Market Sharing Games Applied to Content Distribution in Ad-Hoc Networks

Michel X. Goemans*, Li (Erran) Li†, Vahab S. Mirrokni*, and Marina Thottan†

ABSTRACT

In third generation (3G) wireless data networks, repeated requests for popular data items can exacerbate the already scarce wireless spectrum. In this paper we propose an architectural and protocol framework that allows 3G service providers to host efficient content distribution services. We offload the spectrum intensive task of content distribution to an ad-hoc network. Less mobile users (resident subscribers) are provided incentives to cache popular data items while mobile users (transit subscribers) access this data from resident subscribers through the ad-hoc network. Since the participants of this data distribution network act as selfish agents, they may collude to maximize their individual payoff. Our proposed protocol discourages potential collusion scenarios. In this architecture the goal (social function) of the 3G service provider is to have the selfishly motivated resident subscribers service as many data requests as possible. However, the choice of which set of items to cache is left to the individual user. The caching activity among the different users can be modeled as a market sharing game.

In this work, we study the Nash equilibria of market sharing games and the performance of such equilibria in terms of a social function. These games are a special case of congestion games that have been studied in the economics literature. In particular, pure strategy Nash equilibria for this set of games exist. We give a polynomial-time algorithm to find a pure strategy Nash equilibrium for a special case while it is is NP-Hard to do so in the general case. As for the performance of Nash equilibria, we show that the *price of anarchy* — the worst-case ratio between the social function at any Nash equilibrium and at the social optimum — can be upper bounded by a factor of 2. When the popularity follows a Zipf distribution, the price of anarchy is bounded by 1.45 in the special case where caching any item has a positive reward for all players. We prove that the selfish behavior of computationally bounded agents converges to an approximate Nash equilibrium in a finite number of improvements. Furthermore, we show that even with one improvement by each player, an $O(\log n)$ approximate solution can be obtained. Our simulation scenarios show that the price of anarchy is 30% better than that of the worst-case analysis and that the system quickly (1 or 2 steps) converges to a Nash equilibrium.

*Michel X. Goemans and Vahab S. Mirrokni are with the Department of Mathematics and the Computer Science and Artificial Intelligence Laboratories (CSAIL) at MIT. Email: {goemans,mirrokni}@csail.mit.edu. Research partially supported by NSF contract CCR-0098018.

†Li (Erran) Li, Marina Thottan are with the Center for Networking Research, Bell Labs, Holmdel NJ 07733. Email:{erranlli,marinat}@dnrc.bell-labs.com. Research partially supported by NSF ANI-0335244.

MobiHoc'04, May 24–26, 2004, Roppongi, Japan.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Economics, Theory, Design, Performance

Keywords

3G Wireless Networks, Mobile Ad-Hoc Networks, Unified Architecture, Price of Anarchy, Nash Equilibrium

1. INTRODUCTION

Third-generation (3G) wide-area wireless networks have recently experienced tremendous growth, with the number of subscribers reaching more than 70 million worldwide [1]. Many 3G service providers have started offering content-rich services such as sports replays, news headlines, music videos, and movie trailers [1].

The 3G subscriber market can be categorized into groups with shared interest in location-based services, e.g. the preview of movies in a theater or the scene of the beach nearby. Since the 3G radio resources are limited, it is expensive to repeatedly transmit large quantities of data over the air interface from the base station (BS). It is more economical for the service provider to offload such repeated requests on to the ad-hoc network comprised of its subscribers where some of them recently acquired a copy of the data. In this scenario the goal for the service provider is to give incentives for peer subscribers in the system to cache and forward the data to the requesting subscribers. Since each data item is large in size and transit subscribers are mobile, we assume that the data transfer occurs in a close range of a few hops.

We envision a system consisting of two groups of subscribers: resident and transit subscribers. Resident subscribers are less mobile and mostly confined to a certain geographical area. Resident subscribers have incentives to cache data items that are specific to this geographical region since the service provider gives monetary rewards for satisfying the queries of transit subscribers. Transit subscribers request their favorite data items when they visit a particular region. Since the service provider does not have knowledge of the spatial and temporal distribution of requests, it is difficult if not impossible for the provider to stipulate which subscriber should cache which set of data items. Therefore, the decision of what to cache is left to each individual subscriber. The realization of this content distribution system depends on two main issues. First, since subscribers are selfish agents, they may collude to increase their individual payoffs. Collusion can result in other subscribers being cheated of their rewards. In this work we address the problem of colluding subscribers by providing a protocol framework that discourages or prevents collusions. The second issue is that the payoff of each item for each agent must be a function of the number of cache requests it services. This in turn depends on the number of agents who cache a given item since each agent has limited storage space (due to form factor, current 3G devices only have a couple of MB flash memory [1]). Therefore, each selfish agent may change the set of items it cached in response to the set of items cached by others. This leads to a non-cooperative caching scenario which we model as a market sharing game.

In the market sharing game, the primary questions are whether the system converges, i.e., results in a pure strategy Nash equilibrium and how long it will take to converge. The goal of the service provider is to offload as many cache requests as possible to the ad-hoc network. We refer to this goal as the social optimum. However, given the selfish behavior of the agents, it is unlikely to result in a social optimum. Therefore, we would like to bound the ratio between the optimum solution and the outcome of the selfish behavior of players. We refer to this ratio as the *price of anarchy*. Furthermore, when computing the selfish behavior of individual players it is essential to consider the computational constraint on the individual subscribers. We model computationally bounded agents using approximate algorithms and evaluate how fast the selfish behavior of such agents will converge to an approximate Nash equilibrium or arrive at an approximate solution to the social function.

The main contributions of this paper are as follows: First, we study the applicability of non-cooperative caching in wireless networks and propose a detailed protocol that provides incentives for selfish agents to service other agents while discouraging collusion among participating agents. We model the caching game among the different mobile users as a market sharing game. We show that this is a special case of congestion games. It is known that pure strategy Nash equilibria exist for these games. We give a polynomial-time algorithm to find a pure strategy Nash equilibrium for the special case of uniform market sharing games where all items are of the same size. We know that it is NP-Hard to find such an equilibrium in the general market sharing game. As for the performance of Nash equilibria, we obtain the upper bound on the price of anarchy for two different cases. In the general case where the popularity of the items is arbitrary, the price of anarchy is upper bounded by 2. When the popularity follows a Zipf distribution we have an upper bound of 1.45 in the special case where caching any item has a positive reward for all players. We give an example that shows that the factor 2 is tight in the general case even with Zipf distribution. We also consider the case of computationally bounded agents by modeling them using approximation algorithms. We prove that, after each agent computes its response function once using a constant factor approximation algorithm, the outcome of the game is within a factor of $O(\log n)$ of the social optimum where n is the number of selfish agents. We also show that this bound is tight up to a constant multiplicative factor. We also evaluate our algorithms using simulation scenarios and show that the price of anarchy in our network setting is less than 70% of the worst-case analysis. We also demonstrate that in most cases the system converges to a Nash equilibrium in one or two steps of each player. Step is defined as the greedy action of a player to change to a different set of items given the actions of other players.

The rest of the paper is organized as follows. In section 2, we propose a detailed protocol for offloading content distribution from 3G wireless networks to ad-hoc networks. In section 3, we discuss our incentive mechanism and show that it prevents or discourages collusion. In section 4, we formulate the non-cooperative caching problem as a general market sharing game. In section 5, we analyze the price of anarchy of our market sharing game and in section 6, we show that pure strategy Nash equilibrium exists, but it is NP-complete to find such an equilibrium. We also provide a polynomial time algorithm to find a pure strategy Nash equilibrium for the special case when all the items have the same size. In section 7, we model computationally bounded agents by approximation algorithms and analyze the outcome of their selfish behavior. We observe that such agents converge to an approximate Nash equilibrium. We also show that, after one step of improvements the outcome of the game is within an $O(\log n)$ factor of the optimum. In section 8, we investigate the price of anarchy for a set of sample instances and the convergence rate to exact and approximate Nash equilibria. Related work is provided in section 9. A discussion on our results and other relevant issues are described in section 10. We conclude in section 11.

2. SYSTEM ARCHITECTURE, TRUST MODEL AND PROTOCOLS

In this section, we briefly discuss the system architecture and the trust model for distributed non-cooperative caching. We then outline the protocols required for offloading popular data items from 3G networks to the multi-hop ad-hoc network. Our architecture is largely the same as UCAN [9]. In our system architecture we assume the following:

- A service provider operates a 3G data network, e.g. 1xEV-DO, CDMA2000-1X, etc. The cost of retrieving an item from the service provider is assumed to be significantly higher than obtaining the same item from the ad-hoc network.
- The mobile device of each subscriber has a 3G interface as well as an IEEE 802.11 radio interface so that it can receive data from the 3G data network and can participate in ad-hoc forwarding.
- Users are selfish and act rationally. A user's primary objective is to maximize its own utility. The utility is in terms of obtaining the item of interest or monetary gains for forwarding or serving the requested item to another subscriber in the network.
- Users trust the service provider network for correct accounting, authentication and packet transmission. We do not deal with malicious users in this paper. However, users could collude arbitrarily if they can benefit by doing so.

Each node u has a shared key K_u with the service provider. When node u requests an item from the service provider, the provider first checks whether the item has been cached by other local resident subscribers (subscribers keep the providers updated about the items they cache). If there are no cached copies for the item, the provider will service u directly. However, if the item is available through a subscriber cache, the provider will instruct u to look for the item in the ad-hoc network. Along with this response the

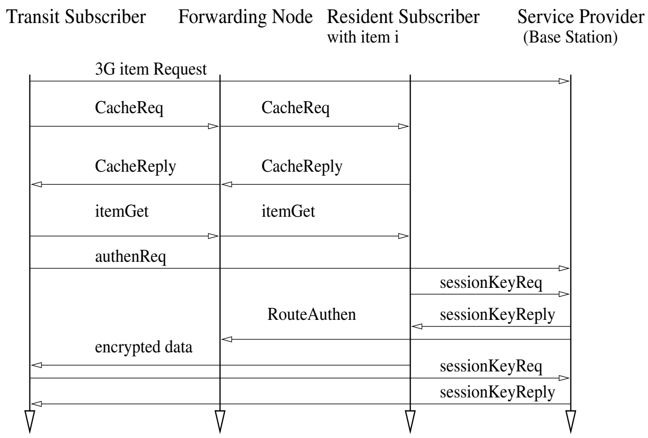


Figure 1: Protocol Messages

provider also sends descriptive information regarding the item requested, for example the item ID and the size. Alternatively, if a node knows the item ID, it can request the item from the ad-hoc network directly. Our protocol messages are illustrated in Figure 1. The first message is optional if the node knows the item ID.

Node u then broadcasts a CacheReq message for the specific item to all k neighbors in the ad-hoc network. Node u includes a Message Authentication Code (MAC) in the CacheReq message. See [9] for details on how to construct MAC codes. The CacheReq message includes the address of u and the requested item ID. Upon receiving the message any resident node that has the requested item will send a CacheReply message to u and attach its MAC computed using its shared key with the provider. The ad-hoc network performs source routing and therefore the final message received by the node u will contain a string of MACs or a layered MAC [19] for all the intermediate hops.

Node u may receive multiple CacheReply messages. It may choose to be serviced by the resident subscriber that has the minimum number of hops. Node u asks the service provider to authenticate the chosen route by sending an authenReq message. The service provider in turn registers the transaction along with the route used. When the BS receives the authenReq and sessionKeyRq message, the BS constructs a RouteAuthen message. The message contains a list of (Session ID (SID), source u, destination v, previous hop, next hop, item size) encrypted using the shared key of each node along the path. An intermediate node w forwards traffic for session SID only if it can decode one (SID, u, v, previous hop, next hop, item size) using K_w shared with the service provider. Once the route authentication has been completed, node u sends an itemGet message to the source node v. When node v receives the itemGet message, it requests a session key K_s from the BS by sending a sessionKeyReq message. This message contains the session ID and the source ID v and destination ID u. The BS responds with a session key encrypted using K_v.

Upon receiving the session key the data item is encrypted using this key. Node v appends a MAC to each packet of the data item using key K_v and it then sends the packet on the ad-hoc network. Each intermediate node w computes a MAC over the received packet using key K_w it shares with the service provider. It replaces the MAC in the received packet with this new MAC (for details of MAC layering, see [19]) and sends the packet to the next hop. During data transfer in the ad-hoc network each intermediate node monitors the packets that are being forwarded by the next node in order to prevent misuse of the authenticated ad-hoc path. When the entire file transfer has been completed node u requests the session key from the BS to decode the data received. When an intermediate relay node forwards data by more than a threshold of the itemSize, it will inform the BS. Forwarding nodes also randomly sample packets with a low probability and report the packet along with its contents to the BS.

Our protocol is lightweight. For one item transfer we have at most 6 short messages between the service provider and the nodes involved plus a small percentage of random sampled packets reported to the BS.

3. CREDIT ACCOUNTING AND AUDITING

In this section, we discuss mechanisms that aid in tracking the credits for resident subscribers and mechanisms that prevent or discourage cheating and collusion among the participants to gain false credits. We also present schemes for charging and rewarding.

There are two types of monetary rewards (1) R_i for successfully servicing the query for item i and (2) f_i for forwarding the query. $R_i > f_i$ and for a given item the total forwarding reward is fixed and shared equally among the forwarding nodes. Both types of rewards are a function of the file size of the item. The receiving node is charged $C_0(i)$ if it obtains the item from subscribers in the ad-hoc network. While if the data item is obtained from the service provider the receiving node is charged $C_S(i)$. $C_S(i)$ is much larger than $C_0(i)$. Participating nodes are charged or rewarded only if BS determines that the item transfer is successfully completed. The service provider can tune R_i and f_i so that a critical number of resident subscribers are willing to participate in caching the popular items.

Our data transfer protocol proposed in the previous section provides no incentive for the following types of behavior.

Stealing reward from forwarding nodes: The repeated collusion of the sender and receiver to leak the session key over a covert channel thus cheating the forwarding nodes of their rewards is prevented by the record keeping done by the base station. This record keeping is accomplished by the random sampling of packets in the intermediate nodes and the information obtained is used to black mark source-destination pairs for attempting to cheat the forwarding nodes.

Refusal to pay by the receiver: The receiver u cannot refuse to pay if it requires the session key from the base station in order to decrypt the content. If it resorts to a covert channel between the sender and receiver to obtain the session key, then BS can determine with high probability that the session is complete through the sampled packets from forwarding nodes.

Impersonate the sender: There is no incentive to impersonate the sender since a sender only gets paid if the receiver acknowledges the service provider about a successful completion of the data transfer. The provider will only pay the original sender that was registered in its database.

Packet dropping: There is no incentive for intermediate nodes to drop packets since they are credited only if the item is transfered completely and the credit is based on the size of the item, not the amount of data relayed. If they try to implicate the sender and receiver by selectively dropping packets, based on session failure statistics from other sessions which involves the same sender-receiver pair, but not these intermediate nodes, the BS can determine with high probability that these intermediate nodes cheated.

Free riding: (1) Since intermediate nodes only forward traffic for a session if it receives authentication from the BS, the source cannot collude with the receiver such that the intermediate nodes do not get paid. (2) The source node or any intermediate node cannot collude with another intermediate node to piggyback data. The "man-in-the-middle" victim knows the size of the item, and it can report this to the provider. Although an individual node is not to be trusted, if many nodes report the same problem with the source node, the provider can punish the source. An intermediate node has no incentive to implicate the source falsely since reporting to the provider incurs a cost. However if all the intermediate relay nodes are colluding nodes, then they merely form a cooperative ad-hoc group. In our protocol we require that intermediate nodes sample packets and report them to the BS. If the BS cannot decrypt the packet correctly due to piggybacked extra data, the BS knows that someone has changed the packet. (3) If a destination node colludes with an intermediate node to get a free ride for their own data and drops legitimate traffic, the destination will have a high session failure rate. The BS can blacklist all such nodes with a high session failure rate. In addition, due to the fact that packets are sampled to the BS, the BS knows that the packet has been changed. (4) If two relay nodes collude and get a free ride on their data and drop legitimate traffic, monitoring by the previous hop node who sent legitimate traffic will reveal that packets other than the one it sent got transmitted. The monitoring node can then report this to the provider. Based on such reports from many sessions, the provider can determine a cheating node with high probability. In all the above cases, if the BS knows that data has been tampered by intermediate nodes from MAC decoding failure, the BS can take immediate actions, e.g. asking the destination to locate another node with the item or find an alternative route.

Suboptimal routes: There is less incentive for a node to add other nodes in the route since a fixed reward is divided among all forwarding nodes. In addition, since the receiver is likely to select the shortest route there is no advantage to announcing a longer route.

4. PROBLEM FORMULATION

In this section, we formalize our caching problem abstractly in terms of a competitive market sharing game.

We are given a bipartite graph $G = (H \cup U, E)$ where U is the set of agents or players and H is the set of markets. There is an edge between agent j and market i if market i is of interest to agent j (we write j is interested in market i). Each market $i \in H$ has a query rate q_i, i.e. the rate at which market i is requested per unit time. Each market i also has a cost C_i corresponding to the cost for servicing this market. Each agent j has a total budget B_j. Each agent should decide which subset of markets to service. Agent j can service a subset S_j of markets, if the sum of the costs C_i of the markets in S_j is less than or equal to B_j. Agent j gets a reward R_i for providing service to market i. This reward for market i depends on the number of agents that serve this market. More precisely, if the number of agents that serve market i is n_i then the reward R_i is equal to $\frac{q_i}{n_i}$. Observe that the total reward received by all players equal the total query rate of the markets being serviced (by at least one player). Let the number of players be n and the number of markets be m, i.e, $|U| = n$ and $|H| = m$.

We now define the necessary game theoretic notations to formally describe the problem. The game is defined as the tuple (U, $\{A_j\}$, $\{P_j()\}$) where U is the set of players or agents. A_j is the set of actions or strategies for player j and $P_j : \Pi_j A_j \to R$ is the payoff or utility function for agent j given the set of actions of all the players. In our model, feasible actions are the set of markets that can be serviced under the given budget constraint. We denote player j's action by S_j. Thus, S_j is a feasible action if $\sum_{i \in S_j} C_i \le B_j$. Given the set, $\mathcal{S}$, of actions for all agents, we can find the number of agents n_i that serve market i, and hence find the reward of each market. The payoff or utility function of player j is the sum of rewards it gets from the markets it serves, i.e., $P_j(\mathcal{S}) = \sum_{i \in S_j} \frac{q_i}{n_i}$ where S_j is the set of markets j serves. In this game, each agent wants to maximize its own payoff. The social function is the total amount of queries satisfied in the market, i.e., $\gamma(\mathcal{S}) = \sum_{i \in \cup_{j \in U} S_j} q_i$ for $\mathcal{S} = (S_1, S_2, \ldots, S_n)$. Notice that this is also the sum of the utility functions of all players. We call the above game a *market sharing game* and the special case of uniform cost for all markets a *uniform market sharing game*.

It is obvious that, in a market sharing game, given the set of actions of other players, the best action of an agent can be obtained by solving a knapsack problem where the value of market (item) i is equal to $\frac{q_i}{n_i}$ or $\frac{q_i}{n_i+1}$ depending on whether market i is currently serviced by this player or not. The size of i in the knapsack instance is C_i, and the knapsack capacity is equal to B_j.

Example: The subscriber caching game described in the introduction is one example of a market sharing game. Agents or players correspond to subscribers and markets correspond to data items. Query rate of market i is the query rate of item i. The cost C_i corresponds to the size of item i and the total budget B_j of agent j corresponds to the available storage space on the subscriber's device.

As mentioned above, in this market sharing game, a feasible action S_j is a feasible solution to the knapsack problem of player j. An action profile is a vector of actions of all players: $\mathcal{S} = (S_1, S_2, \ldots, S_n)$. An action profile $\mathcal{S}$ is a pure strategy Nash equilibrium, if for any player j, given the strategies of all other players, j has no incentive to change its strategy S_j to any other subset to improve its payoff. A mixed strategy Nash equilibrium is a randomization over different pure strategies such that fixing other players' strategy, each player maximizes its expected payoff using the current strategy. Nash equilibria characterize the set of candidate action profiles that selfish players will eventually converge upon if at all. The general market sharing game described here is a special case of congestion games [16]. These games have been studied in the economics literature and the existence of a pure Nash equilibrium has been established.

As mentioned earlier, we focus on market sharing games that have a central authority with a social objective function to be accomplished by means of competitive market sharing. Therefore given a social function, we want to bound the *price of anarchy*, i.e., the worst-case ratio between the social function value at the optimum and at any Nash equilibrium.

5. PRICE OF ANARCHY

5.1 General Distribution

Studying the outcome of games in terms of their price of anarchy is a natural way to evaluate the outcome of a game. Valid games, introduced by Vetta [21], is a set of non-cooperative games for which Vetta proved upper bounds for their price of anarchy. We show that the market sharing game is in this class of games which implies that its price of anarchy is at most 2. In fact, this bound for the price of anarchy can be obtained without referring to valid games and this is done in the appendix as the proof of Corollary 5.5. The main advantage of proving this bound using valid games is that it can be generalized to more general variants of market sharing games. In order to state this result, we need the following definitions.

DEFINITION 5.1. *A function of the form $f : 2^{\mathcal{H}} \to \mathbb{R}$ is called a set function. A set function $f : 2^{\mathcal{H}} \to \mathbb{R}$ is submodular if $f(\emptyset) = 0$ and for any two sets $A, B \subseteq \mathcal{H}$, $f(A) + f(B) \ge f(A \cap B) + f(A \cup B)$. This function is a non-decreasing function, if for any $X \subseteq Y \subseteq \mathcal{H}$, $f(X) \le f(Y)$.*

Given an action profile $\mathcal{S} = (S_1, \ldots, S_n)$, the set $\mathcal{H}_\mathcal{S} = \{(k, i) : 1 \le k \le n, i \in S_k\}$ is called *the pair set for* $\mathcal{S}$. Given a function $f : \Pi_k S_k \to \mathbb{R}$, the corresponding set function, f^s, of f is a set function of the form $2^\mathcal{H} \to \mathbb{R}$ where $\mathcal{H} = \{(k, i) : 1 \le k \le n, 1 \le i \le m\}$ and $f^s(\mathcal{H}_\mathcal{S}) = f(\mathcal{S})$. In other words, $f^s(K) = f((A_1, A_2, \ldots, A_n))$ if $A_j = \{i : (j, i) \in K\}$.

DEFINITION 5.2. *Let $\mathcal{G}(U, \{S_j\}, \{P_j(\Pi_k S_k)\}$ be a noncooperative game with social function $\gamma : \Pi_k S_k \to \mathbb{R}$. $\mathcal{G}$ is called a* valid game *if it satisfies these properties:*

- *γ^s, the corresponding set function of γ, is submodular and non-decreasing.*
- *The payoff of a player is at least equal to the difference in the social function when the player participates versus when it does not participate.*
- *The sum of the utility or payoff functions for any set of strategies should be less than or equal to the social function.*

THEOREM 5.3. *[21] Let $\mathcal{G}$ be a valid game, then for any mixed strategy Nash equilibrium, the social function at this equilibrium is at least half the optimum social function, i.e., in a Nash equilibrium A, $\gamma(OPT) \le 2E[\gamma(A)]$.*

THEOREM 5.4. *A market sharing game is a valid game.*

Proof: We need to show that our social function given by $\gamma(\mathcal{S}) = \sum_{i \in \cup_{j \in U} S_j} q_i$ satisfies the three properties itemized above:

- First, it is clear that γ^s is non-decreasing. To show its submodularity, we use an equivalent definition of submodular functions: A set function f is submodular if for any two subsets A and B such that $A \subset B$ and for any element $i \notin B$, $f(A \cup \{i\}) - f(A) \ge f(B \cup \{i\}) - f(B)$ (see for example Lemma 2.2 of [15]). Thus, in order to prove that γ^s is submodular, it is enough to prove that for two (possibly infeasible) action profiles $\mathcal{S} = (S_1, \ldots, S_n)$ and $\mathcal{S}' = (S'_1, \ldots, S'_n)$ such that $S_j \subseteq S'_j$ for all $j \in U$, by adding a new market i to the action set of any player, j, the increase in γ^s for $\mathcal{S}$ is no less than the increase for $\mathcal{S}'$. If market i is not in $\cup_{j \in U} S'_j$, its addition to one of the S'_j increases γ^s by q_i. In this case, market $i \notin \cup_{j \in U} S_j$, because $\cup_{j \in U} S_j \subseteq \cup_{j \in U} S'_j$. Thus, adding market i to $\mathcal{S}$ also increases γ^s by at least q_i. If i is in $\cup_{j \in U} S'_j$, adding i will not increase γ^s. Thus in any case, the increase for $\mathcal{S}$ is no less than the increase for $\mathcal{S}'$.
- The difference in the social function when j plays S_j or empty (does not play at all) is equal to $\sum_{i \in S_j : n_i = 1} q_i$ and this is indeed less than or equal to $\sum_{i \in S_j} \frac{q_i}{n_i}$.
- By the definition of our social function, we have $\sum_{j \in U} P_j(\mathcal{S}) = \gamma(\mathcal{S})$ and therefore the third property is satisfied as well.

■

To deal with the situation in which agents only compute an approximate solution to the knapsack problem, we consider approximate Nash equilibria. A β-approximate Nash equilibrium is an action profile in which no player can increase its payoff by a factor more than β by switching its strategy (while keeping other strategies unchanged). As an extension of Theorem 5.3, Vetta proved that the social value of a β-approximate Nash equilibrium is at most $\beta + 1$ times the optimal social value [21]. Using this result and Theorem 5.4, we have the following corollary:

COROLLARY 5.5. *The social value of a β-approximate Nash equilibrium is at most $\beta + 1$ times the optimum social function value.*

This corollary can be proved directly without referring to valid games, see appendix. The proof based on valid games can be generalized for more general variants of market sharing game and is discussed in Section 10.

5.2 Zipf Distribution

In the general case, we have a factor of 2 for the price of anarchy. This provides an upper bound on the ratio and is a worst-case analysis. However we can prove better bounds for special cases. One important special case is when query rates q_i follow the power law (Zipf) distribution, namely, $q_i = \frac{1}{i^\alpha}$ for a parameter $0 < \alpha \le 1$. These distributions are important, because it has been observed that in many practical situations demand curves follow these distributions [3].

In this section, we prove that in a uniform market sharing game if all players are interested in all markets, the price of anarchy is less than $1.45 + o(1)$ in the worst case, where $o(1)$ depends on n, i.e. $o(1)$ tends to 0 as $n \to \infty$. Furthermore, for cases in which the given bipartite graph is not complete or markets have different costs, we prove that the factor 2 is tight.

THEOREM 5.6. *In the uniform market sharing game, if the given bipartite graph is complete and query rates are from a Zipf distribution, then*

- *The price of anarchy is less than or equal to $\frac{1}{(1-\alpha)^{1-\alpha}} + o(1)$ for any $\alpha < 1$. In particular, it is less than $e^{\frac{1}{e}} + o(1) < 1.45 + o(1)$ for any $\alpha < 1$ and it tends to $1 + o(1)$ when $\alpha \to 1$.*
- *For $\alpha = 1$, the price of anarchy is $\left(1 + \frac{\ln \ln(n)}{\ln(n)}\right)(1 + o(1))$.*

Proof: Suppose the given bipartite graph is complete. Consider a Nash equilibrium and let p be the least index such that the players do not select p but select all markets 1 to $p-1$. No market beyond p can be selected by any player, since otherwise such a player would have an incentive to switch to market p, thus $\frac{q_i}{n_i} \ge q_p$ or $\frac{1}{n_i i^\alpha} \ge \frac{1}{p^\alpha}$ for $1 \le i \le p-1$. Summing over all markets $1 \le i \le p-1$, we get $\sum_{i=1}^{p-1} \frac{1}{i^\alpha} \ge (\sum_{i=1}^{p-1} n_i) \frac{1}{p^\alpha}$. Letting $Q_\alpha(k) = \sum_{i=1}^{k} \frac{1}{i^\alpha}$, we get $p^\alpha Q_\alpha(p-1) \ge n$. As OPT can at best serve all markets, we have that the price of anarchy is at most $\frac{Q_\alpha(n)}{Q_\alpha(p-1)}$.

We consider the two cases $\alpha = 1$ and $\alpha < 1$ separately. We start with $\alpha = 1$. We need to compute $\frac{Q_1(n)}{Q_1(p-1)}$. From $\ln(n) \le Q_1(n) \le \ln(n) + 1$, it is not hard to see that $p > \frac{n}{\ln(n)}$, for sufficiently large n. Therefore,

$$\begin{aligned}\frac{Q_1(n)}{Q_1(p-1)} &\le \frac{\ln(n)+1}{\ln(\frac{n-\ln(n)}{\ln(n)})} \\ &\le \frac{\ln(n)+1}{\ln(n-\ln(n)) - \ln\ln(n)} \\ &= 1 + o(1),\end{aligned}$$

where the last step can be proved using L'Hopital's rule when $n \to \infty$.

We now consider the case $\alpha < 1$. Let $L_\alpha(k) = \frac{1}{1-\alpha}(k^{1-\alpha} - 1)$, then it is easy to see $L_\alpha(k) \le Q_\alpha(k) \le L_\alpha(k) + 1$. Using this fact, we can bound the ratio $\lim_{n\to\infty} \frac{Q_\alpha(n)}{Q_\alpha(p-1)}$. Observing the facts that $p \to \infty$ and $L_\alpha(p-1) \to \infty$ and α is a positive constant less than 1, we can compute the bound as follows:

$$\begin{aligned}
&\lim_{n\to\infty} \frac{Q_\alpha(n)}{Q_\alpha(p-1)} \\
&\leq \lim_{n\to\infty} \frac{L_\alpha(n)+1}{L_\alpha(p-1)} \\
&= \lim_{n\to\infty} \frac{n^{1-\alpha}-1}{(p-1)^{1-\alpha}-1} \\
&\leq \lim_{n\to\infty} \frac{(p^\alpha Q_\alpha(p-1))^{1-\alpha}}{(p-1)^{1-\alpha}-1} \\
&\leq \lim_{p\to\infty} \frac{(p^\alpha (L_\alpha(p-1)+1))^{1-\alpha}}{(p-1)^{1-\alpha}-1} \\
&\leq \lim_{p\to\infty} \frac{p^{\alpha(1-\alpha)} \frac{1}{(1-\alpha)^{1-\alpha}} (p^{1-\alpha}-1)^{(1-\alpha)}}{(p-1)^{1-\alpha}-1} \\
&= \lim_{p\to\infty} \frac{p^{(1-\alpha)\alpha} \frac{1}{(1-\alpha)^{1-\alpha}}}{((p-1)^{1-\alpha}-1)^\alpha} \\
&= \lim_{p\to\infty} \frac{1}{(1-\alpha)^{1-\alpha}} \left(\frac{p^{1-\alpha}}{(p-1)^{1-\alpha}-1}\right)^\alpha \\
&= \frac{1}{(1-\alpha)^{1-\alpha}}
\end{aligned}$$

Now, one can observe that this bound is less than $e^{\frac{1}{e}}$ for any $\alpha < 1$ and is equal to 1 as α tends to 1. ■

THEOREM 5.7. *The factor 2 for the price of anarchy of uniform market sharing game is tight even for Zipf distributions if the given bipartite graph is not complete. Moreover, this bound is tight for general market sharing game even if the given bipartite graph is complete.*

Proof: We give an example with $q_i = \frac{1}{i}$. There are n^2+n markets and n^2 players. Players are partitioned in n groups of size n. Players in group k are interested in markets $k, kn+1, kn+2, \ldots, kn+n$. All budgets and costs are equal, i.e., for all $1 \leq i \leq n^2+n$ and $1 \leq j \leq n^2$, $B_j = C_i = c$. Now, it is not hard to see that there exists a Nash equilibrium in which all players of group k provide market k. Thus, there exists a Nash equilibrium with the social value $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$. However the optimum is to assign a new market to each player. As a result, all markets are provided except n of them. The value of this assignment is $H_{n^2+n} - \sum_{i=1}^{n} \frac{1}{in+n} = H_{n^2+n} - \frac{H_n}{n+1}$. Thus the ratio is $\frac{H_{n^2+n}}{H_n} - \frac{1}{n+1}$ which is equal to $\frac{\ln(n^2+n)}{\ln(n)} = 2$ as $n \to \infty$.

The proof that this bound is tight for general market sharing games is based on an example similar to the above one. The given graph is complete. The only difference is that the budgets of players in group k are $n-k$. The cost of items $k, kn+1, kn+2, \ldots, kn+n-1$ is also $n-k$ for $1 \leq k \leq n$. The cost of the item $kn+n$ is large. It can be shown that the same Nash equilibrium has the price of anarchy 2. Details are omitted here. ■

6. FINDING A NASH EQUILIBRIUM

Nash proved that any strategic game has a mixed Nash equilibrium [12]. However, there are strategic games that have no pure strategy Nash equilibrium. Furthermore, the complexity of finding a (mixed) Nash Equilibrium for strategic games is still an open problem.

In the market sharing game, if we have only one player, finding a Nash equilibrium corresponds to solving optimally a knapsack problem. Thus, the problem of finding a Nash equilibrium in this market sharing game is NP-hard. However, a Nash equilibrium always exists. An existence proof comes from a general framework defined by Rosenthal [16] for congestion games, as our market sharing game is a special case of such games. In the appendix, we sketch this proof.

THEOREM 6.1. *[16] A pure strategy Nash equilibrium always exists for the market sharing game.*

In the rest of this section, we give a polynomial-time algorithm for finding a pure strategy Nash Equilibrium for the *uniform* market sharing game (and this will at the same time prove the existence of such a Nash equilibrium). One main feature of the uniform variant is that it is easy for player j to determine its optimum strategy, given the set of strategies for other players. Indeed, player j only needs to solve an easy maximization problem corresponding to selecting the k_j most rewarding markets, where $k_j = \lfloor \frac{B_j}{C} \rfloor$. We could therefore let players repeatedly and optimally improve their strategy, but the main issue is to show that such a process converges to a Nash equilibrium in polynomial time. In fact, we will not analyze this algorithm. Instead we analyze an iterative algorithm in which each agent is restricted to a set of changes at each step. This proves that if players changes according to these restrictions, they will converge to a Nash equilibrium in polynomially many steps. It would imply that Nash equilibrium can be found in polynomial time. An iterative algorithm to find a Nash equilibrium navigates the *state graph* defined below.

DEFINITION 6.2. *The* state graph*, $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, is a directed graph. Each vertex in $\mathcal{V}$ corresponds to an action profile. There is an arc from vertex $\mathcal{S}$ to vertex $\mathcal{S}'$ with label j if the only difference between $\mathcal{S}$ and $\mathcal{S}'$ is the action of agent j and the payoff of player j in $\mathcal{S}$ is strictly less than her payoff in $\mathcal{S}'$.*

Any vertex in the state graph without any outgoing arc (a sink) corresponds to a Nash equilibrium. Thus in order to prove that a Nash equilibrium exists, we can show the existence of a sink in the state graph. In the following theorem, we give a polynomial-time algorithm to find such a vertex in the state graph.

THEOREM 6.3. *For the uniform market sharing game, a pure strategy Nash equilibrium always exists and can be found in polynomial time. Furthermore, it can be obtained by traversing a path of length at most $m^2 n$ in the state graph.*

Proof: Our algorithm for traversing the state graph and finding a sink proceeds in rounds. The first round starts at the vertex $\emptyset$ corresponding to the set of empty actions. In each round, the first arc traversed corresponds to a player, say j, switching from S_j to S'_j where $S'_j = S_j \cup \{i\}$. In other words, player j only adds precisely one market to its strategy. We refer to this first arc as an *add* arc. After this first arc, subsequent arcs in a round are *change* arcs. These correspond to a player, say j, replacing S_j by $S_j \cup \{i\} \setminus \{k\}$, where $i \notin S_j$ and $k \in S_j$; player j exchanges market k for market i. Furthermore, given j and k, i is selected among all possible markets i of interest to j and not currently in S_j in order to maximize j's payoff. A round finishes when there are no *change* arcs out of the current vertex. Subsequent rounds start at the vertex where the previous round finishes, unless this vertex has no *add* arc outgoing from it in which case this is the last round.

First, observe that when the last round finishes, the current vertex has no *add* or *change* arcs and therefore must be a pure Nash equilibrium. This implicitly uses the fact that we are dealing with a *uniform* market sharing game and therefore any maximal strategy for player j can be obtained from any other maximal strategy by exchanging in and out two markets at a time. Furthermore, at the end of each round, the current state has no *change* arcs outgoing from it, which implies that it corresponds to a pure Nash equilibrium if we suitably modify the budgets of each player (so that they cannot add markets).

As one player adds a market at the beginning of each round, the number of rounds cannot be greater than mn. We now show that each round ends, and in fact ends after traversing at most $m-1$ *change* arcs (and one *add* arc).

Let us focus on one round and let n_i be the number of players servicing market i at *the beginning* of a round. For simplicity, we assume that the markets are sorted in such a way that $\frac{q_1}{n_1+1} \geq \frac{q_2}{n_2+1} \geq \cdots \geq \frac{q_m}{n_m+1}$. Consider any vertex in the round after the first *add* arc. Let S_j be the markets currently served by j and let T_j be the markets of interest to j (i.e. i with $(j,i) \in E$) not in S_j. For player j, let $s(j)$ denote $\min\{i \in T_j\}$. We show by induction that the following properties hold throughout the round:

1. For any player j, $s(j)$ does not decrease during the round.
2. Every market i is covered by n_i players, except one, denoted by p which is covered by n_p+1.
3. For any player j, any market $i \in S_j$ and any market $k \in T_j$ we have $\frac{q_i}{n_i} \geq \frac{q_k}{n_k+1}$.

Properties 1 and 2 are obviously true after the first *add* arc corresponding to player, say l, adding market p. Property 3 is also true after the first *add* arc. Indeed, the condition reduces to the fact that the last round ended in a pure Nash equilibrium except for the case where $j = l$ and $i = p$ where it follows from the choice of p: $\frac{q_p}{n_p} > \frac{q_p}{n_p+1} \geq \frac{q_k}{n_{k+1}}$.

We see now what happens when we traverse a *change* arc corresponding to player l exchanging two markets. Condition 3 implies that l leaves market p since all other markets do not increase l's payoff. Thus property 2 is maintained after the change (with a different value for p). Secondly, l will now serve market $s(l)$ by definition of $s(l)$. This implies that $\frac{q_p}{n_p+1} < \frac{q_{s(l)}}{n_{s(l)}+1}$, i.e. $s(l) < p$. This means that property 1 is also maintained. To verify that Property 3 is still maintained, we only need to consider the cases in which $j = l$ and either $k = p$ or $i = s(l)$. If $i = s(l)$, property 3 follows from the definition of $s(l)$: $\frac{q_{s(l)}}{n_{s(l)}} > \frac{q_{s(l)}}{n_{s(l)}+1} \geq \frac{q_k}{n_k+1}$. If $k = p$, it follows from $\frac{q_i}{n_i} \geq \frac{q_{s(l)}}{n_{s(l)}+1} > \frac{q_p}{n_p+1}$.

All three properties are maintained during the round. Furthermore, since player l replaces market p by market $s(l)$ and $s(l) < p$, we have that p decreases as we traverse *change* arcs. This implies that we have at most $m-1$ *change* arcs in a round. This proves our bound of nm^2 on the length of the traversal we construct before reaching a pure Nash equilibrium.

■

In order to run the algorithm, we actually do not need to construct the entire state graph. We only need to be able to find the next arc to traverse, and this can be done in $O(m+n)$ time, resulting in a total running time of $O((m+n)m^2n)$.

7. GREEDY BEHAVIOR ANALYSIS

Although we have proved that the price of anarchy for a Nash equilibrium is upper bounded by 2, there are still two main issues that are left unaddressed. First, converging to an exact Nash equilibrium in a way described in Theorem 6.1 requires agents to solve optimally an NP-complete knapsack problem in order to find their best response. Considering the computational constraints of rational agents, we consider agents that choose their action based on an approximation algorithm for knapsack. The second issue is the speed with which the agents can converge to a Nash equilibrium or an approximate Nash equilibrium. We consider both issues in this section.

In the proof of theorem 6.1 given in the Appendix, we see that if agents improve their payoff, they will converge to a Nash equilibrium after finitely many steps (since the potential function can take only finitely many values and increases with every improvement). This result can be generalized to the setting in which each agent chooses its action using a β-approximation algorithm for knapsack (keeping the actions of all the other players unchanged). A β-approximation algorithm is a polynomial-time algorithm guaranteed to return a feasible solution of value (i.e. payoff) at least $1/\beta$ times the optimum value.

THEOREM 7.1. *If all players rely on a β-approximation algorithm to improve their actions, they will converge to a β-approximate Nash equilibrium in finitely many steps.*

From Corollary 5.5, we know that such a β-approximate Nash equilibrium gives a social function value within a factor $\beta+1$ of the optimum social function value.

The proof of the above theorem is similar to that of Theorem 6.1. Using the same potential function, one observes that, after finitely many steps, no agent will be able to improve its action, and therefore, we will have reached a β-approximate Nash equilibrium.

The above theorem, as well as Theorem 6.1, do not bound the number of steps needed for players to converge to an equilibrium. In the remainder of this section, we analyze a greedy behavior of the players, *greedy upon arrival*, in which players enter the game one by one and select their first action by using a β-approximation algorithm for knapsack for some constant β. We show that after the players have selected their first action, i.e. after each player makes just one decision, the resulting action profile has a social function value within a logarithmic factor of the optimum social function value.

THEOREM 7.2. *If each player chooses its first action using a β-approximation algorithm then the social function of the resulting action profile is at least $\frac{1}{(\beta+1)H_n}$ of the optimum social value, where $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \leq \ln(n) + 1$. Moreover, this logarithmic bound is tight, up to a multiplicative factor.*

Proof: For the purpose of the analysis, we consider the following restricted multiple knapsack problem (*RMKP*) corresponding to an instance of our market sharing game. In an instance of the *RMKP* problem, there are m groups of n items each for a total of mn items. The n items in group i all have size C_i, for $1 \leq i \leq m$. The values of items in group i are $(q_i, \frac{q_i}{2}, \frac{q_i}{3}, \ldots, \frac{q_i}{n})$. There are n bins with capacities $(B_1, B_2, \ldots, B_n)$. Bin j can get an item in group i iff agent j is interested in market i ($(j,i) \in E(G)$). We are only allowed to place at most one item of each group in a bin, and the goal is to place (some of the) items in bins so that the size assigned to every bin is at most its capacity and the total value of items assigned is maximized. Given the objective function, we can focus our attention on algorithms for which an item of group i is not assigned to any bin if a higher valued item of the same group is not assigned to any bin. We prove our claims using the following lemma:

LEMMA 7.3. *Consider the following greedy algorithm for* RMKP *problem: arbitrarily order bins, pack bins one at a time using a β-approximation algorithm for the single knapsack problem instance* *for the current bin (considering only the highest valued item remaining for each group) and discard the packed items. Then the resulting assignment is a* $\beta + 1$*-approximation for* RMKP.

Proof: Let **OPT** be an optimum solution of *RMKP*. Let A_j be the set of items in bin j in **OPT** which do not appear in the greedy solution in any bin, and X_j their total value. Let B_j be the set of items in bin j in the greedy solution, and Y_j their total value. Observe that if A_j and B_j each contain a (possibly different) item of group i then the value of the item, say p, in A_j is no larger than the value of the item, say q in B_j (since p does not appear in the greedy solution in any bin). Since we use a β-approximation algorithm for filling each bin, $\beta Y_j \geq X_j$. Thus $\beta \sum_{1 \leq j \leq m} Y_j \geq \sum_{1 \leq j \leq m} X_j \geq$ **OPT** $- \sum_{1 \leq j \leq m} Y_j$, where by abuse of notation $\mathbf{OPT}$ also denotes the value of the optimum solution. Thus, $(\beta + 1) \sum_{1 \leq j \leq m} Y_j \geq$ **OPT** as desired. ■

We are ready to continue the proof of Theorem 7.2. If n_i is the current number of agents servicing market i, the current value of this market is $\frac{q_i}{n_i+1}$ for a new agent. Equivalently, we can consider a new market of value $\frac{q_i}{n_i+1}$. In this way, we add different copies of the same market with different query rates and the same size. The only restriction is that no player can provide two markets of the same group. Now one can observe that after one step of greedy behavior by each player, the resulting action profile is equivalent to the assignment of the greedy algorithm from Lemma 7.3 for *RMKP*. Let the resulting assignment be $\mathcal{G}$. Now we prove that the sum of Y_j's over all players is at most H_n times the sum of payoffs of players in $\mathcal{G}$ (where Y_j is defined in Lemma 7.3). In $\sum_{j \in U} Y_j$, the sum of terms corresponding to markets of group i is at most $q_i + \frac{q_i}{2} + \ldots + \frac{q_i}{n} = q_i * H_n$, while market i contributes to the sum of payoffs in $\mathcal{G}$ as q_i. Thus $\sum_{j \in U} Y_j \leq H_n \sum_{j \in U} P_j(\mathcal{G})$.

In an instance of *RMKP* there are extra markets. Therefore, the optimum solution of *RMKP*, $\mathbf{OPT}'$, is greater than or equal to the optimum social function value, **OPT**. Thus,

$$(1+\beta)H_n \sum_{j \in U} P_j(\mathcal{G}) \geq (1+\beta) \sum_{j \in U} Y_j \geq \mathbf{OPT}' \geq \mathbf{OPT}$$

as desired.

Furthermore, this bound is tight up to a multiplicative because of the following example with n markets and n players: Let $q_i = \frac{n}{i} - \epsilon$ for all $1 \leq i \leq n$ where ϵ is sufficiently small. Player j is interested in item 1 and items $j, j+1, \ldots, n$. All costs and budgets are the same and are equal to 1. It is easy to check that if players choose their initial action in the order $1, 2, \ldots, n$ then in the resulting assignment after each player makes one decision the only provided market is market 1. However, in the optimum assignment all markets are provided. Thus, The ratio between the optimum and the value of the resulting assignment is $\Omega(\log n)$ after one step from each player. ■

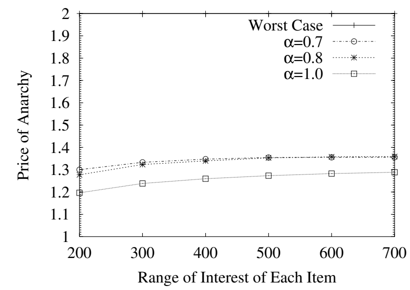
one can show that if β-approximate players continue optimizing their payoff after the first round, the outcome of the game will remain within an $O(\log n)$ of the optimum.

8. EVALUATION

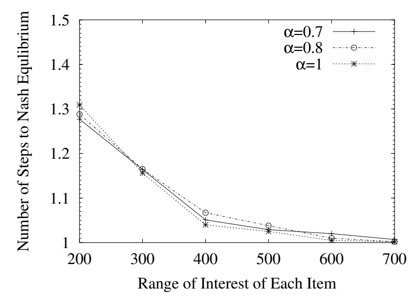
In this section we investigate the efficiency of our decentralized caching scheme in a simulated network scenario. We are interested in obtaining the price of anarchy and determining how quickly the game converges to an exact or approximate pure strategy Nash equilibrium. We investigate the advantages of the greedy behavior in terms of the payoff function as well as evaluate how far it is from the social optimum.

We generated 10 random networks, each with 100 nodes (resident subscribers). These nodes are randomly placed in a 800×800 rectangular region. Each node has an 802.11b interface and has a maximum transmission radius of 115 units. Each item has a radius of interest R_I, i.e., only nodes within that radius of interest benefit from caching the item. We vary this radius from 200 to 700. In our simulation the effect of the transit subscribers is captured by the query rate. We assume a query can originate from anywhere in the network within the item's radius of interest. We assume that we have 1000 items whose popularity follows a Zipf distribution. We vary the α parameter of the Zipf distribution. The items' size is 1 unit for the uniform market sharing game and for the non-uniform market sharing game it follows a lognormal distribution with mean 7.5 and standard deviation 1.5. Each resident subscriber can cache at most 5 items in the uniform case, 20 units for the non-uniform case. We focus our study on issues related to the impact of selfish behavior on network performance and therefore, do not perform packet level simulations. All our results are averaged over 10 random networks.

8.1 Uniform Market Sharing Game



(a) Price of Anarchy



(b) Number of Steps to Convergence to a Nash Equilibrium

Figure 2: Price of Anarchy and Number of Steps to Converge to a Nash Equilibrium for Uniform Game

We would like to investigate the price of anarchy and how quickly the players converge to Nash Equilibrium. Figure 2-a shows the price of anarchy with respect to the item's range of interest R_I. The larger R_I is, the more edges in the bipartite graph between items and resident subscribers. For Zipf distribution with $\alpha = 0.7$, the worst-case price of anarchy from our analysis is 1.45 and 2 for complete and incomplete bipartite graphs respectively. The price of anarchy ρ_a from our simulation is between 1.30 and 1.36. Therefore, ρ_a is within 90% to 95% of 1.45 and 65% to 68% of 2. Therefore, the price of anarchy is far from the worst case for the incomplete bipartite graph case and close to the worst case for the complete bipartite graph. Note that the price of anarchy function $\rho_w = \frac{1}{(1-\alpha)^{(1-\alpha)}}$ increases in interval [0 0.63] and decreases in [0.63 1). However, according to [3], in the case of popular items, α

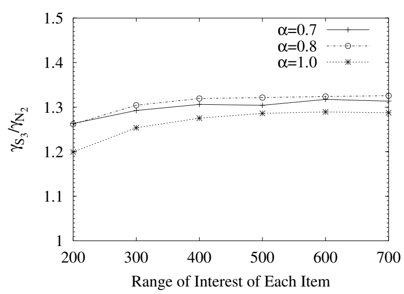
is between 0.65 and 0.8. In this particular range of α we observed that the price of anarchy ρ_a increases only slightly as R_I increases. This is because popular items are within the range of more players and as more players cache common items, ρ_a increases. To see the impact of α on the price of anarchy, we ran our simulation with $\alpha = 0.8$ and $\alpha = 1$. The observations are similar to the case of $\alpha = 0.7$.

	Number of Steps			
	1	2	3	4
percentage of Nodes	78%	20%	1%	1%

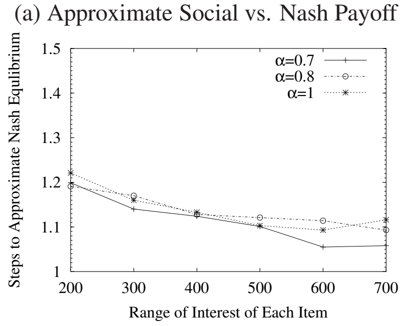
Table 1: Uniform Game: Percentage of Nodes with the Same Number of Steps

Figure 2-b shows the average number of steps required of each player to converge to the pure strategy Nash equilibrium. As we can see, each player takes an average of 1.3 steps or less. Recall that, by a step, we mean that the player changes to a different set of items to cache in response to the action taken by other players. Table 1 shows for a given network setting the percentage of nodes requiring the same number of steps after an equilibrium is attained. We see that, 78% of the players never change the set of items to cache. 20% of them change once. Only 1% change twice and three times. Table 2 shows the percentage of nodes with the same number of cache replacements. We see that 78% of the players stay with the original set of items. Only 19% replaced two items. 2% changed 4 items and 1% changed 6 items in the cache.

8.2 General Market Sharing Game



(a) Approximate Social vs. Nash Payoff



(b) Number of Steps to Converge to a 2-Approximate Nash Equilibrium

Figure 3: The Approximate Social vs. Nash Payoff and Number of Steps to Converge to a 2-approximate Nash Equilibrium for General Market Sharing Game

For the general market sharing game, as finding the best response for a player is NP-complete, we choose to investigate a 2-approximate Nash equilibrium based on a simple 2-approximate greedy algorithm. The algorithm [20] works as follows. Order the items by payoff over size. Let the sorted order of objects be $a_1, a_2, \cdots, a_m$. Find the lowest k such that the size of the first k objects exceeds the cache space. Pick the more profitable of $\{a_1, a_2, \cdots, a_{k-1}\}$ or $\{a_k\}$. We refer to the sum of the payoffs of the items cached when the players reach this 2-approximate Nash equilibrium as 2-approximate Nash payoff and denote it as γ_{N_2}.

Computing the optimum social function involves solving a multiple knapsack problem. As this problem is also NP-hard, we choose a simple 3-approximation algorithm [5]. The algorithm greedily applies the 2-approximate knapsack algorithm for each knapsack. We refer to sum of the payoffs of the cached items computed by this algorithm as the 3-approximate social payoff and denote it as γ_{S_3}.

	Number of Steps	
	1	2
percentage of Nodes	83%	17%

Table 3: General Market Sharing Game: Percentage of Nodes with the Same Number of Steps

	Number of Cache Replacements			
	0	1	2	3
percentage of Nodes	83%	7%	9%	1%

Table 4: General Market Sharing Game: Percentage of Nodes with the Same Number of Cache Replacements

The performance of the approximate Nash payoff with respect to the approximate social payoff is given by the ratio $\gamma_{S_3}/\gamma_{N_2}$. Figure 3-a shows that this ratio is about 1.3 showing that the non-cooperative caching is very efficient. Figure 3 also shows the number of steps it takes to converge to the 2-approximate Nash equilibrium as the range of interest R_I increases. Consider the specific networking setting of $R_I = 200$. Table 3 shows that 83% of the players only take one step while the remaining 17% take 2 steps to reach the approximate Nash equilibrium. Furthermore from Table 4, we see that, 83% of players stayed with the initial set of items, 7% replaced just one item, 9% replaced two and 1% replaced three items.

8.3 One Step of Greedy Behavior

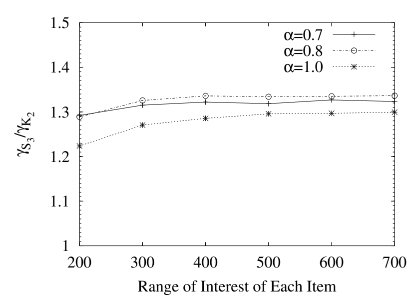


Figure 4: Compare Greedy Payoff with Approximate Social Payoff

	Number of Cache Replacements						
	0	1	2	3	4	5	6
percentage of Nodes	78%	0%	19%	0%	2%	0%	1%

Table 2: Uniform Game: Percentage of Nodes with the Same Number of Cache Replacements. Range of interest $R_I = 200$

We would also like to see the payoff for one step of greedy behavior. Each player, when it decides to participate, caches items according to the 2-approximate knapsack algorithm. Figure 4 compares the corresponding one step greedy payoff γ_{K_2} with the 3-approximate social payoff γ_{S_3}. We see that the behavior of this payoff ratio is similar to the ratio $\gamma_{S_3}/\gamma_{N_2}$ obtained in the previous section. In fact, the one step social payoff γ_{K_2} is only at most 2% worse than the 2-approximate Nash payoff.

9. RELATED WORK

Caching has been widely studied in the context of wired networks [22]. Caching in wireless networks has recently become a popular research area. Sailhan and Issarny [18] propose protocols to enable cooperative caching in ad-hoc networks. They also study cache management strategies that aim to reduce energy consumption and network load. Xiang et al. [23] propose a cooperative cache management scheme in cellular networks. They present a simulation study of cache replication strategies in base stations for streaming services. Nuggehalli et al. [14] present an energy-efficient cache placement scheme. Our work studies caching in a non-cooperative environment. The problem of non-cooperative caching is fundamentally different from its cooperative counterpart.

The problem of how to stimulate cooperation among selfish nodes in ad hoc networks and multi-hop cellular networks has received significant attention recently. Zhong et al. [24] propose a credit-based system which relies on a central authority to collect receipts from forwarding nodes. Charges and rewards are based on the receipts. They assume the availability of public and private keys to compute message signatures as receipts. We cannot directly apply their scheme since we use symmetric keys rather than asymmetric keys in our integrated 3G and ad-hoc networks. Salem et al. [19] propose a charging and rewarding scheme to make collaboration rational for selfish nodes. Their solution is based on symmetric cryptography. However, they require all data packets to go through the base station. Since our goal is to offload the data distribution from the base station as much as possible, their scheme cannot be applied directly in our context. Jakobsson et al. [7] propose a micro-payment scheme for multi-hop cellular networks that encourages collaboration in packet forwarding. The sender attaches a payment for each packet. In our context, the sender gets paid by the receiver through the base station. In addition, it is natural to charge per session in our context. Therefore, their scheme does not apply directly. Reputation-based schemes such as [4] is not intended in our context since it cannot deal with certain collusion scenarios in our context, e.g. collusion between the sender and receiver where forwarding nodes do not get remunerated.

Recently game theory has been found to have applications to wireless networks [6, 2, 8]. Anderegg and Eidenbenz [2] applied the VCG mechanism from mechanism design theory to design a routing protocol that is guaranteed to find the most cost-efficient path and every agent is truthful about their cost. Eidenbenz et al. [6] proposed a topology control game that models user's selfish behavior in forming ad-hoc network topology. Lin et al. [8] used game theory to propose an admission and rate control framework for CDMA data networks. To the best of our knowledge, we are the first to propose a game-theoretic study of caching in wireless networks.

The issues of the efficiency of computing Nash equilibria and considering computational constraints of the central authority and selfish agents, and the outcome of the selfish behavior of these agents have been investigated recently in computer science [15, 13]. The market sharing game is a special case of congestion games introduced by Rosenthal [16]. In these games, each player chooses a particular combination of factors out of a common set of factors. The payoff from each factor is a function of the number of players who have chosen this factor in their set. Milchataich [10] studied more general settings and the length of best response paths in this set of games. Congestion games are in a general class of potential games[11]. None of the these work considered the efficiency of the outcome of the selfish behavior of agents or the speed of convergence to an approximate solution or the computational constraints of selfish agents. We address all the above issues for the special case of market sharing games. We consider a social function and the outcome of selfish behavior of players in terms of this social function. Comparing the value of the social function in the outcome of selfish behavior with the optimum value of the social function is done via the notion of the price of anarchy. For example, for a variant of non-atomic congestion games, Roughgarden and Tardos [17] give some upper bounds on the price of anarchy.

10. DISCUSSION AND FUTURE RESEARCH

There are two main issues discussed in this paper. One is the architecture for enabling mobile content distribution in ad-hoc networks, and the other one involves the game theoretic formulation of the non-cooperative caching problem.

The architectural framework for mobile content distribution described in this paper assumes that each player can determine the amount of requests it will handle if it caches an item. In reality, the player may not know this information. However, this information can be estimated by observing the cache request and reply messages of our protocol. We also assumed that only one player changes the set of cached items at a given time. Once the change is made, all other relevant players are assumed to become aware of the changed payoff. In reality, multiple players may make decisions at the same time. This issue can be addressed by using a random timer to decide when to change thus preventing synchronization effects. Using a timer would allow the effect of the change to stabilize before players respond to the new change. If the estimated payoff changes continuously, the player will have no incentive to change since replacing a cached item will incur a cost. The cost is incurred since the player has to either request the changed item from the 3G service provider or from some other players who have cached the item. We assume the cost of changing cached items are amortized. Therefore, players have an incentive to change if the payoff of an item is stable and increases his utility.

In our current work we have used the market sharing game to capture the essential issues of our caching application. There are more general variants of the market sharing games such as assuming that a market may have different costs for providing services for different players. This problem can be formalized as a more general non-cooperative game which is still a special case of valid congestion games. Thus, the factor 2 result for the price of anarchy and the existence of a pure strategy Nash equilibrium hold for this more general setting as well. Another more general variant of market sharing games can be defined by considering the capacity constraint of individual players. Currently we have assumed that the ad-hoc network has enough capacity to handle all the requests it chooses to serve. If network capacity becomes limited, we can formalize the game appropriately, and the result for the price of anarchy should still hold. However the existence of a pure strategy Nash equilibrium is very much dependent on the definition of the game. Modeling and understanding these more general games is of theoretical and practical interest and we leave it for future research.

We observed that players using a β-approximate algorithm also converge to an approximate Nash equilibrium, but we do not bound the number of steps for this convergence. Bounding the number of steps by a polynomial bound is an open problem and is of theoretical and practical interest. In particular, such a bound would give a polynomial time algorithm to find an approximate Nash equilibrium.

Modeling computationally bounded agents by approximation algorithms and proving bounds on the speed of convergence to an approximate social function or an approximate Nash equilibria is an interesting area of research and can be applied to different sets of games such as potential games, congestion games, and valid games.

11. CONCLUSION

In this paper, we propose a mobile content distribution architecture to offload popular data items from a 3G network to its subscriber based ad-hoc networks. We present a protocol to enable decentralized caching and propose novel incentive mechanisms for proper accounting and crediting. Our incentive mechanisms prevent or discourage players from colluding for selfish gains at the expense of other users.

We study the selfish behavior of the players involved in caching and forwarding by using game theoretic approaches. We formulate the caching problem as a general market sharing game. We observe that pure strategy Nash equilibrium exists in this game and present a polynomial time algorithm to find it in the case when item sizes are the same. Finding a pure strategy Nash equilibrium for the general game is NP-complete. We model computationally bounded agents by using approximation algorithms and observe that they also converge to an approximate Nash equilibrium. We show that after one step of greedy behavior by the players, the social value of the resulting assignment is within an $O(\log n)$ of the optimum. We show that the price of anarchy is bounded by 2 for any arbitrary distribution of popularity of the cached items. For a special case where the popularity of items is according to a Zipf distribution and caching any item has a positive reward for all players, we obtain a bound of 1.45. Our simulation study investigates the price of anarchy and the convergence of exact and approximate Nash equilibria. Our simulations show that for most of real and practical instances of the game, the price of anarchy is better than the worst case. Our results also show that typically most players converge to the equilibrium in one or two steps most of the time.

Acknowledgment

We thank Prof. Joseph Y. Halpern at Cornell University for initial discussions on the problem. We are also very grateful to Dr. Eric Grosse and Dr. Sanjoy Paul at Bell Labs, and Sheng Zhong at Yale University for helpful discussions and comments on the security aspects of the paper.

12. REFERENCES

[1] 3G Today. `http://www.3gtoday.com/`.

[2] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of ACM MOBICOM*, pages 245–259, 2003.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM*, pages 126–134, 1999.

[4] S. Buchegger and J-Y. Le Boudec. Performance analysis of the CONFIDANT protocol:cooperation of nodes–fairness in dynamic ad-hoc networks. In *Proceedings of ACM MOBIHOC*, pages 80–91, 2002.

[5] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 213–222, 2000.

[6] S. Eidenbenz, V. S. Anil Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. In *Proceedings of the joint workshop on Foundations of mobile computing*, pages 2–11, 2003.

[7] M. Jakobsson, J.-P. Hubaux, and L. Buttyan. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In *Proceedings of Financial Cryptography*, pages 15–33, 2003.

[8] H. Lin, M. Chatterjee, S. K. Das, and K. Basu. ARC: An integrated admission and rate control framework for cdma data networks based on non-cooperative games. In *Proceedings of ACM MOBICOM*, pages 326–338, 2003.

[9] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu. UCAN: a unified cellular and ad-hoc network architecture. In *Proceedings of ACM MOBICOM*, pages 353–367, 2003.

[10] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economics Behavior*, 13:111–124, 1996.

[11] D. Monderer and L. S. Shapley. Potential games. *Games and Economics Behavior*, 14:124–143, 1996.

[12] J. F. Nash. Equilibrium points in N-person games. In *Proceedings of National Academy of Sciences (NAS)*, pages 48–49, 1950.

[13] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of ACM STOC*, pages 129–140, 1999.

[14] P. Nuggehalli, V. Srinivasan, and C. Chiasserini. Energy-efficient caching strategies in ad hoc wireless networks. In *Proceedings of ACM MOBIHOC*, pages 25–34, 2003.

[15] C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of ACM STOC*, pages 749–753, 2001.

[16] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[17] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of ACM*, 49(2):236–259, 2002.

[18] F. Sailhan and V. Issarny. Cooperative caching in ad hoc networks. In *Proceedings of the International Conference on Mobile Data Management (MDM)*, 2003.

[19] N. B. Salem, L. Buttyan, J.P. Hubaux, and M. Jakobsson. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *Proceedings of ACM MOBIHOC*, pages 13–24, 2003.

[20] V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Incorporated, Jun 1999.

[21] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of IEEE FOCS*, pages 416–425, 2002.

[22] J. Wang. A survey of web caching schemes for the Internet. *SIGCOMM Comput. Commun. Rev.*, 29(5):36–46, 1999.

[23] Z. Xiang, Z. Zhong, and Y. Zhong. A cache cooperation management for wireless multimedia streaming. In *Proceedings of the IEEE International Conferences on Info-tech and Info-net ICII*, pages 328–333, 2001.

[24] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of IEEE INFOCOM*, pages 1987–1997, 2003.

APPENDIX

Proof of Corollary 5.5.

Proof: Let Y_j be the payoff of player j in the approximate Nash equilibrium. Let X_j be the sum of the q_i's over the set of markets that player j serves in OPT but are not served by any agent in the equilibrium being considered. By the definition of approximate Nash equilibrium, we have $\beta Y_j \geq X_j$. Otherwise, player j would switch action. Furthermore, $\sum_{j \in U} Y_j \geq \gamma(OPT) - \sum_{j \in U} X_j$, as $\sum_{j \in U} X_j$ is at least the total amount of query of the markets serviced by OPT but not by the approximate Nash equilibrium. Thus, $\gamma(OPT) \leq (\beta + 1) \sum_{j \in U} Y_j$, as desired. ■

Proof of theorem 6.1.

Proof: The proof is based on defining a potential function and proving that it decreases as agents improve their payoffs by changing their strategies. The potential function is $\sum_{i=1}^{n} \sum_{t=1}^{n_i} \frac{q_i}{t}$. Consider player i who changes her strategy from set S to S'. Let $A = S - S'$ and $B = S' - S$. Let F be the value of the potential function when i plays S and F' be the value when i plays S'. It is easy to see that $F' - F = \sum_{i \in B} \frac{q_i}{n_i + 1} - \sum_{i \in A} \frac{q_i}{n_i}$. This value is exactly the increase in the payoff of i. Assuming the fact that i improves her payoff, we know $F' > F$. This shows that the pure strategy that maximizes this potential function is a Nash equilibrium, since if some player was able to increase her own payoff, she would improve the potential function by the same amount.

■