Distributed Algorithms in Service Overlay Networks: A Game Theoretic Perspective

Jiang Guo, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{*jguo,bli*}*@eecg.toronto.edu*

***Abstract*— When designing distributed algorithms for application overlay networks, it is usually assumed that the overlay nodes are *cooperative* to collectively achieve optimal global performance properties. However, this assumption does not hold in reality, as nodes generally tend to be *non-cooperative* and always attempt to maximize their gains by optimizing their strategies. With such an assumption, we present extensive theoretical analysis to gain insights from a game theoretic perspective, with respect to the behavior of nodes and the equilibrium of the system. The main idea in our analysis is to *design* appropriate payoff functions, so that the equilibrium of the system may achieve the optimal properties that we desire. Driven by the per-node goal of maximizing gains, such payoff functions naturally lead to distributed algorithms that lead to the desired favorable properties of overlay networks.**

I. INTRODUCTION

Application-layer overlay networks are constructed in the application layer by peer nodes at the edge of a wide-area network, using the underlying IP network topology to forward application-layer messages. Distributed algorithms may be designed to be executed on these peer nodes, so that resources (such as data, services or computational resources) may be efficiently shared, without any modifications to the network protocols. Provisioning data or services in application-layer overlay networks constitutes one of the most important functions of such networks, which may be simply referred to as *Service Overlay Networks* if service provisioning is mandatory.

One critical design objective for distributed algorithms in service overlay networks is to address the problem of *rendezvous* between the service providers and the users: the *timely* delivery of data and services to those in need, with *minimized* topological and message overhead. For example, towards improving the performance of searching for a particular item in overlay networks, there exist two categories of solutions. One approach, based on distributed hash tables (e.g., [1]), associates designated hosts with each data item or service, thus achieving search performance in the order of $\log n$, n being the size of the overlay network. The other approach, pioneered by the work of Cohen *et al.* [2], requires overlay nodes to collaboratively replicate either the actual services in demand or their shortcuts, both of which improve search performance.

However, both approaches rely on a fundamental assumption that the peer nodes (or *overlay nodes*) in a service overlay network are *cooperative* to engage in activities, such as forwarding queries replicating hotspots or storing pre-assigned data item, all of which are critical to the efficiency of the algorithms. In reality, however, nodes belong to different administrative domains and interests, and are inherently *non-cooperative* when it comes to duties that are not of self interests. This is particularly the case when the duties demand local resources (energy, bandwidth, and storage) that the nodes may otherwise conserve for their own uses. Without proper *incentives* for self organization, these distributed algorithms may not be accepted for widespread use. As an example, recent studies [3] have revealed that there exists the problem of free riders in popular peer-to-peer file sharing systems (*i.e.,* most nodes download, but do not share files), demonstrating the validity of such a concern.

In this paper, we take a game theoretic perspective to analyze the effects of rationality and selfishness on the design of distributed algorithms in application overlay networks. The purpose of deriving such game theoretic models is to study the feasibility of *manipulating* rational and selfish per-node behavior so that favorable (or in some cases, *optimal*) global properties may be collectively achieved. In game theoretic terms, our objective is to design appropriate *payoff functions*, such that the derived Nash equilibrium demonstrates certain desired global properties. For example, by customizing payoff functions, we may encourage replication or shortcuts in an overlay topology. In contrast to alternative approaches, a game theoretic analysis leads to fully distributed and robust algorithms with realistic assumptions of rationality and selfishness.

Particularly, we address one specific open problem: how do we design optimal replication strategies to maximize the average query performance? From a game theoretic perspective, we present a *replication game* to study the replication behavior of rational nodes. Each node needs to find the best strategy to maximize its gain, with the knowledge that other non-cooperative nodes seek the same. We observe that such a replication game can be transformed into a *pricing game*, which leads to the solution we proposes. Such a solution leads to our distributed algorithm, which is shown to perform well using simulations.

The remainder of the paper is organized as follows. We start with a brief introduction to game theory and pricing mechanism in Sec. II. In Sec. III, we present our model and analysis of the replication game. We propose a distributed algorithm and show simulation results in Sec. IV. Sec. V concludes the paper.

II. PRELIMINARIES

A. Game theory

We briefly present elements of game theory [4] that we use in this paper. Game theory is a study of multi-person decision problems where a conflict of interests exists. For example, resource sharing and allocation among multiple parties in the context of computer networks may be formulated as a game theoretic problem. Such a problem can be modeled as a *game* with n *players*. Each player i possesses a set of strategies S_i, from which it may arbitrarily choose one strategy s_i. The *payoff function*, $\pi_i(s_1, \ldots, s_n)$, of i, is the payoff to player i if a combination of strategies $(s_1, \ldots, s_n)$ is chosen by all players. The assumption of *selfishness* of players guarantees that, the choices of optimal strategies on each player lead to the maximization of its payoff π_i. In this paper, we restrict our attention to *static games*, *i.e.,* the players simultaneously choose strategies without a particular sequence of play. We consider static games of *complete information*, where the payoff function of each player is considered as *common knowledge* among all players.

In static games of complete information, a particular combination of strategies $(s_1^*, \ldots, s_n^*)$ is a *Nash equilibrium*, if it satisfies

$$\begin{aligned} &\pi_i(s_i^*, s_{-i}^*) \geq \pi_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i, \\ \text{where} \quad & s_{-i}^* \equiv (s_1^*, \ldots, s_{i-1}^*, s_{i+1}^*, \ldots, s_n^*). \end{aligned} \tag{1}$$

There exists a wealth of research results in applying game theory to problems of congestion control [5] and selfish routing [6]. Some of the recent work has applied game theory to examine the free rider problem in popular peer-to-peer file-sharing systems [7] and encourage packet forwarding in wireless ad hoc networks that consist of selfish nodes [8].

Perhaps the work of *algorithmic mechanism design* [9] and its distributed version [10] are closest to our work. Algorithmic mechanism design encourages selfish nodes to behave in a way that leads to a desirable system-wide outcome, by properly designing associated payoffs and specifications that are computationally tractable. It would be interesting to compare our approach with solutions based on mechanism design, which we leave as future work since it is beyond the scope of this paper.

B. Pricing

Pricing has been proposed to allocate resources in wired networks [11] [12] as well as wireless networks [13]. We briefly review the pricing model from [11] and its results. In Sec. III, we transform the replication problem into a resource allocation problem that is mathematically identical to a special case of such pricing model.

Consider a network consisting of a set of links J, and let C_j be the finite capacity of link j, for $j \in J$. Let R be the set of users accessing the network. Associate each user r with a single route that is a non-empty subset of J. Set $A_{jr} = 1$ if $j \in r$, meaning link j lies on route r, and set $A_{jr} = 0$ otherwise. This defines a 0-1 routing matrix $A = (A_{jr}, j \in J, r \in R)$.

Suppose that when a rate x_r is allocated to user r, the user will receive utility $U_r(x_r)$, where $U_r(\cdot)$ satisfies the following assumption:

Assumption 1: $U_r(x_r)$ is increasing, strictly concave and continuously differentiable over the range $x_r \geq 0$.

Let $U = (U_r(\cdot), r \in R)$ and $C = (C_j, j \in J)$, and suppose that the network seeks a rate allocation $x = (x_r, r \in R)$ that solves the following optimization problem:

SYSTEM(U, A, C)

$$\begin{aligned} \max \quad & \textstyle\sum_{r \in R} U_r(x_r) \\ \text{subject to} \quad & Ax \leq C \\ \text{over} \quad & x \geq 0. \end{aligned} \tag{2}$$

In this paper, we are more interested in a special case, listed below as *SYSTEM'(U,A,C)*, where the network consists of *only one single link* k. In this case, users compete for the link capacity C_k.

SYSTEM'(U, A, C)

$$\begin{aligned} \max \quad & \textstyle\sum_{r \in R} U_r(x_r) \\ \text{subject to} \quad & \textstyle\sum_{r \in R} x_r \leq C_k \\ \text{over} \quad & x \geq 0. \end{aligned} \tag{3}$$

In [11], Kelly *et al.* have proposed a price-based mechanism that can drive such selfish users to achieve optimal resource allocations. In our special case, assume that each user r is charged for the single link at a price u for each received packet, then each user r will attempt to maximize its payoff $U_r(x_r) - ux_r$. In this special case, we rephrase Theorem 1 in [11] as follows.

Theorem 1: There exists a price u^* that leads to a unique allocation vector $(x_r^*, r \in R)$, which is the optimal solution to *SYSTEM'(U, A, C)*.

III. THE REPLICATION GAME

A. Problem formulation: replication game

We consider a service overlay network with n identical nodes and m distinct items (data items or services) to be queried. With respect to these items, the query rate distribution is a vector $(q_1, \ldots, q_m)$ with $\sum_{i=1}^m q_i = 1$, known to all nodes. The query rate q_i is the fraction of all queries that are issued for a specific item i. For a particular item i, each node may choose whether or not to replicate i with a probability x_i, which leads to a vector $(x_1, \ldots, x_m)$. We do not consider details such as multi-hop application-layer routing in unstructured networks, and assume all nodes may reach each other. For simplicity of presentation, we assume that there exists an *omniscient observer* which takes responsibility of *randomly* querying the nodes. In each round of queries, the omniscient observer randomly selects a node, until the requested item is found. The *average query path length* is the number of overlay nodes selected in such a query process. It is slightly more complex to assume that a regular overlay node performs the queries.

We model the replication problem in the overlay network as a static game of complete information with n identical players. Since all nodes are identical, if a Nash equilibrium exists in this game, it must be symmetric. We further assume each item is of unit size and each node has a capacity of ρ units. Therefore, the strategy of each player in such a replication game is vector $(x_1, \cdots, x_m)$, which is subject to $\sum_{i=1}^{m} x_i \leq \rho$.

B. Optimal replication

We proceed to examine the case when the *best possible*, or optimal, average query performance is achieved. From the system's perspective, *i.e.*, assume that the system is a single entity, the system chooses to replicate items with a vector of probabilities $(p_1, \cdots, p_m)$ with $\sum_{i=1}^{m} p_i = 1$. To achieve optimal replication, Cohen *et al.* [2] show that we have to solve the following maximization problem:

$$\min \sum_{i=1}^{m} \frac{q_i}{p_i} \tag{4}$$

$$\text{subject to} \quad \sum_{i=1}^{m} p_i = 1, \tag{5}$$

$$p_i \geq 0,\ i = 1, \ldots, m.$$

The solution to such maximization problem is a square-root replication that is defined and proved in Theorem 2.

Theorem 2: The optimal service replication distribution is the square-root replication that satisfies

$$p_i = \frac{\sqrt{q_i}}{\sum_{i=1}^{m} \sqrt{q_i}} \tag{6}$$

for all $i = 1, \ldots, m$.

Proof: Let

$$h(\bar{p}) = \sum_{i=1}^{m} \frac{q_i}{p_i}, \tag{7}$$

$$g(\bar{p}) = 1 - \sum_{i=1}^{m} p_i \tag{8}$$

Using the method of Lagrange multiplier, we solve the equation as follows:

$$\nabla h(\bar{p}) = \lambda \nabla g(\bar{p}) \tag{9}$$

where λ is a Lagrange multiplier.

From the equation, we have

$$\left[\frac{\partial h(\bar{p})}{\partial p_1}, \frac{\partial h(\bar{p})}{\partial p_2}, \cdots, \frac{\partial h(\bar{p})}{\partial p_m}\right]^T = \lambda \left[\frac{\partial g(\bar{p})}{\partial p_1}, \frac{\partial g(\bar{p})}{\partial p_2}, \cdots, \frac{\partial g(\bar{p})}{\partial p_m}\right]^T$$

Then,

$$\left[-\frac{q_1}{p_1^2}, -\frac{q_2}{p_2^2}, \cdots, -\frac{q_m}{p_m^2}\right]^T = -\lambda \left[1, 1, \cdots, 1\right]^T$$

that leads to

$$\frac{q_1}{p_1^2} = \frac{q_2}{p_2^2} = \cdots = \frac{q_m}{p_m^2} = \lambda$$

We thus have

$$p_i = \frac{\sqrt{q_i}}{\sum_{i=1}^{m} \sqrt{q_i}}.$$

□

C. Problem Transformation: A Pricing Game

In the replication game, if all nodes choose the same strategies $(x_1, \ldots, x_m)$, the expected query performance in terms of the number of queried overlay nodes will be $\rho \sum_{i=1}^{m} q_i/x_i$. To obtain the Nash equilibrium $(x_1^*, \ldots, x_m^*)$, we must solve the following optimization problem:

$$\min \sum_{i=1}^{m} \frac{q_i}{x_i} \tag{10}$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_i \leq \rho, \tag{11}$$

$$x_i \geq 0,\ i = 1, \ldots, m.$$

We observe that the above optimization problem is equivalent to a *pricing game* with m players, which has been reviewed in Sec. II. In such a pricing game, each *player* corresponds to an *item* and the corresponding utility function is

$$U_i(x_i) = -\frac{q_i}{x_i} \tag{12}$$

It is clear that $U_i(\cdot)$ satisfies Assumption 1.

The pricing game is mathematically identical to the special case of the pricing model in Sec. II. Furthermore, we observe that such a specific pricing game leads to a service replication distribution, which is both optimal and unique with respect to the resource allocation in each node's capacity. We validate such an observation by formally proving it in Theorem 3.

Theorem 3: If each node is running a pricing game given above, the resulting allocation vector $(x_1^*, \cdots, x_m^*)$ is the best strategy for each node to replicate items, which leads to the square-root replication strategy globally.

Proof: According to Theorem 1, there exists a pricing mechanism that leads to the unique optimal allocation vector $(x_1^*, \cdots, x_m^*)$ for such a pricing game. Furthermore, if all nodes choose such an allocation vector as their replication strategy, it is a Nash equilibrium, since at such point the system achieves the global optimal replication distribution, which is the square-root replication distribution. □

To achieve such a pricing strategy, we need to design specific payoff functions for the players in the pricing game.

D. Pricing game: designing payoff functions

In the pricing game, we propose to design the payoff function of each player i as

$$\pi_i(x_i) = U_i(x_i) - u x_i,\ u > 0 \tag{13}$$

where $U_i(x)$ is defined in Eq. (12), and u is a cost coefficient.

Our goal is to appropriately design the payoff functions — or more precisely, u — such that the resulting equilibrium converges to the optimal allocation for the pricing game.

In addition, for $i = 1, \ldots, m$, we define the function D_i such that $D_i(u)$ is the optimal solution, x^*, to the maximization problem

$$\max_{x \geq 0} \{U_i(x) - ux\} \quad (14)$$

Or, equivalently,

$$D_i(u) = \arg\max_{x \geq 0} \{U_i(x) - ux\}, u \geq 0 \quad (15)$$

An example of the function $D_i(u)$ is illustrated in Fig. 1(a).

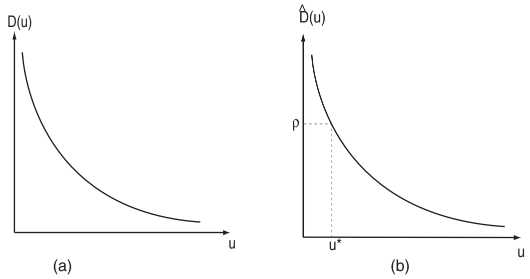

Fig. 1. (a) The function $D(u)$; (b) the function $\hat{D}(u)$.

According to Theorem 1, there exists a unique solution $(x_1^*, \ldots, x_m^*)$. In addition, there exists a parameter u^* such that

$$x_i^* = D_i(u^*) \quad (16)$$

Further, we define $\hat{D}(u)$ as the function

$$\hat{D}(u) = \sum_{i=1}^{m} D_i(u) \quad (17)$$

As shown in Fig. 1 (b), we have

$$\hat{D}(u^*) = \rho \quad (18)$$

The payoff function can be designed as follows:

1) To define the payoff function, we first define a utility function $U_i(x)$ which satisfies Assumption 1. In our case, this step has been done in Eq. (12).
2) Since $U_i(x)$ is increasing, strictly concave and twice differentiable, if we combine Eq. (15) and $x_i^* = D_i(u)$, where x_i^* is the optimal allocation vector, we can solve the function $D_i(u)$.
3) $\hat{D}(u)$ can be solved by its definition in Eq. (17).
4) u^* can be obtained by solving $u^* = \hat{D}^{-1}(\rho)$, where $\hat{D}^{-1}(\rho)$ is the inverse of $\hat{D}(\cdot)$.

With this procedure, the payoff function of each player i may then be defined.

We illustrate the above results using an example. For simplicity of presentation, we assume that

$$D_i(u) = c_i - a_i u, u \geq 0$$

By the definition in Eq. (17), we have

$$\hat{D}(u) = \sum_{i=1}^{m} c_i - u \sum_{i=1}^{m} a_i, u \geq 0$$

Combining the above equation with Eq. (18), we can obtain u_i^* as

$$u^* = \frac{\sum_{i=1}^{m} c_i - \rho}{\sum_{i=1}^{m} a_i}$$

and therefore determine payoff function π_i.

According to our analysis, given the payoff function we have designed for the pricing game, a selfish and rational player will adopt the optimal allocation to maximize its payoff. Furthermore, Theorem 3 shows that such optimal allocation in each node leads to the square-root replication globally. We conclude that, our game theoretic approach may be used to design distributed algorithms and achieve the global property of optimal query performance.

IV. Distributed Algorithm and Performance Evaluation

A. Algorithm design

In Sec. III, we assume that each node can be fully manipulated and possess the knowledge of utility function $U_i(\cdot)$; therefore, according to our analysis, the system will achieve the desired Nash equilibrium that conforms to a square-root replication distribution. It is of interests for us to extend it into a setting that may be more realistic and more flexible, so that we can apply it to other overlay applications.

Assume that we have an overlay application that runs on each node. This application consists of m threads, with each thread in charge of an item. Each thread can be programmed to have a specific payoff function and to behave selfishly to attempt to maximize its own payoff. On the other hand, we expect that each node has no need to know payoff functions of those threads. Therefore, if we change payoff functions of those threads, we can address various global optimization problems in overlay networks.

A mechanism has been proposed by Kelly *et al.* in [11] to address such a problem. We adapt such mechanism to the special case in Sec. II. The *SYSTEM'(U, A, C)* problem can be decomposed into two simpler problems: one for users and one for the network.

Assume that each user r is given the price per packet u. Then r chooses to pay w_r per unit time and receives x_r given by $x_r = \frac{w_r}{u}$. The utility maximization problem for each user r is as follows.

USER(U_r; u)

$$\begin{array}{ll} \max & U_r(\frac{w_r}{u}) - w_r \\ \text{over} & w_r \geq 0. \end{array} \quad (19)$$

The network tries to maximize the function $\sum_{r \in R} w_r \log x_r$ as follows.

NETWORK(A, C; w)

$$\begin{array}{cc} \max & \sum_{r \in R} w_r \log x_r \\ \text{subject to} & \sum_{r \in R} x_r \leq C_k \\ \text{over} & x \geq 0. \end{array} \quad (20)$$

Note that, solving *NETWORK(A, C; w)* does not require the network to know the utilities U.

Kelly *et al.* [11] show that there always exist vectors $(w_r^*, r \in R)$, $(x_r^*, r \in R)$, and u^* that solve *USER(U_r; u)* and *NETWORK(A, C; w)*; furthermore, the vector $(x_r^*, r \in R)$ is the unique solution to *SYSTEM'(U, A, C)*.

Based on the aforementioned design guidelines, we extend the iterative algorithm proposed by Kelly *et al.* in [11]. In this proposed distributed algorithm, listed below, time is discretized into consecutive stages.

Algorithm running in each thread

1. At each stage k, each thread i receives the price $u^{(k-1)}$ that thread i was charged in the previous step $k-1$.
2. Chooses a new $x_i^{(k)}$:

$$x_i^{(k)} = D_i(u^{(k-1)})$$

3. Reports $x_i^{(k)}$ to the Node.

Algorithm running on each node

1. Receives x_i from all threads
2. Updates the price u by setting

$$u^{(k)} = \Big[u^{(k-1)} + \alpha\big(\sum_{i=1}^{m} x_i^{(k)} - \rho\big)\Big]^+$$

where α is a small step size parameter and $\big[z\big]^+ = \max\{z, 0\}$.

3. Reports $u^{(k)}$ to all threads.

B. Simulation results

We have conducted simulation experiments to evaluate our proposed distributed algorithm, using a packet-level event-based C++ simulator. We verify that the algorithm, locally executed on each selfish node, may indeed achieve the desired global replication distribution.

In this set of simulations, we have 1000 nodes in the overlay network, and 10 distinct items. All nodes have the same capacity of 5, and all items are of same size of 1. The step size parameter α is set to 0.025. The four columns in Table I show the item number, query distribution, the desired optimal replication, and the achieved allocation vector at equilibrium.

TABLE I
REPLICATION DISTRIBUTION IN A HOMOGENEOUS NETWORK

i	q_i	p_i	x_i	i	q_i	p_i	x_i
1	0.01	0.035	0.175	6	0.1	0.111	0.555
2	0.02	0.049	0.245	7	0.1	0.111	0.555
3	0.02	0.049	0.245	8	0.2	0.157	0.785
4	0.05	0.078	0.390	9	0.2	0.157	0.785
5	0.05	0.078	0.390	10	0.25	0.175	0.875

Since all nodes are identical, we simply choose one node to plot. We plot the allocation vector $(x_1, \cdots, x_m)$ as well as the price u for the first 50 time units, as it rapidly converges to the stable equilibrium, in most cases within the first 30 time units. We observe that x_i, listed in Table I, is proportional to the desired p_i, validating the conclusion given by Theorem 3. According to Fig. 2, our algorithm[1] is effective and efficient to achieve optimal replication strategy even when all nodes are assumed selfish.

[1]Note that in our simulation, for the sake of simplicity, we do not enforce the constraint that x_i should be less than 1, which does not influence the equilibrium these trajectories converge at.

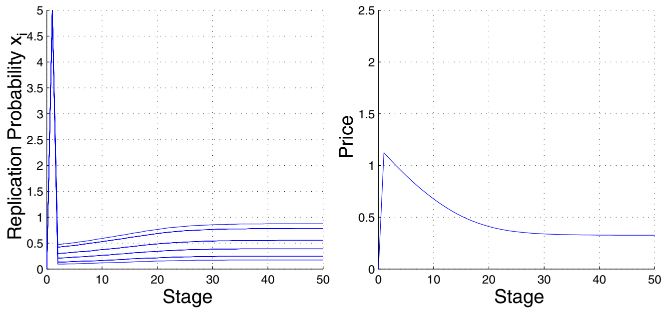

Fig. 2. Trajectories of replication probabilities (left) and price (right).

V. CONCLUDING REMARKS

In this paper, we have presented a game theoretic view of the open and fundamental problems in service overlay networks, where cooperative overlay nodes to implement distributed algorithms are no longer assumed. Rather, we assume that the nodes are inherently non-cooperative, and attempt to maximize gains based on their self interests. In this context, we have extensively analyzed the *replication problem*. We show that, even with the relaxed (and more realistic) assumptions, if appropriate payoff functions are designed and given, it is still possible for overlay nodes to reach an equilibrium that may collectively achieve favorable, or in some cases, optimal global properties that we desire.

REFERENCES

[1] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proc. of ACM SIGCOMM*, 2001.

[2] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," in *Proc. of ACM SIGCOMM*, 2002.

[3] S. Saroiu, P. Gnummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proc. of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN)*, 2002.

[4] R. Gibbons, *Game Theory for Applied Economics*, Princeton Univeristy Press, 1992.

[5] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker, "Selfish behavior and stability of the Internet: A game-theoretic analysis of TCP," in *ACM SIGCOMM*, 2002.

[6] T. Roughgarden and E. Tardos, "How bad is selfish routing?," in *The 41st Annual Symposium on Foundations of Computer Science*, 2000.

[7] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Proc. of the ACM Conference on Electronic Commerce (WELCOM)*, 2001.

[8] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in Wireless Ad Hoc Networks," in *Proc. of IEEE INFOCOM*, 2003.

[9] N. Nisan and A. Ronen, " Algorithmic mechanism design," in *Games and Economic Behavior*, 2001, vol. 35, pp. 166–196.

[10] J. Feigenbaum and S. Shenker, "Distributed Algorithmic Mechanism Design: Recent Results and Future Directions," in *Sixth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial'M)*, 2002.

[11] F. P. Kelly, "Charging and Rate Control for Elastic Traffic," in *European Trans on Telecommun 8*, 1997, pp. 33–37.

[12] S. H. Low and D. E. Lapsley, "Optimization Flow Control: Basic Algorithm and Convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[13] P. Marbach and R. Berry, "Downlink Resource Allocation and Pricing for Wireless Networks," in *Proc. of IEEE INFOCOM*, 2002.