An Advertising Mechanism for P2P Networks

Roy Friedman Alexander Libov
Computer Science Department
Technion - Israel Institute of Technology
Haifa 32000, Israel
{roy,alibov}@cs.technion.ac.il

***Abstract*—In order for P2P systems to be viable, users must be given incentives to donate resources. Such incentives can be in the form of tit-for-tat like mechanisms, in which a user is rewarded with better service for contributing resources to the system. Alternatively, such incentives can be economical, i.e., users get paid for their contribution. In particular, the latter can be achieved through a P2P advertisement mechanism.**

This paper investigates how to incorporate an advertisement mechanism into a P2P system to serve as an incentive to donate resources, especially for services in which users often interact with the system through mobile devices. First, the precise P2P advertisement dissemination model is presented. Second, the paper proposes and explores several advertisement dissemination schemes combined with a few payment models and compares between them through simulations. The reported results are encouraging for this direction and in particular identify payment models whose payment is super-linear with the availability of donated machines. This means that they serve as good incentives for owners of donated machines to keep them connected to the P2P network for long durations.

I. Introduction

P2P networks enable service providers to offer very large scale Internet based services without incurring the tremendous costs associated with hosting the service on their own infrastructure (or leasing it from a cloud provider). By definition, such networks rely on users donating their computers to serve as *peers* in the system, who provide resources such as compute power, storage space, bandwidth, etc., to the benefit of the system. In order to be viable, a P2P system should limit the possibility of *freeloading* [1], [2] - receiving service while not contributing at all, although, some minimal service can be granted to convince users to contribute to get better service [3]. This is where incentive mechanisms step in. Their job is to provide incentives for the peers to contribute resources. Incentives can be non economical - a user can get better service if he contributes more resources [3], [1], [2], and they can also be economical - a user can get paid to provide resources [4], [5], [6], [7].

An advertising mechanism can satisfy the requirements of the latter. Using an advertising mechanism, a user can donate resources to help spread advertisements (alongside the content) and later the advertiser can share revenues with the users that helped earning them.

Traditionally, each peer in the P2P network is a machine capable of doing some computation and is associated with a user of this P2P system. However, nowadays, mobile Internet access is gaining popularity. More and more users connect to the Internet and consume Web content from their mobile devices. This poses a challenge for tit-for-tat like mechanisms, since the device used by a user to access services is usually different than the user's donated machine. While several global incentive mechanisms have been devised, these tend to either be complex and/or suffer from scalability limitations. Consequently, we believe that P2P advertisements mechanisms are a promising alternative for providing incentives to donate resources in P2P networks in which much of the content is consumed through mobile devices. To that end, our model of the P2P network includes user donated machines that are connected by a P2P overlay. In order to interact with the P2P network, users use client devices (e.g., mobile phones or tablets) to directly connect to an arbitrary peer in the P2P network. It is important to note that the client devices are only used to consume content and are not part of the P2P dissemination network. This is further elaborated in section II.

This paper investigates how to implement such advertisement mechanisms. Specifically, we assume that advertisements are spread inside an existing non-dedicated P2P network. Whenever a user accesses a certain content, the peer that served the content will also integrate one or more advertisements from the set of ads it currently has. The advertiser is then notified, and in return pays both to the peer that served the ad and to all other peers that participated in the dissemination of the ad.

Realizing this concept entails several challenges. For example, the dissemination mechanism should disseminate the ads to parts of the network where they are likely to be useful, without knowing who the users that view these ads are. Further, there is a need for a payment model that would encourage the dissemination of ads as well as peer participation in their dissemination. In particular, the payment obtained by each peer should be proportional to its participation time in the P2P network, and ideally, should be super-linear. That is, consistently active peers should be paid much more than sporadically active ones.

The ad dissemination mechanism should work for an existing P2P system. Hence, it cannot modify the P2P overlay for its needs (similar to the work done in [8]), as this may hurt the performance of the real application running on top of the P2P system. Instead, it must rely on the existing P2P overlay and do the best it can given this overlay.

In this work, we present several such approaches to ad dissemination, part of which are random and others are based on machine learning techniques, and investigate their performance in combination with several payment models. We also develop a light weight propagation encoding scheme that prevents peers from cheating about their participation in ad

978-1-4799-0521-8/13/$31.00

forwarding paths. We report on the insights gained from our study regarding when it is preferable to use which techniques and how the payment models affect the dissemination. Our results indicate that indeed P2P advertisement schemes are a promising direction for building incentives mechanisms in P2P networks. In summary, the contributions of this work are as follows:

- A new model of economic incentives in a P2P network.
- A light weight propagation encoding scheme that prevents peers from cheating about their participation in ad forwarding paths.
- An algorithm to find seeding peers in a graph (elaborated further).
- Insights regarding when it is preferable to use which dissemination techniques and how the payment models affect the dissemination.

The rest of this paper is organized as follows: An overview of our solution appears in Section II, while Section III discusses the various payments models and dissemination strategies. The dissemination tracking scheme is introduced in Section IV, and the simulation performance results are presented in Section V. Related work is discussed in Section VI. We conclude with a discussion and future work in Section VII.

II. Overview

A. Goals

As indicated above, the mechanism we are looking for should be an effective incentive for the network's peers as well as attractive for the advertisers. As an incentive mechanism, it should reward the owners of donated peers proportionally to the availability of these peers. Ideally, the reward should be super-linear with the availability in order to encourage keeping a donated machine connected to the network for long periods of time. At the same time, the communication overhead imposed by the protocols should be kept low and they should be computationally and space efficient, in order not to interfere with the main network activities.

As for the advertisers, the mechanism should ensure that every ad reaches every peer that currently serves a potential target user for the ad. In other words, the mechanism should obtain extremely high *reach* rates. At the same time, the mechanism should ensure that the maximal advertiser's budget is not exceeded. Also, the mechanism should account for rogue nodes that may try to attack the system or try to increase their revenue through fraud [9].

B. Basic Concepts

In our work, we assume the existence of a P2P network of *nodes*. The nodes of the network have *neighbors* and can send messages to any other node in the network. All nodes can have *client* devices connected to them. A client device provides a single *user* with access to the network through a sole connection to a node of the P2P network. Each user has specific characteristics such as: age, gender, marital status etc. An advertiser that is interested in disseminating advertisements

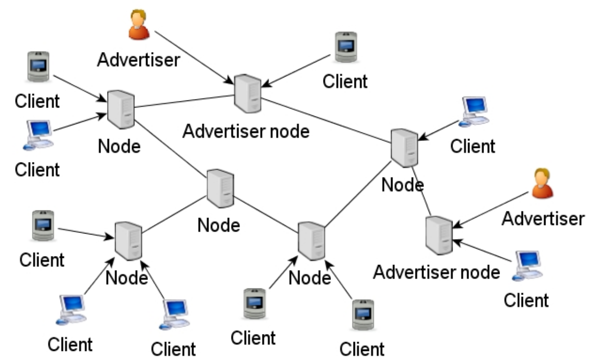


Fig. 1. The entities in the system

on the P2P network must provide machines to function as *advertiser nodes*. Each advertiser node is in charge of disseminating to its portion of the network. Depending on the size of the network, an advertiser must ensure that the advertiser nodes can handle their allocated network portion sizes. The advertiser nodes receive *advertisements* (consisting of bid, target audience, budget and such) from advertisers. An advertiser node stores each advertisement until the budget allocated for the advertisement runs out. The advertiser nodes are in charge of disseminating the advertisements in their portion of the P2P network, tracking impressions and clicks as well as sending payment notices to nodes that participated in the dissemination. The advertiser node disseminates a small message, called *ADM* - advertisement dissemination message, that describes the features of the advertisement (target audience, bid, budget etc.). When a node decides that the advertisement is suitable for one of its clients (by asking the client), the node requests the actual content of the advertisement from the advertiser node. Also, when receiving an ADM, the receiving node decides which of the neighbors it will forward the ADM to. When a user views or clicks on the advertisement, other nodes that contributed to its dissemination and serving are getting paid.

C. Advertiser Node

The advertiser node needs to disseminate the ad to all nodes in its portion of the P2P network that are likely to post it to their clients. Call these nodes the *target nodes*. However, the advertiser node may not know a-priori who are the target nodes, and even if it does, for scalability reasons it may not be practical for the advertiser node to contact each of them directly. Rather, we are looking for a solution in which the advertiser would contact only a small set of peers, which we call the *seeding peers*. These peers initiate the dissemination process to the rest of the network.

As requests for ads and click reports flow back into the advertiser node, it can gradually learn about target nodes as well as growing parts of the P2P overlay in its portion of the P2P network, and collect statistics. In particular, all our schemes are based on *rounds* where each round comprises of disseminating an ad from the advertiser node and then obtaining feedback about its reach, impressions, and clicks. The advertiser node maintains a round based history of seeding peers and for each advertisement a score is being kept, which comprises of the *click through rate* (CTR), number of clicks-per-day, etc. Using the gathered data along with the advertisement properties, an advertiser node can choose seeding peers for new advertisements. Using the principle of time locality, the advertiser node can assume that (most) target nodes of recent ad dissemination rounds remain target nodes for the following rounds. Hence, intuitively, advertiser nodes pick as seeding peers a small number of nodes whose distances to the target nodes are minimal. From these seeding peers, the ads will be propagated on the P2P overlay and will likely meet their target nodes. Yet, since the advertiser node may not be aware of all target nodes, the dissemination process described in the next section enables discovery of additional target nodes and is not limited to the already known ones. Recall that we try to optimize the dissemination on the existing P2P overlay because in our model, the advertisement mechanism is part of an existing P2P system and thus cannot alter the P2P overlay.

The availability of the seeding peers is also recorded and is factored in the score to give preference to nodes that are available the most. For each impression, the node that serves it sends a content request message to the advertiser node. Whenever a user clicks on an ad, the user is taken to the advertiser node that redirects him to the advertiser's page. If the clicks-per-day for the advertisement is too low, the advertiser node may choose different seeding peers and send the advertisement again, or, notify the advertiser. If the ad budget is nearly depleted, all content requests return with a flag that notifies the asking nodes that future requests will be ignored and not be paid for.

1) Seeding Peer Selection: In order for an advertiser node to select seeding peers, it creates a graph depicting the latest state of its P2P overlay portion known to it along with the target nodes that have served advertisements recently. From this graph, the advertiser node selects s seeding peers, where s is a tunable parameter. Nodes that were chosen as seeding peers, but were unavailable are deleted from the graph for a period of time reversely proportional to their overall availability (i.e., a node that is available $1/51$ rounds will have x50 more penalty time than a node that is available $50/51$ rounds). Based on the resulting graph, the advertiser node chooses the seeding peers as the ones minimizing some cost function related to reaching all target nodes. We consider the following optimization function for seeding peer selection: Min sum of distances - select the seeding peers such that the sum of distances from each target node to the closest seeding peer would be minimal.

The problem of finding minimal sum of distances is somewhat similar to K-Medoids [10]. However, the K-Medoids algorithm would minimize the distance between all the nodes to k corresponding seeding peers, while we want to minimize only distances from target nodes to their corresponding seeding peers. We have implemented a variation of the K-Medoids algorithm described in [11] to accommodate these differences, as listed below. On each round, every advertiser node uses the modified algorithm with several ks and chooses the k that resulted in the lowest score; the seeding peers of this advertisement node for that particular round are the resulting seeding ones computed with the chosen k. In order to account for the communication costs of nodes, in the algorithm below we assign a high cost for messages sent between the advertiser node and the seeding peers.

Denote n the number of nodes in the graph of the partial P2P overlay. Let d_{ij} be the distance between every two nodes i and j. d_{ij} is the amount of hops it takes in the graph (or P2P overlay) to reach from i to j (or vice versa). Suppose that, without loss of generality, the first s nodes are target nodes. The new algorithm consists of the following steps:

1) Step 1: (Select initial seeding peers)
   a) Calculate the distance between every pair of all nodes.
   b) Calculate v_j for node j as follows:
   $$v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{s} d_{il}}, \quad j = 1, ..., n$$
   c) Sort v_j's in ascending order. Select k nodes having the first k smallest values as initial seeding peers.
   d) Obtain the initial cluster result by assigning each target node to the nearest seeding peer.
   e) Calculate the sum of distances from all target nodes to their assigned seeding peers.
2) Step 2:(Update seeding peers)
   a) Find a new seeding peer of each cluster – the node minimizing the total distance to target nodes in its cluster.
   b) Update the current seeding peer in each cluster by replacing with the new seeding peer.
3) Step 3:(Assign target nodes to seeding peers)
   a) Assign each target node to the nearest seeding peer and obtain the cluster result.
   b) Calculate the sum of distances from all target nodes to their seeding peers. If the sum is equal to the previous one, then return the current seeding peer (and stop). Otherwise, go back to Step 2.

D. Advertisement Serving

For each ADM that reaches a node with clients, the node asks each client if the advertisement is suitable for it. The node requests the advertisement from the corresponding advertiser node if at least one client is suitable for the advertisement (i.e., the client is part of the target audience of the advertisement). However, the node periodically removes the lowest bidding advertisements to save space. Each time the user pulls new content from the P2P network, advertisements with the highest bid are chosen to be shown to the user and the respective advertiser nodes are notified of the impression.

III. Design Decisions

A. Payment Models

When a user views or clicks on an advertisement, the advertiser node is being notified. We employ the well known GSP mechanism [12] to calculate the amount that the advertiser is paying for the impression or click. All nodes participating in the dissemination should get a part of that amount. We define four different payment models:

- **Equal Share** - we divide the amount equally among all participating nodes. Since each participating node performed the same amount of work (send a message to a neighbor) all nodes should get equal share of the reward.
- **Equal Referral Share** - the last (serving) node gets half of the amount, and the rest is divided equally between the referring nodes. Using this model, the amount that the serving node is receiving is not affected by the dissemination path length.
- **Balloon challenge [13]** - the last (serving) node gets half of the amount, and each node before it on the dissemination path gets half of what the next node got. Using this model, the proximity to the serving node is rewarded.
- **Bounded share [14]** - we choose an upper bound N and pay the amount/N to the first N-1 nodes on the dissemination path. The rest of the amount goes to the last node. Using this model, an advertiser can limit the length of the dissemination path and also each node can know exactly the amount it would get for a referral and for serving the ad.

For each of these payment models we can invoke different payment schemes:

- **Pay for all** - pay for every impression. Since every impression is profitable for the advertiser, all nodes participating in the dissemination are getting paid for each impression.
- **Pay per *useful neighbor*** - all nodes but the serving node sent a single message regardless of the amount of impressions that the advertisement generated. To reflect this, in this scheme we pay the disseminating nodes once for each useful neighbor. A useful neighbor of a node is a neighbor that received an ADM that eventually produced an impression (either by its own client or by disseminating the ADM further). For each advertisement, every node is paid only once per useful neighbor. The serving node is still paid for every impression since it sends messages for every impression. The amount paid for a useful neighbor can be according to the first impression that this neighbor has produced.
- **Pay max revenue per *useful neighbor*** - another option is to pay for the first impression, but if another impression would have paid more for the same neighbor, then the node gets paid for the difference. In this scheme, each node gets paid for one impression per neighbor - the one with the maximal revenue. This scheme employs the same reasoning as the one before, but does not take into account the order in which the impressions occur.

The same schemes can be applied when paying for clicks. Pay per useful neighbor scheme depends on the order in which the impression happen and thus, inserts more uncertainty to the system. Due to lack of space, in the scope of this work, we look into Equal Share, Equal Referral Share and Balloon Challenge models with Pay for all and Pay max revenue per useful neighbor schemes. As mentioned above, it seems that Pay per useful neighbor would not add much over Pay max revenue per useful neighbor, but is much more sensitive and is thus not investigated further. The Bounded share model has an extra tuning parameter. Thus, we have left its exploration and comparison to future work.

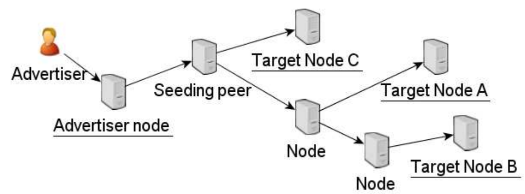


Fig. 2. Payment models example

To get a feel for the different combinations of payment models and payment schemes, consider the scenario depicted in Figure 2 in which a seeding peer has disseminated an ADM with a total payment of $1, which has reached 3 target nodes - A,B and C, each with a single connected client. Suppose that each client has generated 10 impressions. The different models and schemes will generate the following revenues for the seeding peer:

- With the Pay for all payment scheme:
  - Equal Share - the seeding peer would receive $10 * 1/2$ for target node C, $10 * 1/3$ for target node A and $10 * 1/4$ for target node B.
  - Equal Referral Share - the seeding peer would receive $10*1/2$ for target node C, $10*1/4$ for target node A and $10 * 1/6$ for target node B.
  - Balloon challenge - the seeding peer would receive $10*1/4$ for target node C, $10*1/8$ for target node A and $10*1/16$ for target node B.
- With Pay max revenue per useful neighbor:
  - Equal Share - the seeding peer would receive $1/2$ for target node C and $1/3$ for target node A. No payment would be received for node B, since target nodes A and B were reached by the same useful neighbor, and the pay for A is higher than the pay for B.
  - Equal Referral Share - the seeding peer would receive $1/2$ for target node C and $1/4$ for target node A.
  - Balloon challenge - the seeding peer would receive $1/4$ for target node C and $1/8$ for target node A.

B. Dissemination Strategies

Whenever a node receives a new ADM, the receiving node decides to which of the neighbors it will forward the ADM. Lacking prior studies on dissemination strategies for our setting, we first consider the following simple strategy:

- **Random with parameter** r - a trivial solution is whenever a node receives a new ADM, it sends the message to part of the neighbors probabilistically. For each neighbor, the probability to forward an ADM is

r. There is an immediate tradeoff between reach and amount of messages. The higher r is, the higher the reach would be, but, also making the cost higher.

The above simple strategy treats all neighbors the same and mostly serves as a reference to compare with more sophisticated ones. In particular, learning strategies can inspect past ADMs passed to a specific neighbor and decide based on the revenue of those advertisements if the current ADM should be passed to that neighbor. If there is not enough history for a specific neighbor, the advertisement is passed to that neighbor to gather information. Only recent history is considered when making the decision whether to pass the ADM. This ensures that only the latest network state is taken into account. We propose the following learning strategies:

- **AdPrice with parameter** k - a strategy that for every neighbor p employs machine learning techniques on recent advertisements sent to p to predict the revenue for the current ADM. The revenue prediction is then compared to the average revenue of recent ADMs sent to p. The ADM is passed if the revenue prediction divided by the average revenue is higher than k.
- **Probabilistic AdPrice with parameter** $prob$ - a strategy that for every neighbor p employs machine learning techniques on recent advertisements sent to p to predict the revenue for the current ADM. The ADM is passed with probability of $prob$ + revenue prediction divided by average revenue of recent ADMs sent to p .

As an example, suppose a node p has to decide whether to pass an ADM to a neighbor q. The node p predicts a revenue of rev if the ADM is passed to q. The average revenue for p of advertisements recently passed to q is $avgRev$. The different strategies will operate as follows:

- Flood - will pass the ADM.
- AdPrice with parameter k - will pass the ADM if $rev/avgRev > k$.
- Probabilistic AdPrice with parameter $prob$ - will pass the ADM with probability $(rev/avgRev) + prob$.

IV. Dissemination Tracking

A. The Mechanism

When a node receives an ADM, the ADM should include the route this ADM has traveled. This information is needed for the advertiser so he can pay all participating nodes. However, if we simply pass this information as is, then any node on the route can delete all preceding nodes before it from this info, thus getting more credit for all the clicks and impressions of that advertisement further down the line. Hence, a mechanism is needed that will guarantee that no node would be able to alter the route of the ADM.

To that end, the advertiser node includes in the ADM an array, called *path array*, with a length of k. In each cell of the array, the advertiser node generates a random one time pad (OTP) that will be used to encrypt a node ID, an index of a cell, and a bit specifying whether this cell is already used. The advertiser marks $m < k$ cells as used and sends the message to the seeding peers specifying an index of a free cell to use (the advertiser keeps record of this as well). Upon receiving the ADM and an index of the cell to use, each node P does the following:

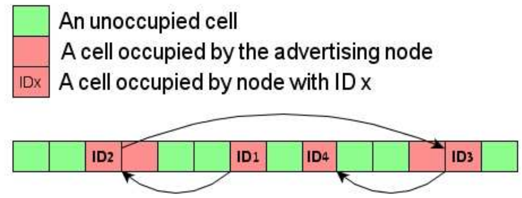


Fig. 3. An example of a path array

1) Chooses randomly a free cell from the array in the ADM for the next node to use.
2) Encrypts the ID of P and the index of the chosen cell for the next node in the cell received from the sending node (by XORing with the OTP in that cell).
3) Marks that cell as used.
4) Chooses neighbors to send the ADM to, and sends them the ADM specifying the unoccupied path array cell chosen for them.

When requesting ad content to present to the user, the node also sends the path array to the advertiser node. The advertiser node decrypts all occupied cells in the path array that the advertiser node did not mark as occupied.

The cells in the path array are linked so that the advertiser would know the order of the dissemination. Some payment models (Balloon challenge) may require the exact order of the node. Also, when a node is paid for an advertisement, it should know the successor that is responsible for that payment in order to evaluate which nodes are good candidates to disseminate the next advertisement.

B. Security Analysis

As mentioned before, all clicks go through the advertiser node. The advertiser node can employ click fraud detection mechanisms [9] to filter fraudulent clicks.

A malicious node may take on different roles to try and gain profit using the advertisement mechanism. This is explored below.

1) Imposing as an advertiser node: All messages sent by legitimate advertiser nodes can be signed by the advertisers making it unfeasible to impose as an advertiser node. We assume that there cannot be a malicious advertiser.

2) A malicious referring node: When receiving an ADM, a node cannot know in which cell its predecessor stored its info in since all the cells containing the path information are encrypted. Also, the node does not know the order between its predecessors. Since the advertiser marked m cells as used, the pth ($p > 1$) node will have a $1/(p+m-1)$ chance of guessing the position of its predecessor (or any other specific cell). Even when colluding with other neighbors of the sending node, the colluding malicious node does not gain any more information since the cell index chosen for all neighbors by the sending node is the same (the cell index is encrypted in step 2).

If a node receives an ADM that the node has already participated in its dissemination, the node can learn the path that this ADM has traveled since it has sent it. But the node cannot use that information to increase its revenue or reduce the revenue of others (without hurting itself).

When receiving an ADM, a malicious node can alter the contents of an occupied cell in the path array. In this case, there is a slight chance that the altered cell will hold a different legitimate ID; otherwise, the altered cell will hold non-existent ID. If an advertiser node receives a path array with non-existent ID, it can ask all the participating nodes to send the path array and the ID of the node that they had sent the ADM to. That way, the advertiser node will find out the missing ID and also will narrow down the nodes that are suspicious of malicious behavior to two - the first node that reported the array with an altered cell and the node before it. Both nodes can be notified of this so that the legitimate node could punish the malicious one (for example by not forwarding ADMs to it). To handle the former case (altered cell holds legitimate ID), an advertiser node may once in a while ask the participating nodes to send the path array and the ID of the node that they had sent the ADM to, even if the path array holds only legitimate IDs.

If two nodes (A and B) are working together to cheat the system, every time A receives an advertisement, A can write B also in the array. There is a chance that B is already in the array. In that case, the advertiser can again ask all the participating nodes to send the path array and the ID of the node that they had sent the ADM to, and, narrow down the nodes that are suspicious of malicious behavior to two. Otherwise, (if B is not present) the advertiser node can not tell that cheating has occurred. To prevent this scenario, the path array can be altered to hold ID of node that the ADM was received from, ID of current node and ID of node that the ADM is sent to. That way, a node cannot add any other IDs before or after it in the path array at the cost of increased message size. Another option is for A to check if B is already in the array. This communication would require A to send a message to B and wait for a response, while sending the ADM to B would potentially have the same effect and use only one message. In other words, this type of cheating costs more than following the rules.

3) A malicious serving node: A malicious serving node can request for advertisements even though there are no users interested in the advertisement (or no users at all) connected to the node. It that case, the advertiser node should notice that the CTR of the node is marginally lower than the CTR of other nodes and suspect the node of malicious requests. Suspected nodes can be punished (for instance, have all their requests rejected for some time). As mentioned before, all clicks also go through the advertiser node, so fraudulent clicks can also be detected.

V. Performance Evaluation

A. Model and Setup

The simulation of the different elements is divided into dissemination rounds. Each round starts with the advertiser nodes sending new ADMs to the respective seeding peers. The bids for the ADMs are generated using information from [15]. During the round, the ADMs are being disseminated and impressions are being simulated (a constant amount of impressions for the 3 highest bidding advertisements for every online user). When the round ends, the advertising peers send a payment notice to all nodes that are entitled to any payment specifying the advertisement for which the node is being paid. If the payment is for a referral, the referred node is also specified.

In order to model churn, every node receives on start up a number of consecutive rounds that the node should be responsive for until it fails, called *session length*. These lengths are assigned to the nodes randomly from a Weibull distributed variable, which was reported to represent well real P2P networks' churn [16], [17], with parameter $k = 0.5$ and a specific mean. Further, in order to avoid a situation in which all nodes are up on the first round, every node also receives on start up the first round in which it should fail. After a round in which a node was in a failed state, the node rejoins the network as a brand new node with new users, no history and the same session length.

The various dissemination strategies, payment models and the tracking mechanism are fully implemented in Java and the source code is available online[1]. Only the messages between the peers are simulated using the PeerSim simulator[2]. We simulate a P2P network with 512 nodes and the network is wired as a hypercube, which functions as a portion of a P2P network and thus has only one advertiser node per advertiser. 5 such nodes act as advertiser nodes for target audiences of various sizes. The history length of the learning strategies is set to 20 rounds (as elaborated below).

B. Definitions

- *Round reach rate* is defined to be the ratio of users receiving an advertisement out of all users that are the target audience for that advertisement for a specific round.
- *Round miss rate* is defined to be the ratio of users not receiving an advertisement out of all users that are the target audience for that advertisement for a specific round.
- *Average reach rate* is the average round reach rate.
- *Average miss rate* is the average round miss rate.
- *History length* - the amount of latest rounds the learning strategies take into account when making predictions.

C. Results

a) Random r parameter tuning: We have tested the random strategy with different r parameters. We have used the Balloon Challenge payment model with pay for all impressions payment scheme. As we can see from Figure 4, we get low (< 0.05) round miss rate starting from $r = 0.4$. There is a

[1] https://sourceforge.net/projects/adflow/

[2] http://peersim.sourceforge.net/

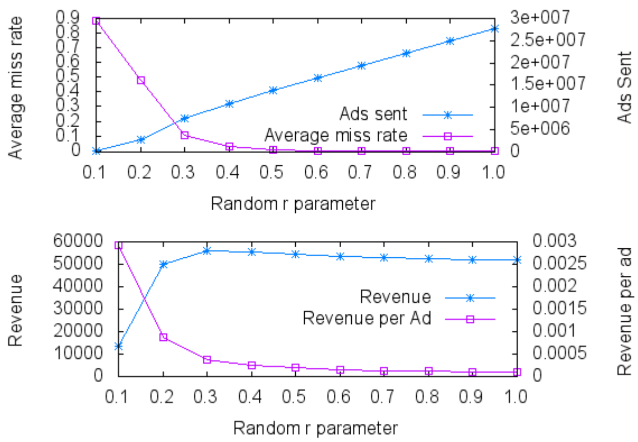


Fig. 4. Overall miss rate, ads sent, revenue and revenue per ad as a function of the value of the r parameter of the random strategy

steady increase in the amount of advertisements sent with the increase of r. Also, we note a decrease in the revenue per ad with the increase of r. The total revenue is increasing with the reach up to $r = 0.3$. Then, when the reach is high (> 0.9), the total revenue decreases. This happens because the Pay for all payment scheme pays for referrals proportionately to the path lengths.

b) AdPrice p parameter tuning: For all values of $p < 1$ that we have tested, the behavior is nearly identical; it can be seen as the second (righthand side) set of graphs in Figure 5. That is, initially the miss rate is high and the message cost is relatively low. Yet, as time goes by and the learning improves, the miss rate drops to almost 0 but the message cost increases. When $p >= 1$, the round miss rate becomes much higher than with any of the other schemes we tested, more than 0.2 (graph omitted for brevity).

c) Average miss rate versus messages sent: In Figure 5, we analyze closely the behavior of the different dissemination strategies. The payment model is Balloon Challenge with pay for all impressions payment scheme. The Random strategies send approximately the same amount of messages for every advertisement. They also maintain the same round miss rate with little variance from the average. The behavior of the AdPrice strategies is cyclic. Every cycle the average amount of messages sent per advertisement increases and the round miss rate decreases. The size of the cycles is exactly the history length of each strategy. However, the amount of messages sent per advertisement increase logarithmically. As shown in Figure 5, for every round miss rate, there are random and learning strategies that can eventually provide that round miss rate. Yet, in doing so, the learning strategies eventually send fewer messages per advertisement than the random ones while providing the same round miss rate.

d) Impact of churn: We have tested the AdPrice0.0 and Random0.6 strategies with different churn settings. We have ran tests with session length means of 75, 150 and 300 rounds. Figure 6 shows that both strategies maintain their average miss rate behavior. However, when increasing the session length mean (decreasing the churn), the random strategy sends out more messages per advertisement while the AdPrice strategy sends the same amount.

e) Impact of payment model: We have tested the randomStrategy0.6 with different payment models. In all payment modes, the average miss rate and the amount of messages sent per advertisement are similar to Balloon Challenge pay model with pay for all payment scheme shown in Figure 5. In Figure 7, we present an exponential fit to the revenue results. The variance of the exponential fit is very low (0.00001-0.02) and is lower than fitting to a linear plot (which is in turn lower than fitting to a logarithmic plot). We can observe the desired result of nodes receiving payment proportionally to their availability. In particular, methods whose curve is higher in this graph are better as incentives to participate, since with these schemes the reward for continuous participation is higher.

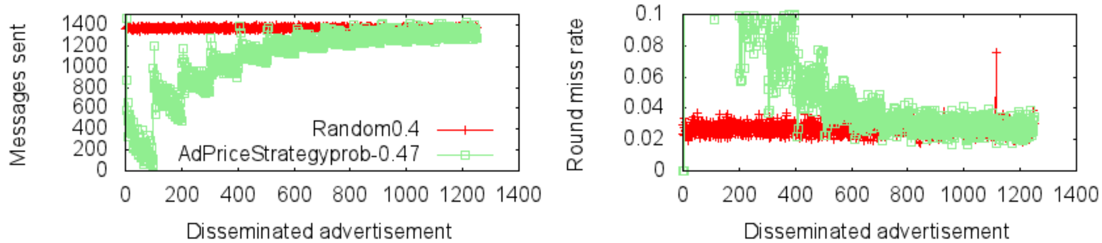
As can be further seen in Figure 7, payment models that pay for all impressions pay more for availability and their base of the exponent is higher. Equal reference and Balloon challenge models behave in a similar manner. Both models pay half of the amount to the serving node. When paying for all impressions, Equal reference model pays a little more since Balloon challenge model does not distribute all the amount between the disseminating nodes ($1/2^{disseminationLength}$ of the pay amount is always left undistributed). However, when paying only for maximal impression, Balloon challenge model pays more for availability than Equal reference model. Using the Pay max revenue for useful neighbor scheme, Equal reference model pays every node for the shortest dissemination path that the node participated in, while Balloon challenge pays for the closest target node. Equal share model pays marginally more for availability than both Balloon challenge and Equal reference models in both payment schemes. When employing the AdPrice strategy, the payments per availability are a bit lower than those in Figure 7 due to the time it takes for the strategy to reach a comparable round miss rate. Further, the variance of the fit is larger (0.00005-0.03). This exponential behavior is enforced by the seeding peers that take into account nodes that are unavailable when choosing potential seeding peers.

VI. Related Work

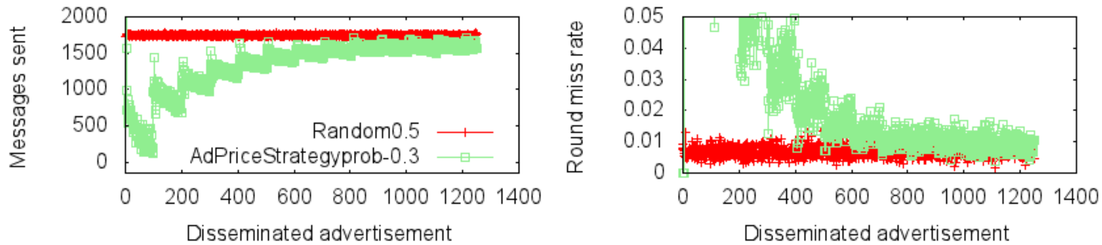
Pub/Sub systems [18] are used to efficiently disseminate events generated by publishers to subscribers. Yet, in our model, clients do not explicitly subscribe to a topic or content as done in Pub/Sub systems. Instead, when a client machine is made aware of a disseminated ad, it can decide whether the ad is suitable for its user or not. Further, existing Pub/Sub systems do not track the dissemination paths since they need not worry about paying nodes that participated in the dissemination. Similarly, known works on Pub/Sub did not investigate payment models since it is not in their scope.

In [19] and the followup work [20], the authors disseminate messages to target audiences. However, the dissemination is carried out over the social links rather than the P2P links. Also, in order to disseminate, the users share information with their immediate social neighbors. Lastly, the authors of these works did not simulate churn.

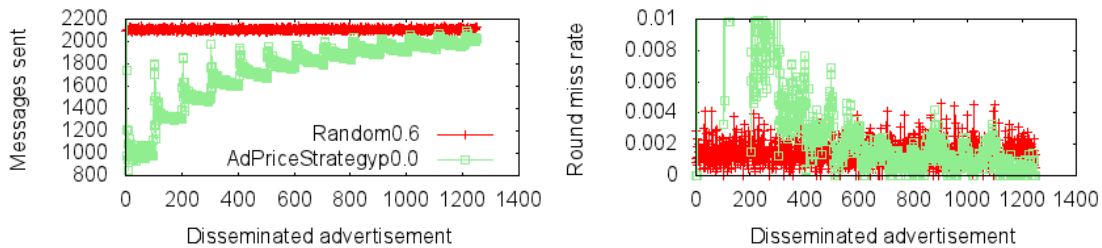
In [21], the authors discuss instant and location-aware commercials. The paper presents an opportunistic gossiping model for disseminating instant advertisements, including a



(a) Algorithms reaching round miss rate of about 0.03



(b) Algorithms reaching round miss rate of about 0.005



(c) Algorithms reaching round miss rate of about 0.0001

Fig. 5. Round miss rate and messages sent per advertisement throughout the rounds

probability function for determining advertisement forwarding probability at different locations. In their work, they were able to provide high delivery rate of advertisements while keeping the delivery time and the number of messages low. However, [21] addresses dissemination between a network of mobile devices. Hence, they used distance and velocity information to optimize the gossiping model, information that cannot be used to optimize our P2P dissemination strategies.

[22] introduces a new social gossip protocol. As a recommendation travels from one user to the next, its relevance decreases. Once a certain hop-count limit is reached, the relevance goes to zero and the message dissemination stops. The adoption criterion of accepting only $f+1$ disjoint gossip paths protects the network from spam recommendations. The main contribution is a practical path verification protocol whose computation and storage complexities are polynomial in n. In our work, we assume that no spam advertisements are passed (they could be signed by the advertisers). Hence, we accept the first ad received as genuine.

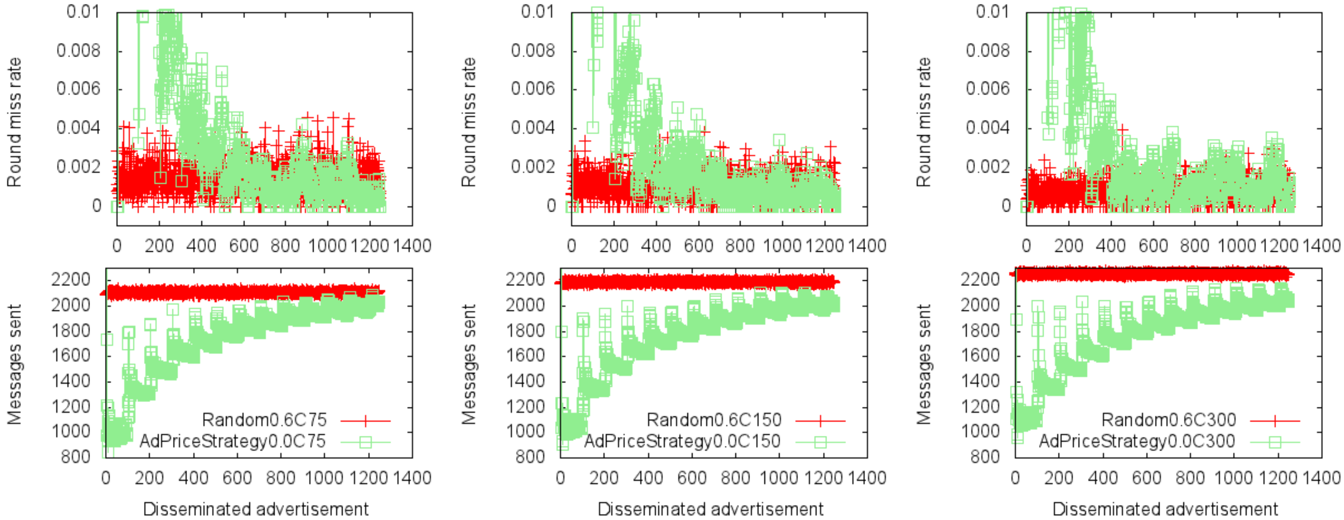
A recent incentive mechanism for P2P is presented in [23]. However, their model is designed to run over any type of graph structure that can be sub-grouped and managed by a super-peer, which is not the case in our work.

A P2P publish/subscribe technique called Pub-2-Sub [24] can be used to disseminate advertisements in a P2P network. Pub-2-Sub assigns to each node a unique binary string called a virtual address so that the virtual addresses of all the nodes form a prefix tree. Based on this tree, each node is assigned a unique zone partitioned from the universe of binary strings. Then, later queries and publications are hashed to binary strings and, based on their overlapping with the node zones, subscription and notification paths are chosen appropriately and deterministically. Unfortunately, Pub-2-Sub can work well only when the network is stable and cooperative, since any churn (peers leaving and joining) triggers a restructure of parts of the prefix tree. Also, rogue nodes (if placed high enough in the prefix tree) can affect many peers.

A well conceived non economical incentive mechanism for message relaying of service requests in a P2P network is described in [25]. In that mechanism, promised rewards are passed along the message propagation process and after a service provider was reached - a rewarding process is propagated backwards on the same route. However, in our model, the advertiser disseminates the advertisements and is not a service provider whom the users are trying to reach.

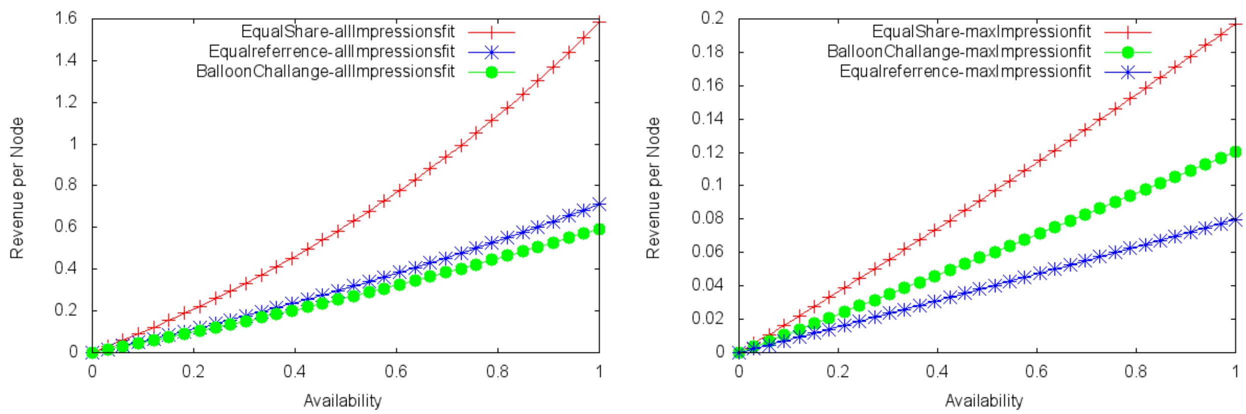
VII. Discussion

In this paper, we have provided a realistic model of a P2P social network advertisement dissemination mechanism. We have designed and implemented a dissemination method so



(a) Mean session length of 75 (b) Mean session length of 150 (c) Mean session length of 300

Fig. 6. The effect of churn on round miss rate and messages sent per advertisement



(a) All impressions payment (b) Max impression payment (note Y-axis scale)

Fig. 7. Nodes' revenue as a function of nodes' availability for all tested payment models

that the dissemination is carried out while cheating is difficult. We have presented a heuristic for finding a minimum sum of distances between the seeding peers and the target nodes. We have defined several payment models and compared between them. We have also introduced and thoroughly tested different dissemination methods.

When trying to make general observations about the benefit of the learning dissemination schemes vs. the random dissemination ones, we can point out the following: For every required minimal average miss rate, we can eventually achieve this miss rate with a learning strategy while sending fewer messages per round than the random strategy that achieves the required average miss rate. Further, the benefit of the learning strategies over the random ones increases as the churn rates decrease. This is because when the network changes too fast, by the time something is learned about the network, it is no longer relevant. Hopefully, with a paying advertisements based incentive mechanism, the churn rates in a P2P network will indeed not be too high.

Finally, we have shown mechanisms that reward peer nodes in an exponential proportion to their availability. This was an important goal of our work, since it shows that ads can serve as a real incentive for users to keep their donated machines connected to the network for long periods of time.

In the future, we would like to examine different optimization functions for seeding peer selection and possibly different heuristics for each one of them. More sophisticated gossip algorithms [26] can be used to disseminate the ads. We would like to test how the different payment models affect the dissemination. More work can be done to ensure the exponential payments for availability, i.e., that nodes can take availability into account when dissemination ADMs. Different network overlays can also affect the mechanism. For instance, the overlay achieved by Kademlia [27] is not a hypercube. Hence, we would like to test the effect of the overlay on these mechanisms. Finally, we intend to further explore the tradeoffs between desired miss rates and communication overhead.

The model and mechanisms described in this paper may be extendible to other types of streams of messages disseminated over an existing P2P overlay to interested audiences. In order for these methods to be feasible, the disseminated message (ADM) should always be kept small while the actual content (advertisements in our case) can be larger since they are sent directly from the source (advertiser node) to the client. This is left for future work.

Acknowledgements: We would like to thank the anonymous reviewers and our shepherd Toru Hasegawa for their useful comments and insights. This work is partially supported by ISF grant 1247/09 and the Technion Hasso Platner Institute (HPI) Research School.

REFERENCES

[1] M. Feldman and J. Chuang, "Overcoming free-riding behavior in peer-to-peer systems," *ACM SIGecom Exchanges*, vol. 5, no. 4, pp. 41–50, 2005.

[2] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *IEEE Jr. on Selected Areas in Communications*, vol. 24, no. 5, pp. 1010–1019, 2006.

[3] B. Cohen, "Incentives build robustness in bittorrent," 2003.

[4] A. Mondal, S. Madria, and M. Kitsuregawa, "An economic incentive model for encouraging peer collaboration in mobile-p2p networks with support for constraint queries," *Peer-to-peer networking and applications*, vol. 2, no. 3, pp. 230–251, 2009.

[5] J. Rius, F. Cores, and F. Solsona, "A new credit-based incentive mechanism for p2p scheduling with user modeling," in *Advances in P2P Systems, 2009. AP2PS'09. First International Conference on*. IEEE, 2009, pp. 85–91.

[6] M. Salek, S. Shayandeh, and D. Kempe, "You share, i share: network effects and economic incentives in p2p file-sharing systems," *Internet and Network Economics*, vol. 6484, pp. 354–365, 2010.

[7] G. Tan and S. Jarvis, "A payment-based incentive and service differentiation scheme for peer-to-peer streaming broadcast," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 7, pp. 940–953, 2008.

[8] T. Nguyen, S. Cheung, and D. Tran, "Efficient p2p data dissemination in a homogeneous capacity network using structured mesh," in *IEEE International conference on multimedia access networks*, 2005.

[9] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar, "Click fraud resistant methods for learning click-through rates," in *Internet and Network Economics*, ser. Lecture Notes in Computer Science, X. Deng and Y. Ye, Eds. Springer Berlin / Heidelberg, 2005, vol. 3828, pp. 34–45.

[10] L. Kaufman, P. Rousseeuw, D. D. o. M. Technische Hogeschool, and Informatics., "Clustering by means of medoids," Technische Hogeschool, Delft. Dept. of Mathematics and Informatics., Tech. Rep., 1987.

[11] H. Park and C. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[12] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords," National Bureau of Economic Research, Working Paper 11765, November 2005. [Online]. Available: http://www.nber.org/papers/w11765

[13] J. C. Tang, M. Cebrian, N. A. Giacobe, H.-W. Kim, T. Kim, and D. B. Wickert, "Reflecting on the darpa red balloon challenge," *Commun. ACM*, vol. 54, no. 4, pp. 78–85, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1924421.1924441

[14] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar, "On bitcoin and red balloons," *ArXiv e-prints*, nov 2011.

[15] "Facebook advertising performance benchmarks & insights," http://www.brandchannel.com/images/papers/522_2011-01_webtrends_facebook-ads.pdf.

[16] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, ser. IMC '06. New York, NY, USA: ACM, 2006, pp. 189–202. [Online]. Available: http://doi.acm.org/10.1145/1177080.1177105

[17] M. Steiner, T. En-Najjary, and E. Biersack, "Long term study of peer behavior in the kad dht," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1371–1384, 2009.

[18] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[19] A. Datta and R. Sharma, "Godisco: Selective gossip based dissemination of information in social community based overlays," *Distributed Computing and Networking*, pp. 227–238, 2011.

[20] R. Sharma and A. Datta, "Decentralized information dissemination in multidimensional semantic social overlays," *Distributed Computing and Networking*, pp. 473–487, 2012.

[21] Z. Chen, H. T. Shen, Q. Xu, and X. Zhou, "Instant advertising in mobile peer-to-peer networks," *Data Engineering, International Conference on*, vol. 0, pp. 736–747, 2009.

[22] Y. Fernandess and D. Malkhi, "On spreading recommendations via social gossip," in *Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*. ACM, 2008, pp. 91–97.

[23] J. Rius, I. Barri, F. Cores, and F. Solsona, "A formal credit-based incentive model for sharing computer resources," *Euro-Par 2010-Parallel Processing*, pp. 497–509, 2010.

[24] D. A. Tran and C. Pham, "Pub-2-sub: A content-based publish/subscribe framework for cooperative p2p networks," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, ser. NETWORKING '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 770–781. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01399-7_60

[25] C. Li, B. Yu, and K. Sycara, "An incentive mechanism for message relaying in unstructured peer-to-peer systems," *Electronic Commerce Research and Applications*, vol. 8, no. 6, pp. 315 – 326, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1567422309000234

[26] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[27] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *Peer-to-Peer Systems*, pp. 53–65, 2002.