad-hosting, where $L \le K$.
6. Every request has the form: <peer id, name of the company, url, peer's semantic subcategory, host semantic subcategories (or "all"), Φ>.
7. Receive requests for Ad-hosting. A request from a peer j would be served with probability Φ_j only if : peer i has empty slots and peer j is not in the same subcategory with peer i.
8. Store the user-clicks of ads.
9. The Eigen-Test run (in a randomly chosen period) in each quality category and after the end of period r.

Eigen-Test

1. If the Eigen-Test run at period t .
2. Compute the Eigen-Test values based on user-clicks from the t periods.
3. Compare the Eigen-Test values in each quality category.
4. If are homogenous,
5. Ad-Hosting run in period t+1
6. If not,
7. End of Ad-Hosting.
8. Cluster anew in quality categories all the peers in the semantic category based on Eigen-Test values.

IV. Evaluation of the Method

In this section we present the experimental results evaluating the Ad-method.

A. Initial Clustering

First, we assess the Eigen-Clustering. We simulate each semantic category with a homogenous network (i.e. all nodes have the same connectivity) of 200 peers using the GTITM topology.

As we have already mentioned, the action "open K-slots" implies that the peer is willing to accept to host K-ads and we assume that the peers are honest with their choices. We make a similar assumption considering all the peers as pre-trusted. We define the sets A_i and B_i of a peers i as the 10% of the peers using a uniform distribution. We define $\sum_{j\in B_i} c_{ij} = 1, \ \forall i, j$ and each peer has to give a reputation value to each peer in B_i . We define the local trust values representing the trust a peer attributes to the peers in B_i - as follows: Assume peer i has to evaluate four peers: k, l, m, n, then we choose randomly 3 values between 0 and 1, e.g. 0.156, 0,627, 0.256. We divide the range (0,1) into the intervals (0, 0.156), (0.156, 0.256), (0.256, 0.627), (0.627, 1). The size of each interval represents the trust values c_{ik} , c_{il} , c_{im} and c_{in} .

Then, we estimate the convergence of the Eigen-Clustering for 200 peers, we set b=0.1 in the equation 2. In Fig. 1(a) we present the Eigen-Clustering values for 20 randomly selected peers and for 10 iteration of the algorithm. Fig.1(b) shows an estimation of the convergence rate of the Eigen-Clustering, it convergences fast, after the 6th iteration, similar with the original EigenTrust [11] .

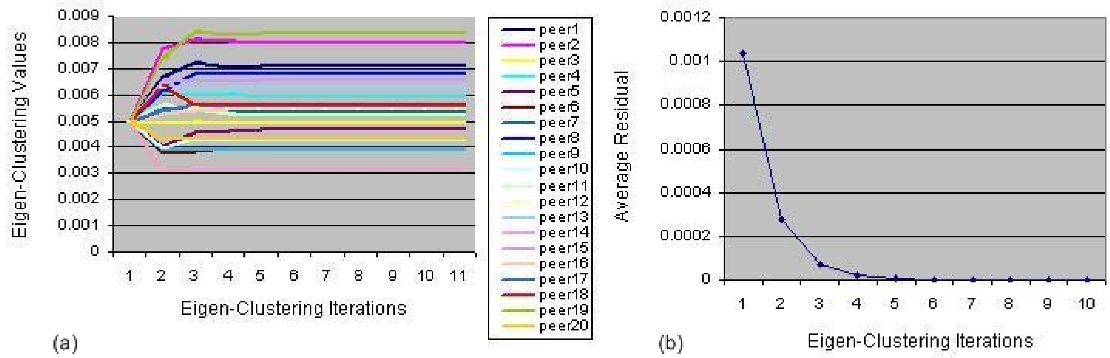

Fig. 1. (a) The Eigen-Clustering values for 20 random peers. b) Eigen-Clustering convergence

We further estimate the quality categories Low, Medium, High for a semantic category from the frequency of the Eigen Values for iteration after the convergence of the algorithm.

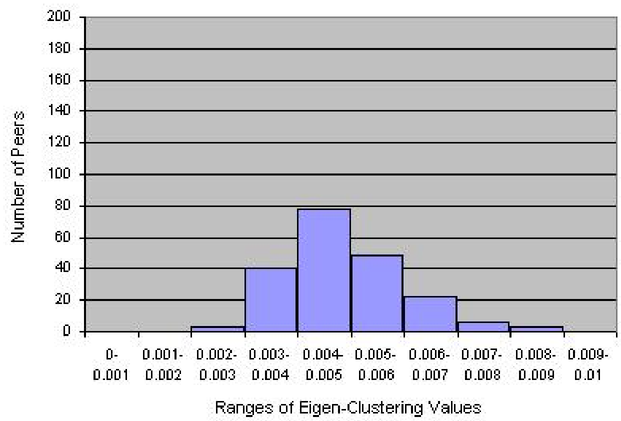

Fig. 2. The frequency distribution of Eigen-Clustering values after the 6th iteration and for 200 peers.

We compute the number of peers with Eigen-Clustering values in the intervals [0, 0.001),…, [0.009, 0.01] respectively. The results in Fig. 2 show the frequency distribution of the Eigen values. Small number of peers has Eigen-Clustering values into the interval [0,002, 0,004) and (0,006, 0,009], those set of peers are the clusters Low and High. More than 50% of peers are drawn to the interval [0.004, 0.006] and constitute the cluster Medium. Finally, we define the quality categories Low, Medium and High as the sets of peers with Eigen-Clustering values in the intervals [0,002, 0.004), [0.004, 0.006] and (0.006, 0.09] respectively.

B. Ad-Hosting

The experiments in this section address the convergence of function Φ for different values of α. We simulate an unstructured homogenous network with 200 peers and an average connectivity 4. Each semantic category includes 10 subcategories, we assign the peers to sub categories randomly using the uniform distribution. In every ad-period the peers choose to advertise others with probability p, according to mixed Nash Equilibrium (7). In (7) we set $C/U = 1/100$. Each peer has to decide independently with probability p if it will open K-slots, with K=4. In every ad-period every peer could request for advertisements. The maximum number of request per peer is K. The requests have the form: <peer's id, name of the company, url, peer's semantic subcategory, host semantic subcategories (or

"all"), Φ>. At the field "Host sub categories" we set "all", i.e. the peers are indifferent about the subcategory of their advertisers. The requests are propagated using a version of the m-random walks [8] method as follows: The request is propagated to m randomly chosen neighbours of the peer. Thus, m random paths-requests are created. Each neighbour propagates the request to only one randomly chosen neighbour. The process continues until the time to live (TTL) of the request reaches zero. If a request is served from a peer then all the others m-1 sub-requests stopped. This ensures than every request would be served from one peer.

For our simulation experiment we set TTL=10 and we implement a 4-random walks algorithm.

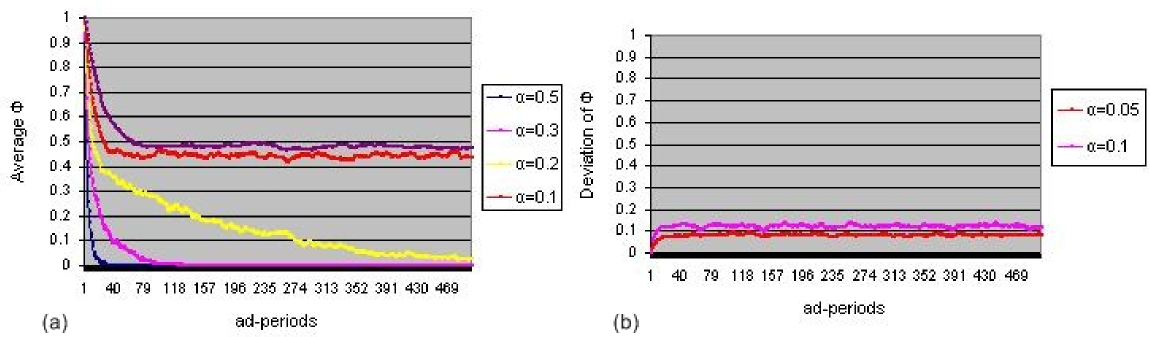

Fig. 3. (a) Average Φ values for 200 peers for various α. (b) Deviation of Φ for α=0.05 and α=0.1.

Fig. 3 depicts the convergence of function Φ for different values of α. If $\alpha > 0{,}1$ Φ convergences to zero. We study further the deviation of Φ, for α=0,1 and 0,05 (Fig. 5). For small values of α the reputation values converge to 0,5.

The next experiment computes the average p value and the deviation for 500 Ad-periods and α=0,05 and 0,1 (Fig. 4, 5).

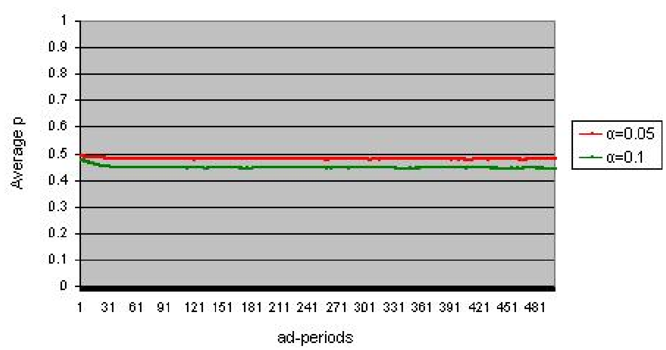

Fig. 4. Average p for α=0.05 and α=0.1.

Fig. 4 shows that for α=0,05 the probability p converges very close to 0,5. We recall that the upper bound of p is 0,5. In Fig. 5 we present the deviation of p for α=0,1 and 0,05. The deviation of p for α=0,05 is very small, less than 0,01. Thus, for the rest of the experiments we set α=0,05.

In the following experiments we estimate the average number of Ads per advertiser and the average number of Ads per advertisee. The number of Ads per advertiser is very important for the efficiency of the method since we want to ensure that the sources of the system (the open slots) will not remain unused.

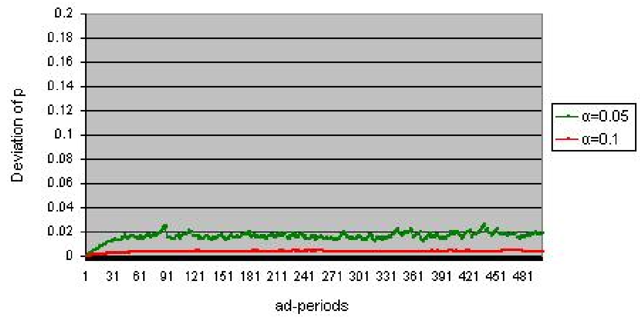

Fig. 5 Deviation of p for α=0.05 and α=0.1

As can be seen from Fig. 6 the average number of Ads for the advertisers is almost 4 from the first ad-period. This implies that the peers who are willing to accept to host K-Ads they host eventually almost 4 Ads per ad-period.

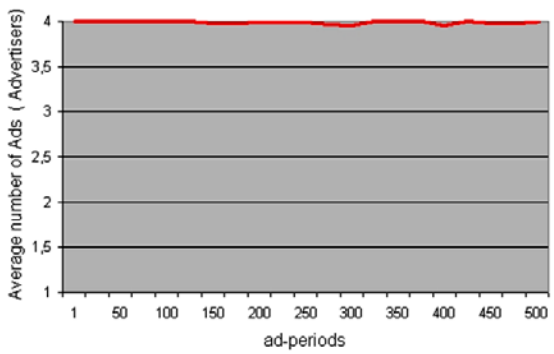

Fig.6. Average number of Ads for the advertisers

In Fig. 7(a) and (b) we present the average number of Ads for the advertisees (500 ad-periods) and the deviation. The average number of Ads is significant of the efficiency of Ad-method. The method must ensure that the participating entities eventually will be advertised at every ad-period. The experimental results shows that in every ad-period each peers will be advertised at least in one web site.

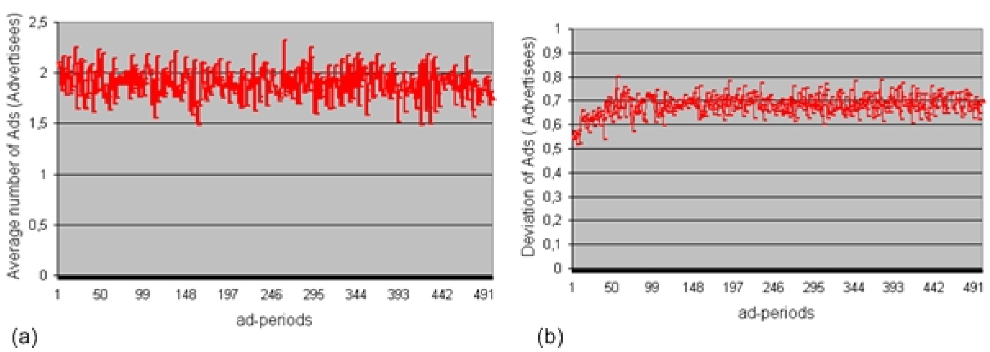

Fig. 7. (a) Average number of Ads for the advertisees. (b) Deviation of Ads for the advertisees.

C. Homogeneity of clusters

In the last section we assess the performance of Eigen-Test. The Eigen-Test algorithm runs after μ Ad-periods, where μ randomly chosen, and after Φ has converged. We implement the same simulation network as in section IV-B. We set the following values for the involved parameters: α=0,05, b=0,1 and 0,5. We set α=0,05 because for α=0,05 the probability p converges very close to 0,5. Also we experiment with b=0,1 and 0,5. The parameter b (2) determines the percentage of the pre-trust value of peer in an iteration of EigenTrust. We define the overall quality of a category as proportional to the total number of clicks its advertisements collected in one Ad-period. We evaluated the Eigen-Test for a quality category of 10.000 user-clicks. We assume that the advertisements in the category are equivalent with respect to their penetration to the market, i.e. they gather similar amounts of user clicks. We assign the clicks to the advertisements uniformly as follows; each advertisement gains a click with probability 1/Y, where Y is the total number of advertisements. Fig. 8(a) shows that the Eigen-Test converges after the 6th iteration. The result is similar with those for the convergence of Eigen-Clustering and EigenTrust. Figures 8(b), 9(a) and 9(b) depict the frequency distribution of Eigen-Test values for 50, 100 and 200 ad-periods, assuming b=0,1.

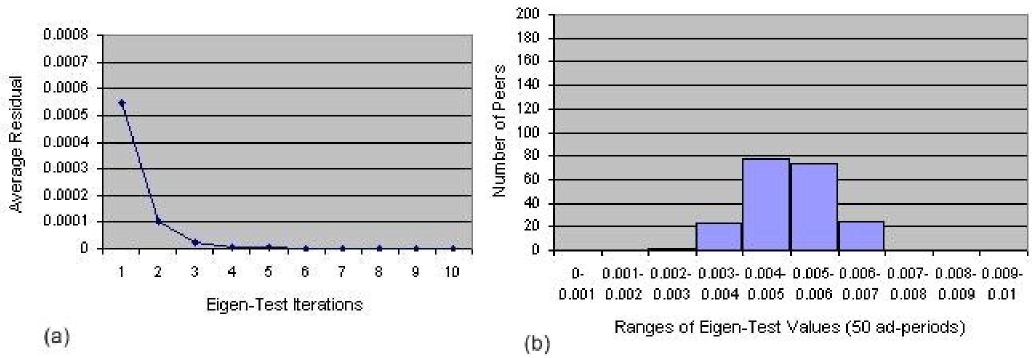

Fig. 8. (α) Eigen-Test convergence. (b) The frequency distribution of Eigen values (b=0,1)

We observe that as the number of ad-periods (parameter μ) increases the Eigen-Test values tend to concentrate in the interval [0.004, 0.006]. As μ increases the sets A_μ and B_μ of the peers grow larger. Thus, the peers evaluate a larger part of the network and the global Eigen-Test values tend to concentrate in the same interval.

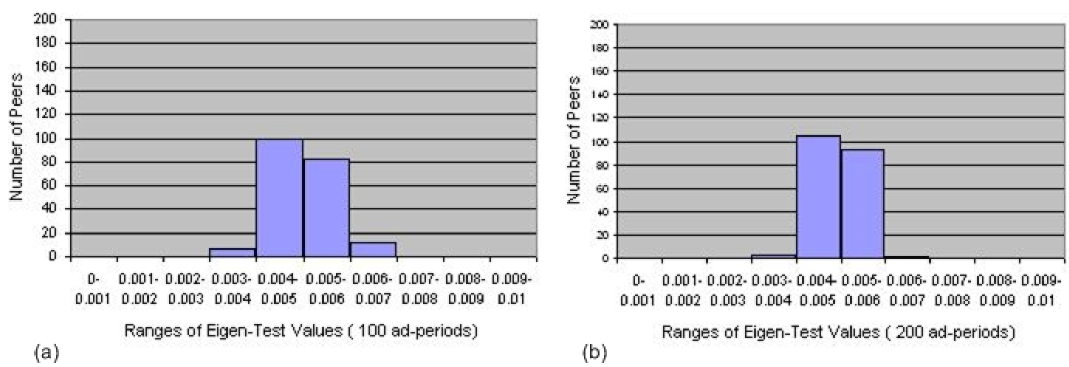

Fig.9. The frequency distribution of Eigen values for 100 and 200 Ad-periods respectively (b=0,1).

We experiment with b=0,5 and the main observation is that the Eigen-Test values tend to the concentrate into the same intervals as for b=0,1 but in a much smaller number of periods. Fig. 10(a) and (b) shows the experimental results for b=0,5.

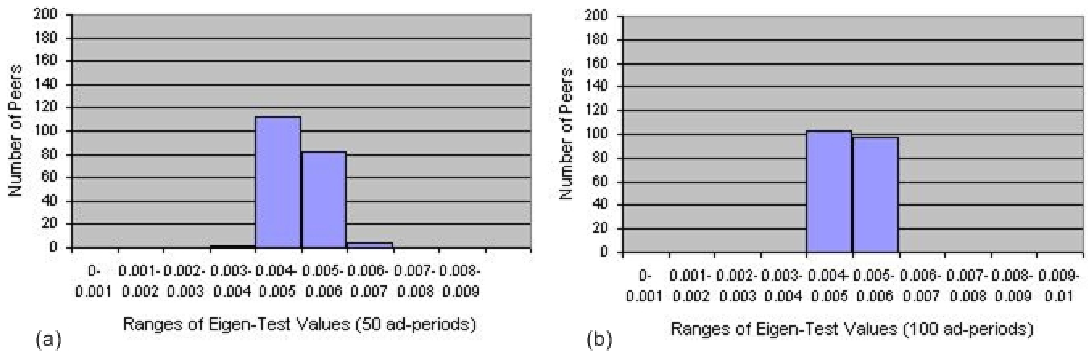

Fig.10. The Eigen-Test values distribution, for b=0,5, are drawn faster to the same intervals.

V. CONCLUSION

In this paper we have proposed a distributed online advertising method for P2P systems. Our method based on two variations of EigenTrust algorithm and an incentive-based scheme. The contribution of our method is that it constitutes a novel approach towards an online distributed advertising method lacking any payment scheme. We evaluate the method extensively and the experimental results show that Ad-Share can effectively provide fair and robust advertisements assignments.

REFERENCES

[1] Androutsellis - Theotokis S. and Spinellis D, A Survey of Peer-to-Peer Content Distribution Technologies, ACM Computing Surveys, Vol. 36, No. 4, December 2004, pp. 335–371.
[2] C. Buragohain, D. Agrawal, S. Suri. A Game Theoretic Framework for Incentives in P2P Systems. In Proc. of the Third International Conference on Peer-to-Peer Computing (P2P'03), 2003.
[3] A. Crespo and H. Garcia-Molina. Semantic Overlay Networks for P2P Systems. Technical report, Stanford University, 2002.
[4] C. Dellarocas. Reputation Mechanisms, in Handbook on Information Systems and Economics, T. Hendershott (ed.), Elsevier Publishing, forthcoming, 2006
[5] Despotovic, Z. and Aberer, K., 2004, Possibilities for Managing Trust in P2P Networks. Swiss Federal Institute of Technology (EPFL) Technical Report IC/2004/84, Lausanne, Switzerland.
[6] Doulkeridis Christos, Norvag Kjetil and Vazirgiannis Michalis, DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks, IEEE Journal On Selected Areas In Communications, Vol. 25, No. 1, January 2007
[7] Feldman Michal and Chuang John, Overcoming Free-Riding Behavior in Peer-to-Peer Systems, ACM SIGecom Exchanges, Vol. 5, No. 4, July 2005, Pages 41-50
[8] C. Gkantsidis, M. Mihail, A. Saberi, RandomWalks in P2P Networks, IEEE INFOCOM '04, HK, Mar.2004.
[9] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. Incentives for sharing in peer-to-peer networks. In Proc. of 2001 ACM Conference on Electronic Commerce, 2001
[10] Gupta R. and Somani A. K., Game Theory As A Tool To Strategize As Well As Predict Nodes' Behavior In Peer-to-Peer Networks, Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05).
[11] Kamvar S. D., Schlosser M. T. and Garcia-Molina H., The EigenTrust Algorithm for Reputation Management in P2P Networks, WWW2003, May 20–24, 2003, Budapest, Hungary
[12] Osborne M. J., A Course in game theory, Cambridge, Mass.: MIT Press