International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:7, 2009

A Competitive Replica Placement Methodology for Ad Hoc Networks

Samee Ullah Khan and Cemal Ardil

***Abstract*—**In this paper, a mathematical model for data object replication in ad hoc networks is formulated. The derived model is general, flexible and adaptable to cater for various applications in ad hoc networks. We propose a game theoretical technique in which players (mobile hosts) continuously compete in a non-cooperative environment to improve data accessibility by replicating data objects. The technique incorporates the access frequency from mobile hosts to each data object, the status of the network connectivity, and communication costs. The proposed technique is extensively evaluated against four well-known ad hoc network replica allocation methods. The experimental results reveal that the proposed approach outperforms the four techniques in both the execution time and solution quality.

***Keywords*—**Data replication, auctions, static allocation.

I. INTRODUCTION

THERE is continuing interest to study and improve technologies related to ad hoc networks, as they are constructed from only mobile hosts and require no particular infrastructure [1]. In such a setup, every mobile host acts as a router and communicates with other mobile hosts. As a consequence even if the source and destination mobile hosts are not in each other's communication range, data is still forwarded to the destination mobile host by relaying transmission through other mobile hosts which exist between the two mobile hosts.

Due to the continuous mobility of hosts, an ad hoc network suffers from frequent disconnections. This phenomenon is undesirable when mobile hosts are accessing data from each other. Since one cannot control network disconnections, an alternative solution to this problem is to replicate data onto mobile hosts so that when disconnections occur, mobile hosts can still access data [7].

The decision where to place the replicated data must trade off the cost of accessing the data, which is reduced by additional copies, against the cost of storing and updating the replicas. These costs have severe implications in ad hoc networks since mobile hosts have poor resources (storage and processing power). In general the mobile hosts would experience reduced access latencies provided that data is replicated within their close proximity. However, this is applicable in cases when only read accesses are considered. If updates of the contents are also under focus, then the locations of the replicas have to be 1) in close proximity to the mobile hosts, and 2) in close proximity to the primary (assuming a "master" replication environment [8]) copy. Therefore, efficient and effective replication schemas strongly depend on how many replicas to be placed in the system, and more importantly where [10].

Ad hoc data replication problem (ADRP) was first introduced by Hara [5], which was further extended ([6], [7]) to incorporate various network connectivity related issues. Although the above mentioned works are plausible in the sense that they advance the study of ADRP, yet none involve reasoning via a concrete mathematical model. Thus, it is imperative to derive and understand an optimization model that is general, flexible and modifiable to cater for various applications of ADRP. In this paper, we will first focus on deriving a mathematical model for ADRP and show that this problem in general is NP-complete. We then follow it up by proposing "Mosaic-Net" (acronym of game theoretical method for selection and allocation of replicas in ad hoc networks), which exhibits fast execution time and guarantees excellent solution quality.

The reminder of this paper is organized as follows. In Section II, we give a brief description of the related works and discuss how our approach is different from others. Section II focuses on deriving a mathematical model for ADRP. Section IV depicts the Mosaic-Net technique for ADRP, followed by experimental evaluations in Section V. Finally, in Section VI, we summarize this paper.

II. RELATED WORK

Several strategies for caching data contents in mobile computing environments have also been proposed [1], [2], [8], [9], [11], [13]. Most of these strategies assume an environment where mobile hosts access contents at sites in a fixed network, and cache data on the mobile hosts because wireless communication is more costly than wired communication. Such strategies address the issue of keeping consistency between original data and its replicas or caches with low communication costs. They are considered to be similar to our approach, because both approaches replicate data on mobile hosts. However, these strategies assume only one-hop wireless communication, and thus, they are

S. U. Khan is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58102, USA (phone: 701-231-7615; fax: 701-231-8677; e-mail: samee.khan@ ndsu.edu).

C. Ardil is with the National Academy of Aviation, Baku, Azerbaijan, (e-mail: cemalardil@gmail.com).

completely different from our approach that assumes multi-hop ad hoc communication.

Another closely related research topic is of push-based information systems in which a server repeatedly broadcasts data to clients using a broadband channel. Several caching strategies have been proposed to improve user access [4], [12]. In these strategies, clients are typically mobile hosts, and the cache replacement is determined based on several parameters such as the access frequency from each mobile host 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, because both approaches replicate data on mobile hosts. However, comparing the strategies for caching or replicating, both approaches are completely different because the strategies in push-based information systems do not assume that the clients cooperatively share cached data in ad hoc networks.

Hara's work ([5], [6], [7]) is the closest among all the related works on ADRP compared to this paper. However, our work differs from Hara's in: 1) deriving a mathematical problem formulation for ADRP, 2) proving that the generalized form of ADRP is NP-complete, 3) 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 host" policies, 4) extensively evaluating the techniques under varying system parameters, 5) using game theoretical techniques.

III. Problem Formulation

Consider an ad hoc network comprising m mobile hosts, with each mobile host having its own processing power and storage. Let M^i and s^i be the name and the total storage capacity (in simple data units e.g. blocks), respectively, of mobile host i where $1 \le i \le m$. The m mobile hosts can communicate with each other using a wireless communication network. A wireless channel between two mobile hosts M^i and M^j (if it exists) has a positive integer $c(i,j)$ associated with it, giving the communication cost for transferring a data unit between mobile hosts M^i and M^j. If the two mobile hosts 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 site M^i to the site M^j. Without the 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 objects, each identifiable by a unique name O_k and size in simple data unites o_k where $1 \le k \le n$. Let r_k^i and u_k^i be the total number of reads and updates, respectively, initiated from M^i for O_k during a certain time period. In other words, r_k^i and u_k^i are the read and update frequencies, respectively.

Our replication policy assumes the existence of one primary copy for each object in the system. Let P_k, be the mobile host which holds the primary copy of O_k, i.e., the only copy in the network that cannot be de-allocated, hence referred to as primary mobile host of the k-th object. Each primary mobile host P_k, contains information about the whole replication scheme R_k of O_k. This can be done by maintaining a list of the mobile hosts where the k-th object is replicated at, called from now on the *replicators* of O_k.

When a mobile host M^i initiates a read request r_k^i for a data object O_k, the request is redirected to the nearest neighbor NN_k^i mobile host that holds either the original or the copy of the data object O_k. In other words, NN_k^i is the mobile host for which the reads from M^i for O_k, if served there, would incur the minimum possible communication cost. It is possible that $NN_k^i = M^i$, if M^i is a *replicator* or the primary mobile host of O_k. Another possibility is that $NN_k^i = P_k$, if the primary mobile host is the closest one holding a replica of O_k.

For the updates we assume that only the mobile host P_k (that owns the data object(s)) can perform such operations. Such an assumption is justifiable when considering sensor networks, which is a special class of ad hoc networks. P_k updates a data object O_k by sending broadcasts to the set of mobile hosts that hold the replicas of O_k, i.e., $M^i \in R_k$.

For the ADRP under consideration, we are interested in minimizing the total network transfer cost (NTC) due to data object movement, *i.e.* the data object movement due to the read and update accesses. It is to be noted that the minimization of NTC in turn leads to increased data accessibility. There are two components affecting NTC. The first component of NTC is due to the read requests. Let R_k^i denote the total NTC, due to M^is' reading requests for object O_k, addressed to the nearest site NN_k^i. This cost is given by the following equation:

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

where NN_k^i = {*Mobile host* j | $j \in R_k$ and min $c(i,j)$}. The second component of NTC is the cost arising due to the updates. Let U_k^i be the total NTC, due to P_ks' updates requests for object O_k. This cost is given by the following equation:

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

The cumulative NTC, denoted as $C_{overall}$, due to reads and writes is given by:

$$C_{overall} = \sum_{i=1}^{m} \sum_{k=1}^{n} (R_k^i + U_k^i). \tag{3}$$

Let X_{ik}=1 if M^i holds a replica of object O_k, and 0 otherwise. X_{ik}s define an $m \times n$ replication matrix, named X, with boolean elements. Mobile hosts which are not the *replicators* of data object O_k create NTC equal to the communication cost of their reads from the nearest *replicator*. Sites belonging to the replication scheme of O_k, are associated with the cost of receiving all the updated versions of it. Using the above formulation, the ADRP can be defined as:

"Find the assignment of 0, 1 values in the X matrix that minimizes $C_{overall}$, subject to the storage capacity constraint:

$$\sum_{k=1}^{n} X_{ik} o_k \le s^i \quad \forall (1 \le i \le m),$$

and subject to the primary copies policy:

$$X_{P_k k} = 1 \quad \forall (1 \le k \le n)$$."

IV. Proposed Technique

In this section we will first describe the basics of our proposed game theoretical model followed by extending the model to ADRP.

The Setup: The ad hoc network described in Section 3 is considered, where each mobile host is represented by a player, i.e., the mechanism contains *m* players. In the context of the ADRP, a player holds two key elements of information: a) the available mobile host storage, as^i, and b) the read access frequency, r_k^i, for a data object *k*

Intuitively, if players know the available storage of each other, that gives them no advantage whatsoever. However, if they come about to know the read frequency, then they can easily modify their bids and alter the algorithmic output, as the read frequency directly represent the popularity of data objects [10]. It is to be noted that a player *i* can only calculate a data object *k*'s benefit (that it brings by reducing the cost of read accesses due to replication) by making use of the access frequency, and thus, everything else such as the network topology, latency on communication lines, and even the mobile host capacities can be public knowledge. Therefore, DRP[π] is the only natural choice.

Communications: The players in the mechanism are assumed to be greedy (or selfish) and therefore, they project a bid b^i (as apposed to t^i) to the mechanism. At this moment we want to clarify that the mechanism has no way of knowing that b^i is t^i or some other value. However, in the subsequent text we will rigorously prove that if the mechanism provides proper incentive (payments to compensate for holding replicas of data objects) to the players, then they are in fact forced to bid $b^i = t^i$.

Components: The mechanism has two components: 1) the algorithmic output $x(\cdot)$, and 2) the payment mapping function $p(\cdot)$.

Algorithmic output: In the context of the ADRP, the algorithm accepts bids from all the players, and outputs the maximum beneficial bid, i.e., the bid that incurs the minimum NTC due to object movement (Equation 3). (We will give a detailed description in the subsequent text.)

Monetary cost: When a data object is allocated (for replication) to a player *i*, the player becomes responsible to entertain (read and write) requests to that object. For example, assume object *k* is replicated to player *i*, then the amount of traffic that the player has to entertain due to the replication of object *k* is exactly equivalent to the NTC cost, i.e., $c^i = R_k^i + U_k^i$. This fact is easily deducible from Equation 3.

Payments: To offset c^i, the mechanism makes a payment $p^i(b)$ to player *i*. This payment is equivalent to the second highest bid that is submitted to the mechanism for a data object *k*. The readers would immediately note that in such a payment function, a player *i* can never obtain a net profit greater than 0. This is exactly what we want. In a selfish (or greedy) environment, it is possible that the players bid higher than the true value. The mechanism creates an illusion to negate that, by compensating the players with the payment that is lower than their incurred cost. This leaves no room for the players to overbid or underbid. For example, 1) If a player *i* overbids for an object *k*, it gives *i* no advantage since it receives payment equivalent to the second highest bid; 2) If a player *i* underbids for an object *k*, it lessens its chances to (win and) replicate *k*. In literature, such a payment function is termed as Vickrey payment [14]. For more details on the optimality of such type of a payment functions see [12]. In that paper, the authors have identified many such scenarios, but all fail to exploit this (Vickrey) payment option.

We want to emphasis that each player's incentive is to replicate data objects so that queries can be answered locally. If the replicas are made available elsewhere, the cost to access data would be much higher.

Bids: Each player reports a bid that is the direct representation of the true data that it holds, i.e., $b^i = r_k^i o_k c(i, NN_k^i) = R_k^i$. Readers will immediately notice that b^i only represents one-half of the NTC. This is because U_k^i is the private information for P_k.

In essence, the mechanism $\varsigma(x(b), p(b))$, takes in the vector of bids *b* from all the players, adjusts the bids by injecting the cost of updates and then selects the highest bid. The highest bidder is allocated the data object *k* which is added to its allocation set x^i. The mechanism then pays the bidder p^i. This payment is equivalent to the second highest bid.

At a relocation period, a mobile host might not connect to another mobile host which has an original or a valid replica of a data item that the host should allocate. In this case, the memory space for the replica is temporarily filled with one of the replicas that have been allocated since the previous relocation period but are not currently selected for allocation. This temporarily allocated replica is chosen from among the possible replicas according to Mosaic-Net. If there is no replica that can be temporarily allocated, the memory space remains free. When data access to the data item whose replica should be allocated succeeds, the memory space is filled with the valid replica as depicted in Fig. 1.

Description of Algorithm: At a relocation period, each mobile host broadcasts its host identifier. After all mobile hosts complete their broadcasts, every host knows its connected mobile hosts. In every connected ad hoc network, the Mosaic-Net is invoked. We maintain a list L^i at each

```
A Competitive Technique for Replica Placement in Ad hoc Networks (Mosaic-Net)

Initialize:
01 LM, L^i, T_k^i, ς, MT

02 WHILE LS ≠ NULL DO
03    OMAX = NULL; MT = NULL; P^i = NULL;
04         PARFOR each M^i∈ LS DO
05                  FOR each O_k∈ L^i DO
06                        B_k^i = compute (t^i); /*compute the valuation corresponding to the desired object*/
07                  ENDFOR
08              b^i = argmax_k(B_k^i);
09              SEND t^i to ς; RECEIVE at ς b^i and immediately adjust b^i to v^i and save in MT;
10          ENDPARFOR
11   OMAX = argmax_k(MT);    /*Choose the global dominate valuation*/
12   p^i = argmax_k(delete(argmax_k(MT))) ;                /*Calculate the payment*/
13   BROADCAST OMAX;
14   SEND P^i to M^i;        /*Send payments to the agent who is allocate the object OMAX*/
15   Replicate O_OMAX;
16   as^i=as^i - o_k;                        /*Update capacity*/
17   L^i = L^i - O_k;                   /*Update the list*/
18   IF L^i = NULL THEN SEND info to ς to update LM = LM - M^i;     /*Update mechanism players*/
19         PARFOR each M^i∈ LM DO
20               Update NN^i_OMAX           /*Update the nearest neighbor list*/
21         ENDPARFOR                 /*Get ready for the next round*/
22 ENDWHILE
```

Fig. 1 Pseudo-code for Mosaic-Net

mobile host. This list contains all the data objects that can be replicated by player i onto the mobile host M^i. We can obtain this list by examining the two constraints of the ADRP. List L^i would contain all the data objects that have their size less then the total available storage as^i. Moreover, if the mobile host M^i is the primary host of some object k', then k' should not be in L^i. We also maintain a list LM containing all mobile hosts that can replicate an object, i.e., $M^i \in LM$ if $L^i \neq$NULL. The algorithm works iteratively. In each step the mechanism (ς) asks all the players to send their preferences (first PARFOR loop). Each player i recursively calculates the true data of every data object in list L^i. Each player then reports the dominant true data (line 09) to the mechanism. The mechanism receives all the corresponding entries, injects it with the update cost and then chooses the globally dominant true data. This is broadcasted to all the players, so that they can update their nearest neighbor table NN_k^i, which is shown in Line 20 (NN^i_{OMAX}). The object is replicated and the payment is made to the player. The mechanism progresses forward till there are no more players interested in acquiring any data for replication (Line 18).

V. Discussion of Results

Mobile hosts exist in a size 1000×1000 flatland. Each host randomly moves in all directions, and the movement speed is randomly determined from 0 to d. The radio communication range of each mobile host has a radius of R. The number of mobile hosts was set to 200, and the number of data objects was set to 2000. The primary data object's original mobile host was mimicked by choosing random locations. The size of data objects was obtained using the highly cited [3] hybrid lognormal-pareto distribution, where the distribution's body follows lognormal and tail follows pareto. In all the experiments, the basic storage capacity of mobile hosts (C%) was proportional to the total size of data objects. In order to ensure that the system had mobile servers with diverse enough storing capabilities, the actual storage capacity, s^i, of a mobile host was a random value between (C/2)% and (3C/2)%. For simulation purposes, the replicas were Mobile hosts exist in a size 1000×1000 flatland. Each host randomly moves in all directions, and the movement speed is randomly determined from 0 to d. The radio communication range of each mobile host has a radius of R. The number of mobile hosts was set to 200, and the number of data objects was set to 2000. The primary data object's original mobile host was mimicked by choosing random locations. The size of data objects was obtained using the highly cited [3] hybrid lognormal-pareto distribution, where the distribution's body follows lognormal and tail follows pareto. In all the experiments, the basic storage capacity of mobile hosts (C%)

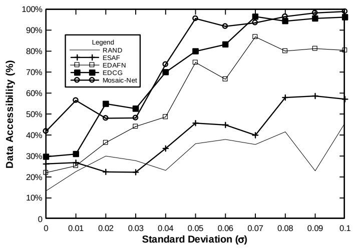

Fig. 2 (a) Scatter of access and data accessibility

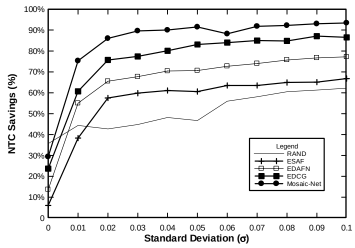

Fig. 2(c) Scatter of access and NTC

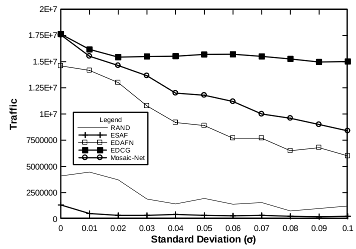

Fig. 2(b) Scatter of access and traffic

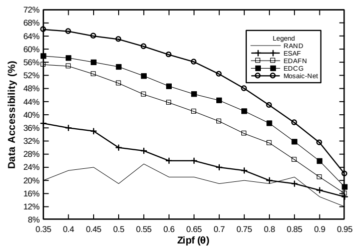

Fig. 3 (a) Updates and data accessibility

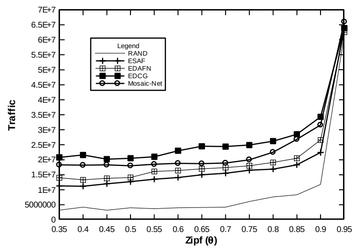

Fig. 3 (b) Updates and traffic

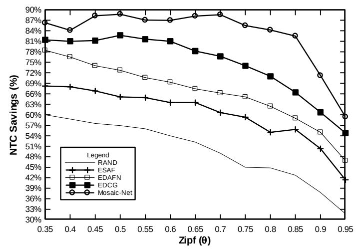

Fig. 3 (c) Update and NTC

TABLE I
RUNNING TIME IN SECONDS

Problem Size	RAND	ESAF	EDAFN	EDCG	Mosaic-Net
m=200, *n*=2000 [*d*=1, *R*=7, *T*=256, *C*=30%, σ=0.01, θ=0.45]	**13.27**	23.28	21.28	29.42	18.33
m=200, *n*=2000 [*d*=1, *R*=7, *T*=256, *C*=30%, σ=0.01, θ=0.65]	**15.05**	22.95	22.64	29.83	18.23
m=200, *n*=2000 [*d*=1, *R*=7, *T*=256, *C*=30%, σ=0.01, θ=0.75]	17.29	27.15	23.54	31.52	**15.40**
m=200, *n*=2000 [*d*=1, *R*=7, *T*=256, *C*=30%, σ=0.01, θ=0.85]	17.82	30.58	26.05	32.38	**13.51**
m=200, *n*=2000 [*d*=1, *R*=7, *T*=256, *C*=30%, σ=0.01, θ=0.95]	20.01	35.33	30.96	40.45	**9.06**

was proportional to the total size of data objects. In order to ensure that the system had mobile servers with diverse enough storing capabilities, the actual storage capacity, s^i, of a mobile host was a random value between (C/2)% and (3C/2)%. For simulation purposes, the replicas were the assumptions and system parameters were kept the same in all the approaches. The techniques studied include: 1) randomized, 2) extended static access frequency (ESAF) [6], 3) extended dynamic access frequency and neighborhood (EDAFN) [6], and 4) extended dynamic connectivity grouping (EDCG) [6]. Note that the proposed game theoretical technique has an acronym Mosaic-Net. Due to space restrictions, we could not provide the detailed workings for each of the comparative techniques. Readers are encouraged to find more about the techniques from the referenced papers.

We use three performance metrics defined as follows:

1. Data accessibility: Percentage number of successful hits/access to data objects.
2. Traffic: The total hop count of data transmission for allocating/relocating replicas.
3. NTC savings: Network transfer cost (in percentage) that is saved under the replication scheme found by the algorithm, compared to when only primary copies exists.

We examine the characterization of scattering the read access frequency of mobile hosts. To observe this the read access frequencies of mobile hosts are determined based on case 3, and the standard deviation, σ, is changed. When σ = 0, access frequencies to data objects is equivalent to case 1. As σ increases, the scatter if read access frequencies increases. As a consequence the difference in read access characteristics among the mobile hosts also increases. We fix m = 200, n = 2000, T = 256, C = 30%, d = 1, R = 7 and θ = 0.90. Figs. 2(a)-2(c) show the simulation results.

Fig. 2(a) shows that as the difference of read access characteristics gets larger, the relative difference in data accessibility increases. This is because when the scatter of access characteristics is larger, each mobile host allocates more replicas, and thus, the mobile hosts share a wide range of data objects. This sharing of data objects in turn generates traffic (for accessing data objects) as depicted in Fig. 2(b). However, notice that with the increase in read access characteristic, the traffic generated by Mosaic-Net method reduces. This phenomenon is attributed to the fact that when the access characteristic is larger, the degree of duplication is small in Mosaic-Net compared to that of EDCG.

With the increase in read access characteristics, all methods showed increase in NTC savings. Mosaic-Net showed (Fig. 2(c)) the most savings followed by EDCG. Observe that nearly every method showed almost 90% of its total improvement in NTC savings with an initial increase in σ (as little as 0.01). This is due to the fact that σ injects much more diversity than the number of objects available in the system. Thus, with the later increase in σ not much of NTC savings is observed.

We examine the effects of change in updates. We fix m = 200, n = 2000, T = 256, C =30%, d = 1, R = 7 and σ = 0.01. The increase in the frequency of updates in the system requires the replicas be placed as close as to the primary site as possible (to reduce the update broadcast). This phenomenon is also interrelated with the system storage capacity, as the update ratio sets an upper bound on the possible traffic reduction through replication. Thus, if we consider a system with unlimited storage capacity, the "replicate everywhere anything" policy is strictly inadequate [10]. Fig. 3(a)-3(c) show the results.

Fig. 3(a) shows that as the Zipf parameter (θ) increases, the accessibility of every method decreases. The reason behind this is that when the update frequency increases, the replica contents change with respect to their originals frequently. This in turn shows that when the update frequency is higher, all the comparative methods (RAND, ESAF, EDAFN and EDCG) discussed in this paper do not perform well. This is attributed to the fact that the replicas quickly become invalid, and thus, relocation considering the time remaining until each data object is updated next has rather adverse effects. On the other hand if the update frequency is lower, relocation considering the time remaining until each data object is updated next is meaningless and it might also result in adverse effects.

Fig. 3(b) shows the relationship between traffic generated and the Zipf parameter (θ). EDCG method produces the highest amount of traffic and Mosaic-Net method produces the next highest. Notice that as the update frequency gets higher the traffic soars. This is because each mobile host must frequently refresh the replicas that they hold after the originals have been updated. This fact is also realized in Fig. 3(c) where all the methods gradually loose NTC savings as the Zipf parameter increases. From the Figs. 3(a)-3(c) we can observe that the idea value for the Zipf parameter is 0.90.

Finally, we compare the termination time of the algorithms. We randomly choose the largest size problem instances with varying parameters. The entries (Table I) in bold represent the fastest time recorded over the problem instance. It is observable that RAND and Mosaic-Net terminated faster than all the other techniques.

VI. Conclusion

The proposed game theoretical replica allocation mechanism in ad hoc networks (Mosaic-Net) is a protocol for automatic replication and migration of objects in response to demand changes. Mosaic-Net aims to place objects in the proximity of a majority of requests while ensuring that no mobile hosts become overloaded.

We compared Mosaic-Net with some well-known techniques, as: randomized, extended static access frequency, extended dynamic access frequency and neighborhood and extended dynamic connectivity group. Mosaic-Net compared to other techniques exhibited up to 20% better solution quality and considerable savings in termination timings.

As future work, we would like to extend Mosaic-Net to incorporate the phenomenon of unstable radio links [7]. That is, when mobile hosts are connected by unstable radio links, which are likely to be disconnected after a short time, it is inefficient to allocate different replicas on them because they cannot share the replicas after disconnections.

References

[1] D. Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," in *ACM SIGMOD*, 1994, pp. 1-12.
[2] J. Cay, K. L. Tan and B. C. Ooi, "On Incremental Cache Coherency Schemes in Mobile Computing Environments, in *IEEE International Conference on Data Engineering*, 1997, pp. 114-123.
[3] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, 5(6), pp. 835-846, 1997.
[4] V. Grassi, "Prefetching Policies for Energy Saving and Latency Reduction in a Wireless Broadcast Data Delivery System," in *ACM International Workshop on Modeling*, 2000, pp. 77-84.
[5] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," in *IEEE INFOCOM*, 2001, pp. 1568-1576.
[6] T. Hara, "Replica Allocation in Ad Hoc Networks with Data Update," *Mobile Networks and Applications*, vol. 8, pp. 343-354, 2003.
[7] T. Hara and S. K. Madria, "Dynamic Data Replication Using Aperiodic Updates in Mobile Adhoc Networks," in *9th International Conference on Database Systems for Advance Applications*, 2004, pp. 869-881.
[8] Y. Huang, P. Sistla and O. Wolfson, "Data Replication for Mobile Computer," in*ACM SIGMOD*, 1994, pp. 13-24.
[9] J. Jing, A. Elmagarmid, A. Helal and R. Alonso, "Bit-sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *Mobile Networks and Applications*, 2(2), pp. 115-127, 1997.
[10] S. Khan and I. Ahmad, "A Powerful Direct Mechanism for Optimal WWW Content Replication," in *19th IEEE International Parallel and Distributed Processing Symposium*, 2005.
[11] E. Pitoura and B. Bhargava, Maintaining Consistency of Data in Mobile Distributed Environments, in *IEEE International Conference on Distributed Systems*, 1995, pp. 404-413.
[12] S. Saurabh and D. Parkes, "Hard-to-Manipulate VCG-Based Auctions," Technical Report, Harvard University, 2004.
[13] K. L.Wu, P. S. Yu and M. S. Chen, "Energy-efficient Caching for Wireless Mobile Computing, in *IEEE International Conference on Data Engineering*, 1996, pp. 336-343.
[14] W. Vickrey, "Counterspeculation, Auctions and Competitive Sealed Tenders," *Journal of Finance*, pp. 8-37, 1961.
[15] G. Zipf, *Human Behavior and the Principle of Least-Effort*, Addison Wesley, Cambridge, MA, 1949.