A Game Theoretical Data Replication Technique for Mobile Ad Hoc Networks

Samee Ullah Khan[1], Anthony A. Maciejewski[1], Howard Jay Siegel[1,2]
[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science
Colorado State University
Fort Collins, CO 80523
{samee.khan, aam, hj}@colostate.edu

Ishfaq Ahmad
Department of Computer Science and Engineering
University of Texas
Arlington, TX 76019
iahmad@cse.uta.edu

Abstract

Adaptive replication of data items on servers of a mobile ad hoc network can alleviate access delays. The selection of data items and servers requires solving a constrained optimization problem, that is in general NP-complete. The problem is further complicated by frequent partitions of the ad hoc network. In this paper, a mathematical model for data replication in ad hoc networks is formulated. We treat the mobile servers in the ad hoc network as self-interested entities, hence they have the capability to manipulate the outcome of a resource allocation mechanism by misrepresenting their valuations. We design a game theoretic "truthful" mechanism in which replicas are allocated to mobile servers based on reported valuations. We sketch the exact properties of the truthful mechanism and derive a payment scheme that suppresses the selfish behavior of the mobile servers. The proposed technique is extensively evaluated against three ad hoc network replica allocation methods: (a) extended static access frequency, (b) extended dynamic access frequency and neighborhood, and (c) extended dynamic connectivity grouping. The experimental results reveal that the proposed approach outperforms the three techniques in solution quality and has competitive execution times.

This research was supported by the NSF under grant CNS-0615170 and by the Colorado State University George T. Abell Endowment.

978-1-4244-1694-3/08/$25.00 ©2008 IEEE

1 Introduction

Due to the continuous mobility of servers, an ad hoc network suffers from frequent disconnections. This phenomenon is undesirable when mobile servers must access data from each other. Because one cannot control network disconnections, an alternative solution to this problem is to replicate data onto mobile servers so that when disconnections occur, mobile servers can still access data [14]. With replication, data retrieval becomes faster because requests are processed locally or from mobile servers in close proximity. Replication also improves data reliability and fault tolerance. Furthermore, replication also conserves energy by traversing fewer radio links to access the closest replicated data.

The ad hoc data replication problem (ADRP) was first introduced by Hara [12], and further extended, e.g., [13–15], to incorporate various consistency related issues. We build on the above mentioned work and address the selfish behavior of mobile servers in our solution concept. In ad hoc networks, resources may belong to different self-interested servers [19]. These servers may manipulate the resource (replica) allocation mechanism for their own benefit by misrepresenting their preferences, which may result in severe performance degradation [10]. Such scenarios are often modeled and studied using game theory (GT).

As indicated by Arrow's impossibility theorem for satisfactory voting systems [5], aggregating player valuations (or preferences) to reach a collective decision is a difficult problem. It is further complicated by the possibility that the players might try to manipulate the mechanism. Nissan and

Ronen [29] were the first to consider discrete optimization problems in the context of GT, where the correct valuation is not directly available to the mechanism. Instead, players report some valuation to the mechanism, but they might lie. Their contribution is significant in the sense that they break the traditional barriers of GT by explicitly stating that the mechanism's objective function may have nothing to do with social welfare, which is the crux of the theory of collective decision making. Hence, many optimization problems can now be studied by relating player valuations to the objective function of the mechanism because both rely on the player's valuation and both determine the player's strategies. Nissan and Ronen termed their framework as Algorithmic Mechanism Design (AMD).

AMD can be used to force the players to always tell the truth and follow the rules by laying out a set of incentives and repercussions. In the literature, such mechanisms are known as "truthful" mechanisms [3]. Each player in AMD has some valuation function that quantifies its benefit or loss. Every player reports the output of its valuation function to a centralized mechanism, which chooses an outcome that optimizes a given objective function and makes payments to players. The design of payments is important because side payments are used to provide incentives to the players to report to the mechanism truthfully.

In this paper, we design a truthful mechanism for the ADRP, called Data Replication Game (DRG), where each player's valuation is naturally expressed by a single positive real number. In DRG we restrict the form of valuations but allow general objective functions. This is contrary to Vickrey-Clarke-Groves (VCG) mechanisms, which allow arbitrary valuation functions but apply only to utilitarian objective functions [10]. Thus, our results are also applicable to optimization problems other than ADRP. The outcome of the mechanism we consider will always define some set of replica allocations at the player's mobile server. A player's valuation will always be the cost it incurs per replica, and this valuation will have some physical significance in terms of the amount of traffic (read requests from servers that do not hold replicas and are the closest to the player's mobile server plus updates) that it processes. The goal of a player is to maximize its profit, which is payment minus cost. The goal of the mechanism is to minimize the total *data item transfer cost* in the network due to the read and update accesses. In DRG, we use side payments to encourage players to tell the truth. It is known that for some output functions no side payments can make the resulting mechanism truthful [3,29]. However, Archer and Tardos [2] have shown that output functions that can be used in truthful mechanisms are those where the resource allocation made to a player decreases as the cost increases, where each player's valuation function is a composed of only one parameter. We build on that result and show that the same outcome holds when each player's valuation function is composed of multiple parameters.

Our contributions: This paper is principally concerned with designing an efficient and effective data replication technique for ad hoc networks.

1. We derive a data access cost model for ADRP that is general, flexible, and adaptable for various applications of ADRP.
2. We design a truthful mechanism to suppress the selfish behavior of the servers and derive a payment scheme where each player's valuation function is a composition of multiple parameters.
3. It is known that even when we have a truthful mechanism and a payment scheme the problem to find an optimal allocation algorithm is NP-complete [29]. We derive a polynomial-time allocation algorithm that in the worst case guarantees an allocation that is $(2+\epsilon)$-optimal (twice the cost of an optimal allocation). The worst case running time of the allocation algorithm is $\mathcal{O}(mn^2)$, where $\underline{m}$ and $\underline{n}$ are the total number of mobile servers and data items, respectively.
4. The proposed GT technique is extensively evaluated against three state-of-the-art ad hoc network replica placement methods: (a) extended static access frequency [13], (b) extended dynamic access frequency and neighborhood [14], and (c) extended dynamic connectivity grouping [15]. Experimental data is recorded to measure the effects of: (a) relocation period, (b) read access frequency, (c) update access frequency, and (d) server storage capacity. The comparative studies reveal that the proposed approach outperforms the three methods in solution quality and has competitive execution times.

The remainder of this paper is organized as follows. In Section 2, we give a brief description of the related works and discuss how our approach is different from previous work in this field. We describe the system model and derive a mathematical model for ADRP in Section 3. Section 4 explains the DRG technique for ADRP, followed by experimental evaluations in Section 5. Finally, in Section 6, we summarize this paper.

2 Related Work

The ADRP can be considered as an extension of the classical file allocation problem (FAP) [6], but is more complicated because of factors such as the mobility of servers and potential network disconnections. Several extensions of the FAP have been studied for static networks such as the data allocation problem (DAP) [1], the data replication problem (DRP) [27], the data staging problem (DSP) [35, 36], and video allocation problem (VAP) [25]. All of the above mentioned methods and problem formulations are related to the

ADRP, because all address the improvement of data accessibility by replicating data items. The current work differs from the previous work in that different target systems are considered with different communication and storage characteristics. Furthermore, because server mobility is not an issue in static networks, it is usually sufficient to create a few replicas of a data item, and thus no special strategy is required that addresses network disconnections.

Several strategies for caching data contents in mobile computing environments have been proposed [17, 18], that address the issue of keeping consistency between original data and its caches. They are considered to be similar to our approach, because both allocate data on mobile servers. However, these strategies assume only one-hop wireless communication. Thus, they are completely different from our approach which allows multi-hop communication in ad hoc networks.

Another closely related research topic is that of push-based information systems in which a server repeatedly broadcasts data to clients using a broadband channel [9,31]. In these strategies, clients are typically mobile servers and cache replacement is determined based on several parameters such as the access frequency from each mobile server to each data item, the broadcast frequency of each data item, and the time remaining until each item is broadcast next. They are also considered to be similar to our approach. However, when comparing the strategies for caching and replication, both approaches are completely different because in push-based information systems the clients do not cooperatively share cached data items.

Hara's work [12–15], is the closest among all of the related work on ADRP. However, our work differs from Hara's in: (a) deriving a mathematical problem formulation for ADRP, (b) proposing an optimization technique that allocates replicas so as to minimize the network traffic under storage constraints with "read from the nearest" and "push based update through the primary mobile server" policies, and (c) using a strict consistency model as opposed to an opportunistic consistency model.

The game theory literature contains an enormous body of work on mechanism design, also called *implementation theory*. See [28] for an introduction to the field, or the survey [16]. The Gibbard-Satterthwaite theorem [8] is the main negative result. It states that, when the players' domain of possible preferences is sufficiently rich, truthful non-dictatorial mechanisms do not exist. In light of this, it is common to specialize by allowing side payments to the players, and assuming each player tries to maximize the sum of its payment plus its intrinsic valuation of the outcome. The celebrated Vickrey-Clarke-Groves (VCG) mechanism [10] is the main general positive result here. It handles arbitrary valuation functions, but only the utilitarian objective function, that maximizes the sum of the players' valuations. Nevertheless, this objective function captures some interesting combinatorial problems, in addition to the more usual social welfare functions [4]. For instance, the shortest path in a graph with respect to edge costs maximizes social welfare because it minimizes the total cost incurred.

Recently, game theoretical analysis has gained a considerable amount of popularity for the study of self-interested entities in distributed computing systems [20–24,26]. These studies are considered to be similar to our approach because they all use game theoretical VCG analysis; however, the above mentioned research solves the DRP, whereas our focus is the ADRP. More specifically, there is a significant difference between the definition of truthfulness used in [23, 24] and the one used in this paper. Indeed, the randomized replica allocation technique in [23, 24] yields a truthful dominant strategy for any possible random choice of the algorithm, however, the mechanism is *truthful in expectation*, a weaker notion of truthfulness [28]. A randomized mechanism can be seen as a probability distribution over deterministic mechanisms: an element x is selected randomly and the corresponding mechanism is used. So, the mechanism in [23, 24] is truthful for every fixed x. Moreover, in [20], the notion of utility is replaced by the *expected utility* one: even though the expected utility is maximized when telling the truth, for some x there might exist a better (untruthful) strategy. We distinguish ourselves from the above mentioned work by deviating in the definition of truthfulness, and say that the replica placement mechanism is deterministic and *truthful in implementation*, the strongest form of truthfulness [28].

The work reported in [26] is notable because it explicitly chooses not to maximize the social welfare, but maximize revenue instead. In their model, the cost of creating an extra copy (replica) of data is assumed to be negligible. Therefore, the socially optimal allocation is to allocate replicas to everyone, but the replica placement technique does not do this because it generates no revenue. Highlighting the fact that revenue is a major concern, Ref. [23] suggests examining auctions (or mechanisms) that do not necessarily maximize the social welfare, and characterizes all truthful mechanisms for this problem. The characterization of [23] is a special case of ours, for bounded (predefined number of replicas in the system) replica allocations, and it also appears implicitly in [20–22]. In [24], the authors also ignore the social welfare, instead they attempt to compute various functions of the agents valuations (such as the order statistics) using auctions of minimal communication complexity. Our current paper differs from the above in considering a game theoretical replica allocation mechanism that implements social welfare within which each players valuations accurately describes their preferences and needs.

All of the above referenced papers ([20–24, 26]) are in-

Table 1. Notations and their meanings.

symbols	meaning
m	number of servers in the ad hoc network
n	number of data items in the ad hoc network
O_k	k-th data item
o_k	size of O_k
M^i	i-th mobile server
s^i	storage capacity of M^i
r_k^i	read frequency for O_k from M^i
R_k^i	aggregation of r_k^i
u_k^i	update frequency for O_k from P_k
U_k^i	aggregation of u_k^i
NN_k^i	nearest neighbor of M^i holding O_k
$c(i,j)$	communication cost between M^i and M^j
P_k	primary mobile server of O_k
R_k	replication scheme of O_k
NTC	network transfer cost
D	total NTC
DRG	data replication game
$\mathcal{M}$	game theoretical mechanism
P^i	player i representing M^i in DRG
t^i	true data of P^i
b^i	bid reported by P^i to $\mathcal{M}$
ra^i	replica allocation for P^i

spired by the work reported in [2], where one-parameter players are considered for the problem of combinatorial auction. In our current work, truthfulness is achieved with respect to the expected utility and with high probability (i.e., the probability that an untruthful declaration improves the player utility is infinitesimally small). The game theoretical results of our current paper make a considerable leap in extending the results of [2] by showing that the same outcome holds when each player's valuation function is composed of multiple parameters. To the best of our knowledge, studies to replicate data items in ad hoc networks using game theoretical techniques have never been made.

3 System Model and Problem Description

3.1 Ad Hoc Data Replication Problem Formulation

Consider an ad hoc network comprising m mobile servers, with each mobile server having its own processing power and storage. Let $\underline{M^i}$ be the i–th mobile server and $\underline{s^i}$ the total storage capacity (in simple data units, e.g., blocks) of M^i where $1 \leq i \leq m$. The m mobile servers can communicate with each other using a wireless communication network. A wireless channel between two mobile servers M^i and M^j (if it exists) has a positive integer $\underline{c(i,j)}$ associated with it, giving the communication cost for transferring a data unit between mobile servers M^i and M^j. If the two mobile servers are not one-hop connected by a wireless channel, then the above cost is given as the sum of the costs of all the wireless channels (multi-hop) in a chosen path from server M^i to server M^j. Without loss of generality, we assume that $c(i,j) = c(j,i)$. Such an assumption can be relaxed when the upstream and downstream bandwidths vary for a wireless channel.

Let there be n data items, each identifiable by a unique name $\underline{O_k}$ and size in simple data units $\underline{o_k}$ where $1 \leq k \leq n$. Let $\underline{r_k^i}$ be the total number of reads initiated from M^i for O_k during a given period of time. Our replication policy assumes the existence of one primary copy for each data item in the system. Let $\underline{P_k}$ be the mobile server that holds the primary copy of O_k, i.e., the only copy in the network that cannot be de-allocated, henceforth referred to as the primary mobile server of the k-th object. Let $\underline{u_k}$ be the total number of updates initiated from P_k for O_k during a given period of time. Each primary mobile server P_k, contains information about the whole replication scheme $\underline{R_k}$ of O_k, i.e., R_k is the set of mobile servers that hold a replica of O_k.

When a mobile server M^i initiates a read request r_k^i for a data object O_k, the request is redirected to the nearest neighbor mobile server $\underline{NN_k^i}$ that holds either the original or a copy of the data item O_k. That is, NN_k^i is the mobile server where reads from M^i for O_k, if served there, would incur the minimum possible communication cost. We assume that only the mobile server P_k (that "owns" the data item) can perform such operations. P_k updates a data object O_k by sending broadcasts to the set of mobile servers that hold the replicas of O_k, i.e., to all $M^i \in R_k$.

For the ADRP under consideration, we are interested in minimizing the total <u>n</u>etwork <u>t</u>ransfer <u>c</u>ost (<u>NTC</u>) due to data movement, i.e., the data item movement due to the read and update accesses. There are two components that affect NTC. The first is due to the read requests. Let $\underline{R_k^i}$ denote the NTC due to the read requests by M^i for object O_k:

$$R_k^i = r_k^i o_k c(i, NN_k^i). \tag{1}$$

The second component is the cost that occurs due to the updates. Let $\underline{U_k}$ be the NTC due to the update requests from P_k for object O_k:

$$U_k = u_k o_k \sum_{\forall i \in R_k} c(i, P_k). \tag{2}$$

The total NTC, denoted as $\underline{D}$, due to reads and updates is given by

$$D = \sum_{k=1}^{n} \left[U_k + \sum_{i=1}^{m} R_i^k \right]. \tag{3}$$

D is meaningful only when the entire ad hoc network is connected. If we use the same cost model when considering disconnections, the cost represented by D would go to infinity because $c(i,j)=\infty$. To avoid such a phenomenon, we replace the case when $c(i,j)=\infty$ with a very large number $\underline{N}$. N can be considered as the cost of using certain connectivity related measures when an ad hoc network is disconnected such as reconfiguration of the working area, enlargement of the transmission radius, or temporary enhanced connection periods where an external entity is introduced in the system to reconnect partitioned networks [15].

Let $\underline{X_{ik}}=1$ if M^i holds a replica of object O_k, and 0 otherwise. The X_{ik}s define an $m\times n$ replication matrix, named $\underline{X}$, containing boolean elements. Using the above formulation, the ADRP can be stated as: "Find the assignment of 0/1 values in the X matrix that minimizes D; subject to the storage capacity constraint, $\sum_{k=1}^{n} X_{ik}o_k \leq s^i\ \forall(1\leq i\leq m)$; and subject to the primary copies policy, $X_{P_k k}=1\ \forall(1\leq k\leq n)$."

Application scenario: In many military operations, ad hoc networks are used as a communication medium to facilitate mobility. One key condition for a successful operation is to have a collaborative workspace with sufficient data exchanges so that collective decision making is possible. This may be problematic when accesses must be made through potentially expensive or slow connections. Data replication can alleviate access delays. However, when an ad hoc network is partitioned, access to the updated data may not be possible. To ensure reliability, an external entity (e.g, an unmanned aerial vehicle) can be used to reconnect the network. Using the ADRP formulation, our goal is to find a replica allocation technique that effectively address the issues of (a) mobility, (b) accessibility, and (c) reliability, while minimizing the NTC.

4 Game Theoretical Technique

4.1 Notations and Building Blocks

In DRG, the mechanism asks each player to report its valuation for a data item, it then allocates replicas to the players using some allocation algorithm and hands a payment to each player. The players know the mechanism and the payment scheme in advance. We assume that the players are self-interested, hence each player chooses a strategy that will maximize its profit (payment received by the mechanism minus cost incured to host the replica for entertaining traffic).

Let $\underline{P^i}$ be the i–th player representing the i–th mobile server. Each P^i has some private data $\underline{t^i}\in\Re$. In the literature, t^i is commonly known as player's *true data*, *true value*, or simply *type*. Each P^i reports to the mechanism a value $\underline{b^i}$, called bid. Let $\underline{t}$ and $\underline{b}$ represent the vector of true data and bids, respectively. It is sometimes convenient to write vectors such as t as $t=(t^i,t^{-i})$, where t^{-i} lists the types of the agents other than i, i.e., $t^{-i}=(t^1,\cdots,t^{i-1},t^{i+1},\cdots,t^m)$.

The mechanism has the capability to select an outcome $\underline{o}$ among $\underline{O}$ allowable outcomes using some algorithm that computes an outcome function $\underline{o(b)}\in O$. The mechanism optimizes a given objective function $\underline{f(o(b),t)}$, but it does not know t directly. Each P^i incurs some monetary cost, $\underline{cost^i(t^i,o(b))}$. To offset this cost, the mechanism makes a payment $\underline{pay^i(b)}$ to P^i. Each self-interested player *always* attempts to maximize its profit (hereafter referred to as utility), $\underline{U^i(t^i,b)}=pay^i(b)-cost^i(t^i,o(b))$. To keep things relatively simple, we will assume $cost^i(t^i,o(b))=t^i\cdot ra^i(b)$. Therefore, P^i's private data t^i measures its cost per replica allocation $\underline{ra^i(b)}$ in terms of the amount of traffic that it processes (read requests from servers that do not hold replicas and are the closest to P^i plus updates).

We say that truth-telling is a *dominating strategy* for P^i if bidding t^i always maximizes its profit, regardless of what other players bid, i.e., $U^i(t^i,(b^{-i},t^i))\geq U^i(t^i,(b^{-i},b^i)),\forall\, b^i\wedge b^{-i}$ [32]. Our goal is to make the system robust against incorrect dissemination of information by the players. This is accomplished by designing mechanisms such that truth-telling is a dominating strategy for every player.

Formally, a mechanism $\underline{\mathcal{M}}=(o(b),pay)$ consists of an outcome function $o(b)$ and $\underline{pay}$ is the payment scheme, i.e., the vector of payment functions $pay^i(b)$. We say that an outcome function admits a truthful payment scheme if there exist a pay such that $\mathcal{M}=(o(b),pay)$ is truthful. In reality not all outcome functions can admit a truthful payment scheme [29]. We have to design an outcome function that: (a) admits truthful payments, (b) achieves optimization for $f(o(b),t)$, and (c) converges in polynomial time.

Assume that $\mathcal{M}$ is truthful and each payment $pay^i(b^{-i},b^i)$ and replica assignment $ra^i(b^{-i},b^i)$ is twice differentiable with respect to b^i, for all values of b^{-i}. To derive a payment for P^i, we follow the same approach as described in [2]. Fixing other players' bids b^{-i}, we can consider the payment pay^i, replica assignment ra^i, and profit to be functions of just b^i. We start by deriving necessary conditions for truthfulness, that also turn out to be sufficient. Let us define $U^i(t^i,(b^{-i},t^i))=pay^i(t)-t^i\cdot ra^i(t)$. Truthfulness is equivalent to, for all $b,t\geq 0$,

$$U^i(t^i,(b^{-i},t^i))\geq pay^i(b)-t^i\cdot ra^i(b),$$

$$U^i(t^i,(b^{-i},t^i))\geq pay^i(b)-b^i\cdot ra^i(b)+b^i\cdot ra^i(b)-t^i\cdot ra^i(b),$$

$$U^i(t^i,(b^{-i},t^i))\geq U^i(t^i,(b^{-i},b^i))+(t^i-b^i)(-ra^i(b)). \tag{4}$$

Thus, $U^i(t^i,(b^{-i},t^i))$ is a convex function, and $-ra^i(t)$ is a subgradient at t^i. Standard results from analysis (as reported in [2]) show that $U^i(t^i,(b^{-i},t^i))$ is continuous on $[0, \infty)$, differentiable almost everywhere, and is equal to the integral of its derivative. Also, $U^i(t^i, (b^{-i}, t^i))' = -ra^i(t^i)$ wherever $U^i(t^i, (b^{-i}, t^i))$ is differentiable. Thus, we have $U^i(t^i, (b^{-i}, t^i)) = U^i(0) - \int_0^{t^i} ra^i(u)du$. Writing $U^i(t^i, (b^{-i}, t^i)) = pay^i(t) - t^i \cdot ra^i(t)$ and rearranging gives,

$$pay^i(t) = pay^i(0) + t^i \cdot ra^i(t) - \int_0^{t^i} ra^i(u)du. \quad (5)$$

Because $U^i(t^i, (b^{-i}, t^i))$ is convex, its derivative is increasing, which implies that $ra^i(t)$ is decreasing. Thus, for the mechanism to be truthful, it is necessary that $ra^i(t)$ is decreasing and $pay^i(t)$ is given by Equation 5 [3]. Decreasing allocation curves means nothing more than the notion that the allocation of a resource is inversely proportional to the reported true data. If a player bids higher than its true data, then it is allocated a smaller resource but its cost to host that resource increases. When it bids lower than its true data, then it is allocated a larger resource but its cost to host that resource increases considerably compared to the worth of the allocated resource. The above results hold when we know that players always report true data. For the case when the mechanism is skeptical about the reported true data, the following results are sufficient.

Theorem 4.1 ([2,10]) *The outcome function $o(b)$ admits a truthful payment scheme if and only if it is decreasing and the payments $pay^i(b^{-i}, b^i)$ are of the form $h^i(b^{-i}) + b^i \cdot ra^i(b^{-i}, b^i) - \int_0^{b^i} ra^{-i}(b^{-i}, u)du$, where h^i is an arbitrary function.* ■

Theorem 4.1 gives little flexibility in identifying a feasible payment scheme. Consider the profit of telling the truth if we set h^i to zero. The cost incured $(t^i \cdot ra^i(t))$ by P^i cancels out the second term and incurs a loss equal to the area under the allocation curve from zero to b^i. Since the players cannot even hope for a profit under this scheme, they presumably would not participate in such a mechanism. Hence, the following definition.

Definition 4.2 ([2,4]) *$\mathcal{M} = (o(b), pay)$ is a voluntary participation mechanism if players who bid truthfully never incur net loss, i.e., $U^i(t^i, (b^{-i}, t^i)) \geq 0$, for all P^is, t^is, and b^{-i}s.*

Theorem 4.3 ([2,10]) *A decreasing $o(b)$ admits a truthful payment scheme for a voluntary participation mechanism if and only if $\int_0^{\infty} ra^i(b^{-i}, u)du < \infty$ for all i and b^{-i}; hence, the payments become $pay^i(b^{-i}, b^i) = b^i \cdot ra^i(b^{-i}, b^i) + \int_{b^i}^{\infty} ra^i(b^{-i}, u)du$.* ■

The payment scheme described in Theorem 4.3 is equivalent to the celebrated Vickrey payment scheme [10], where the mechanism allocates a replica to the lowest bidder and pays it the amount of the second lowest bid. In [33] the authors have shown that the Vickrey payment scheme is robust against manipulative players. We will use the payment scheme of Theorem 4.3 for our GT technique that is outlined in the following section.

4.2 Game Theoretical Methodology for Ad Hoc Network (DRG)

Let $\mathcal{V}_k^i$ given as $(R_k^i - u_k o_k c(i, P_k))/o_k$ represent the valuation function of a self-interested player P^i for a data item O_k. $\mathcal{V}_k^i$ represents the expected benefit gained by P^i in terms of NTC when O_k is replicated at P^i. This benefit is computed by using the difference between the NTC incured from read requests, which would be eliminated if we made a replica, and the NTC arising due to updates to that replica. Because we want to consider the benefit per storage data unit, we divide the difference by the data item size. Negative values of $\mathcal{V}_k^i$ mean that replicating the k-th object is inefficient from the "local view" of P^i. This does not necessarily mean that we are unable to reduce the total NTC by creating such a replica, but that the local NTC observed by P^i will be increased.

When a replica is allocated at P^i, it experiences additional traffic at_k^i that is given as $\sum_j r_k^j o_k c(i,j) + u_k o_k c(i, P_k)$, $\forall j \notin R_k \wedge \min c(i,j)$, i.e., the cost incured due to read requests from servers that do not hold replicas and are the closest to P^i plus updates for O_k. We take the true data to be $t^i = 1/\mathcal{V}_k^i$, hence P^is' cost becomes

$$cost^i(t^i, o(b)) = t^i \cdot ra^i(b) = o_k \left(\frac{\sum_j r_k^j c(i,j) + u_k c(i, P_k)}{r_k^i c(i, NN_k^i) - u_k c(i, P_k)} \right), \forall j \notin R_k \wedge \min c(i,j). \quad (6)$$

The goal of a player is to maximize its profit defined as payment minus cost. The goal of the mechanism is to minimize the total data transfer cost due to read and update accesses, i.e., $\min \sum_i^m ra^i(b)$. Recall Theorem 4.1 states that an outcome function admits a truthful payment scheme if and only if it is decreasing. An outcome function would result in an allocation vector $(ra^1, ra^2, \cdots, ra^m)$. We say that this allocation vector is lexicographically smaller than $(\bar{ra}^1, \bar{ra}^2, \cdots, \bar{ra}^m)$ if for some i, $ra^i < \bar{ra}^i$ and $ra^k = \bar{ra}^k, \forall\ k < i$. Hence "our" outcome function $o(b)$ among all possible allocations should select an outcome in which the allocation of data items $(ra^1, ra^2, \cdots, ra^m)$ is lexicographically minimum. If such an outcome is selected, then P^i raising b^i would not cause the allocation to change unless P^i is the bottleneck [3]. In that case, raising b^i will only cause P^i to get a data item that is not beneficial for replication. Thus, the outcome function $o(b)$ is decreasing and by Theorem 4.1 it admits a truthful payment scheme. Also recall that in our analysis, we keep $ra^i(b^{-i}, \cdot)$ constant ("$\cdot$" could be replaced with either b^i or t^i), so the analysis can be aggregated over all P^is. This means that all of the P^is must participate in the mechanism, if they are to be compensated. Therefore, we have the following result.

Theorem 4.4 *For ADRP, we can always have a voluntary participation mechanism with decreasing outcome function that admits truthful payment scheme.* ■

Theorem 4.4 is a positive result; however, it does not say if we can have a polynomial-time algorithm to solve the ADRP. To obtain a polynomial-time algorithm, we observe that the ADRP is equivalent to the *Generalized Assignment Problem* [7].

Generalized Assignment Problem: *Instance*: A pair $(\mathcal{B}, \mathcal{S})$ where $\mathcal{B}$ is a set of m bins and $\mathcal{S}$ is a set of n items. Each bin $i \in \mathcal{B}$ has a capacity s^i, and for each item k and bin i, we are given a size o_k and a profit $p(i,k)$. *Objective*: Find a subset $U \subseteq \mathcal{S}$ that has a feasible packing in $\mathcal{B}$ and maximizes the profit of packing.

This equivalence implicitly gives us a 2-optimal upper bound on the optimality (for a formal proof see [34]). This bound is tight since data items cannot be further sub-divided into fractions. If fractional assignments were possible then $\frac{11}{9}$-optimal would have been our upper bound [7].

The allocation algorithm for DRG: Before the allocation algorithm is invoked, each mobile server broadcasts its identifier to all other mobile servers. After all mobile servers complete their broadcasts, every server knows its connected mobile servers, from the received identifiers. If the network is partitioned, then we make use of an external entity (e.g., a drone or a satellite) so that the partitioned network can be reconnected. Using such an entity will come at a very high cost, but acceptable because we implement a strict data consistency model for our system. For that purpose, when M^i and M^j are disconnected, we take the the communication cost $c(i,j)$ to be equal to N, where N is a very large number.

To present our algorithm, each P^i maintains a list $\underline{L^i}$ containing all of the data items that can be replicated. A data item O_k can be replicated at P^i, only if the remaining available storage capacity $\underline{as^i}$ of P^i is greater than its size, i.e., $as^i \geq o_k$ and the valuation for a data item is positive. We also keep a list $\underline{LP}$ containing all of the P^is that have the "opportunity" to replicate a data item. More succinctly, $P^i \in LP$ if and only if $L^i \neq \emptyset$. The DRG Algorithm performs in steps. At each step all P^is compute their bids b^i based on $\mathcal{V}_k^i$. Each of these bids reflects the highest benefit for the corresponding P^i. Each P^i sends its own bid to the mechanism $\mathcal{M}$, that selects from the list $\underline{MT}$ the P^i with the minimum bid. P^i is compensated with a payment and the data item is replicated. The lists LP and L^i together with the corresponding nearest neighbor value NN_k^i are updated correspondingly. The continuous nearest neighbor updates ensure decrease allocation curves. The selection of the lowest bid ensures that P^i with the highest possible benefit is allocated a replica. The second lowest price selection warrants that no P^i deviates from its true data. The DRG Algorithm is outlined as follows:

Algorithm 4.1: DATA REPLICATION GAME (DRG)

$$
\begin{array}{l}
\textbf{while } LP \neq \emptyset \\
\textbf{do} \begin{cases}
\textbf{for each } P^i \in LP \textbf{ in parallel} \\
\quad \textbf{do} \begin{cases}
\textbf{for each } O_k \in L^i \\
\quad \textbf{do} \text{ compute } \mathcal{V}_k^i; \\
b^i \leftarrow \frac{1}{\max(\mathcal{V}_k^i)}, \ \forall \mathcal{V}_k^i \geq 0; \\
\textbf{Send } b^i \text{ to } \mathcal{M}; \textbf{Receive } MT \leftarrow b^i \text{ at } \mathcal{M};
\end{cases} \\
\mathcal{K} = \text{argmin}_k(MT); \\
pay^i = at_{\mathcal{K}}^i \times \min(MT - (\min(MT))); \\
\textbf{Send } pay^i \textbf{ to } P^i; \\
\text{Replicate } O_{\mathcal{K}}; \\
as^i \leftarrow as^i - o_{\mathcal{K}}; \\
L^i \leftarrow L^i - \{O_{\mathcal{K}}\}; \\
\textbf{for each } P^i \in LP \textbf{ in parallel} \\
\quad \textbf{do} \text{ Update } NN_{\mathcal{K}}^i; \\
\textbf{if } L^i = \emptyset \\
\quad \textbf{then } LP \leftarrow LP - \{P^i\};
\end{cases}
\end{array}
$$

In the worst case where each server has enough capacity to hold all the data items and the number of updates is zero, there are $m \times n$ such iterations for the *while* loop. The time complexity for each iteration is governed by the two *for* loops executed in parallel. The first loop uses at most n iterations, while the second loop performs the update in constant time for each iteration. Hence, we conclude that the worst case running time of the DRG Algorithm is $\mathcal{O}(mn^2)$.

Theorem 4.5 *Given a instance of the DRG Algorithm: (a) the profit obtained by the players is no less than the optimal profit, (b) each data item O_k allocated to a player P^i satisfies $s^i \leq o_k$, (c) if s^i is violated, then there exists a single item that is allocated to P^i whose removal ensures feasibility, (d) a 2-optimal result is always guaranteed, and (e) a $(2+\epsilon)$-optimal is the worst case upper bound.*

Proof (a) Follows from Theorem 4.3.
(b) and (c) List L^i ensures that the constraint $s^i \leq o_k$ is satisfied and never violated.
(d) Consider the following greedy approach as reported in [34]. For a player P^i, let s^i be violated by the allocation of $\bar{O}_k$. If ra^i is obtained by accepting the allocation

of $\bar{O}_k$ and ra^i is at least half the accumulated ra^i by P^i, then we retain $\bar{O}_k$ and remove the rest of the data items of P^i. In the other case, we leave out $\bar{O}_k$. This results in a feasible solution of at least half the optimal allocation — a 2-optimal result. (The results of [7] warrant that this is a tight bound unless fractional assignments are possible.)
(e) Follows from (d) and the standard approximation algorithms analysis [37]. ■

5 Experimental Results and Discussion

Here, we present the results of our experiments carried out on a 440MHz Ultra 10 machine with 512MB memory. DRG was implemented using Ada and Ada GNAT's distributed systems annex GLADE [30].

Four different types of experiments were conducted. The purpose of the first experiment is to study the balance between how frequently we want to replicate data as opposed to the savings in network traffic. In the next three, we study the effects of increases in read accesses, increases in update accesses, and the storage capacity of mobile servers. The solution quality in all cases is measured according to the NTC percentage that is saved under the replication scheme found by the algorithms, compared to the initial one, i.e., when only primary copies exist.

5.1 Workload

We generate the structure of the ad hoc network in the following manner. We define an enclosed polygonal region in which the mobile servers exist. We choose this region to be a square grid of size 1000×1000 with distance between two adjacent points in the grid equal to 1. We define the total number of mobile servers and data items in the network. For simplicity, we assume that each P_k holds only one data item as the original. This of course is not always the case, as P_k can hold multiple original data items or none. Modeling such scenarios also can be done using our framework.

Each mobile server randomly moves horizontally or vertically and the movement speed is randomly determined from a uniform distribution between 0 and a maximum value $\underline{d}$. The random movement of mobile devices will make the ad hoc network disconnections more frequent than the case when one could anticipate the movement of mobile devices. If two mobile servers M^i and M^j are within their radio communication range $\underline{Rd}$, then there exists a bi-directional link between them. The communication cost of this link $c(i,j)$ is equal to the distance between M^i and M^j plus a processing overhead constant whose values are taken from a uniform distribution between 1 and 10. When two mobile servers M^i and M^j are not one-hop connected, then the communication cost is given as the sum of the communication costs (distance plus the processing overhead constant) of all the links in a chosen path from M^i to M^j. (This is a very common link cost model used for simulating large-scale networks. For an elaborate discussion see [11].) For the case when M^i and M^j are disconnected, we take $c(i,j)$ to be equal to $N = 1,000,000$.

For a single simulation time unit the number of reads for all (server, data item) pairs is either 0 or 1. The number of updates generated by each P_k is determined in a similar manner.

The initial size of each data item is generated using a uniform distribution between 10 and 60. Changes in the size of data items due to updates is simulated by adding to the current size a value from a random distribution between -5 and 5. For the case when the current size of a data item is less than 6, the added size is selected using a random distribution between 1 and 5.

In all of the experiments, the capacity of mobile servers is proportional to the total size of data items ($\underline{TS}$). The capacity of a mobile server is generated using a uniform distribution from $(\frac{1}{2}TS)C$ and $(\frac{3}{2}TS)C$, where $0 \leq \underline{C} \leq 1$ is a parameter that reflects the storage capacities of the mobile servers. For example, when $TS = 100$ and $C = 0.30$ the capacities of the mobile servers are uniformly distributed between $(\frac{1}{2} \times 100 \times 0.30 =)15$ and $(\frac{3}{2} \times 100 \times 0.30 =)45$.

For all methods, the replicas are periodically allocated after a specific period of time, called the relocation period $\underline{T}$ [12]. In all experiments, we examine the average percentage NTC savings $((\frac{D_{no\ replicas} - D_{replicas}}{D_{no\ replicas}}) \times 100$, where $\underline{D_{no\ replicas}}$ is the NTC of the ad hoc network when there are only primary copies of the data ietms and $\underline{D_{replicas}}$ is the NTC when the ad hoc network has replicas placed by an algorithm) of each of the four methods during 10,000 units of simulation time.

Table 2 shows the parameters and their values used in the simulation experiments. We want to clarify that the relative effect of changing C is similar to changing the number of mobile servers and data items. Therefore, we keep both m and n constant during our experimental studies. Moreover, authors in [15] have shown that the change in the movement speed d has a similar impact on the system as the change in the relocation period T. Therefore, d is also not altered for the simulation study.

5.2 Comparative Techniques

For comparisons, we selected three state-of-the-art ad hoc replica placement techniques. To provide a fair evaluation, the assumptions and system parameters were kept the same in all the approaches. Due to space limitations, we will only give a brief overview of the comparative techniques. Details for a specific technique can be obtained from the referenced papers.

Extended Access Static Frequency method (ESAF) [13]**:**

Table 2. Parameter variance intervals for simulations.

parameter	value [range]
m	200
n	200
d	1
Rd	7
o_k	uniform distribution [10, 60]
TS	$\sum_{k=1}^{n} o_k$
C	[0.10, 0.40]
s^i	uniform distribution $[(\frac{1}{2}TS)C,(\frac{3}{2}TS)C]$
T	256 [1, 8192]
N	1,000,000

Each mobile server is allocated replicas based on the descending order of the prefetching technique $\underline{PT}$ values. PT is given as $p_k^i \cdot \tau_k = p_k^i \cdot (T_k - t_k)$, where p_k^i denotes the probability that an access request for data item O_k from mobile server M^i is issued at a unit of time, τ_k denotes the time remaining until O_k is updated next, T_k denotes the update period of O_k, and t_k denotes the time that has passed since O_k has been updated at the most recent update period. The PT value represents the average number of access requests that are issued for O_k, until O_k is updated next, and it takes the maximum value $p_k^i \times T_k$ when O_k is updated.

Extended Dynamic Access Frequency and Neighborhood method (EDAFN) [14]**:** Each mobile server determines the preliminary allocation of replicas based on the ESAF method. In each set of mobile servers that are connected to each other, starting from the mobile server with the lowest suffix (i), the following procedure is repeated in the order of the breadth first search. When there is duplication of a data item between two neighboring mobile servers, and if one of them is the original, then the server which holds the replica replaces it with a different replica. If both of them are replicas, then the server whose PT value for the data item is lower replaces the replica with a different replica. The replacement replica is chosen from among data items whose replicas are not allocated at either of the two servers and whose PT value is the highest.

Extended Dynamic Connectivity based Grouping method (EDCG) [15]**:** In each set of mobile servers that are connected to each other, an algorithm to find bi-connected components is executed. Then, each bi-connected component is put in a group. In each group, a group access frequency to each data item is calculated as a summation of access frequencies of mobile servers in the group. Then, the PT value of the group for each data item is calculated. The rest of the procedure is exactly the same as EDAFN.

We select the above mentioned techniques for our comparisons because the results reported in the referenced papers show that the three methods work very well in environments when: (a) updates follow no fixed pattern, (b) reads are more predominate than updates, and (c) the relocation period is comparable to that assumed in this study. The performance metrics used by the three methods were (a) the maximization of data accessibility and (b) the minimization of traffic in the system. Comparisons among DRG, ESAF, EDAFN, and EDCG are appropriate because data accessibility can always be guaranteed in our problem formulation and reduction of traffic is equivalent to the NTC.

5.3 Results and Discussion

We assess the effects of the relocation period on each of the four methods. Figure 1 summarizes the results. The horizontal axis indicates the relocation period T and the vertical axis indicates the NTC savings. From the plot, we can see that DRG gives the highest NTC savings followed by EDCG. DRG and EDCG give a significant improvement over ESAF because they selectively replicate data items. In ESAF, replicas are allocated based on local benefits; thus ESAF is equivalent to the Least Frequently Used (LFU) cache replacement technique. We observe that the savings in NTC obtained by the methods decreases as the relocation period increases. This is because a shorter relocation period can detect the changes of network topology. However, the difference in the NTC savings by having a smaller relocation period as opposed to larger relocation period is not that large. For the rest of the simulations, we choose $T = 256$, as done in [13–15].

To illustrate the main merits of DRG, we consider the case when either reads, updates, or storage capacities increase. Let $\underline{Ch}$ denote the percentage of increase in either reads or updates for a data item. Let $\underline{OCh}$ represent the percentage of data items in the network with changing read or update patterns. Let $\underline{R}$ and $\underline{U}$ represent the percentage of changes being performed toward a read or update increase, respectively. For example, the network with parameters $Ch = 600\%, OCh = 30\%, R = 80\%$, and $U = 20\%$ would mean that among the 200 total objects $(200 \times \frac{30}{100} \times \frac{80}{100} =)48$ experience an increase by 600% in their reads, while $(200 \times \frac{30}{100} \times \frac{20}{100} =)12$ a same increase in their updates.

Increase in the number of reads in the system means that there is a need to replicate as many data items as possible. However, an increase in the number of updates means that the number of replica placements have to be controlled in the system and the replicas have to be placed closer to the primary mobile server. Because changes in both the read and update parameters have a complementary effect on the

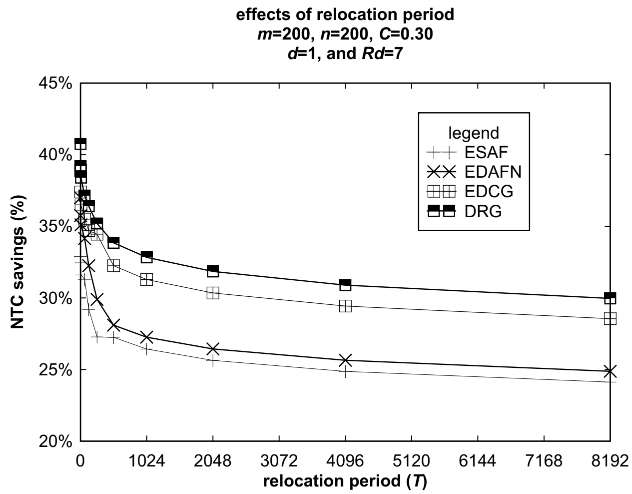

Figure 1. Measuring the effects of the relocation period.

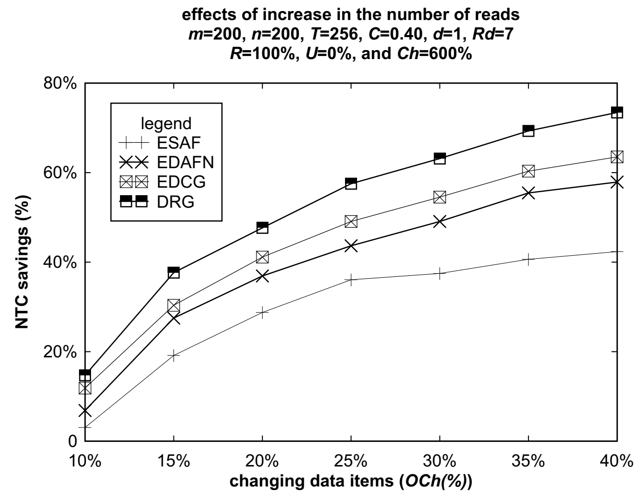

Figure 2. Measuring the effects of the increase in the number of reads.

system, we describe them together. Figures 2 and 3 summarize the results for the change in the read and update frequencies, respectively. In both figures, the horizontal axis indicates the change in the number of objects that had their reads or updates increased. The vertical axis indicates the NTC savings.

From the plots, a clear classification can be made between the algorithms. DRG incorporates the effect of increase in the number of reads by replicating more intensively in the beginning to exploit the available storage capacity to the maximum. After a certain point, further replication is constrained due to the storage limitations, thus, savings tend to increase less rapidly. EDAFN and EDCG also a give competitive increase in NTC savings, while ESAF does not seem to that receptive to the changes. To understand why there is such a gap in the performance between ESAF and the rest of the algorithms, recall that the degree of data duplication is extremely high in ESAF. Hence, ESAF replicates data items that are actually not beneficial from the system point of view.

An increase in the frequency of updates results in data items having a decreased local significance unless the mobile server under consideration is in close vicinity to the primary mobile server. This forces ESAF, EDAFN, and EDCG that rely on the PT values to discard data items that have a shorter update time. In contrast, DRG with its *decreasing allocation mechanism* is able to select better replication schemes and maintains a good bound on the number of beneficial replicas.

An increase in the storage capacity means that a large number of data items can be replicated. Replicating a data item that is already extensively replicated, is unlikely to result in significant traffic savings as only a small portion of the mobile servers will be affected overall. Moreover, since data items are not equally read intensive, increase in the storage capacity would have a great impact at the beginning (initial increase in capacity), but has little effect after a certain point, where the most beneficial ones are already replicated. This phenomenon is observable in Figure 4. All techniques showed an immediate initial increase (the point after which further replication is inefficient) in its NTC savings, but afterward showed a near constant performance. EDAFN, although did not perform that well compared to DRG, observably gained the most NTC savings of 50% followed by ESAF with 43%.

Finally, Table 3 shows the average execution times of the algorithms. It is observable that ESAF terminated faster than all of the other techniques, followed by EDAFN, DRG, and EDCG.

Summarizing the above, DRG achieves more traffic savings (in many cases even 80%), than the rest of the techniques and responds better to changes in the network toplogy, the reads and update changes, and the servers' storage capacities. The ESAF method apart from achieving mediocre solution quality, runs in about 2 orders of magnitude less time than the DRG method.

6 Concluding Remarks

In this paper, we addressed the data replication problem in ad hoc networks by developing a cost model that minimizes the total data item transfers due to reads and updates.

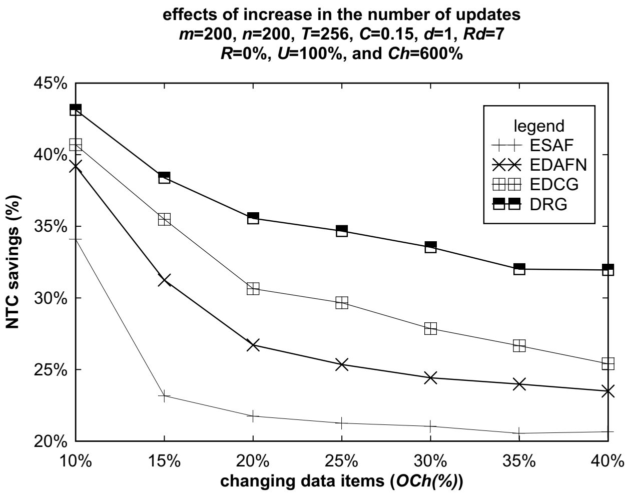

Figure 3. Measuring the effects of the increase in the number of updates.

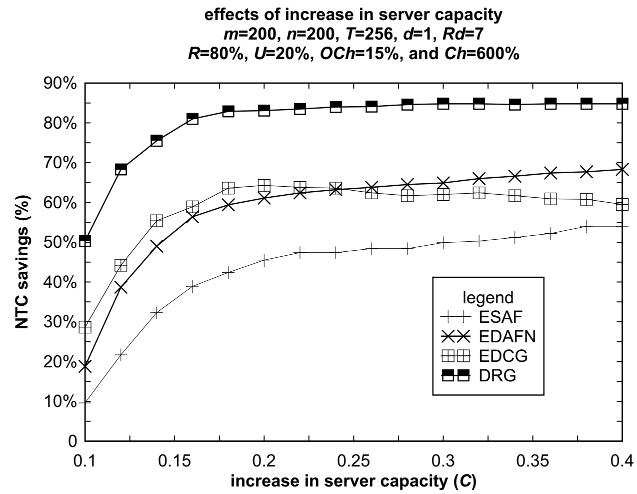

Figure 4. Measuring the effects of the increase in the storage capacities.

Table 3. Average execution times, in seconds, of a relocation period for the studied techniques

technique	average execution time (sec.)
DRG	18.63
ESAF	6.98
EDAFN	18.46
EDCG	23.41

We developed a replica allocation mechanism based on a game theoretic approach (DRG) to solve the problem in the presence of self-interested mobile servers. Such techniques are meaningful to develop when servers have the capability to manipulate the mechanism for their own benefit by misrepresenting their preferences, which may result in severe performance degradation. We demonstrated the effectiveness of our proposed technique by deriving a payment scheme that suppresses the selfish behavior of the participating servers. Moreover, we guarantee that all mobile servers are better off participating in the mechanism as opposed to operating in isolation in a greedy local manner. This keeps the essence of the data replication problem in tact, which is the minimization of the total data item transfer cost.

We compared DRG with three state-of-the-art ad hoc replica placement techniques: (a) extended static access frequency, (b) extended dynamic access frequency, and (c) neighborhood and extended dynamic connectivity group. The experimental results revealed that DRG outperformed the techniques in solution quality and had competitive execution times.

Throughout this paper, we assumed that when the ad hoc network is disconnected, we have the flexibility to make use of an external entity (if needed) to reconnect the network. In future work, we will relax this assumption and incorporate the phenomenon of unstable radio links, i.e., to study an environment where obstacles to radio waves may exist, e.g., buildings and mountains.

Associated with the mobile ad hoc networks is the key issue of energy consumption. Data replication implicitly conserves energy because requests traverse fewer communication links to access the closest replicated data. It is our plan to extend this work to study the effect of replica schemes on energy consumption. We also would improve our simulation model by incorporating various mobility models and the usage of real-world access traffic patterns.

Acknowledgments: The authors would like to thank Luis Diego Briceño and Jay Smith for their valuable comments.

References

[1] I. Ahmad, K. Karlapalem, Y.-K. Kwok, and S.-K. So. Evolutionary algorithms for allocating data in distributed database systems. *Distributed and Parallel Databases*, 11(1):5–32, January 2000.

[2] A. Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *42^{nd} Annual Symposium on Foundations of Computer Science (FOCS '01)*, pages 482–491, 2001.

[3] A. Archer and Éva Tardos. Frugal path mechanisms. In *13^{th} Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '02)*, pages 991–999, 2002.

[4] A. Archer, C. Papadimitriou, K. Talwar, and Éva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *14^{th} Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pages 205–214, 2003.

[5] K. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58(4):328–346, April 1950.

[6] W. W. Chu. Optimal file allocation in a multiple computer system. *IEEE Transactions on Computers*, 18(10):885–889, October 1969.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[8] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, September 1972.

[9] V. Grassi. Prefetching policies for energy saving and latency reduction in a wireless broadcast data delivery system. In *3^{rd} ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '00)*, pages 77–84, 2000.

[10] D. Grosu and A. T. Chronopoulos. Algorithmic mechanism design for load balancing in distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(1):77–84, January 2004.

[11] J. S. Gwertzman and M. Seltzer. The case for geographical push-caching. In *5^{th} International Workshop on Hot Topics in Operating Systems (HotOS '95)*, pages 51–55, 1995.

[12] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *20^{th} IEEE Conference on Computer Communications (INFOCOM '01)*, pages 1568–1576, 2001.

[13] T. Hara. Replica allocation methods in ad hoc networks with data update. *Mobile Networks and Applications*, 8(4):343–354, August 2003.

[14] T. Hara and S. K. Madria. Dynamic data replication using aperiodic updates in mobile ad hoc networks. In *9^{th} International Conference Database Systems for Advances Applications (DASFAA '04)*, pages 869–881, 2004.

[15] T. Hara and S. K. Madria. Data replication for improving data accessibility in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(11):1515–1532, November 2006.

[16] W. Hildenbrand. *Advances in Economic Theory*. Oxford University Press, Liverpool, UK, 1982.

[17] Y. Huang, A. P. Sistla, and O. Wolfson. Data replication for mobile computers. In *ACM International Conference on Management of Data (SIGMOD '94)*, pages 13–24, 1994.

[18] J. Jing, A. K. Elmagarmid, A. Helal, and R. Alonso. Bit-sequences: An adaptive cache invalidation method in mobile client/server environments. *Mobile Networks and Applications*, 2(2):115–127, April 1997.

[19] D. Johnson. Routing in ad hoc networks of mobile hosts. In *Workshop on Mobile Computing Systems and Applications (HotMobile '94)*, pages 158–163, 1994.

[20] S. U. Khan and I. Ahmad. A powerful direct mechanism for optimal www content replication. In *19^{th} International Parallel and Distributed Processing Symposium (IPDPS '05)*, 2005.

[21] S. U. Khan and I. Ahmad. A pure nash equilibrium guaranteeing game theoretical replica allocation method for reducing web access time. In *12^{th} International Conference on Parallel and Distributed Systems (ICPADS '06)*, pages 169–176, 2006.

[22] S. U. Khan and I. Ahmad. Replicating data objects in large-scale distributed computing systems using extended vickrey auction. *International Journal of Computational Intelligence*, 3(1):14–22, January 2006.

[23] S. U. Khan and I. Ahmad. Discriminatory algorithmic mechanism design based www content replication. *Informatica*, 31(1):105–119, January 2007.

[24] S. U. Khan and I. Ahmad. A semi-distributed axiomatic game theoretical mechanism for replicating data objects in large distributed computing systems. In *21^{st} International Parallel and Distributed Processing Symposium (IPDPS '07)*, 2007.

[25] Y. Kwok, K. Karlapalem, I. Ahmad, and N. M. Pun. Design and evaluation of data allocation algorithms for distributed multimedia database systems. *IEEE Journal on Selected Areas in Communications*, 14(7):1332–1348, July 1996.

[26] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis. Distributed selfish replication. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1401–1413, December 2006.

[27] T. Loukopoulos and I. Ahmad. Optimizing download time of embedded multimedia objects for web browsing. *IEEE Transactions on Parallel and Distributed Systems*, 15(10):934–945, October 2004.

[28] A. Mas-Collel, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, Liverpool, UK, 1995.

[29] N. Nisan and A. Ronen. Algorithmic mechanism design. In *31^{st} Annual ACM Symposium on Theory of Computing (STOC '99)*, pages 129–140, 1999.

[30] L. Pautet and S. Tardieu. Glade: A framework for building large object-oriented real-time distributed systems. In *3^{rd} International Symposium on Object-Oriented Real-Time Distributed Systems (ISORC '00)*, pages 244–251, 2000.

[31] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[32] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, February 2004.

[33] S. Sanghvi and D. C. Parkes. Hard to manipulate combinatorial auctions. Technical report, Division of Engineering and Applied Sciences, Harvard University, 2004.

[34] D. B. Shmoys and É. Tardos. An approximate algorithm for the generalized assignment problem. *Mathematical Programming, Part A*, 62:461–474, February 1993.

[35] M. D. Theys, N. Beck, H. J. Siegel, and M. Jurczyk. An analysis of procedures and objective functions for heuristics to perform data staging in distributed systems. *Journal of Interconnection Networks*, 7(2):257–293, June 2006.

[36] M. D. Theys, H. J. Siegel, and E. K. P. Chong. Heuristics for scheduling data requests using collective communications in a distributed communication network. *Journal of Parallel Distributed Computing*, 61(9):1337–1366, September 2001.

[37] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Germany, 2001.