Static Replication Strategies for Content Availability in Vehicular Ad-hoc Networks[*]

Shyam Kapadia[1,†], Bhaskar Krishnamachari[1,2], Shahram Ghandeharizadeh[1]
[1]Department of Computer Science
[2]Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, USA
Email:{kapadia,bkrishna,shahram}@usc.edu

Abstract

This study investigates replication strategies for reducing latency to desired content in a vehicular peer-to-peer network. We provide a general constrained optimization formulation for efficient replication and study it via analysis and simulations employing a discrete random walk mobility model for the vehicles. Our solution space comprises of a family of popularity based replication schemes each characterized by an exponent n. We find that the optimal replication exponent depends significantly on factors such as the total system storage, data item size, and vehicle trip duration. With small data items and long client trip durations, $n \sim 0.5$ i.e., a square-root replication scheme provides the lowest aggregate latency across all data item requests. However, for short trip durations, n moves toward 1, making a linear replication scheme more favorable. For larger data items and long client trip durations, we find that the optimal replication exponent is below 0.5. Finally, for these larger data items, if the client trip duration is short, the optimal replication exponent is found to be a function of the total storage in the system. Subsequently, the above observations are validated with two real data sets: one based on a city map with freeway traffic information and the other employing encounter traces from a bus network.

[*] A shortened version of this study, specifically, Section 7 was presented at ACM MDM in 2005.

[†] Corresponding author

Table 1: Organization of this paper

1 Introduction

Advances in computer processing, data storage, and wireless communications have made it feasible to envision on-demand delivery of content such as audio and video clips between mobile vehicles. The content exchanged between the vehicles may vary from traffic information such as accident notifications and emergency vehicle arrival notifications to multimedia for entertainment such as audio files, cartoons, movies and other video files. A vehicle is equipped with an AutoMata [2](formerly known as a C2P2 for Car-to-Car Peer-to-Peer [14]) device consisting of several gigabytes of storage, a fast processor and a short-range wireless interface with bandwidths of several tens of Mbps [3]. The AutoMata-equipped vehicles, forming an intermittently connected network [33, 29], collaborate to realize an application for on-demand delivery of entertainment content.

In such a system, when a client vehicle issues a request for desired content not found in its local storage, this request can be satisfied only when it is in the vicinity of another vehicle that carries the requested item. Therefore, a key metric is availability latency, defined as the time between request issuance and request satisfaction. Clearly, the more the number of vehicles carrying a certain data item, the lower will be its expected availability latency. However, constraints on storage limit the number of unique data items that each vehicle can carry.

All data items in the system repository are not likely to be equally requested[1]. The aggregate availability latency across all items is therefore the average latency for individual data items weighed by their respective popularities. Hence, in order to minimize this aggregate latency metric, the popular data items warrant more replicas. We address the following key question in this study: **how many replicas should be allocated for each item in the repository in order to minimize the aggregate availability latency?**

Several parameters impact the optimal replication scheme. For instance, the mobility model of the vehicles impacts how the expected availability latency for a data item decreases with the number of replicas for that item. The size of a data item and the available bandwidth between devices dictates whether it is possible to download the entire data item in a single encounter between a client and a vehicle carrying the requested item. Moreover, the client trip duration bounds the maximum amount of time that a client is willing to wait for a request to be satisfied. The distribution of the data item popularities is a key component of the aggregate latency metric. The available storage directly affects the constraints under which the optimal replication strategy can be found.

Our primary contributions are as follows. We first provide a general optimization formulation for minimizing the

[1] It is found that Zipf's law controls many of the features observed with the Internet, primarily because of different user preferences for different files [4].

average availability latency subject to a storage constraint per vehicle. To solve this optimization, the solution space we explore comprises a family of popularity-based replication schemes each characterized by an exponent n. This exponent, n, defines the relation between the replicas of a data item and its popularity. We are interested in the optimal replication exponent that minimizes the aggregate availability latency. With small data items and long client trip durations, we find that $n \sim 0.5$ i.e., a square-root replication scheme provides the lowest aggregate latency. However, for short trip durations, we find that the optimal replication exponent (n) moves toward 1, making a linear replication scheme more favorable. For larger data items and long client trip durations, we find that the optimal replication exponent is below 0.5. In the limit for extremely large data items, a random ($n = 0$) replication scheme that allocates the same number of replicas to all data items yields the minimum aggregate latency. Finally, for such data items, if the client trip duration is short, we find that the optimal replication exponent is a function of the total storage in the system. Specifically, in low storage scenarios, a linear scheme shows superior performance, while for moderate to high storage scenarios a square-root replication scheme is preferred. Moreover, if the storage is abundant even a random replication scheme is good enough to provide a low aggregate latency.

While the above results are based on a 2D random walk based mobility model for the vehicles, in the second part of this study, we validate our model observations with vehicular movements obtained from real data sets. Specifically, two independent validation phases are presented employing (a) A real map of an urban environment that dictates the mobility transitions of the Markov model and (b) Fine grained mobility traces from a real environment comprising buses moving around a university campus area. The observations from these studies indicate that a random walk-based mobility model captures performance trends that may be applicable for a wide range of scenarios.

The snapshot of this paper with the section headings and the corresponding section numbers is listed in Table 1. The rest of this paper is organized as follows. Section 2 gives a brief overview of the related work in the area. Section 3 presents the general optimization formulation and details about the family of frequency-based replication schemes explored in our study. Section 4 employs mathematical analysis and simulations to determine the optimal replication scheme for small data items and long client trip durations. The results for small data items are extended to consider short client trip durations in Section 5. Sections 6 and 7 present the optimal replication scheme for larger data items with long and short client trip durations, respectively. The above sections employ a 2D random walk-based mobility model for the vehicles. Subsequently, representative results obtained with the random walk model are validated on a map of the city of San Francisco in Section 8. This is followed by exploration of the performance of the replication schemes on a real data set comprising of movement traces of buses in a small neighborhood in Amherst (Section 9). Finally, Section 10 presents brief conclusions, some discussion and concrete future research directions.

2 Related work

Techniques to determine number of replicas are similar to assigning seats to different parties as a function of their popularity, i.e., ratio of votes casted for a party to the total vote. Webster's divisor method and its alternatives attributed to Hamilton, Adams and Jefferson allocate storage to each replica (assign seats to a party) as a linear function of its frequency of access (popularity). The divisor technique has been employed to determine the number of replicas of a video clip in a distributed video-on-demand architecture [34]. Our proposed framework captures this technique by setting a key parameter, denoted as n (see Section 3), to one.

Techniques to compute number of replicas for objects have been studied for both peer-to-peer networks [8] and mesh community networks [10, 15]. Mobility of nodes is our primary contribution and separates these prior studies from the work presented here. In the following, we provide an overview of these prior studies. Replication of objects is important to their discovery in an un-structured peer-to-peer network. A smart replication technique minimizes search size, defined as the number of walks required to locate a referenced data item. In [8], the divisor method is compared with one that employs the square root of the frequency of access, demonstrating the superiority of the later.

Replication in MANETs has been explored in a wide variety of contexts. Hara [17] proposes three replica allocation methods. The first one that allocates replicas to nodes only on the basis of their local preference to data items. The second technique extends the first by considering the contents of the connected neighbors while performing the allocation to remove some redundancy. The last technique discovers bi-connected components in the network topology and allocates replicas accordingly. The frequency of access to data items is known in advance and does not change. Moreover, the replica allocation is performed in a specific period termed the relocation period. Several extensions to this work have been proposed where replica allocation methods have been extended to consider data items with periodic [19, 21] and aperiodic [24, 20] updates. Further extensions to the proposed replica allocation methods consider the stability of radio links [26], topology changes [27] and location history of the data item access log [22, 23]. In [25], the authors consider data items that are correlated by virtue of being requested simultaneously and present the performance of the replica allocation methods proposed earlier. Related studies [31, 35, 18] have appeared in the context of cooperative caching in mobile ad-hoc networks where the objective is to use caching to reduce the mobile node's latency in accessing data items. All the above studies are based on simulations of the proposed replica allocation methods.

Our study differs from the above in the following ways. We formulate a general optimization problem that minimizes an aggregate latency metric subject to a storage constraint per vehicle. We propose a family of replication schemes and explore which scheme provides optimal latency under a variety of scenarios by solving the optimization formulation. The mobility of vehicles is represented by a Markov-based mobility model that is general enough to capture a wide variety of mobility models such as Freeway, Highway, Random Way-point etc. We have analyzed the latency performance obtained with such a mobility model via mathematical analysis and extensive simulations. The different scenarios encompass vehicles with unbounded as well as finite trip durations, data items with different display times etc.

In prior work [11], we have explored data discovery in an AutoMata system and proposed PAVAN as a policy framework to predict the list of items available to a client during its journey along with their respective availability latencies. The effectiveness of the predictions are then evaluated by the comparing the predicted lists with the actual lists employing suitable utility models. In other work [12], we explore three popular replication schemes, namely, linear, square-root and random schemes for larger data items with short trip durations. This study significantly extends the prior work along a number of dimensions such as different data item sizes, different client trip durations etc. Finally, in our zebroid study [13], vehicles are employed as data carriers to further alleviate the availability latency of active requests thereby causing dynamic data re-distribution across the system storage to better match the currently active requests. Appropriate static replication schemes allocate replicas for data items at the bootstrap phase, after which data carriers may be scheduled to improve client latency.

Another related area is that of Intelligent Transportation Systems (ITS). The vehicular network is viewed as a MANET and messages are forwarded from one vehicle to another realizing several applications like vehicle accident notification broadcast, pre-emptive emergency vehicle arrival information etc [5, 6]. The lessons learnt from this study may be directly applicable to some ITS problems.

3 General framework

In this section, we first introduce some definitions and associated terminology used repeatedly in this paper. Then, we provide a general optimization formulation for minimizing availability latency in the presence of storage constraints. Table 2 summarizes the notation used in this study.

Assume a network of N mobile AutoMata devices, each with storage capacity of α bytes. The total storage capacity of the system is S_T=$N \cdot \alpha$. There are T data items in the repository, each with a display time of Δ_i seconds and display bandwidth requirement of β_i. Hence, the size of each item is given by $S_i = \Delta_i \cdot \beta_i$. The frequency of access to item i is denoted as f_i with $\sum_{j=1}^{T} f_j = 1$. Let the trip duration of the client AutoMata under consideration be γ.

We now define the normalized frequency of access to the item i, denoted R_i, is:

Database Parameters	
T	Number of data items.
S_i	Size of data item i.
Δ_i	Display time of data item i.
β_i	Bandwidth requirement of data item i.
f_i	Frequency of access to data item i.
Replication Parameters	
R_i	Normalized frequency of access to data item i, $R_i = \frac{(f_i)^n}{\sum_{j=1}^{T}(f_j)^n}; 0 \leq n \leq \infty$
r_i	Number of replicas for data item i, $r_i = \min\left(N, \max\left(1, \lfloor \frac{R_i \cdot N \cdot \alpha}{S_i} \rfloor\right)\right)$
n	Characterizes a particular replication scheme.
δ_i	Average availability latency of data item i
δ_{agg}	Aggregate availability latency for replication technique using the n^{th} power, $0 \leq n \leq \infty$, $\delta_{agg} = \sum_{j=1}^{T} \overline{\delta_j} \cdot f_j$
AutoMata System Parameters	
N	Number of AutoMata devices in the system.
α	Storage capacity per AutoMata.
γ	Trip duration of the client AutoMata.
S_T	Total storage capacity of the AutoMata system, $S_T = N \cdot \alpha$.

Table 2: Terms and their definitions

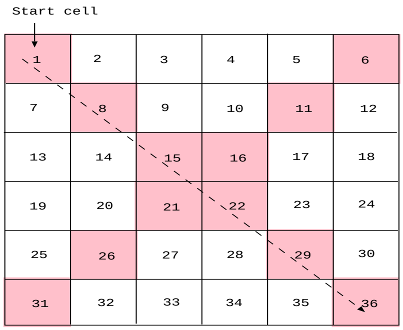


Figure 1: An example 6×6 map.

$$R_i = \frac{(f_i)^n}{\sum_{j=1}^{T}(f_j)^n}; 0 \leq n \leq \infty \tag{1}$$

R_i is normalized to a value between 0 and 1. The number of replicas for title i, denoted as r_i, is:

$$r_i = \min\left(N, \max\left(1, \lfloor \frac{R_i \cdot N \cdot \alpha}{S_i} \rfloor\right)\right) \tag{2}$$

This defines a family of replication schemes that computes the degree of replication of item i as the n^{th} power of its frequency of access. The exponent n characterizes a particular replication scheme. Hence, r_i lies between 1 and N. Note that r_i includes the original copy of a clip. One may simplify Equation 2 by replacing the $\max$ function with $\lfloor \frac{R_i \cdot N \cdot \alpha}{S_i} \rfloor$. This would allow the value of r_i to drop to zero for item i. This means that there is no copy of the clip in the AutoMata network. In this case, a hybrid framework might provide access to the clip i. For example, a base station employing IEEE 802.16 [16] might facilitate access to a wired infrastructure with remote servers containing the item i. However, in this study, we assume at least one copy of every data item must be present in the ad-hoc network at all times.

The availability latency for a given request for item i, denoted as δ_i, is defined as the time after which a client AutoMata will find at least one replica of its requested item accessible to it, either directly or via multiple hops. Additionally, δ_i must be greater than Δ_i, the data item display time. If a replica for item i is not encountered for a given request, we set δ_i to γ. This indicates that item i was not available to the client during its journey. Also, if Δ_i exceeds γ for a certain item i then we set δ_i to γ. We are interested in the availability latency observed across all the data items. Hence, we augment the δ_i for every item i with its f_i. This is termed the aggregate availability latency (δ_{agg}) metric. It is computed as follows. For each item i, calculate the average availability latency ($\overline{\delta_i}$) based on the particular replication scheme of interest. Then these availability latencies are combined into a single metric:

$$\delta_{agg} = \sum_{i=1}^{T} \overline{\delta_i} \cdot f_i \tag{3}$$

The aggregate availability latency, δ_{agg}, depends on the value chosen for n, since n determines the replicas per data item. Intuitively, a higher number of replicas for data item i reduces the availability latency experienced by a request for that data item. The core problem of interest here is to keep the aggregate availability latency as low as possible by tuning the data item replication levels, in the presence of storage constraints. We assume that the data item repository size is smaller than the total storage capacity of the system, $\sum_{i=1}^{T} S_i \leq S_T$. Otherwise, data items cannot be replicated when at least one replica of a data item must be present in the system. More formally, the optimization problem can be stated as,

$$\textbf{Minimize } \delta_{agg}\textbf{, subject to } \sum_{i=1}^{T} S_i \leq S_T \tag{4}$$

Implicit in this formulation is the design variable, namely, the desired replication for each data item. The value of n in Equation 4 determines a r_i value for each data item i with the objective to minimize δ_{agg}. This minimization is a challenge when the total size of the database exceeds the storage capacity of a car, $\sum_{i=1}^{T} S_i > \alpha$. Otherwise, the problem is trivial and can be solved by replicating the entire repository on each device.

The optimization space that defines what value of n provides the best δ_{agg} is quite large and consists of the following parameters: (i) density of cars, (ii) data item display time, (iii) size of the data item, (iv) display bandwidth per data item, (v) data item repository size, (vi) storage per car, (vii) client trip duration, (viii) frequency of access to the data items, and (ix) mobility model for the cars. In this study, we determine how the aggregate availability latency and hence the optimal replication scheme is affected by each of these parameters.

3.1 Model Preliminaries

In this section, we provide details about the discrete model adopted in this study for both analysis and simulations. We assume a repository of homogeneous data items with identical bandwidth requirement, display time, and size ($\beta_i = \beta$, $\Delta_i = \Delta$, $S_i = S$). Figure 1 shows an example map used in our study. The map constitutes a torus with fixed size cells, a torus is considered to avoid border effects [32, 28]. The vehicles[2] themselves are assumed to be distributed across these cells of the map. Only AutoMatas within a cell communicate with each other either directly if they are in radio-range or via other AutoMatas using multi-hop transmissions. In other words, the AutoMatas within a cell form a connected sub-network. AutoMatas in adjacent cells cannot communicate with each other.

A Markovian mobility model describes the movement of the vehicles where the vehicles perform a 2D random walk on the surface of the torus. Hence, in each time step, a vehicle in a given cell may transition to any one of its neighboring 4 cells. Each cell of the map constitutes a state. A map of size $G \times G$ yields G^2 states. These states are self-contained and a transition from one state to another is independent of the previous history of a car in that state. The aggregate of the transitions from each cell (state) to every other state gives the $G \times G$ probability transition matrix $Q = [q_{ij}]$ where q_{ij} is the probability of transition from state i to state j.

Using Markov chains, it is possible to estimate the distribution of the steady-state probabilities of being in the various cells, by solving $\Pi = \Pi * Q$, where Π is the vector representing the steady-state probabilities of being in the various cells (states). While for the 2D random walk mobility model, due to symmetry, $\Pi = \frac{1}{G}$, in general, the map may represent an underlying city area where the transition probabilities for each cell may not be symmetric in each of the 4 possible directions. For example, in the example map depicted in Figure 1, the mobility model is weighted toward the diagonal both from the left to right and vice-versa. This may be due to the presence of two major freeways running across a city area that the map represents. Note that in order to incorporate directionality, the transition for a vehicle from its current cell may incorporate the current as well as the previous location of the vehicle (see Section 8). In this case, the model may not be strictly Markovian, however, by incorporating previous as well as current vehicle location information in the state, the mobility model can still be solved employing Markov chains.

Without any loss of generality, to reduce the dimensionality of the problem, we express the data item display time, Δ, as the amount of time required by an AutoMata equipped vehicle to travel Δ cells. We express α as the number of storage slots per AutoMata. Each storage slot stores a data item fragment equivalent to a single cell worth of data item display time. Moreover, we assume the amount of data displayed in each cell is identical. Now, we represent both the size of a data item and the storage slots in terms of the number of cells. This means that a data item has a display time of Δ cells and an AutoMata has α units of cell storage. For example, a data item with display time of 4 cells ($\Delta = 4$) requires 4 storage slots and an AutoMata provides 100 storage slots ($\alpha = 100$). Hence, from hereon, we assume that the size of the data item is indicated by its display time.

The trip duration (γ) is the maximum amount of time that a client vehicle is willing to wait for request satisfaction. Here, it is expressed as the maximum number of discrete time steps that the client is willing to traverse before it gives up on a given request for a data item. We also define availability latency (δ_i) for data item i in discrete terms, as the number of steps after which a client AutoMata will encounter a replica of the requested data item i, either directly or via multiple hops, for the data item display time (Δ). Hence, the possible values of the availability latency are between 0 and γ. While in most cases, the trip duration is usually short, occasionally a client may specify an extremely large value of γ indicating that it is willing to wait as long as it takes to satisfy an issued request. As we shall show in the following sections, the long versus short client trip durations have a profound impact on the optimal replication scheme that minimizes the aggregate availability latency.

[2] We use the term AutoMata and a vehicle (or car) interchangeably in this study, with the assumption that each vehicle is equipped with a single AutoMata device.

4 Data items with small size and long client trip duration

In this section, we consider small data items with a display time of one, where the client trip duration is long. We first present analytical approximations that capture the performance of availability latency for an item as a function of the number of replicas for that item for both a low and high density of replicas. Subsequently, we employ simulations to determine the optimal replication exponent that minimizes the aggregate availability latency.

4.1 Analysis of data items with display time = one

In this section, we assume data items with a display time of one cell and for a scenario with a sparse density of data item replicas, derive a closed-form expression for the aggregate availability latency. Subsequently, we use this expression to solve the optimization problem to reveal that a square-root replication scheme minimizes this latency. Then, we derive an expression that approximates the aggregate availability latency in case of a high density of data item replicas.

4.1.1 Sparse Scenario

In this section, we provide a formulation that captures scenarios with a low density of vehicles. For a given mobility model of the vehicles, the relationship between δ_i and r_i can be obtained using simulations. For illustration, we have considered that vehicles follow a random walk-based mobility model on a 2D-torus. Aldous *et al.* [1] show that the mean of the hitting time for a symmetric random walk on the surface of a 2D-torus is $\Theta(GlogG)$ where G is the number of cells in the torus. Moreover, the mean of the meeting time for 2 random walks is half of the mean hitting time. Furthermore, the distribution of the meeting times for an ergodic Markov chain can be approximated by an exponential distribution of the same mean [1]. Hence,

$$P(\delta_i > t) = \exp\left(\frac{-t}{c \cdot G \cdot logG}\right) \tag{5}$$

where the constant $c \simeq 0.34$ for $G \geq 25$. Now since there are r_i replicas, there are r_i potential servers. Hence, the the meeting time, or equivalently the availability latency for the data item i is the time till it encounters any of these r_i replicas for the first time. This can be modeled as a minimum of r_i exponentials. Hence,

$$P(\delta_i > t) = \exp\left(\frac{-t}{c \cdot \frac{G}{r_i} \cdot logG}\right) \tag{6}$$

Note, however that this formulation is valid only for the cases when $G >> r_i$, which is the case for sparse scenarios. The expected value of δ_i is given by:

$$\overline{\delta_i} = \frac{c \cdot G \cdot logG}{r_i} \tag{7}$$

For a given 2D-torus, G is constant, hence we have $\overline{\delta_i} \propto \frac{1}{r_i}$ or equivalently, $\overline{\delta_i} = \frac{C}{r_i}$ where $C = c \cdot G \cdot logG$.

Hence, we have the following optimization formulation,

$$\text{Min } \left[\sum_{i=1}^{T} f_i \cdot \frac{C}{r_i}\right] \tag{8}$$

Subject to:

$$\sum_{i=1}^{T} r_i = N \cdot \alpha \tag{9}$$

$$r_i \leq N \;; \forall\, i = 1 \text{ to } T \tag{10}$$

$$r_i \geq 1 \;; \forall\, i = 1 \text{ to } T \tag{11}$$

Theorem 1. *In case of a sparse density of vehicles, a replication scheme that allocates data item replicas as a function of the square-root of the frequency of access to data items minimizes the aggregate availability latency.*

$$r_i = \begin{cases} \frac{\sqrt{f_i} \cdot N \cdot \alpha}{\sum_{j=1}^{T} \sqrt{f_j}} & \frac{1}{N \cdot \alpha} \leq \frac{\sqrt{f_i}}{\sum_{j=1}^{T} \sqrt{f_j}} \leq \frac{1}{\alpha} \\ \max\left(1, \min(\sqrt{\frac{f_i \cdot C}{\gamma_0}}, N)\right) & \textit{in the general case} \\ & \textit{where } \gamma_0 \textit{ is s.t. } \sum_{i=1}^{T} r_i = N \cdot \alpha \end{cases} \tag{12}$$

Proof. We solve the above optimization using the method of Lagrange multipliers. First, we prove part(i) of the theorem.

The Lagrangian for the optimization can be written as:

$$H = \sum_{i=1}^{T} \frac{f_i \cdot C}{r_i} + \varphi \left[\sum_{i=1}^{T} r_i - N \cdot \alpha \right] \tag{13}$$

We solve for r_i as follows:

$$\frac{\partial H}{\partial r_i} = -f_i \cdot \frac{C}{r_i^2} + \varphi = 0 \tag{14}$$

$$r_i = \sqrt{\frac{C \cdot f_i}{\varphi}} \tag{15}$$

Substituting r_i in the constraint, we get:

$$\varphi = \left(\frac{\sum_{i=1}^{T} \sqrt{C \cdot f_i}}{N \cdot \alpha} \right)^2 \tag{16}$$

Finally, we get the optimal value of r_i as,

$$r_i = \frac{\sqrt{f_i} \cdot N \cdot \alpha}{\sum_{j=1}^{T} \sqrt{f_j}} \tag{17}$$

The constraints are satisfied if $\frac{1}{N \cdot \alpha} \leq \frac{\sqrt{f_i}}{\sum_{j=1}^{T} \sqrt{f_j}} \leq \frac{1}{\alpha}$ which proves part (i) of the theorem.

Without this condition on f_i, the above optimization can be re-written as the following Lagrangian taking all the constraints into account as:

$$G = \sum_{i=1}^{T} \frac{f_i \cdot C}{r_i} + \gamma_0 \left[\sum_{i=1}^{T} r_i - N \cdot \alpha \right] - \sum_{i=1}^{T} \gamma_i \cdot (r_i - N \cdot \alpha) - \sum_{i=1}^{T} \beta_i \cdot (-r_i + 1) \tag{18}$$

The Kuhn Tucker Conditions for the modified Lagrangian are:

$$- f_i \cdot \frac{C}{r_i^2} + \gamma_0 - \gamma_i + \beta_i = 0; \forall\, i = 1 \text{ to } T \tag{19}$$

$$\sum_{i=1}^{T} r_i \leq N \cdot \alpha,\, \gamma_0 \geq 0, \text{ and } \gamma_0 \left[\sum_{i=1}^{T} r_i - N \cdot \alpha \right] = 0 \tag{20}$$

$$r_i \leq N,\, \gamma_i \geq 0, \text{ and } \gamma_i \cdot (r_i - N) = 0; \forall\, i = 1 \text{ to } T \tag{21}$$

$$- r_i \leq -1,\, \beta_i \geq 0, \text{ and } \beta_i \cdot (-r_i + 1) = 0; \forall\, i = 1 \text{ to } T \tag{22}$$

Solving Equation 19, we get,

$$r_i = \sqrt{\frac{f_i \cdot C}{\gamma_0 - \gamma_i + \beta_i}} \tag{23}$$

Equations 21 and 22 imply that either $\gamma_i = 0$ or $r_i = N$ and also either $\beta_i = 0$ or $r_i = 1$ respectively. Therefore, the optimum solution for r_i is given by,

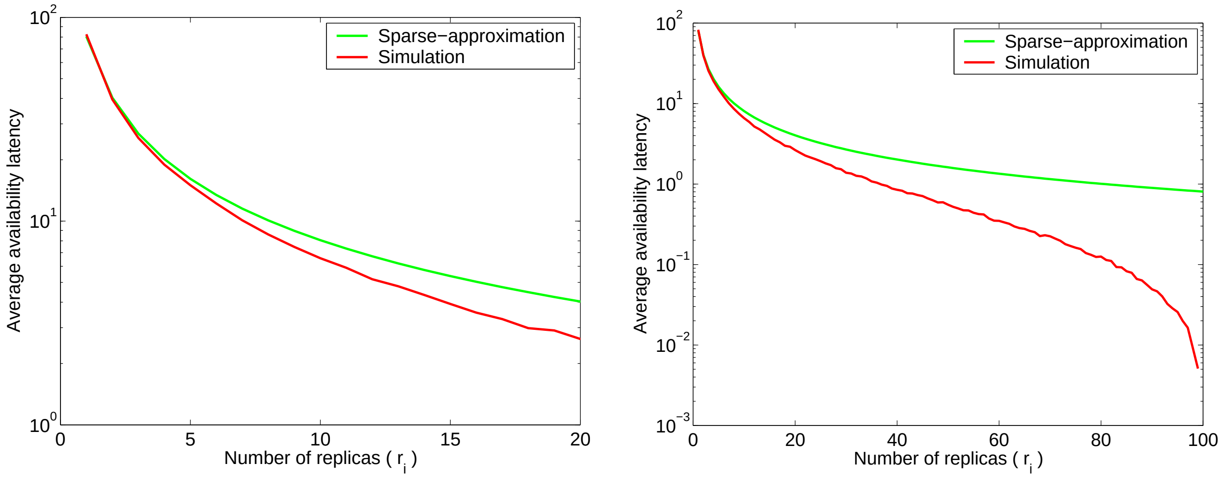
$$r_i = \max\left(1, \min(\sqrt{\frac{f_i \cdot C}{\gamma_0}}, N)\right) \tag{24}$$

where γ_0 is such that $\sum_{i=1}^{T} r_i = N \cdot \alpha$ proving part (ii) of the theorem. □

Hence, in a sparse network, the optimal replication that minimizes the aggregate availability latency is obtained if the number of replicas for a data item is proportional to the square root of the frequency of access for that data item. Cohen *et al.* [8] proved that for unstructured peer-to-peer networks the expected search size is minimized using a square-root replication strategy which is shown to be optimal. The aggregate availability latency metric in wireless mobile ad-hoc networks is analogous to the expected search size used in peer-to-peer networks.

It should be noted that, in general, the optimal replication depends on how δ_i is related to r_i i.e. $\delta_i = F(r_i)$ and $F(\cdot)$ is the function that will determine the optimal replication strategy. The above methodology can be used to be obtain the optimal number of replicas as long as $F(\cdot)$ is differentiable.

Figure 2 shows the typical trend shown by δ_i for a 10×10 torus, where r_i is increased from 1 to N where $N = 100$. In other words, in a $G = 100$ cell torus, $N = 100$ cars are deployed, with r_i of them having a replica for the data item. We only consider a single data item, a request for that item can be issued at any vehicle chosen uniformly at random among all the cars. If the item is stored locally, the latency is 0. This result is independent of the storage per car



(a)

(b)

Figure 2: Sparse analysis (Equation 7) versus simulation obtained average availability latency for a data item as a function of its replicas for a 10×10 torus, when the number of cars is set to 100. Figure 2(a) zooms into the sparse replica part of the entire spectrum depicted in Figure 2(b).

because recall that a maximum of one copy of a given data item may be stored in a car. Figure 2(a) indicates that when r_i is small, ($r_i \leq 20$) the analytical approximation in Equation 7 is valid. Subsequently, latency reduces at a much faster rate when compared to that predicted by the sparse approximation (see Figure 2(b)). This is because for a given G, as r_i increases, the latency till any one of the r_i replicas is encountered can no longer be modeled as the minimum of r_i independent exponentials. In the next section, we provide an approximation that captures the high density case.

4.1.2 Dense scenario

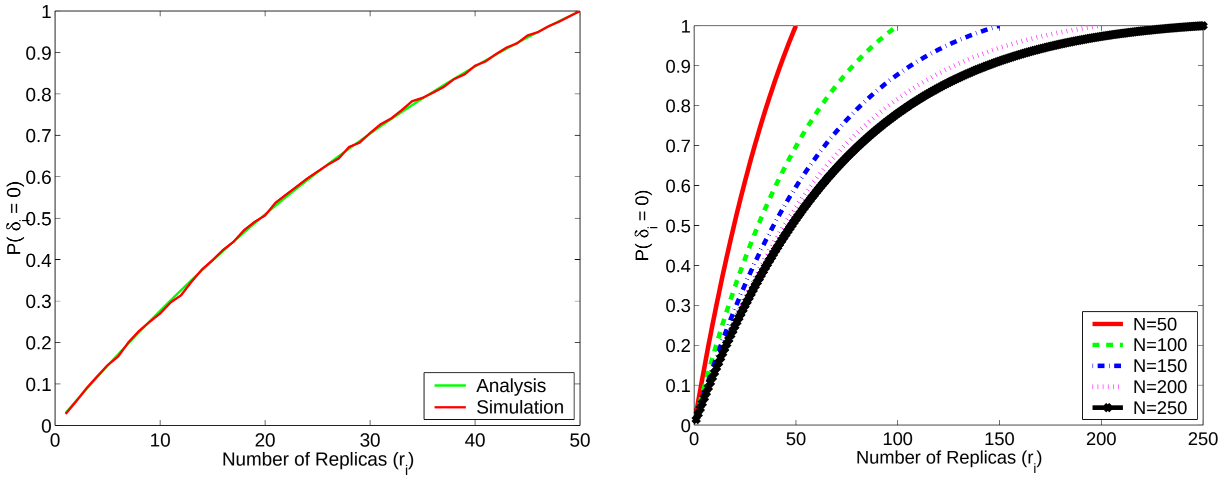
In this section, we provide an analytical formulation that captures the trends shown by the availability latency in the presence of a high density of replicas. Recall that N cars are distributed uniformly at random across G cells, r_i of the N cars carry a copy of the data item of interest. Here, we use the traditional definition of the expected availability latency for title i, namely,

$$\overline{\delta_i} = \sum_{k=0}^{\infty} k \cdot P(\delta_i = k) \tag{25}$$

We first determine an expression for the case when the latency is 0. This occurs if the data item is locally stored at a client or a data item replica is located in the same cell as the client at which the request is issued. Hence, the probability that the latency experienced by a client is zero is given by the following expression:

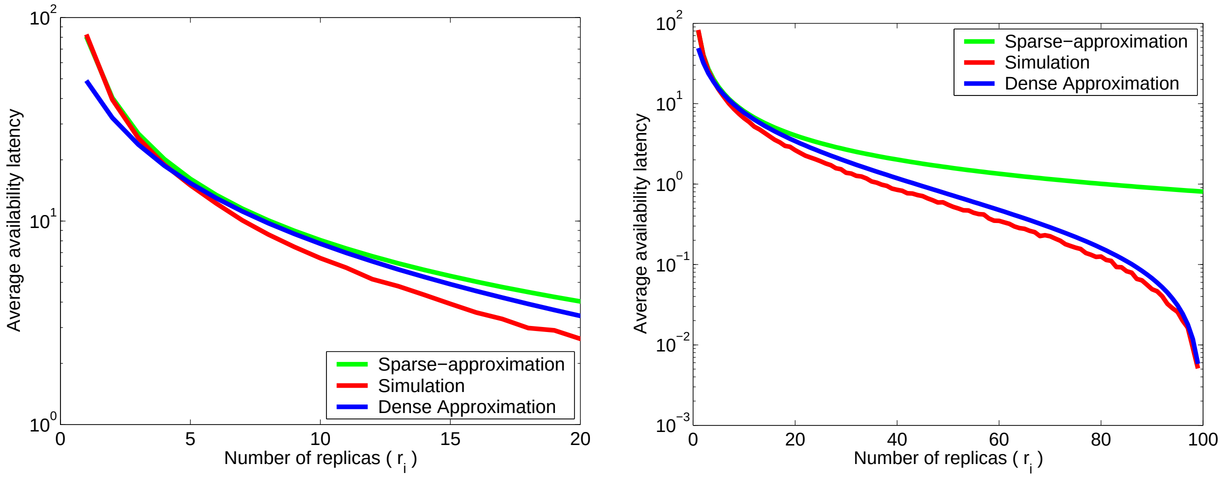
$$P(\delta_i = 0) = \frac{r_i}{N} + (1 - \frac{r_i}{N}) \cdot \left(1 - \left(1 - \frac{1}{G}\right)^{r_i}\right) \tag{26}$$

Figure 3(a) indicates that the analytical expression above matches the simulation results quite well. For a given car density N, as the density of replicas increases, the probability that the availability latency experienced by a client is zero also increases. Figure 3(b) shows how this probability varies with increasing car density. Given a torus comprising G cells, increase in $P(\delta_i = 0)$ shows a decreasing steepness as N increases. This is because with increasing N, the number of potential clients from which a request can be issued also goes up. Hence, a given value of r_i implies a greater percentage of the vehicles store the requested item i locally for a smaller N as compared to a larger one. Consequently, the corresponding $P(\delta_i = 0)$ is lower for a smaller N as compared to a larger one.



(a) (b)

Figure 3: Figure 3(a) shows the validation of the analytical expression in Equation 26 for the probability that availability latency is zero when $N = 50$. Figure 3(b) shows the probability that the availability latency is zero as a function of the replicas for the data item for 5 different car densities {50, 100, 150, 200, 250}.



(a) (b)

Figure 4: The complete picture depicting the availability latency for a data item obtained via simulations as compared with its sparse and dense approximation as a function of its replicas for a 10×10 torus, when the number of cars is set to 100. Figure 4(a) zooms into a part of Figure 4(b) that represents a sparse density of replicas.

We provide an approximation assuming a memoryless mobility model without regards to the shape of the region across which the vehicles are moving. Define, A_k, the event that a data item i is encountered by the client for the *first* time in the k^{th} cell. This implies that the item i was not encountered in any of the previous $k-1$ cells. Let $P(A_k)$ denote the probability of event A_k occurring. Note $P(A_k)$ is a joint probability function. Let p_k denote the probability of encountering data item i in the k^{th} cell, given that it was not encountered in the previous $k-1$ cells. Note that p_k is a conditional probability. Also, $p_1 = P(\delta_i = 0)$ as defined by Equation 26. Then,

$$p_k = 1 - \left(1 - \frac{1}{G-k+1}\right)^{r_i} \; ; 2 \leq k \leq G \tag{27}$$

Note that the model assumes that not encountering the data item in the $(k-1)^{th}$ cell increases the probability of encountering it in the k^{th} cell. Moreover, when $k = G$, $p_k = 1$ no matter what the value of r_i, meaning that the maximum latency that a client will encounter will always be $<= G$. Although this is true for a high density of replicas, this approximation is not valid for a sparse replica density where p_k may not increase as k increases especially for the first few steps of the client.

Recall, $P(A_k)$ is a joint probability since encountering a data item for first time in the k^{th} cell indicates that it was not encountered in any of the previous $k-1$ cells. Clearly, p_k and p_{k-1} are conditional probabilities that are not independent, hence, we use the following generalized multiplication rule to obtain the value of $P(A_k)$ as,

$$P(E_n...E_1) = P(E_n|E_{n-1}...E_1) \cdot P(E_{n-1}|E_{n-2}...E_1)...P(E_2|E_1) \cdot P(E_1) \tag{28}$$

$$P(A_k) = p_k \prod_{j=1}^{k-1} (1-p_j) \; ; 2 \leq k \leq G \tag{29}$$

Then, the average availability latency $(\overline{\delta_i})$ for data item i is given by,

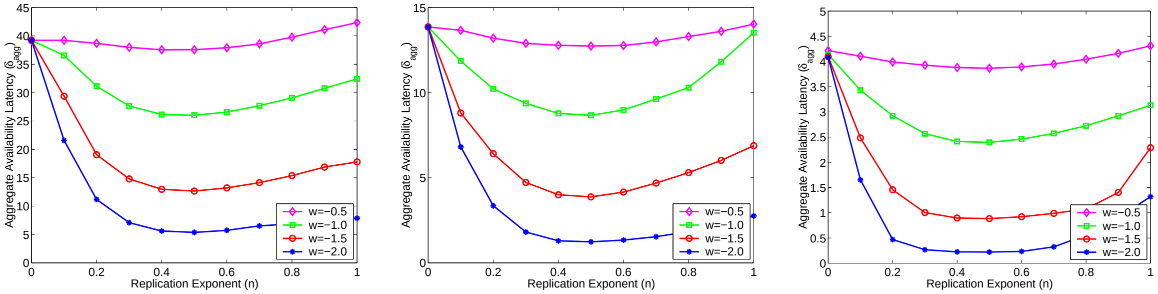
$$\overline{\delta_i} = \sum_{k=1}^{G} (k-1) P(A_k) \tag{30}$$

Figure 4 shows that the above equation captures the trend depicted by the average availability latency for higher replica densities where the sparse and dense approximations are plotted together with the latency obtained via simulations. However, when the dense approximation obtained in Equation 30 is plugged into the optimization presented in Section 3, the resulting numerical solution that does not lend itself directly to determining the optimal replication exponent n. Hence, we employ simulations to obtain the optimal replication exponent that minimizes the aggregate availability latency.

4.2 Simulation results of data items with display time = one

We present simulation results indicating the replication exponent range that provides near optimal aggregate availability latency. We also show how this latency is affected individually by the different parameters in the optimization space.

In all our experiments, we assume that the various data item popularities are distributed as per the Zipf's law [36]. This means that the frequency of the r^{th} popular data item is inversely proportional to its rank i.e.



(a) $\alpha = 4$ (b) $\alpha = 10$ (c) $\alpha = 25$

Figure 5: Aggregate availability latency for different replication strategies for a 10×10 torus when $T = 100$ and $N = 50$. Figures (a), (b), and (c) depict three different storage values per car: {4,10,25}.

$$f_i = \frac{\frac{1}{i^v}}{\sum_{j=1}^{T} \frac{1}{j^v}}; 1 <= i <= T \tag{31}$$

Here, the exponent v controls the skewness in the popularity distribution of the data items. We denote $w = -v$ as the skewness parameter. A higher absolute value of w indicates that most of the popularity weight is spread across the first few popular titles. Note that the data item repository size is T and the denominator is simply a normalization constant.

Figure 5 depicts the latency performance for different replication schemes when storage per car is increased from 4 to 25 slots. The title repository size is $T = 100$ and the car density is $N = 50$ which implies that the total storage S_T is increased from 200 to 1250 slots. As expected the latency decreases as storage is increased. The replication schemes with exponent values 0, 0.5, and 1 have been popularly studied in the literature and are labelled **random**, **square-root**, and **linear** respectively. The value of w captures the skewness in the data item popularities, a higher value of w indicates that most of the popularity weight is spread across the first few popular titles. Below we describe the main observations from this figure.

The random scheme allocates the same number of replicas per data item irrespective of their popularity. Hence, in all cases, it yields the same aggregate availability latency irrespective of the value of w. As the replication exponent increases from 0 to 1 progressively more replicas are allocated for the popular data items. This increase in the replicas is accelerated for higher values of w that provide a bias for the popular titles. Hence, we see a sharp decrease in the availability latency from $n = 0$ to $n = 0.3$ for $w = -1.5$ and $w = -2$. However, the maximum number of replicas per data item can never exceed N. For a value of $w = -0.5$ in which case the popularity weight is spread more evenly among all the data items, it almost doesn't matter what the replication scheme is as seen by the flat latency curves for $w = -0.5$.

When storage per car is low, $\alpha = 4$, this represents a scenario with a sparse density of data item replicas. In this case, the square-root replication scheme provides the minimum latency. Also, the range where the replication exponent n varies from 0.4 to 0.6 shows a latency very close to the square-root scheme. This is true even when the data item popularities are skewed. Moreover, the range $0.4 \leq n \leq 0.6$ shows near optimal latency performance even when the storage is increased (see Figures 5(b) and (c)). In other words, through the entire spectrum of the replica density, a replication scheme defined by an exponent in this range will provide near optimal performance. For the rest of this chapter, we will consider the square-root ($n = 0.5$) scheme as representative of this range and compare its performance to the two extremes namely, random ($n = 0$) and linear ($n = 1$).

4.3 Scale-up experiments

In these set of experiments, we maintain a constant ratio of the total storage to the data item repository size ($S_T : T$). Figure 6 presents the performance of the three replication schemes when $S_T = 3000$ and $T = 600$. Since $S_T = N \cdot \alpha$, we vary the values of (N,α) as {(50,60), (100,30), (150,20), (200,15), (250,12)}. Since the total storage in the system S_T remains the same the number of replicas allocated per data item also remains. As N increases, the number of potential clients increases, this accounts for the slight upward trend in the latency curves for the different replication schemes. With increasing skewness, for the same total storage, the latency realized by the square-root and linear schemes reduces. This is because replicas assigned to the more popular data items result in lower latency for those items because as the skewness parameter w increases, a higher popularity weight assigned to these data items. The random replication scheme is blind to the popularity of the data items and hence shows similar latency performance independent of the value of w. For all but $w = -0.5$ it performs an order of magnitude worse as compared to the square-root scheme.

4.4 Variation in car density

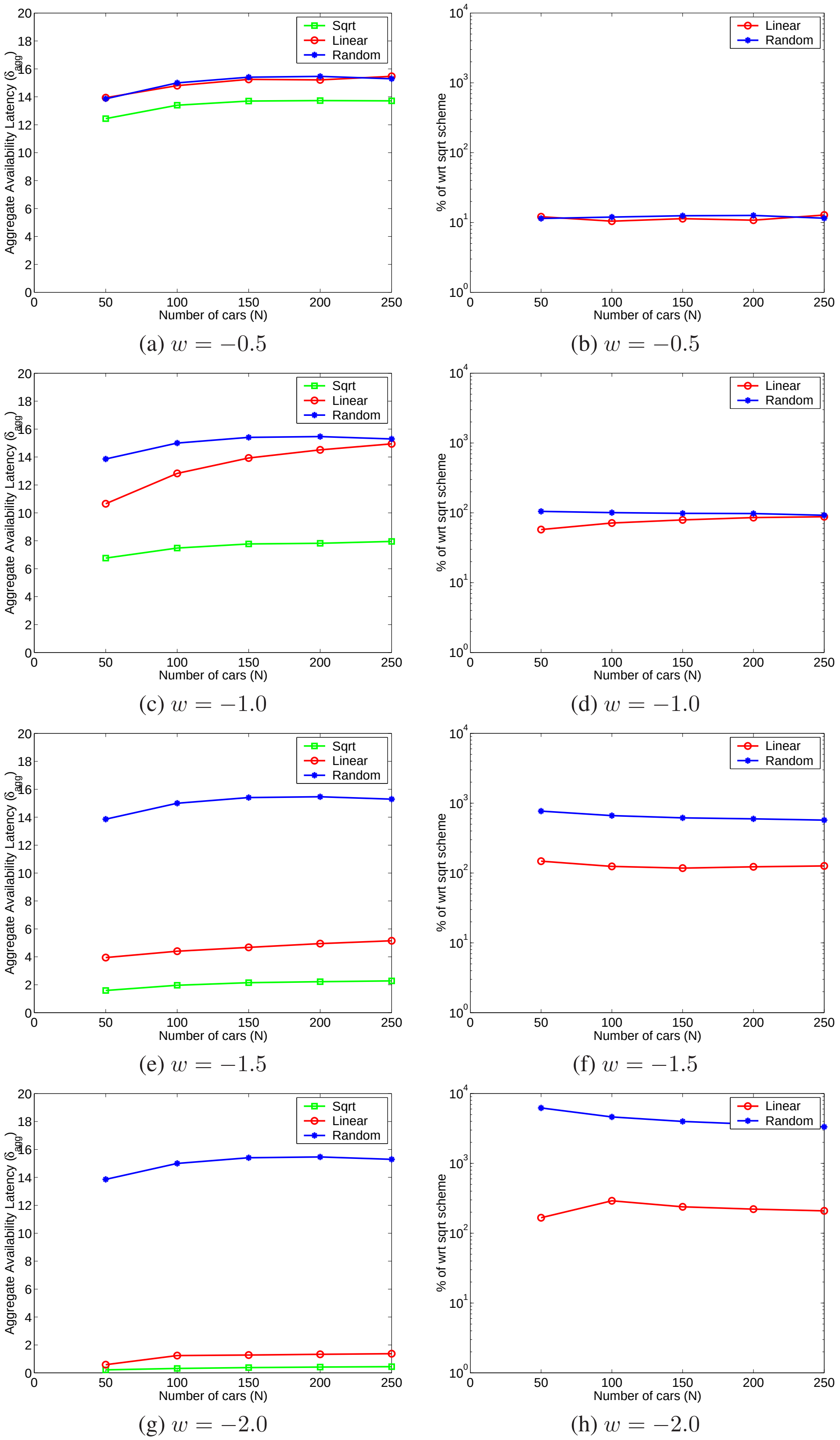
Next, we study the effect of car density on the performance of the replication schemes. Figure 7 presents the performance of the three replication schemes as a function of the car density when the storage per car is held constant at 3 for $T = 100$. Increase in the car density increases the total storage in the system. Hence, more replicas per data item can be allocated resulting in an overall decrease in the aggregate availability latency. This is true for all replication schemes. However, here for $w = -0.5$ and $w = -1$ (beyond $N = 100$), the random scheme shows slightly better performance than the linear scheme. This is because for low skewness parameters assigning equal number of replicas per data item is better than providing higher replicas for the popular data items which do not have a sufficiently high popularity weight. However, for higher skewness in popularity, the behavior of the linear scheme starts paying richer dividends in reducing the overall latency, hence, it outperforms the linear scheme. In all cases, the square-root scheme always yields the lowest aggregate availability latency.

4.5 Variation in storage per car

The total storage in the system can also be increased by keeping the car density constant and increasing the storage per car. Figure 8 shows the performance of the three replication schemes as a function of the storage per car when the car density is held constant at 50 for $T = 50$. As expected increasing storage reduces the latency for all the schemes. In case of $w = -0.5$, the random and linear scheme show a latency performance within $10 - 20\%$ of the square-root scheme. However, with higher w values this difference blows up with the square-root scheme providing a much lower latency as compared to the other two.

4.6 Variation in data item repository size

Finally, we consider the effect of increasing the data item repository size for a given value of car density and storage per car. Figure 9 depicts the latency performance of the replication schemes as a function of T when $N = 50$ and $\alpha = 15$ giving $S_T = 750$. For the same total storage, as the data item repository size increases, lesser replicas are assigned per data item, resulting in an increase in the overall availability latency. With $w = -0.5$, all the schemes show an almost linear increase in the latency as T increases. The increase in the latency becomes less significant with increasing skewness because enough replicas can still be assigned to the popular data items which have a major contribution to the aggregate availability latency. With $w = -1.5$ and $w = -2$, the random scheme shows a step function like behavior because increase in data item repository size from 100 to 400 first causes a reduction in the replicas for the popular data items. However, further increase in T from 400 to 700 does not change the number of replicas for the popular data items causing minimal change in the aggregate availability latency values.



(a) $w = -0.5$

(b) $w = -0.5$

(c) $w = -1.0$

(d) $w = -1.0$

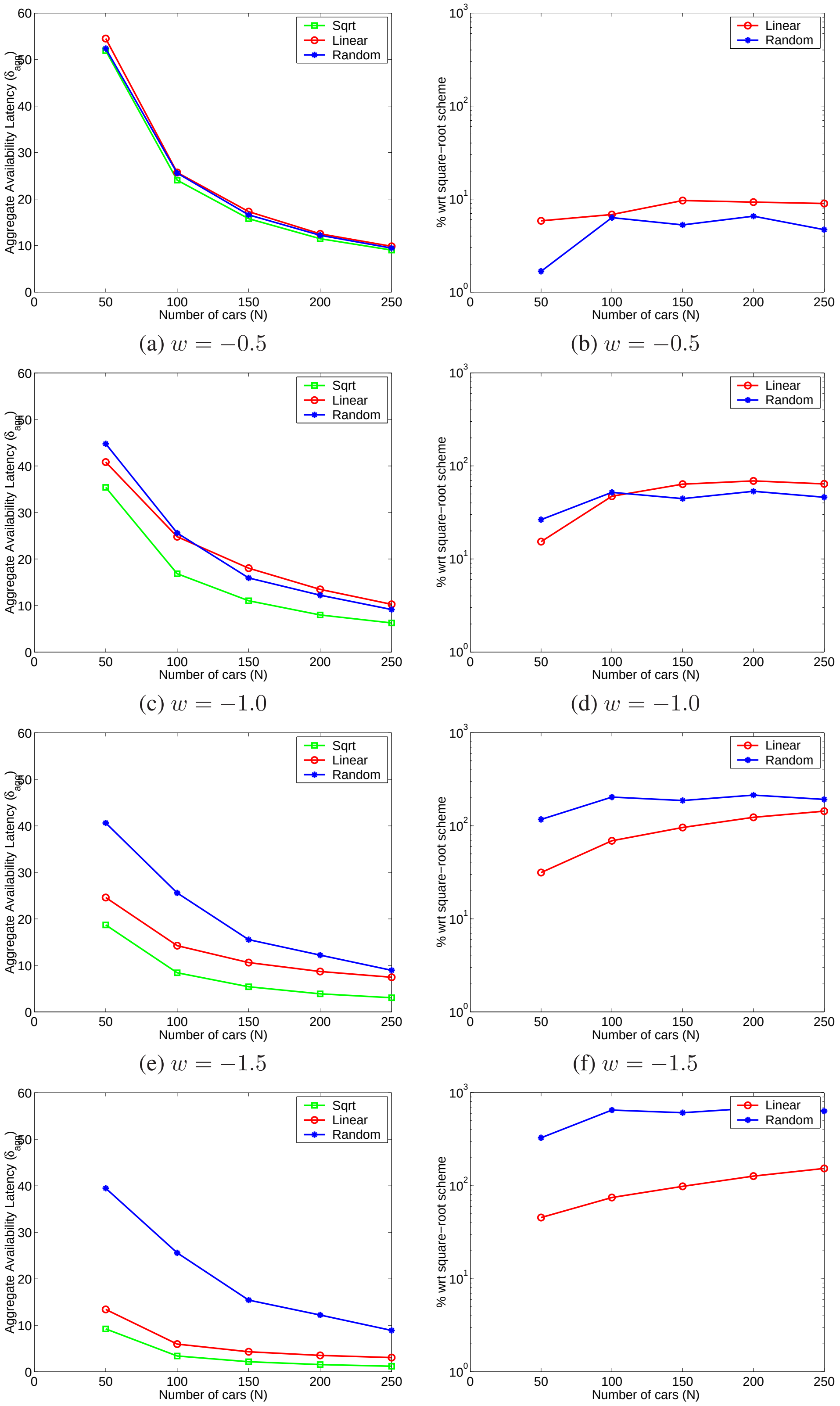
(e) $w = -1.5$

(f) $w = -1.5$

(g) $w = -2.0$

(h) $w = -2.0$

Figure 6: Scale-up experiments where the total storage to the data item repository size is held constant at $\frac{S_T}{T} = \frac{3000}{600}$. The number of cars and the storage per car are varied to realize $S_T = 3000$.



(a) $w = -0.5$

(b) $w = -0.5$

(c) $w = -1.0$

(d) $w = -1.0$

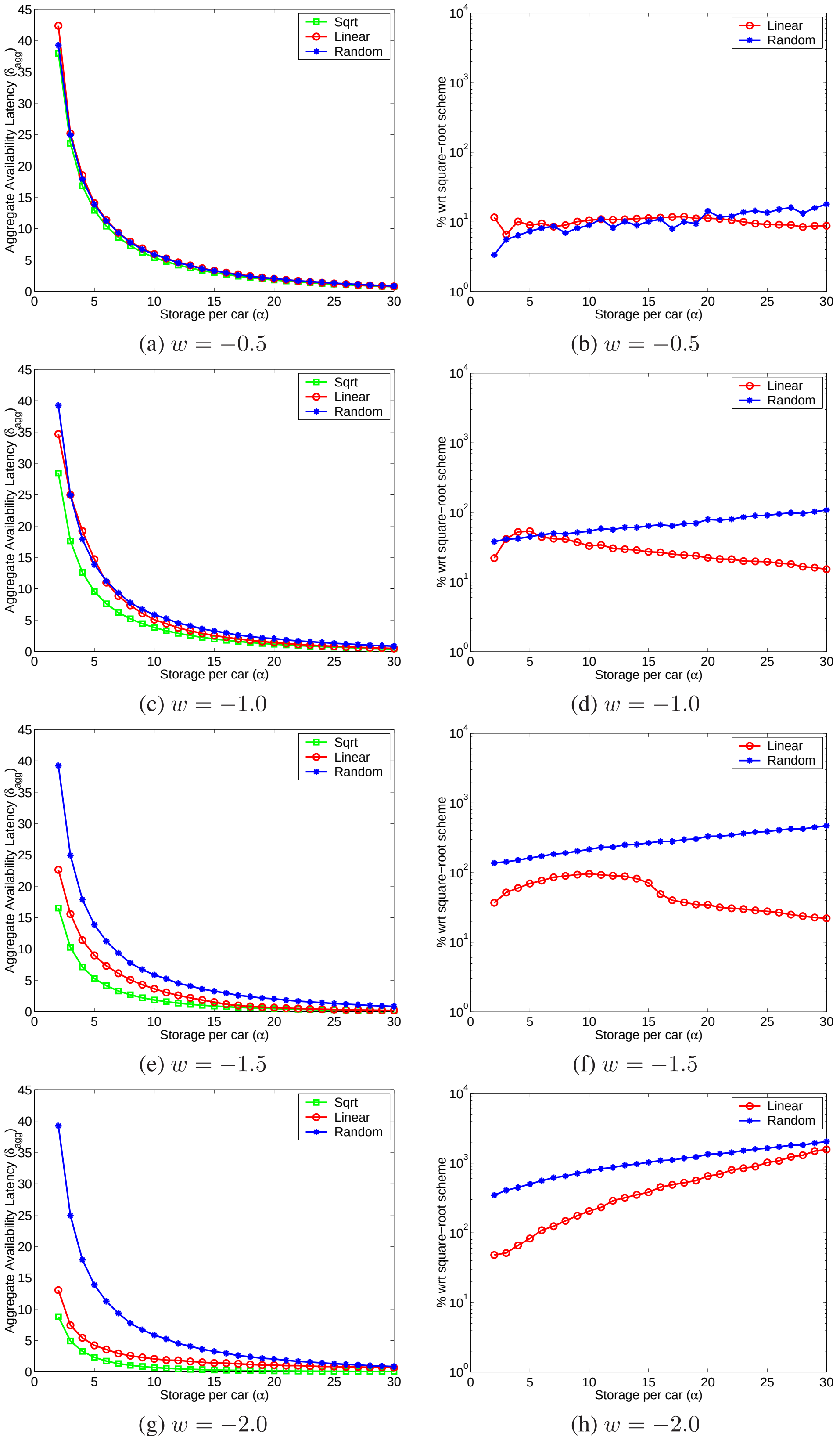
(e) $w = -1.5$

(f) $w = -1.5$

(g) $w = -2.0$

(h) $w = -2.0$

Figure 7: Aggregate availability latency for the three replication schemes as a function of the car density when the storage per car is fixed at 3. Here $T = 100$.



(a) $w = -0.5$

(b) $w = -0.5$

(c) $w = -1.0$

(d) $w = -1.0$

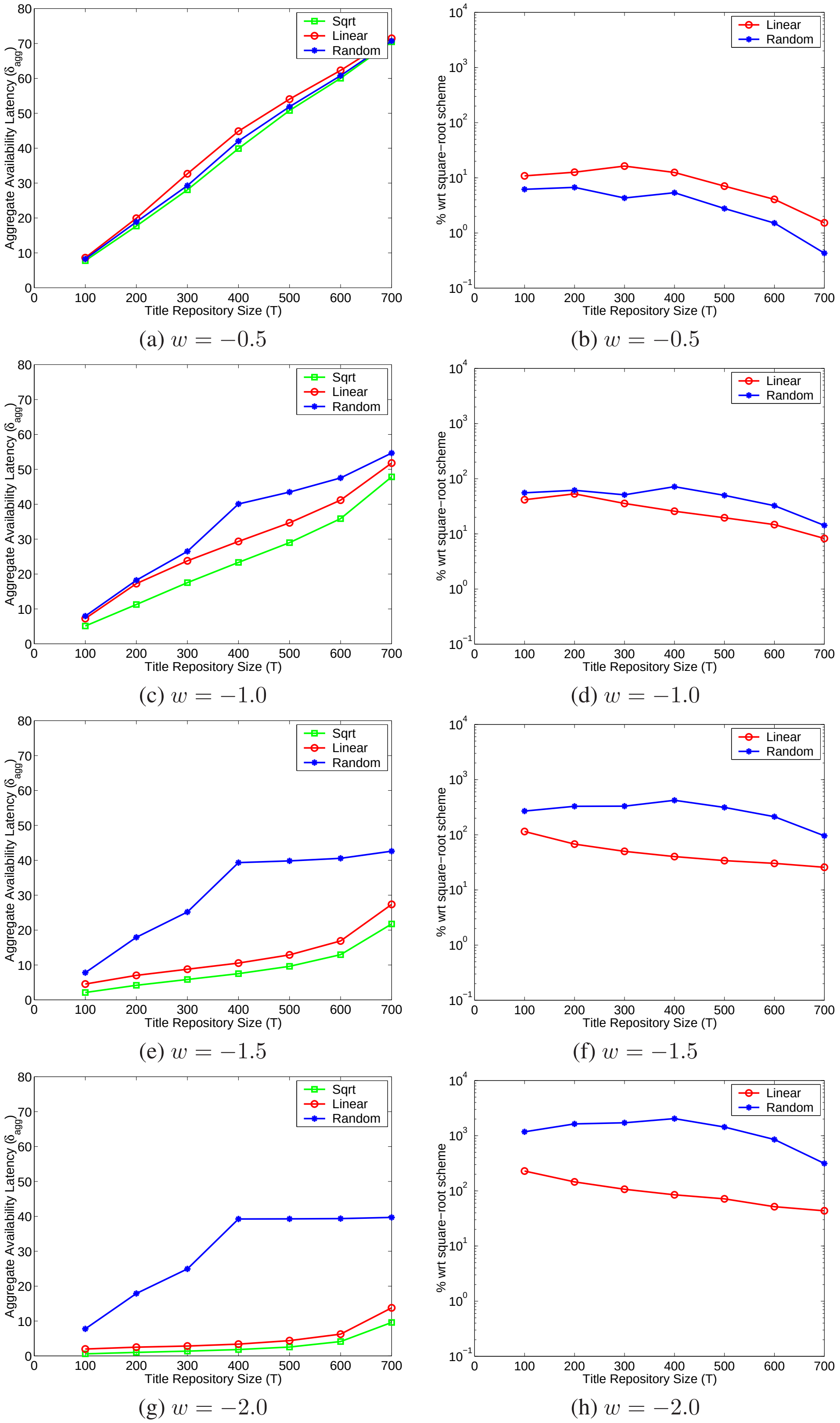
(e) $w = -1.5$

(f) $w = -1.5$

(g) $w = -2.0$

(h) $w = -2.0$

Figure 8: Aggregate availability latency for the three replication schemes as a function of the storage per car when data item repository size is 50 and car density is 50.



(a) $w = -0.5$

(b) $w = -0.5$

(c) $w = -1.0$

(d) $w = -1.0$

(e) $w = -1.5$

(f) $w = -1.5$

(g) $w = -2.0$

(h) $w = -2.0$

Figure 9: Aggregate availability latency for the three replication schemes as a function of the data item repository size for a car density of 50 and storage per car of 15.

5 Data items with small size and short client trip duration

The analysis and simulation results presented so far assumed that once a request is issued at a client, it is willing to wait as long as it takes for its request to be satisfied. In other words, the client trip duration was assumed to be unbounded. For the specific mobility model under consideration, namely 2D random walk on a torus, the maximum latency experienced by a client is bounded [1] as long as at least one replica of every item is present in the system at all times. However, in more practical scenarios, the client may have a certain maximum time it is willing to wait for request resolution. This is captured by considering a finite trip duration, γ, for the client. The availability latency for item i, δ_i, can be any value between 0 and $\gamma - 1$. If the client's request is not satisfied, we set $\delta_i = \gamma$ indicating that the client's request for item i was not satisfied.

5.1 Analysis

As before with Section 4.1, here, we derive expressions for average availability latency of a data item as a function of its replicas for a short client trip duration. Below, we present approximations in the case of low and high density of replicas.

5.1.1 Sparse Approximation

Recall that latency in the case of a 2D-random walk on a torus can be modeled as an exponential distribution as:

$$P(\delta_i > t) = \lambda \cdot \exp{(-\lambda \cdot t)} \tag{32}$$

where $\lambda = \frac{r_i}{c \cdot G \cdot \log G}$. The average availability latency with finite trip duration γ is then given by,

$$\overline{\delta_i} = \int_0^{\gamma} x \cdot \lambda \cdot \exp{(-\lambda \cdot t)} dx + \int_{\gamma}^{\infty} \gamma \cdot \lambda \cdot \exp{(-\lambda \cdot t)} dx \tag{33}$$

Hence, we get

$$\overline{\delta_i} = \frac{c \cdot G \cdot \log G}{r_i} \cdot [1 - \exp{(\frac{-\gamma \cdot r_i}{c \cdot G \cdot \log G})}] \tag{34}$$

5.1.2 Dense Approximation

Recall that as defined in Section 4.1.2, A_k is the event that a data item i is encountered by the client for the *first* time in the k^{th} cell and $P(A_k)$ is the probability that event A_k occurs. Also, p_k is the probability of encountering data item i in the k^{th} cell, given that it was not encountered in the previous $k-1$ cells. Then,

$$p_k = 1 - \left(1 - \frac{1}{G - k + 1}\right)^{r_i} ; 2 \leq k \leq \gamma \tag{35}$$

Also, we rewrite $P(A_k)$ incorporating the finite trip duration constraint as,

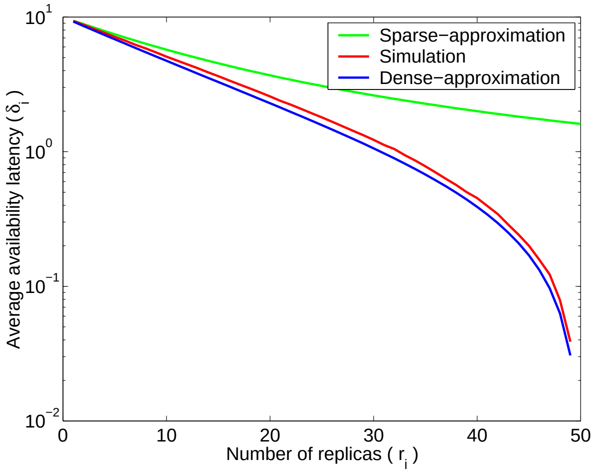


Figure 10: Average availability latency for a data item as a function of its replicas for a finite trip duration γ of 10. The simulation curves are plotted along with the sparse and dense approximations for finite trip duration for a 10×10 torus, when the number of cars is set to 50.

$$P(A_k) = p_k \prod_{j=1}^{k-1} (1 - p_j) \; ; 2 \leq k \leq \gamma \tag{36}$$

Let $P(A_{\gamma+1})$ denote the probability of not encountering the data item i during the entire trip duration γ. Hence,

$$P(A_{\gamma+1}) = \prod_{j=1}^{\gamma} (1 - p_j) \tag{37}$$

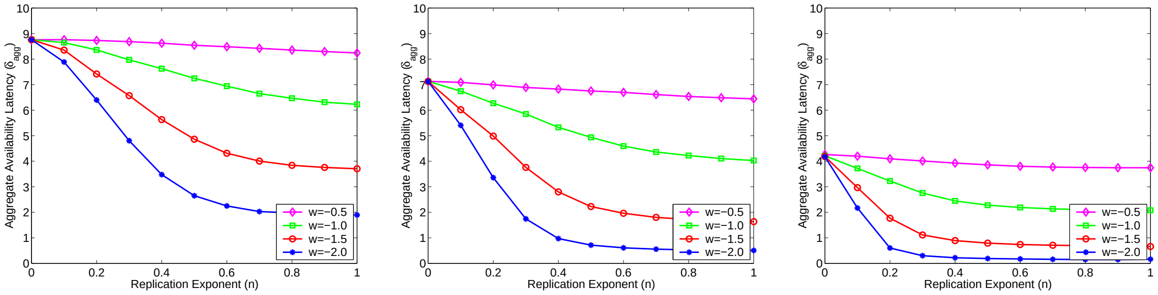
Then, the availability latency (δ_i) for data item i is given by,

$$\overline{\delta_i} = \sum_{k=1}^{\gamma+1} (k-1) P(A_k) \tag{38}$$

Figure 10 shows that the above approximations for low and high density of replicas matches the latency obtained by simulations. In this case, the dense approximation is also valid for a low density of replicas because the finite trip duration γ limits the maximum value of the availability latency. For a low density of replicas in most cases the latency will be higher than γ and hence it will be bounded by γ. For a higher replica density, the value of γ is not as significant since the latency for that item will be much lower than γ.

5.2 Simulation Results

Figure 11 depicts the latency performance for different replication schemes when storage per car is increased from 4 to 25 slots when the trip duration is set as 10. When storage per car is low, $\alpha = 4$, this represents a constrained storage scenario. The linear scheme that allocates more replicas to the popular data items shows superior performance as compared to the square-root scheme. This is because in such scenarios the replicas per data item is small, hence, only data items having a larger number of replicas will provide a latency less than γ. Since the popular data items are



(a) $\alpha = 4$ (b) $\alpha = 10$ (c) $\alpha = 25$

Figure 11: Aggregate availability latency for different replication strategies for a 10×10 torus for a finite trip duration of 10 when $T = 100$ and $N = 50$. Figures (a), (b), and (c) depict three different storage values per car: $\{4,10,25\}$.

the ones that requested more often allocating more replicas for these items lowers the aggregate availability latency. Contrast this scenario with the case of unbounded trip duration where a square-root replication scheme always provided the minimum latency (see Figure 5).

The optimal scheme here is a super-linear one which allocates most of the replicas to the first few popular data items after satisfying the constraint that at least one copy of every data item must be present in the network. For a highly skewed scenario, $w = -2$, allocating all the remaining storage for the most popular data item minimizes the latency. This is because most of the popularity weight is associated with the most popular data item which is requested very often.

As the storage per car is increased further the curves start becoming flatter and at $\alpha = 25$, see Figure 11(c), a replication scheme characterized by an exponent in the range, $0.3 \leq n \leq 1.0$, shows near optimal performance. This is because the storage is abundant enough for all these schemes to allocate a copy of the popular data items to every car bringing the latency for these items to 0. The difference in the replicas allocated for the lesser popular data items has minimal effect on the aggregate availability latency on account of their lower request rate. Recall, the frequency of access to the data items follows a zipf distribution that depicts a heavy-tailed behavior.

6 Data items with large size and long client trip duration

All the results presented so far considered a homogeneous repository of data items with a display time (Δ) of one. In this section, we consider data items with a higher Δ. In such cases, the latency encountered by a client's request is given by the earliest time when a contiguous block of Δ cells containing at least one replica of the requested item is encountered. Here, we consider scenarios with long client trip durations and present a curve-fit based approximation that captures the relation between the average availability latency and the number of data item replicas. Hence, we present the optimal replication exponent that minimizes aggregate availability latency in the case of large data items and long client trip durations.

Figure 12 depicts the average availability latency for a data item with a higher display time ($\Delta = \{2, 3, 4, 5\}$) as a function of the replicas for that item. For a given data item replica density, the latency increases with the display time. As expected the latency reduces with increase in the replicas. A simple curve-fit on the latency curves for all the Δ values yields a close-match with the expression of the form

$$\overline{\delta_i} = \frac{C}{r_i^{\sigma}} \tag{39}$$

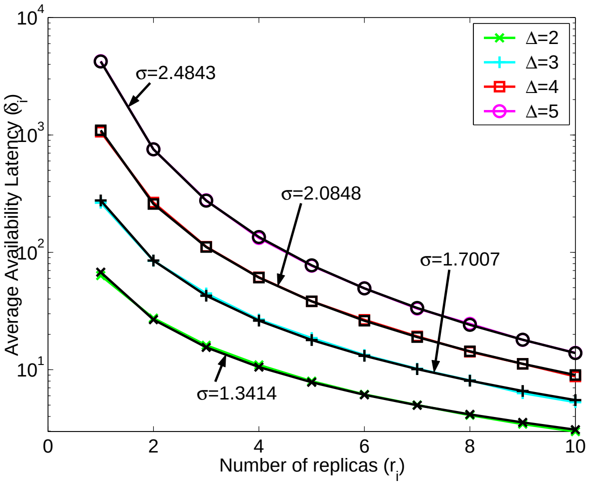


Figure 12: Average availability latency for a data item as a function of its replicas for different data item display times for a 10×10 torus. The latency is given by $\frac{C}{r_i^\sigma}$ where the exponent σ increases with data item display time.

Δ	σ	$n = \frac{1}{\sigma+1}$
2	1.3414	0.4271
3	1.7007	0.3703
4	2.0848	0.3242
5	2.4843	0.287

Table 3: Approximate optimal replication exponents for data items with higher data item display times.

where σ represents the exponent for a given data item display time and C is a constant that is a function of the size of the torus. Note that the value of σ for data items with a display time of one is one (see Equation 7). The values of σ increase with data item display time. This indicates that an increase in the replicas provides a larger drop in the latency for a data item with a higher display time. Intuitively, encountering a replica in a contiguous block of Δ cells becomes more and more difficult as Δ increases. Hence, an increase in the replica density provides a faster reduction in the latency for the higher Δ items. This is captured by the increasing value of σ with Δ.

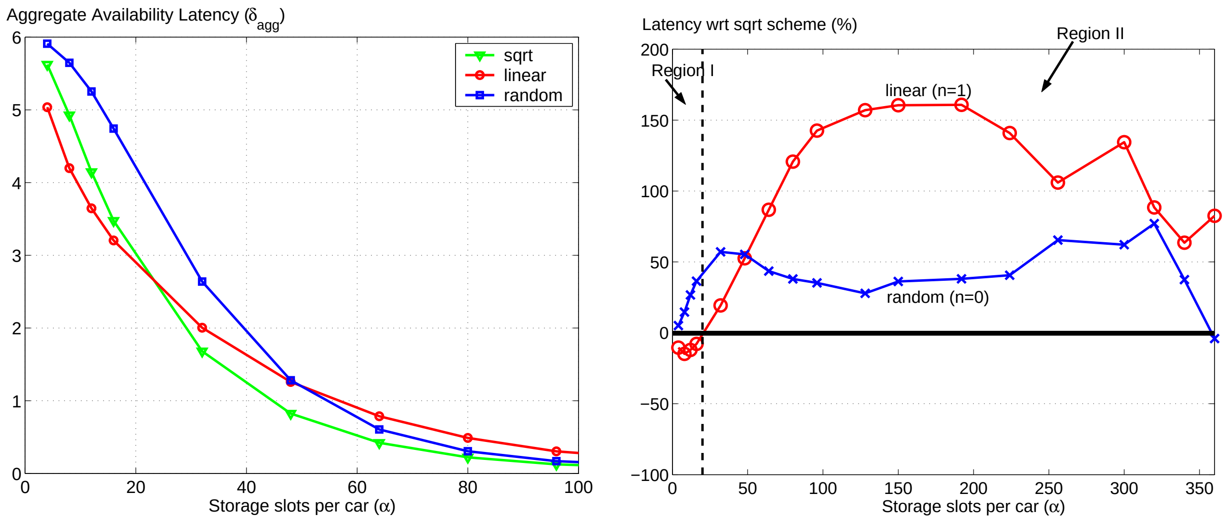
The specific formulation of Equation 39 has special significance. Equation 39 can be plugged in directly into the optimization formulation in Section 4.1.1 to determine the optimum replication scheme that minimizes the availability latency in case of data items with higher display times.

Corollary 1. *In case of a sparse density of vehicles, with a repository of data items with higher display times ($\Delta > 1$), replication exponent n, such that $n < 0.5$, minimize the aggregate availability latency.*

Proof. Following a similar procedure as the proof listed in Theorem 1, we obtain the optimal replication exponents for the σ values capturing higher data item display times in Figure 12. Table 3 lists the display times and the corresponding approximate optimal exponent values. □

7 Data items with large size and short client trip duration

In this section, we consider scenarios with short client trip durations with data items having higher display times. This implies that the client is only willing to wait for a short period for its request to be satisfied (denoted by γ). Otherwise



(a) δ_{agg}

(b) % Comparison wrt sqrt scheme (Γ)

Figure 13: Figure 13(a) shows δ_{agg} of the sqrt, linear and random replication schemes versus α for $\Delta = 4$ and $N = 200$. Figure 13(b) shows the % comparison of the linear and random schemes wrt the sqrt scheme for this scenario. Region I and Region II, respectively, indicate the parameter space where $n = 1$ and $n = 0.5$ perform the best.

the request is tagged with a latency equal to γ.

As with the previous simulations, we assume that the cars employ a 2D random walk based mobility model. Here, we set the client trip duration, γ, as 6, $N = 200$, and $T = 100$. We simulated a skewed distribution of access to the T data items using a zipf distribution with a mean of 0.27. The distribution is shown to correspond to sale of movie theater tickets in the United States [9].

Initially, all cars are distributed across the cells of the map as per the steady state distribution which is determined by a random number generator initialized with a seed. Depending on the particular replication technique, the replicas for each data item are calculated using Equation 2 and then distributed across the car. A car only contains a maximum of one replica for a particular data item. The allocation of data item replicas across the cars is uniform. At each step, depending on the current car location, it moves to one of its adjoining cell (including itself) as governed by the mobility model. Another seed determines the choice of which cell a car moves to. Since $\gamma = 6$, each car performs six transitions according to the mobility model. We performed the comparisons for several different data item distribution seeds starting from the same initial car positions. Next, we varied the initial car positions by changing the initial seed. Specifically, we chose 50 different initial seeds and for each of these we used 50 seeds that decide the distribution of the data item replicas among the cars. Thus, each point in all the presented results is an average of 2500 simulations.

Below is an overview of the key lessons learned from these experiments with higher data item display times and a short trip duration.

(a) The optimal value of n varies as a function of the scarcity of the network storage (b) When storage is scarce, the optimal aggregate availability latency is realized by using a higher value of n. (c) Even a random scheme with $n = 0$ is good enough when storage is abundant relative to the repository size.

When storage is extremely scarce, with larger data item sizes ($\Delta > 1$), linear ($n = 1$) scheme provides the best performance. This is because it allocates more replicas for the popular data items at the cost of assigning very few for the remaining data items. In this case, the contribution to δ_{agg} is a function of the δ for the more popular data items since for the less popular data items there will be insufficient replicas to reduce their δ. On the other hand, since the random scheme is blind to the data item access frequencies, on an average, it assigns equal number of replicas for each data item thereby providing the worst performance.

The square root ($n = 0.5$) scheme assigns fewer replicas for the popular data items than the linear scheme. As we increase the amount of storage, there is a cut-off point along the storage axis, where allocating more replicas for the popular data items provides negligible improvement in δ_{agg}. It is beyond this point that the square root scheme starts outperforming the linear scheme. This is because the square root scheme can use the extra storage savings for allocating replicas for the less popular data items thereby reducing their δ.

To illustrate, Figure 13 shows the variation of δ_{agg} as a function of α for $\Delta = 4$. Since δ_{agg} is a function of the value of n, hence, here we denote it as $\delta_{agg}(n = i)$. For Figure 13(b), the y-axis represents the percentage comparison of the linear ($n = 1$) and the random ($n = 0$) schemes with respect to the square root ($n = 0.5$) scheme calculated as, $\Gamma = \left(\frac{\delta_{agg}(n=i)-\delta_{agg}(n=0.5)}{\delta_{agg}(n=0.5)}\right) \times 100$; where $i = \{0, 1\}$.

Figure 13(b) shows two distinct regions in which the schemes with $n = 0.5$ and $n = 1$ perform well under certain parameter settings within the design space. For $\alpha <= 20$, the linear scheme ($n = 1$) performs the best. For $20 <= \alpha <= 360$, the square root scheme ($n = 0.5$) performs the best. Beyond this value, even a random scheme ($n = 0$) provides a competitive latency performance.

With $\Delta = x$ and $T = y$, the value of α needed to replicate the entire database on each car is $\alpha_{db} = x \cdot y$. At a certain storage threshold (earlier than α_{db}), the random scheme assigns enough replicas to the popular data items to bring their δ down. In this case, all the data items have the same number of replicas, thereby producing a low δ for every data item. Hence, from this point onward, even a random scheme provides adequate performance. However, this point requires sufficient storage per car and hence a random scheme may be appropriate only for over-provisioned scenarios. As illustrated in Figure 13(b) with $N = 200$, $T = 100$, and $\Delta = 4$, the storage threshold is around 360 slots per car. For $\Delta = 5$, and 6, this threshold is approximately 450 and 540, respectively. These are loose upper bounds.

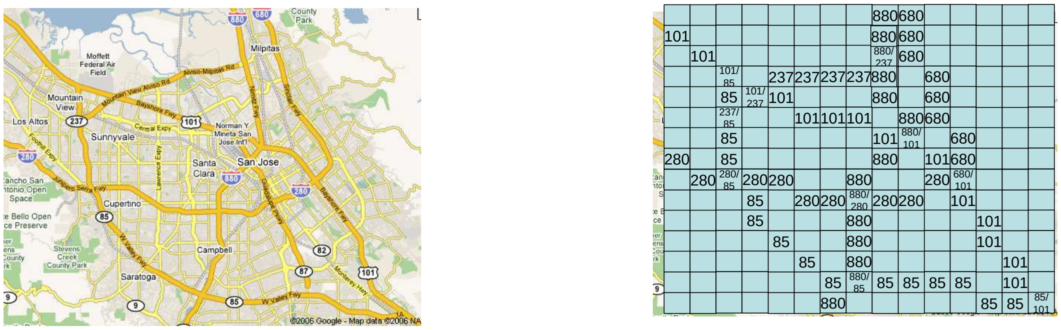
7.0.1 Aggregate availability latency as a function of car density (N)

Car density, which in turn affects the available storage in the system, has a major impact on the performance of δ_{agg} for all the schemes. With the decrease in the car density to $N = 100$, the number of replicas allocated by the schemes is reduced thereby giving comparatively larger values of δ_{agg} across the same storage axis. As α is increased, the drop in the δ_{agg} curves for all the schemes is not as steep as seen in the case with $N = 200$. Again, this is because the number of replicas is not increasing at such a high rate. The storage is reduced by an order of 2, hence a higher value of α is needed to produce the same drop in δ_{agg} as was seen in the case with $N = 200$ cars. This is observed across all values of Δ.

For all experiments, we also calculated the standard deviations (SD) and the standard error of the mean (SEM). The 95% confidence intervals determined as $1.96 * SEM$ are quite small and accordingly the curves are quite smooth. However, the standard deviation is quite large, especially for the cases when δ_{agg} is low for high values of Δ and α. This is because a low latency requires the data item to be present in every cell along the journey depending on the value of Δ. As Δ increases, it becomes increasingly difficult to meet this condition thereby showing a high variance in δ_{agg}. The large SD value is an empirical observation about the nature of the random process.

8 Evaluation with a Map of San Francisco

In this section, we describe the performance of the various replication schemes when the vehicular movements are dictated by an underlying map of the San Francisco Bay Area. Figure 14(a) depicts a section of the San Francisco Bay area with the major freeways and their intersections. We superimpose a 2D-grid on this map and the individual cells are labelled with the respective freeway id that they cover as shown in Figure 14(b). This 2D-grid serves to capture the underlying map at a coarse granularity. Most of the probability mass is concentrated on the cells that represent the



								880	680					
101								880	680					
	101							880/ 237	680					
		101/ 85		237	237	237	237	880		680				
		85	101/ 237	101				880		680				
		237/ 85			101	101	101		880	680				
		85						101	880/ 101		680			
280		85						880		101	680			
	280	280/ 85	280	280			880			280	680/ 101			
			85		280	280	880/ 280	280	280		101			
			85				880					101		
				85			880					101		
					85		880						101	
						85	880/ 85	85	85	85	85		101	
						880						85	85	85/ 101

(a) (b)

Figure 14: A map of the San Francisco Bay Area obtained from http://maps.google.com is shown in Figure 14(a). Figure 14(b) superimposes a 15 × 15 grid on this map and labels the cells appropriately with the freeway IDs that they overlap with.

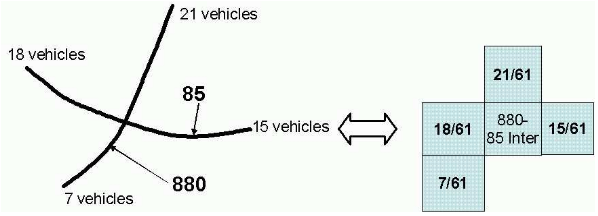
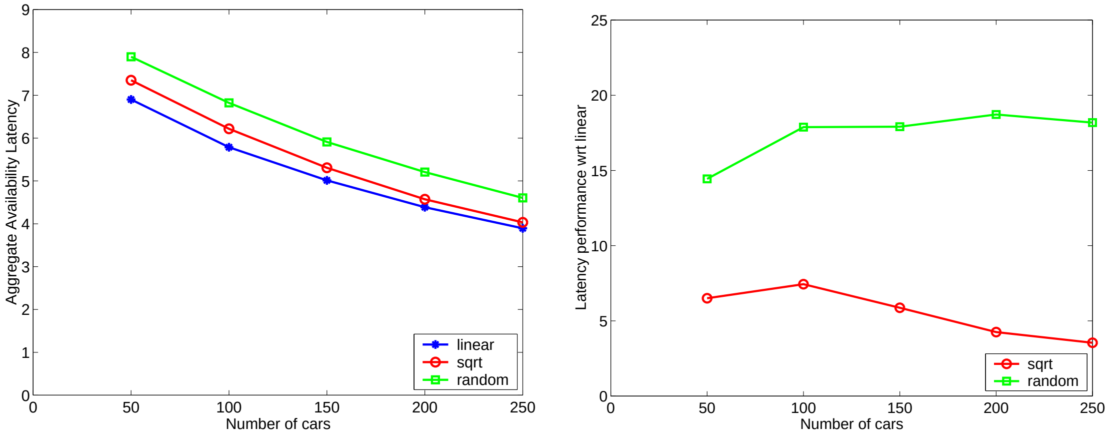


Figure 15: The intersection between freeways 880 and 85 is captured in the figure along with the equivalent probability transitions in the Markov model based on data obtained from Caltrans regarding the vehicular densities.

major freeways. The non-labelled cells have equal transition probabilities to each of its neighboring eight cells.

The outgoing transition probabilities at a cell that represents an intersection between two freeways are calculated as follows. As an example, consider the intersection of the freeways 880 and 85 as shown in Figure 15. We obtained the traffic density seen on the freeways before and after the intersection from Caltrans data provided by the California department of transportation [30]. The website allowed real time gathering of vehicle traffic data. We considered a time window between 7-8 pm for a particular week and averaged the vehicular density seen during this period. The day-to-day statistics were quite similar, here, we show an example of how the actual data was converted into the probability transition values that formed the basis of the Markov mobility model. Similar calculations were employed to populate the entire transition probability matrix. Finally, we converted the 15 × 15 grid into a torus by allowing cars at the boundaries to appear at the opposite ends with equal transition probabilities.

The transition matrix was used to generate the car movements. We provide a notion of directionality to the car movements by ensuring that the next step for a cars movement takes into account both the current cell as well as the previous cell which a car traversed. This is done by storing both the cell ids as part of the state of the Markov chain. Consequently, the flip-flop movements of the cars is avoided thereby ensuring that car movements are constrained by the underlying freeway structure of the map and are not entirely random. We used these car movements to investigate the relative performance of the various replication schemes under such a scenario.



(a) (b)

Figure 16: Performance of various replication schemes as a function of car density when $T = 25$, $\alpha = 2$, and $\gamma = 10$. Figure (b) shows the performance wrt the linear scheme.

8.1 Results with replication schemes

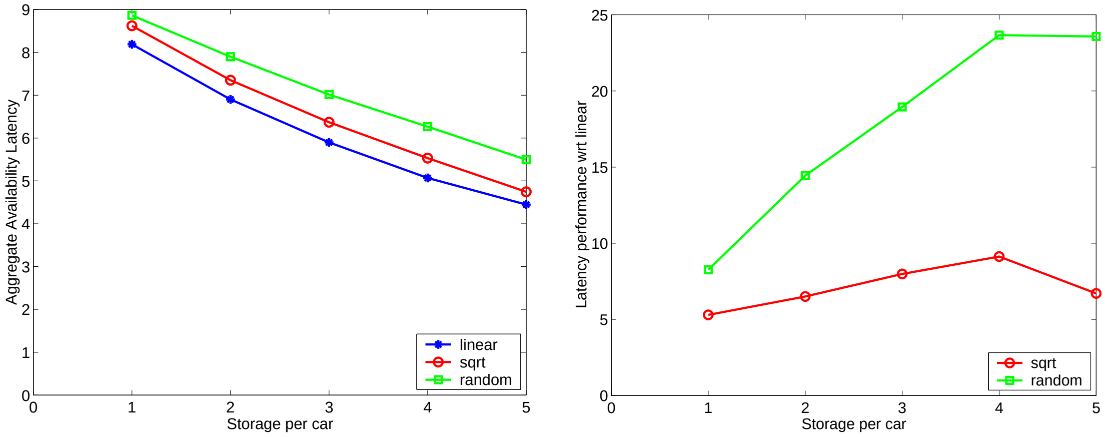
In this section, we present some representative results for the various replication schemes obtained by employing the Markov mobility model previously derived from a map of the San Francisco Bay area. Below we describe the three sets of experiments used in our evaluation for comparison of the linear, square-root, and random replication schemes. As before, requests are issued, one at a time at each time-step at vehicles in a round-robin manner, as per a Zipf distribution with a mean of 0.27. The three sets of experiments were:

- For data item repository size T set as 25, client trip duration, γ, set as 10, storage per car, α, set as 2, the latency performance with the various replication schemes is studied as a function of increasing car density N (see Figure 16).
- For data item repository size T set as 25, client trip duration, γ, set as 10, car density, N, set as 50, the latency performance with the various replication schemes is studied as a function of increasing storage per car α (see Figure 17).
- For car density, N, set as 50, client trip duration, γ, set as 10, storage per car, α, set as 2, the latency performance with the various replication schemes is studied as a function of increasing data item repository size T (see Figure 18).

In all cases, the main conclusion is that the linear replication scheme shows superior performance as seen in Section for the case with data item size=1 and finite client trip duration. The trends seen with this model are similar to those seen with a uniform Markov mobility model with equal transition probabilities. This is because the map for the former causes some cells to be visited more than others, however, the movements of the vehicles still remain essentially Markovian. This result suggests that the uniform probability transition matrix based Markov model may be a good indicator of the performance that may be seen with a model derived from real maps.

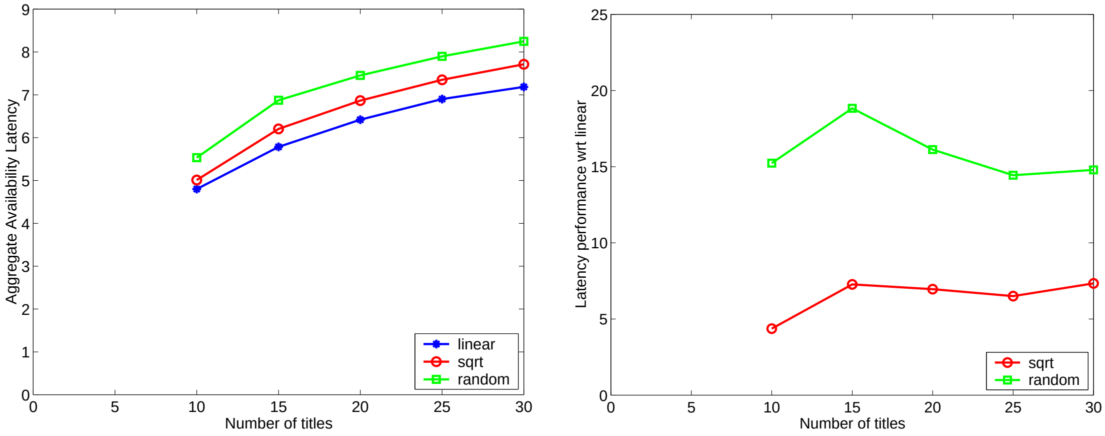
9 Evaluation with real movement traces

In this section, we evaluate the latency performance of the static replication schemes using traces obtained from a bus-based DTN test-bed called UMassDieselNet [7]. First, we briefly describe the test-bed and present some properties



(a) (b)

Figure 17: Performance of various replication schemes as a function of storage per car when $T = 25$, $N = 50$, and $\gamma = 10$. Figure (b) shows the performance wrt the linear scheme.



(a) (b)

Figure 18: Performance of various replication schemes as a function of data item repository size when $N = 50$, $\alpha = 2$, and $\gamma = 10$. Figure (b) shows the performance wrt the linear scheme.

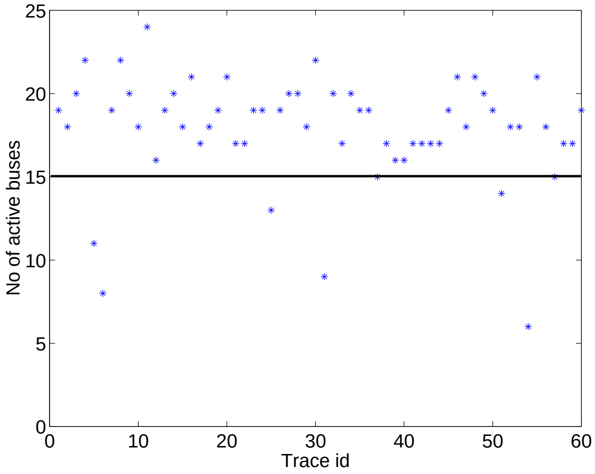


Figure 19: The number of active buses for each trace representing the bus encounters for each day of a 60-day period. The buses operated from 7am to 5pm.

of the mobility model followed by the buses. Then, we describe the details of the experimental set-up and the results comparing the square-root, linear, and random replication schemes under different parameter settings using these traces.

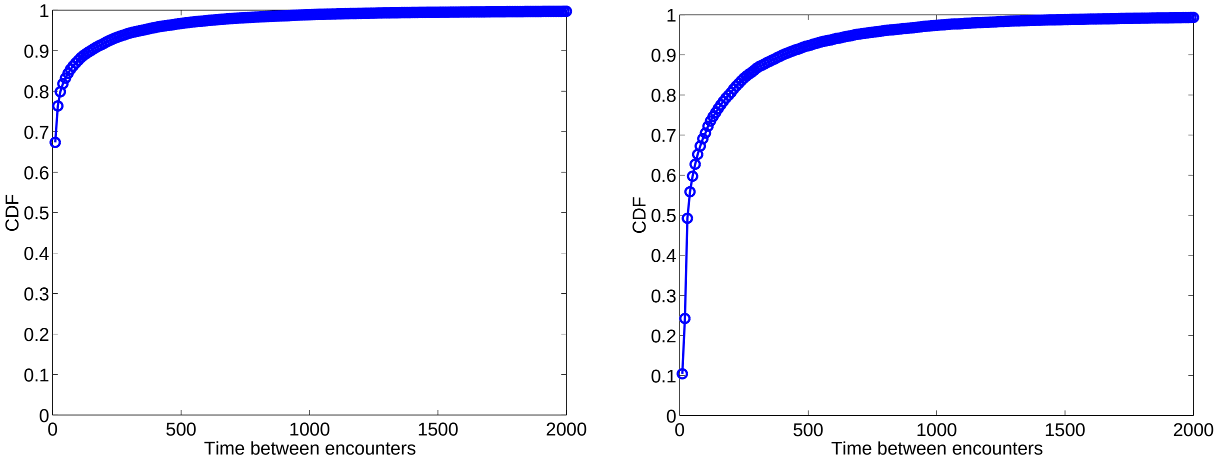
9.1 UMassDieselNet Traces

In this section, we briefly describe the details of the UMassDieselNet test-bed and present some properties of the mobility model that characterizes the movement of the vehicles that are part of the test-bed. The UMassDieselNet network operates daily around the UMass campus and the surrounding county. It comprises of 30 buses equipped with a Linux based computer coupled with a IEEE 802.11b wireless interface that permits ad-hoc communication between the buses when they are in radio range. An IEEE 802.11b access point is also connected to the brick computer that allows DHCP access to passengers within the bus. The traces are available for a period of 60 days, the logs describe every encounter between every pair of buses that occurred during the day. These traces do not contain the logs for accesses made between the passengers and the access point within the bus. The identity of the buses involved in the encounter, the time of encounter and amount of data transmitted during the encounter are logged in the trace files. Certain buses had long routes while others had short ones. Unfortunately, due to technical difficulties, the GPS device on the buses were unable to provide details about the bus locations during the encounter.

Figure 19 shows the number of buses that were active on each day of the 60 day period during which the traces were collected. We only considered traces where the number of active buses was greater than 15. This accounted for 52 traces. In general, the traces indicated a sparse density of buses where there was a high degree of locality in the encounters. In other words, if 2 buses encounter each other at the beginning then they will continue to encounter each other more frequently than other buses. This is captured in Figure 20 where we set a minimum separation time between two consecutive encounters of the same pair of buses to consider it a different encounter. In Figure 20(b), the separation time is set as 20 seconds as compared to 0 seconds set for Figure 20(a).

9.2 Experimental Set-up

In this section, we describe the details of the simulation set-up used for evaluation of the replication schemes employing the UMassDieselNet traces. Each trace represents the movements of buses during that particular day. There is no



(a) Separation = 0s

(b) Separation = 20s

Figure 20: The CDF of the time between encounters averaged across all the traces for 2 different separation times 0s (Figure 20(a)) and 20s (Figure 20(b)).

correlation between trace movements across days. Hence, we process each trace at a time and then average the results observed across all the days noting that the average is indicative of the performance seen on most days. However, certain days do appear as outliers since the number of active buses differs from day-to-day.

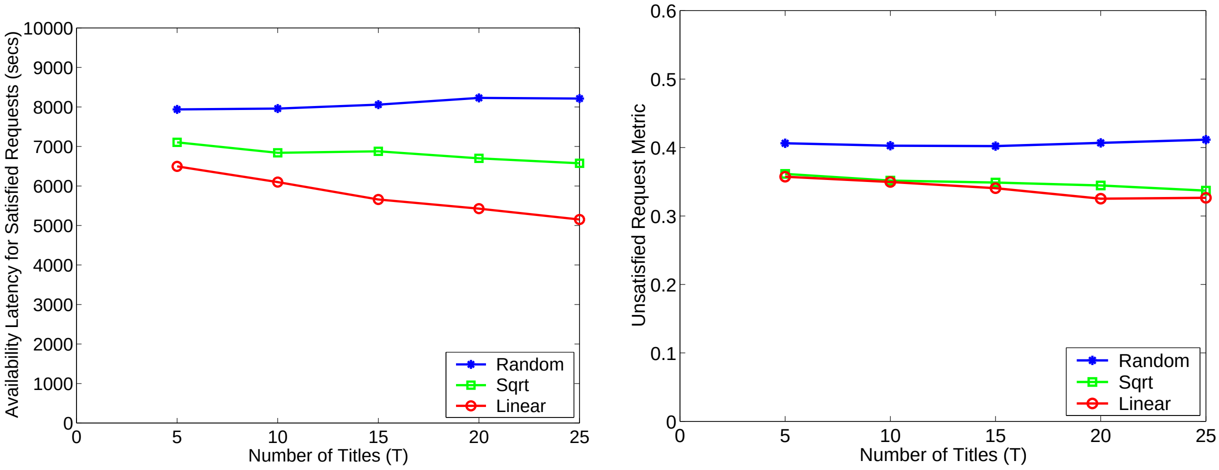
As before we consider a finite data item repository of size T. Each bus is assumed to carry α storage slots. Replicas for each data item are determined based on a replication scheme and then allocated across the buses uniformly at random. The constraint is that at least one copy of every data item must be present in the network at all times. We consider the three representative replication schemes: random, square-root, and linear and study the relative performance of the schemes.

Since the buses only operate for a finite amount of time we consider two separate metrics (i) Average availability latency for satisfied requests (ii) Normalized unsatisfied request rate. Requests for the T titles are generated as per a zipf distribution with an exponent $w = -0.73$. The duration during which the buses were active during a day is determined apriori and subject to this duration requests are issued at equal inter-arrival times. A generated request is assigned to a bus chosen uniformly at random. A request is assumed to be satisfied either if the data item requested is locally stored or another bus carrying the requested item is encountered at some point after the request is issued. Those requests that are not satisfied at the end of the day are tagged as unsatisfied requests.

9.3 Results

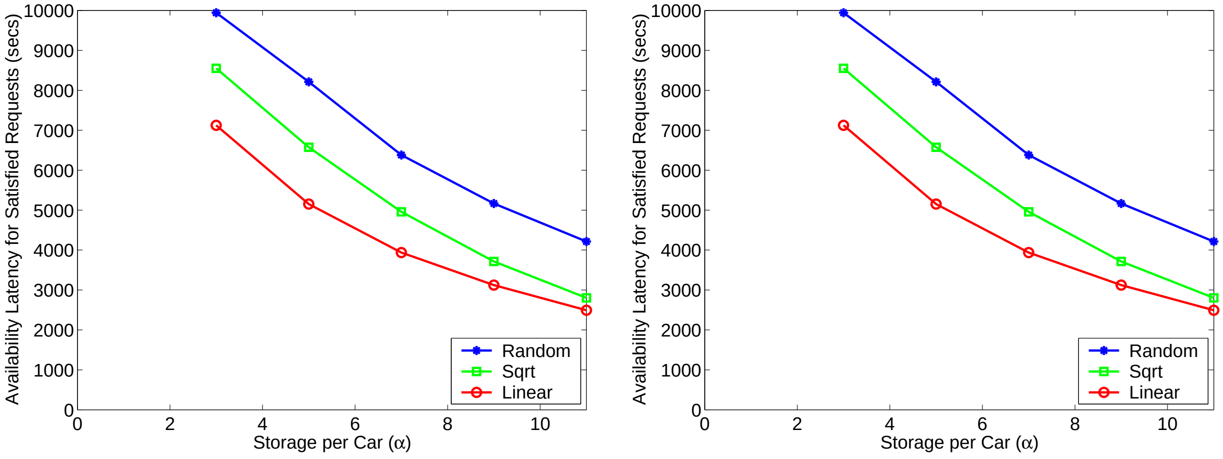
In this section, we briefly describe the main results from evaluation of the performance of the replication schemes using the UMassDieselNet traces. For the first set of experiments we vary the values of (T,α) as $\{$(5,1), (10,2), (15,3), (20,4), (25,5)$\}$, see Figure 21. The linear replication scheme provides the lowest average availability latency for satisfied requests (about $10-25\%$ better than the square-root scheme). The linear and square-root scheme show similar performance in terms of the normalized unsatisfied requests. The random scheme shows poor performance both in terms of latency as well as the normalized unsatisfied requests.

Figure 22 shows the performance of the replication schemes when the data item repository size is fixed at 25 and the storage per bus is increased. Increase in storage leads to increase in the replicas per data item, hence, as expected for all schemes, the latency and the normalized unsatisfied requests go down. The linear scheme continues to show superior performance with respect to both metrics.



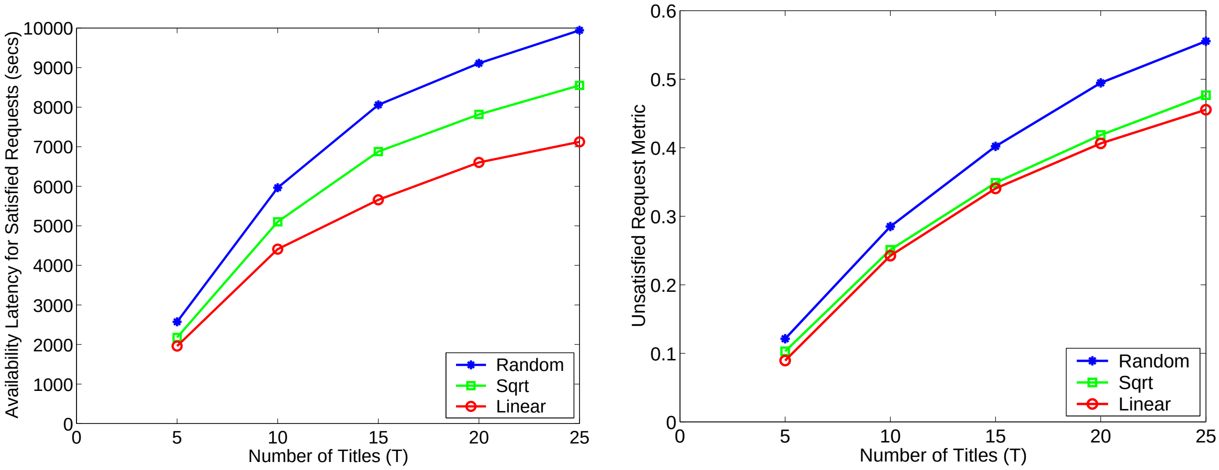
(a) (b)

Figure 21: Aggregate availability latency for satisfied requests and the aggregate unsatisfied request metric for the random, square-root, and linear replication schemes are shown in Figure 21(a) and (b) respectively. The ratio of the storage per car to the data item repository size, $\frac{\alpha}{T}$ is maintained as 1:5.



(a) (b)

Figure 22: Aggregate availability latency for satisfied requests (Figure 22(a)) and the aggregate unsatisfied request metric (Figure 22(b)) as a function of the storage per car for a data item repository size of 25.



(a) (b)

Figure 23: Aggregate availability latency for satisfied requests (Figure 23(a)) and the aggregate unsatisfied request metric (Figure 23(b)) as a function of the data item repository size when storage per car α is fixed at 3.

Figure 23 shows the performance of the replication schemes when the storage per bus is fixed at 3 and the data item repository size is increased. As the data item repository size is increased, lesser replicas are allocated per data item resulting in an increase in the latency as well as the unsatisfied requests. Initially the increase in latency is linear but slowly becomes sub-linear. Note that the data item repository size cannot be made arbitrarily large since the number of active buses is constrained. As before the linear scheme always shows the best performance.

The mobility model provided by the traces represents an extremely sparse density of buses where inherently there is a limit to the maximum amount of time for request satisfaction (namely the last encounter time on the trace). The finite trip duration in conjunction with the low density and encounter model favors a linear scheme which allocates more replicas for the popular data items. The popular data items are requested more frequently and within the finite time for request satisfaction, have a higher probability of being satisfied on account of the larger number of replicas. The square-root scheme tries to allocate replicas less aggressively to the more popular data items in favor of the less popular ones. This hurts its performance since the less popular data items have a very low probability of being satisfied. Notwithstandingly, in such scenarios, a random scheme that allocates replicas equally across the data items shows the worst performance.

We now consider an equivalent scenario with the Markov mobility and study its properties in terms of the time between encounters (see Figure 24). The aim is to capture a similar scenario as depicted by the UMassDieselNet traces (compare with Figure 20(a)). We consider a similar set-up to experimental scenario described in Figure 21. Similar to the trends seen with the traces, Figure 25 shows that the linear replication scheme outperforms the square-root and the random schemes in terms of the latency for satisfied requests. The performance in terms of the normalized unsatisfied requests is quite similar for the three schemes. These results suggest that the results obtained from the Markov mobility model may be applicable across a vast range of scenarios comprising different mobility models. Adequate adjustment to the transition probabilities of the Markov model may enable this model to suitably capture the mobility trends of other models such as Manhattan, Highway, Random way-point etc.

10 Conclusions, Discussion, and Future Research Directions

In this study, we have systematically explored how data replication impacts the user latency to desired content in a vehicular network. We present an optimization formulation that minimizes availability latency across a repository of

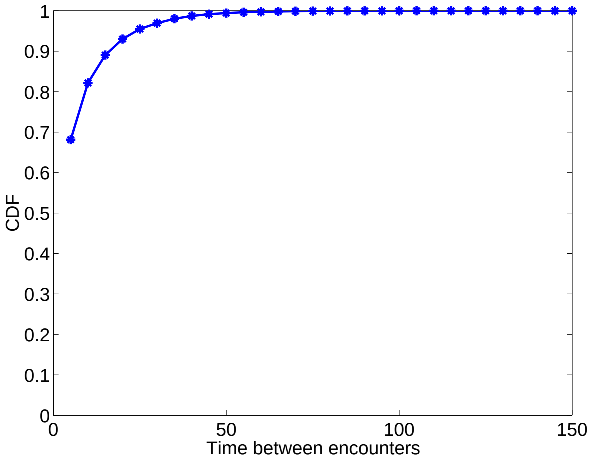
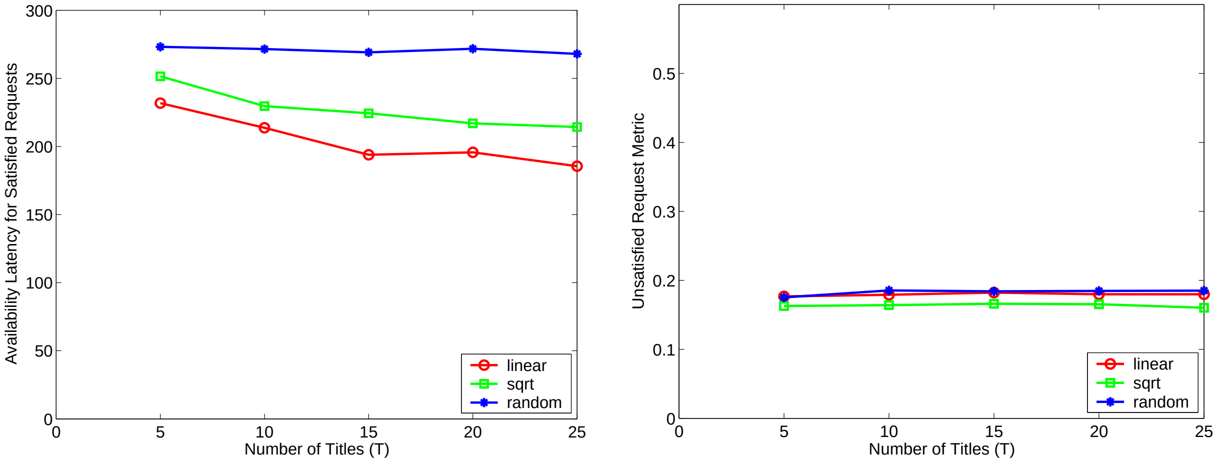


Figure 24: CDF of the time between encounters from the Markov model for a 25×25 torus with a car density of 15.



(a)

(b)

Figure 25: Aggregate availability latency for satisfied requests and the aggregate unsatisfied request metric as obtained from an equivalent scenario employing the Markov model. The ratio of the storage per car to the data item repository size, $\frac{\alpha}{T}$ is maintained as 1:5.

data items subject to storage constraints per vehicle. The solution to this optimization yielded the optimal replication scheme, characterized by an exponent n that defines the number of replicas for a data item as a function of its popularity. Using mathematical analysis and simulations, we showed how the optimal replication exponent varied as a function of the three critical factors: data item size, the client trip duration, and the total storage in the system. While the evaluation above was based on the assumption of a random walk mobility model for the vehicles, we validated a subset of our observations employing two real data sets. The first was based on a city map with freeway traffic information while the second employed traces listing encounters between buses that were part of a small test-bed.

While our study indicates that replication has a significant impact on latency performance in an AutoMata-based application, we did not directly address how a replication scheme may be realized. A promising approach is to employ a two-tier architecture [11] comprising of a low-bandwidth control plane between the base-station and vehicles, and a high-bandwidth data plane representing the ad-hoc, peer-to-peer network between vehicles. While all the data exchange takes place via the data plane, the control plane enables centralized information gathering at a suitable dispatcher. The dispatcher is aware of information such as the total number of cars, the available storage per car, the data item requests etc. On the basis of the target replication scheme to be realized, the dispatcher computes the total number of replicas to be maintained for each data item. Hence, every time a request is satisfied, depending on the current replicas of the requested item in the system, the dispatcher decides whether a client vehicle should create a new copy of that item. If such a process is followed, then after a cold-start phase where replicas are being created and deleted, the data item replica distribution will approximate the targeted one.

Second, this study can extended in terms of heterogeneity with respect to several parameters. The data item repository may have a mix of different data item sizes, the vehicles may have different amounts of available storage and different trip durations depending on their data item request preferences, and popularity distribution of the data items may change over time, especially when new items are introduced in the system. Each of these extensions adds another dimension in terms of the practicality of realizing such a system.

Third, we do not explicitly address contention and bandwidth issues within our model. Recently, Jindal *et al.* [28] have provided a contention model that can be easily incorporated in our study promising a richer evaluation.

The methodology by which we handle data items with larger sizes is slightly restrictive, in that once the download of the item has begun, the entire download should finish un-interrupted. However, another approach may be to employ suitable buffers, so that partial download of an item can be suitably resumed at a later time, as and when other vehicles with a copy of that item appear in the vicinity of the requesting client. It may be useful to compare results obtained with this approach with the existing results listed in this study.

Fourth, our study is based on static replication schemes, in that we do not explore any dynamic data re-organization schemes that reactively or pro-actively change the replica distribution depending on the requested items. In a related study, once static replication schemes have allocated replicas for the various data items, vehicles are employed as data carriers [13] to deliver data items from server vehicles carrying items to clients requesting them. The study shows that significant latency improvements can be obtained by employing these intermediate data carriers.

Finally, vehicular ad-hoc networks are an emerging area and even though this study considers audio and video content, the nature of the content and also the application domain in general might vary. Specifically, many of the results may be general enough to be applicable in a wide variety of contexts such as in intermittently connected mobile networks, mobile sensor networks, and other delay tolerant networks.

11 Acknowledgments

This research was supported in part by NSF grants numbered CNS-0435505 (NeTS NOSS), CNS-0347621 (CAREER), IIS-0307908, National Library of Medicine LM07061-01, and an unrestricted cash gift from Microsoft Research.

References

[1] D. Aldous and J. Fill. Reversible markov chains and random walks on graphs. Manuscript under preparation, http://stat.www.berkeley.edu/users/a ldous/RWG/book.html.

[2] F. Bar, W. Baer, S. Ghandeharizadeh, and F. Ordonez. Automata. Video at http://www.youtube.com/watch?v=nrRFbLRjeZM.

[3] S. Bararia, S. Ghandeharizadeh, and S. Kapadia. Evaluation of 802.11a for streaming data in ad-hoc networks. In *Proc. of the 4th Workshop on Applications and Services in Wireless Networks (ASWN)*, 2004.

[4] L. Breaslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. of IEEE Infocom*, pages 126–134, 1999.

[5] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Proc. of ACM MobiHOC*, pages 45–50, August 2000.

[6] L. Briesemeister, L. Schafers, and G. Hommel. Disseminating Messages among Highly Mobile Hosts based on Inter-Vehicle Communication. In *Proc. of IEEE Intelligent Vehicle Symposium*, pages 522–527, October 2000.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. of IEEE Infocom*, April 2006.

[8] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *Proc. of ACM SIGCOMM*, pages 177–190. ACM Press, 2002.

[9] A. Dan, D. Dias, R. Mukherjee, D. Sitaram, and R. Tewari. Buffering and Caching in Large-Scale Video Servers. In *Proc. of COMPCON*, 1995.

[10] S. Ghandeharizadeh and T. Helmi. An Evaluation of Alternative Continuous Media Replication Techniques in Wireless Peer-to-Peer Networks. In *Proc. of the 3rd International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE, in conjunction with MobiCom'03)*, September 2003.

[11] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari. PAVAN: A Policy for Content Availability in Vehicular Ad-hoc Networks. In *Proc. of the 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET)*. ACM Press, 2004.

[12] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari. Comparison of Replication Strategies for Content Availability in C2P2 networks. In *Proc. of the 6th International Conference on Mobile Data Management (MDM)*, May 2005.

[13] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari. An evaluation of availability latency in carrier-based wehicular ad-hoc networks. In *Proc. of the 5th ACM international workshop on Data engineering for wireless and mobile access*, pages 75–82, New York, NY, USA, 2006. ACM Press.

[14] S. Ghandeharizadeh and B. Krishnamachari. C2P2: A Peer-to-Peer Network for On-Demand Automobile Information Services. In *Proc. of the 1st International Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (Globe'04)*, 2004.

[15] S. Ghandeharizadeh, B. Krishnamachari, and S. Song. Placement of Continuous Media in Wireless Peer-to-Peer Networks. *IEEE Transactions on Multimedia*, April 2004.

[16] IEEE 802.16 Working Group. Emerging IEEE 802.16 WirelessMAN Standards for Broadband Wireless Access. *Intel Technology journal*, 8(3), 2004.

[17] T. Hara. Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In *Proc. of IEEE Infocom*, pages 1568–1576, 2001.

[18] T. Hara. Cooperative caching by mobile clients in push-based information systems. In *Proc. of the 11th International Conference on Information and Knowledge Management (CIKM)*, 2002.

[19] T. Hara. Replica allocation in ad hoc networks with periodic data update. In *Proc. of the 3rd International Conference on Mobile Data Management(MDM)*, pages 79–86, Washington, DC, USA, 2002. IEEE Computer Society.

[20] T. Hara. Replicating Data with Aperiodic Update in Ad Hoc Networks. In *Proc. of IASTED International Conference on Communications, Internet and Information Technology (CIIT)*, pages 242–247, 2002.

[21] T. Hara. Replica allocation methods in ad hoc networks with data update. *ACM/Kluwer Journal on Mobile Networks and Applications (MONET)*, 8(4):343–354, 2003.

[22] T. Hara. Location management of data items in mobile ad hoc networks. In *Proc. of the 2005 ACM symposium on Applied computing (SAC)*, pages 1174–1175, New York, NY, USA, 2005. ACM Press.

[23] T. Hara. Strategies for data location management in mobile ad hoc networks. In *Proc. of IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2005.

[24] T. Hara and S. Madria. Dynamic data replication using aperiodic updates in mobile ad-hoc networks. In *Proc. of International Conference on Database Systems for Advanced Applications (DASFAA)*, pages pp.869–881, Jeju Island, Korea, 2004.

[25] T. Hara, N. Murakami, and S. Nishio. Replica allocation for correlated data items in ad hoc sensor networks. *SIGMOD Rec.*, 33(1):38–43, 2004.

[26] T. Hara, Y.-H.Loh, and S.Nishio. Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks. In *Proc. of International Workshop on Mobility in Databases and Distributed Systems (MDDS)*, pages 969–973, 2003.

[27] H. Hayashi, T. Hara, and S. Nishio. A Replica Allocation Method Adapting to Topology Changes in Ad Hoc Networks. In *Proc. of International Conference on Database and Expert Systems Applications (DEXA)*, 2005.

[28] A. Jindal and K. Psounis. Performance analysis of epidemic routing under contention. In *Proc. of International Conference on Communications and Mobile Computing*, pages 539–544, New York, NY, USA, 2006. ACM Press.

[29] C. Lochert, H. Hartenstein, J. Tian, H. Fler, D. Hermann, and M. Mauve. A routing strategy for vehicular ad hoc networks in city environments. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 156–161, Columbus, OH, USA, June 2003.

[30] California Department of Transportation. Caltrans. http://www.dot.ca.gov/.

[31] F. Sailhan and V. Issarny. Cooperative caching in ad hoc networks. In *Proc. of the 4th International Conference on Mobile Data Management (MDM)*, 2003.

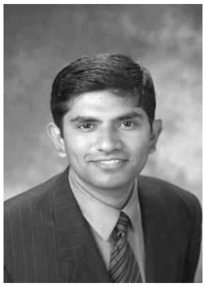
[32] A. Spyropoulous, K. Psounis, and C. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Proc. of IEEE SECON*, 2004.

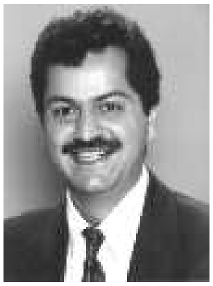
[33] W. Sun, H. Yamaguchi, and S. Kusumoto. A study on performance evaluation of real-time data transmission on vehicular ad hoc networks. In *Proc. of the 7th International Conference on Mobile Data Management (MDM)*, page 126, 2006.

[34] J. Wolf, P. Yu, and H. Shachnai. DASD Dancing: A Disk Load Blanacing Optimization Scheme for Video-on-Demand Computer. In *Proc. of ACM SIGMETRICS*, pages 157–166, May 1995.

[35] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. In *Proc. of IEEE Infocom*, 2004.

[36] G. K. Zipf. The Psychobiology of Language. *Boston: Houghton-Mifflin*, 1935.





Shyam Kapadia received his Ph.D and M.S. degrees in Computer Science from the University of Southern California in 2003 and 2007 respectively. Prior to that he received his B.E. in Computer Engineering from the Mumbai University in India in 2000. He is currently a software engineer at the Internet and Services Business Unit at Cisco Systems. His primary research interests include designing efficient data delivery mechanisms for wireless ad-hoc and sensor networks including vehicular networks.

Bhaskar Krishnamachari received his B.E. in Electrical Engineering at The Cooper Union, New York, in 1998, and his M.S. and Ph.D. degrees from Cornell University in 1999 and 2002 respectively. He is currently the Philip and Cayley MacDonald Early Career Chair Assistant Professor in the Department of Electrical Engineering at the University of Southern California. His primary research interest is in the design and analysis of efficient mechanisms for operating wireless sensor networks.

Shahram Ghandeharizadeh received his Ph.D. degree in computer science from the University of Wisconsin, Madison, in 1990. Since then, he has been on the faculty at the University of Southern California. In 1992, Dr. Ghandeharizadeh received the National Science Foundation Young Investigator's Award for his research on the physical design of parallel database systems. In 1995, he received an award from the School of Engineering at USC in recognition of his research activities. His primary research interests include design and implementation of multimedia storage managers, parallel database systems, and active databases. He has served on the organizing committees of numerous conferences and was the general co-chair of ACM Multimedia 2000. His activities are supported by several grants from the National Science Foundation, Department of Defense, Microsoft, BMC Software, and Hewlett-Packard. He is the director of the database laboratory at USC. He was a member of ACM SIGMOD executive committee and the Editor-in-Chief of ACM SIGMOD DiSC from 2003 to 2006. He also serves as a member of the Council of Directors of Kodaikanal International School.