Targeted Coupon Distribution Using Social Networks

Srinivas Shakkottai
Dept. of ECE
Texas A&M University
sshakkot@tamu.edu

Lei Ying
Dept. of ECE
Iowa State University
leiying@iastate.edu

Sankalp Sah
Dept. of ECE
Texas A&M University
sankalpsah@neo.tamu.edu

ABSTRACT

We propose to exploit online social networks in order to reveal user preferences such that goods and services might be targeted appropriately. We consider goods and services that are consumed periodically, and a user model where users could buy different numbers of goods at a marked ("high") price and at a discounted ("low") price. The problem is to identify users who are interested in the good, and to discount them sufficiently that the entire demand is realized, but not so much that revenue is compromised. Using techniques from "backpressure routing" used for control of multihop communication networks, we develop a multi-hop coupon forwarding scheme in which users are incentivized to forward the right number of coupons to appropriate neighbors such that the above target is achieved. We also develop a simple delay-based heuristic algorithm that also shows a near-optimal performance.

1. INTRODUCTION

Online social networks are increasingly being seen by marketing firms as a means of obtaining information about user preferences. For example, displaying selected advertisements and observing user responses can be used to explore an individual user's tastes. Such information would help marketers to target selected goods and services at particular users, so increasing the efficiency of an advertising campaign.

Online social networks could also be used in a slightly different fashion to create a market with differentiated prices, based on users' consumption capacity. Consider goods or services that are consumed on a periodic basis such as movie tickets, car washes, fitness club visits etc. There could be users who would purchase goods at the marked ("high") price, and others who might need to be subsidized to some extent ("low" price) in order to make a purchase. Such differentiated prices can be achieved through the distribution of discount coupons. However, it is not clear which users should be given such coupons, or how many coupons should be given to them. The questions are not easy to answer since the online retailer is unlikely to know the consumption capacity of users, and perhaps not be even aware that they exist. Flooding coupons is not a good solution since giving too many coupons to users would be a disincentive to purchase goods at the marked price, even if they are capable of doing so. Similarly, restricting the coupon generation rate to a small number might not achieve the true market potential. Finally, user preferences and available goods change with time, which makes the whole system dynamic.

In this paper we propose a scheme wherein some number of users register with online stores in order to receive coupons for goods that they may not necessarily be interested in. Each coupon has a unique identity and may be redeemed at a store in order to purchase the good it corresponds to. Users may forward such coupons to any of their friends on the social network, who in turn could forward it to their friends and so on. Eventually, the coupon must reach a user (a leaf-node) who actually would use it to make a purchase. Questions that require algorithmic solutions are (i) at what rate should coupons be injected into the system? (ii) to which neighbor should a user who possesses coupon forward it to, and what incentives would cause her to do so? and (iii) how do we ensure that customers do not obtain too many or too few coupons? An example of such a system in practice is *mGinger* [1] that acts as a multi-hop advertising and discount distribution system using SMS messages, with rewards being paid in a pyramidal fashion to forwarders. In this paper, we develop revenue maximizing, incentive compatible schemes based on ideas of backpressure [2] that has been used as a throughput optimal scheme for packet routing in multihop wireless networks [3, 4, 5, 6]. Our system is described as follows.

2. SYSTEM MODEL

Denote by $\mathcal{N}$ the set of nodes and $\mathcal{L}$ the set of links. There are three different node types — coupon distributor, users, and store — in the network. The links represent social connections. A link from a coupon source to a user represents the idea that the user has registered with the source to receive its coupon periodically. A link from a user to a product means that the user buys that product periodically. The links between users are assumed to be bidirectional, and represent friendship between the connected users. In this paper, we assume that the coupon sources and the store are managed by the same entity. We use s to denote the store and d to denote the coupon distributor. We define $\mathcal{S}$ to be the set of products and $\mathcal{B}_i$ is the set of users who will buy product i (these are the customers). A customer who presents a coupon obtains a discount. Users are rewarded if they forward coupons to the "right" neighbor (how this neighbor is chosen, and how much reward is given will be described in Section 4)[1]. We consider a synchronized slotted-time system. We define μ_j to be the coupon transmission capacity of node j, which is the maximum number of coupons user j can send out in one time-step.

We consider two different time scales in this paper. The small time scale t is the one in which purchases are made and coupons are delivered. The large time-scale, consisting of T small time slots, is the buying interval after which coupons expire and the customers start afresh. For example, we may think of t as being of the order of an hour, while T could be on the order of a week. Good i might be sold at the marked ("high") price p^i or at the discounted ("low") price q^i. We assume that these prices are such that the store makes a profit regardless of whether the good is sold at the high or low price. Of course, the profit made by selling goods at the high price is higher than at the low price. We assume that the capacity for consumption of a good i by user j (per unit time at scale t), can be divided naturally into two parts—one at at the high price $\hat{x}_j^{ih}$, and one at the low price $\hat{x}_j^{il}$. We denote the total as $b_j^i = \hat{x}_j^{ih} + \hat{x}_j^{il}$. We assume that these values are fixed for some multiple of T, and so can be learned. The number of coupons given to a customer must be carefully regulated; if it is larger than $\hat{x}_j^{il}$, the store loses profits due to excessive discounting (since the customer would not buy as many high-price goods as possible), while if it is less than $\hat{x}_j^{il}$, the entire demand is not realized.

3. PROFIT MAXIMIZATION

We first develop expressions corresponding to the profit made by the store. We say a coupon is *valid* if it is eventually used to purchase a product. We denote by $y^i_{(m,n)}$ the average number of valid type-i coupons sent from user m to user n in a time slot. The profit the store extracts from user j is

$$q^i y^i_{(j,s)} + p^i \min\left\{\hat{x}_j^{ih}, b_j^i - y^i_{(j,s)}\right\}.$$

Thus, the maximum profit the store can extract is defined by the following optimization problem:

OPT 1:

$$\max \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{B}_i} \left(q^i y^i_{(j,s)} + p^i \min\left\{\hat{x}_j^{ih}, b_j^i - y^i_{(j,s)}\right\}\right) \quad (1)$$

$$s.t. \quad \sum_{i\in\mathcal{S}} \sum_{j:(m,j)\in\mathcal{L}, j\neq s_i} y^i_{(m,j)} \leq \mu_m, \forall m \in \mathcal{N} \quad (2)$$

$$\sum_{j:(m,j)\in\mathcal{L}} y^i_{(m,j)} = \sum_{n:(n,m)\in\mathcal{L}} y^i_{(n,m)} \forall m \in \mathcal{N} \quad (3)$$

$$y^i_{(m,j)} = 0 \text{ if } m \in \mathcal{B}_i \text{ and } j \neq s \quad (4)$$

where (2) is the capacity constraint, which indicates node m cannot send more than μ_m coupons in a time-slot, (3) is the flow-conservation constraint for the coupons, and (4) indicates that user j will not forward type-i coupons to her neighbors if she uses type-i coupons. Note that we have not explicitly included the total reward paid out to coupon forwarders since this value is a constant and can be subsumed into q^i.

We need to answer the following two questions in order to maximize revenue. First, the value of $\hat{x}_j^{ih}$ and $\hat{x}_j^{il}$ are unknown at the store, and need to be learned. Second, since all users interested in a good may not be registered to receive coupons directly from the coupon generator, the store cannot directly control the number of coupons sent to such users. The next section develops a coupon distribution scheme that consists of two control loops corresponding to the two time scales.

4. COUPON DISTRIBUTION

The control loops in our coupon distribution algorithm can be described as follows:

1. **Choice of Coupon Forwarding Reward Rate:** At the large time scale, each store must adapt the target rate γ_j^i for the next buying interval using the information gathered about the customers' preferences over the past intervals. In our algorithm, γ_j^i is an estimate of $\hat{x}_j^{il}$. If γ_j^i is set too low, customer j may not purchase all the goods that she potentially could, and if γ_j^i is too high, customer j may be being discounted excessively and the store is not extracting the maximum extractable revenue.

2. **Coupon Routing at Target Rate:** At the small time scale, a store attempts to meet the current target rate of coupon delivery γ_j^i for each product i and each customer j that purchases goods from it. As we will see below, it does this by setting the rate at which it rewards neighbors of customer j as γ_j^i. Mathematically, we will guarantee that the coupon distribution algorithm solves the following optimization problem:

OPT 2:

$$\max \sum_{i\in\mathcal{S}} q^i \left(\sum_{j\in\mathcal{B}_i} y^i_{j,s_i}\right) \quad (5)$$

$$s.t. \quad \sum_{i\in\mathcal{S}} \sum_{j:(m,j)\in\mathcal{L}, j\neq s_i} y^i_{(m,j)} \leq \mu_m, \forall m \in \mathcal{N} \quad (6)$$

$$\sum_{j:(m,j)\in\mathcal{L}} y^i_{(m,j)} = \sum_{n:(n,m)\in\mathcal{L}} y^i_{(n,m)} \forall m \in \mathcal{N} \quad (7)$$

$$y^i_{(j,s)} \leq \gamma_j^i \quad \forall j, i \quad (8)$$

$$y^i_{(m,j)} = 0 \text{ if } m \in \mathcal{B}_i \text{ and } j \neq s \quad (9)$$

It is easy to show that given that $\gamma_j^i = \hat{x}_j^{il}$ for all i and j, OPT 1 is equivalent to OPT 2. Next, we develop a distributed coupon routing algorithm that solves OPT 2.

4.1 Backpressure-based coupon routing at the small time scale

We first introduce the coupon management scheme which consists of three parts: (i) each user maintains a per-product queue, and monitors the lengths of the queues; (ii) store rewards the neighbors that forwarded type i coupons used by each customer j at a rate γ_j^i, and monitors the number of unrewarded coupons at each customer; (iii) coupon generator i monitors the number of coupons it has not yet sent out, and generates additional coupons based on this value.

A1: Coupon Management:

9[1]Note that we use the term "discount" to indicate customer remuneration when a coupon is redeemed during purchase of goods, and "reward" to indicate remuneration to users who forwarded the coupon.

(1) Per-product queues are maintained at each user, and the number of type i coupons user j has at a finer time-step t is denoted by $Q_j^i[t]$. Thus, the dynamic of $Q_j^i[t]$ is as follows: If $j \notin \mathcal{B}_i$, then

$$\begin{aligned} & Q_j^i[t+1] \\ = & \left(Q_j^i[t] + \sum_{m:(m,j)\in\mathcal{L}} y_{(m,j)}^i[t] - \sum_{n:(j,n)\in\mathcal{L}} y_{(j,n)}^i[t] \right)^+ ; \end{aligned}$$

otherwise

$$Q_j^i[t+1] = Q_j^i[t] + \sum_{m:(m,j)\in\mathcal{L}} y_{(m,j)}^i[t] - y_{(j,s)}^i[t],$$

where

$$\begin{aligned} y_{(j,s)}^i[t] &= \min\left\{ Q_j^i[t] + \sum_{m:(m,j)\in\mathcal{L}} y_{(m,j)}^i[t], \right. \\ & \left. \left(b_j^i T - \sum_{\tau=0}^{t-1} y_{(j,s)}^i[\tau] \right)^+ \right\}, \end{aligned}$$

i.e., user j will use up all available coupons unless she has already bought enough ($b_j^i T$) products.

(2) Store maintains a queue for unrewarded coupons corresponding to each of product i and its customers j. We may think of these as *virtual coupons* that are used to maintain a pressure on j's neighbors. Note that it is only the *neighbors* of j that are not rewarded for forwarding these coupons, the rest of the users involved in forwarding coupons would be guaranteed a reward (and, of course, j has already redeemed these coupons for a discount). Denote by $\tilde{Q}_j^i[t]$ the number of such unrewarded coupons corresponding to customer j. We have

$$\tilde{Q}_j^i[t+1] = \left(\tilde{Q}_j^i[t] + y_{(j,s)}^i[t] - \gamma_j^i \right)^+,$$

where γ_j^i is the coupon forwarding reward rate for neighbors of customer j.

(3) Coupon distributor d maintains a separate queue for each type of coupons that have not been sent out. The length of the queue is denoted by $\tilde{Q}_d^i[t]$ for each product i. We have

$$Q_d^i[t+1] = \left(Q_d^i[t] + \Theta^i[t] - \sum_{j:(d_i,j)\in\mathcal{L}} y_{(d,j)}^i[t] \right)^+,$$

where $\Theta^i[t]$ is the number of new type i coupons generated by coupon distributor i at time t.

(4) We also assume that when user j receives a type i coupon such that $j \notin \mathcal{B}_i$, user j will insert her identity and the coupon queue length $Q_j^i[t]$ in the coupon before sending the coupon to her neighbor. This information allows the store to reconstruct path and reward the coupon relays based on $Q_j^i[t]$.

In our system the store needs to reward coupon fowarding in order to motivate users to forward coupons to their friends. The efficiency of a coupon distribution scheme is determined by: (i) the incentive scheme that the stores use, and (ii) the users' decisions under the incentive scheme. Next, we propose a coupon rewarding scheme, under which a rational user will distribute the coupons according to a backpressure policy. The optimality of the coupon distribution scheme will be proved in Theorem 1.

A2: Reward Scheme for Coupon Forwarding: The store rewards the users involved in forwarding each used type i coupon with a total of α^i dollars. Consider a specific coupon associated with product i, and assume $\mathcal{R}$ is the path (consisting of the sequence of transmissions used to distribute the coupon) over which the coupon was transferred. Then node m gets a reward

$$\left(Q_m^i - Q_{n:(m,n)\in\mathcal{R}}^i \right)^+ \frac{\alpha^i}{\sum_{l\in\mathcal{R}} \left(Q_{s(l)}^i - Q_{r(l)}^i \right)^+}, \qquad (10)$$

where l is a link on path $\mathcal{R}$, $s(l)$ is the sender, and $r(l)$ is the receiver. Note that this queue length information is inserted by the users before they forward the coupons to their neighbors. Furthermore, note that the amount of reward user m obtains is proportional to the queue difference. The idea is to motivate user m to send her coupon to a neighbor who has the least number of coupons and hence is most likely the one who needs the coupon. Under this scheme, the user has the motivation to follow the backpressure-like coupon distribution scheme.

A3: User Behavior:

(1) First, if node j is interested in buying product i, then user j uses all available type i coupons up to her purchasing limit b_j^i. Thus, at finer time-step t, user j purchases $y_{(j,s_i)}^i[t]$ products with coupons such that

$$\begin{aligned} y_{(j,s)}^i[t] &= \min\left\{ Q_j^i[t] + \sum_{m:(m,j)\in\mathcal{L}} y_{(m,j)}^i[t], \right. \\ & \left. \left(b_j^i T - \sum_{\tau=0}^{t} y_{(j,s)}^i[\tau] \right)^+ \right\}, \end{aligned}$$

We assume that user j never forwards type-i coupons to her neighbors if user j buys product i.

(2) If user j is not a customer buying product i, then user j needs to distribute type i coupons to her neighbors. We assume that at the beginning of finer time-step t, user j requests $Q_m^i[t]$ if user m is her neighbor, and also polls the store to obtain $\tilde{Q}_m^i[t]$. Since the amount of coupon forwarding reward is determined by the queue difference as described in (10), user j selects a coupon type i^* and neighbor m^* such that

$$\begin{aligned} & (i^*, m^*) \in \\ & \arg\max_{(j,m)\in\mathcal{L}} \left(Q_j^i[t] + \tilde{Q}_j^i[t] - Q_m^i[t] - \tilde{Q}_m^i[t] \right), \end{aligned}$$

and transfers $\min\left\{ \mu_j, Q_j^{i^*}[t] \right\}$ of type i^* coupons to node m^*.

Since $\tilde{Q}_m^i[t]$ is the number of coupons that have been used by user m but have not yet been rewarded, $\tilde{Q}_m^i[t] = 0$ if user m is not a customer buying product i. A store maintains this unrewarded coupon queue to prevent a customer receiving too many coupons. When user j uses too many coupons, the unrewarded coupon queue becomes large. After a neighbor of user j finds a large $\tilde{Q}_j^i[t]$, the neighbor realizes that user j has received too many coupons and the store might not reward him for forwarding coupons to user j. Then the neighbor will stop forwarding more coupons to user j.

A4: Coupon Generation Scheme: The coupon distributor needs to decide the number of coupons to inject into the network. We assume that coupon distributor generates μ_d type-i coupons when $Q_d^i[t] \leq Q_T q^i$, and zero type-i coupon otherwise. Here, Q_T is a constant threshold value. In other words, $\Theta^i[t] = \mu_d$ if $Q_d^i[t] \leq Q_T q^i$, and $\Theta^i[t] = 0$ otherwise.

In the following theorem, we analyze the performance of the backpressure coupon routing, and prove that

THEOREM 1. *Assume that $\gamma_j^i \leq b_j^i$ for all i and j. Under the coupon management, coupon rewarding and generating scheme, and user behavior defined above, we have*

$$\lim_{Q_T \to \infty} \lim_{T \to \infty} \frac{\sum_{t=1}^{T} \Theta^i[t]}{T} = \left(\sum_{j \in \mathcal{B}_i} \check{y}^i_{(j,s)} \right), \tag{11}$$

and

$$\lim_{Q_T \to \infty} \lim_{T \to \infty} \frac{\sum_{t=1}^{T} y^i_{(j,s)}[t]}{T} = \check{y}^i_{(j,s)}, \tag{12}$$

where $\check{\mathbf{y}}$ is the optimal solution of OPT 2.

PROOF. We skip the proof details due to space constraints. Interested readers can find the details in [7]. □

Note that although the theorem is an asymptotic result, the algorithm itself works for any value of T. A finite value of T may result in a sub-optimal solution. In our simulations, we choose $T = 300$ and the final coupon allocation is very close to the optimal.

4.2 Coupon rate selection at the large time scale

We assume that the algorithm for coupon delivery at the small time scale converges quickly to the target rate, and now consider how to choose this target rate. Recall that our means of implementing coupon delivery at rate γ_j^i is to reward the neighbors of a customer j for forwarding coupons to j at rate γ_j^i. In this section all dynamics take place at the large time. Thus, we have the sequence of target rates $\gamma_j^i[0], \cdots, \gamma_j^i[k-1], \gamma_j^i[k], \gamma_j^i[k+1], \cdots$, and the large time scale algorithm needs to guarantee that $\lim_{k\to\infty} \hat{\gamma}_j^i[k] = \hat{x}_j^{il}$.

Denote by $z_j^{ih}[k]$ and $z_j^{il}[k]$ the number of product i that user j buys from store in the interval $[k, k+1]$ at the marked price and the discounted price, respectively. Let the total number of goods purchased in the interval $[k, k+1]$ be denoted $z_j^i[k] = z_j^{ih}[k] + z_j^{il}[k]$. Further, denote the difference in purchases made by user j over intervals $[k, k+1]$ and $[k-1, k]$ by $\Delta z_j^i[k] = z_j^i[k] - z_j^i[k-1]$ corresponding to a difference in the coupon delivery rate $\Delta\gamma_j^i[k] = \gamma_j^i[k] - \gamma_j^i[k-1]$. We use the following controller to choose γ_i^i.

$$\begin{aligned} \gamma_j^i[k+1] \quad &= \quad (\gamma_j^i[k] + \delta)\chi_{\{\Delta\gamma_j^i[k]\Delta z_j^i[k]>0\}} \\ &\quad + (\gamma_j^i[k] - \delta)\chi_{\{\Delta\gamma_j^i[k]\Delta z_j^i[k]=0\}}. \end{aligned} \tag{13}$$

Here, $\delta > 0$ is a constant small amount by which we increase or decrease γ_j^i. The controller increases γ_j^i if the customer has a greater potential of buying goods at the discounted price, and decreased if the customer has already reached her maximum potential. We can now easily prove that the controller converges to within $\delta/2$ of the desired value of $\hat{\gamma}_j^i$.

THEOREM 2. *Under the time scale separation assumption, using the controller (13) we have*

$$\lim_{k\to\infty} |\gamma_j^i[k] - \hat{x}_j^{il}| \leq \delta/2 \quad \forall\ i \in \mathcal{S},\ j \in \mathcal{R}.$$

PROOF. We use a Lyapunov argument, with the Lyapunov function

$$J[k] = \left(\gamma_j^i[k] - \hat{\gamma}_j^i\right)^2.$$

Then we have

$$\begin{aligned} J[k+1] \quad - \quad & J[k] \\ = \quad & (\gamma_j^i[k+1])^2 + (\hat{\gamma}_j^i)^2 - 2\hat{\gamma}_j^i\gamma_j^i[k+1] \\ & -(\gamma_j^i[k])^2 - (\hat{\gamma}_j^i)^2 + 2\hat{\gamma}_j^i\gamma_j^i[k] \\ = \quad & \left(\gamma_j^i[k+1] - \gamma_j^i[k]\right)\left(\gamma_j^i[k+1] + \gamma_j^i[k] - 2\hat{\gamma}_j^i\right) \end{aligned}$$

We have two cases.
Case I: If $\Delta\gamma_j^i\Delta z_j^i[k] > 0$, i.e., $\gamma_j^i[k] < \hat{\gamma}_j^i$, we have from (13)

$$J[k+1] - J[k] = \delta(2(\gamma_j^i[k] - \hat{\gamma}) + \delta),$$

which is non-positive except in $\gamma_j^i[k] - \hat{\gamma}_j^i \in [-\delta/2, 0]$.
Case II: $\Delta\gamma_j^i\Delta z_j^i[k] = 0$, i.e., $\gamma_j^i[k] \geq \hat{\gamma}_j^i$, we have from (13)

$$J[k+1] - J[k] = -\delta(2(\gamma_j^i[k] - \hat{\gamma}) + \delta),$$

which is non-positive except in $\gamma_j^i[k] - \hat{\gamma}_j^i \in [0, \delta/2]$.

Thus, the system is globally asymptotically stable and $\gamma_j^i - \hat{\gamma}_j^i$ will converge to the interval $[-\delta/2, +\delta/2]$. □

Note that when δ is smaller enough and the algorithm starts with a small $\gamma_j^i[0]$, we can guarantee that $\gamma_j^i[k] \leq b_j^i$ for all k. Combining Theorem 1 and Theorem 2, we conclude that *the number of coupons consumed under the two time-scale algorithm converges to the optimal solution to OPT 1.*

5. DELAY-BASED COUPON FORWARDING

Suppose that the store rewards relays only after a coupon has been used to make a purchase. The insight that we obtain from the optimality of backpressure is the following:

- If coupons are not used on a particular path, queues build up. This would cause the average delay in being rewarded to all relays on the path to increase.
- If a higher rate of coupons than that set by the store are transferred along a path, the store does not reward the relays for some fraction of coupons and virtual coupons build up. Again, this would mean that the average delay in being rewarded to all relays on the path would increase.

The observation immediately suggests that perhaps a simpler algorithm would be to replace the backpressure based user control of Section 4 A3 with a much simpler scheme. Users need only keep track of the average delay experienced in obtaining rewards when they forward coupons to each of their neighbors. They choose to forward coupons to that neighbor who has the lowest such delay. The scheme is intuitively incentive compatible, since users might want to obtain rewards as soon as possible. Thus, we may replace the reward scheme of Section 4 A2 with an equal reward for all users in the path. Further, we choose a threshold value of delay and users do not forward coupons to any neighbor that yields a delay larger than this threshold.

6. SIMULATION RESULTS

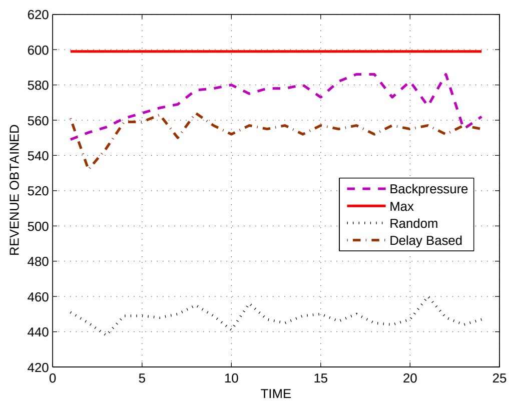

Figure 1: Trajectory of revenue obtained by the store using different schemes.

We first consider a simple tree topology with 9 nodes, where the root is the store, and the leaves are customers. We plot the total revenue obtained by the store for the two different schemes, and compare them to the maximum possible revenue in Figure 1. The upper bound is the value $\sum_j p\hat{x}_j^h + q\hat{x}_j^l$, which is the maximum extractable revenue. We see that the randomized algorithm does significantly worse than backpressure as well as delay-based schemes. Even accounting for the fact that a constant part of the revenue would have to be used to incentivize the scheme, the performance improvement is still significant, although the delay-based scheme performs worse than backpressure.

We then consider a power-law topology that might bear a closer resemblance to real-world social networks. The graph consists of 100 nodes with 200 links. Nodes are connected to the coupon source, are relays, and are customers with probabilities 0.2, 0.7 and 0.1, respectively. Customers have arbitrary spending capacities. We show the upper bound and the performance of the three schemes in Figure 2. Backpressure clearly performs the best, followed by the delay-based and random schemes. The results indicate that using such coupon distribution schemes could significantly increase the revenue obtained.

7. CONCLUSION

In this paper we studied the problem of creating differentiated prices by means of disbursing discounts in a distributed fashion using online social networks. We designed algorithms in which some users act as sources of coupons, which they then forward to other uses based on appropriate incentives. We used the concept of "backpressure routing" borrowed from the control of communication networks, which

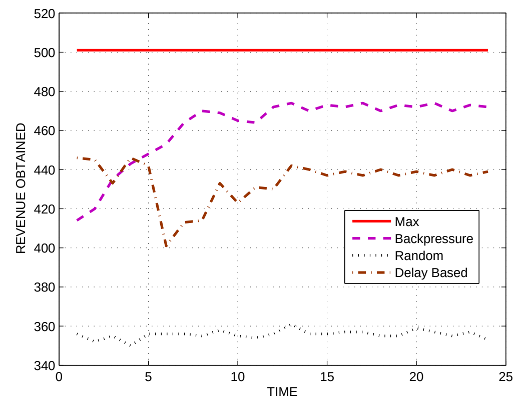

Figure 2: Trajectory of revenue obtained by the store using different schemes for the power-law topology.

we modified appropriately to our scenario. We also designed a simpler heuristic that sacrifices some performance, and showed how both schemes outperform a simple random coupon forwarding scheme.

8. REFERENCES

[1] "mGinger," http://www.mginger.com/, 2009.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, pp. 1936–1948, December 1992.

[3] M. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE Infocom.*, vol. 3, Miami, FL, March 2005, pp. 1723–1734.

[4] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, August 2005.

[5] A. Eryilmaz and R. Srikant, "Joint Congestion Control, Routing and MAC for Stability and Fairness in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514–1524, August 2006.

[6] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *IEEE Infocom*, Barcelona, Spain, April 2006.

[7] S. Shakkottai, L. Ying, and S. Sah, "Implicit Learning for Explicit Discount Targeting in Social Networks," Tech. Report available at http://www.ece.tamu.edu/∼sshakkot/coupon-tech.pdf.