Exploring Group Mobility for Replica Data Allocation in a Mobile Environment

Jiun-Long Huang and Ming-Syan Chen
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, ROC

Wen-Chih Peng
Department of Computer Science and Information Engineering
National Chiao-Tung University
Hsinchu, Taiwan, ROC

E-mail: jlhuang@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw, wcpeng@csie.nctu.edu.tw

ABSTRACT

The growth in wireless communication technologies attracts a considerable amount of attention in mobile ad-hoc networks. Since mobile hosts in an ad-hoc network usually move freely, the topology of the network changes dynamically and disconnection occurs frequently. These characteristics make a mobile ad-hoc network be likely to be separated into several disconnected partitions, and the data accessibility is hence reduced. Several schemes are proposed to alleviate the reduction of data accessibility by replicating data items. However, little research effort was elaborated upon exploiting the group mobility where the group mobility refers to the phenomenon that several mobile nodes tend to move together. In this paper, we address the problem of replica allocation in a mobile ad-hoc network by exploring group mobility. We first analyze the group mobility model and derive several theoretical results. In light of these results, we propose a replica allocation scheme to improve the data accessibility. Several experiments are conducted to evaluate the performance of the proposed scheme. The experimental results show that the proposed scheme is able to not only obtain higher data accessibility but also produce lower network traffic than prior schemes.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems*; H.2.4 [**Database Management**]: Systems—*distributed databases*

General Terms

Algorithms

Keywords

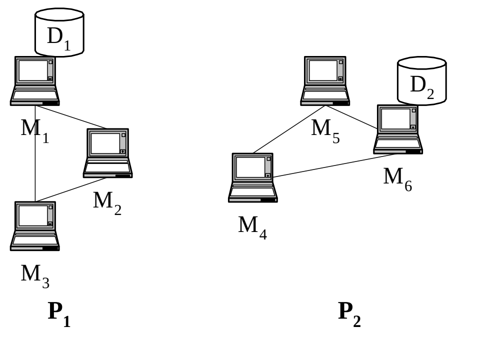
Ad-hoc networks, replica allocation, data accessibility, mobile computing

1. INTRODUCTION

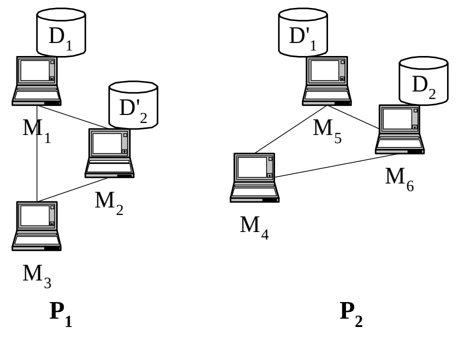
The growth in wireless communication technologies attracts a considerable amount of attention in mobile ad-hoc networks (MANETs). A MANET is a self-organizing, rapidly deployable network which consists of wireless nodes without infrastructure. All nodes in a MANET are capable of moving actively and can be connected dynamically. Due to the lack of infrastructure, mobile nodes of a MANET also function as routers which discover and maintain routes, and forward packets to other nodes.

With the advance of technology, a mobile device is equipped with a small storage to store data items. In a collaborative work scenario, several mobile devices may work together and access data items stored in other devices. Since each node in a MANET can move freely, the topology and connectivity of the network often change dynamically. In addition, disconnection often occurs and causes frequent network division. Both characteristics will separate the network topology into several disconnected *partitions*, and hence decrease the accessibility of data items. Consider the scenario in Fig. 1a, where there are six mobile devices (M_1, M_2, $\cdots$, M_6), and M_1 and M_6 contain data items D_1 and D_2, respectively. Assume that the link between M_1 and M_4 disconnects due to the movement of mobile devices and then the network is divided into two partitions, $P_1 = \{M_1, M_2, M_3\}$ and $P_2 = \{M_4, M_5, M_6\}$. The data access to D_2 from a mobile devices in P_1 (e.g., M_1) will fail since P_1 and P_2 are disconnected. Similarly, the data access to D_1 from mobile devices in P_2 will also fail.

Data replication is a promising technique to improve data accessibility and system performance in distributed database systems. However, in the literature of mobile computing, only a few works addressed the issues of data replication in MANETs. Authors in [17] and [18] addressed the problem of the update of replicas in a MANET. However, each node is expected to replicate all data items in the MANET whose practicability may need further justifications since the memory space is a scarce resource of mobile devices. Note that with the global topology of a fixed network, the replica allocation problem can be transformed into a minimum K-median problem which is NP-hard [7][13]. Clearly, the problem of replica allocation in a MANET is even more complicated since the network topology changes frequently. Authors in [3] proposed scheme DCG to address the problem of replica allocation in a MANET where the data items



(a) Without replication



(b) With replication

Figure 1: The effect of data replication in a MANET

are assumed to be readonly. The experiments in [3] show that data replication can greatly improve the data accessibility for a partitioned network. Authors in [5] proposed an extension of scheme DCG, named scheme E-DCG, for data items the periodic update.

In reality, the moving behavior of mobile users is usually regular and follows some mobility patterns [12][16]. Group mobility refers to the scenario where several mobile nodes tend to move together. In practice, group mobility usually occurs in collaborative works [6][9]. For example, a group of visitors visiting an art gallery with the same guide usually has similar movement behavior. In military, a team of soldiers usually moves together to accomplish a task. As a consequence, a number of research works have been elaborated upon the impact of group mobility [10][11][14][15]. Although reducing the degradation of data accessibility to some extent, both scheme DCG and scheme E-DCG have the following two major drawbacks.

1. *Generation of a large amount of traffic.*

   In essence, because of requiring the global network connectivity, both schemes are *centralized* and require all data nodes to broadcast their information to all other nodes, which will undesirably cause a significant amount of network traffic. In this paper, we call this phenomenon as *blind flooding*. This situation is more severe in a MANET due to the low network bandwidth.

2. *Negligence of group mobility.*

   Consider the scenario shown in Fig. 1b. Suppose that M_1 and M_3 are in the same mobility group M_2 is in another. We also assume that both groups are of different moving behaviors. Therefore, M_1 and M_2 tend to disconnect in the near future and only allocating $D_2^{'}$ in M_2 will lead to an unsuccessful access to D_2 from M_1. As a result, replica allocation schemes without considering group mobility are not able to allocate replicas effectively.

In view of this, we address in this paper the problem of replica allocation by exploring group mobility. The underlying group mobility model is assumed to be RPGM (standing for Reference Point Group Mobility model). By analyzing the characteristic of the employed group mobility model, several theoretical results are derived. In light of these results, we propose a replica allocation scheme (referred to as scheme DRAM) to allocate replicas by considering group mobility. To avoid blind flooding, scheme DRAM takes a *bottom-up* approach without requiring the global network connectivity. In scheme DRAM, each mobile node first exchanges its motion behavior with some neighbors. The coverage of the information exchange is limited by a predetermined parameter. Then, a *decentralized* clustering algorithm is proposed to cluster mobile nodes with similar motion behavior into mobility groups. Hence, clusters which are likely to connect with one another later will be merged into an allocation unit to save the aggregate storage cost. Finally, data items are replicated according to the resulting allocation units. Moreover, scheme DRAM maintains the mobility groups in an *adaptive* manner which keeps the number of information broadcasts as small as possible and hence reduces the generated network traffic. These characteristics distinguish our work from others. To evaluate the performance of the proposed schemes, several experiments are conducted. The experimental results show that scheme DRAM is able to not only achieve higher data accessibility but also produce less network traffic than prior schemes.

The rest of this paper is organized as follows. Section 2 describes the employed mobility and system models. The proposed replica allocation scheme DRAM is described in Section 3. The performance study of DRAM is given in Section 4, and finally, Section 5 concludes this paper.

2. PRELIMINARIES

2.1 Mobility Models

A mobility model is used to model the mobility behavior of mobile nodes. Waypoint mobility model [8] is a widely-used mobility model to describe the moving behavior of a single mobile node. Waypoint mobility model breaks the entire movement of a mobile node into repeating *pause* and *motion* periods. In a pause period, a mobile node will stay at the current location for a certain time period. When changing to a new movement period, one node will move toward a new random-chosen destination at a speed uniformly distributed between $[v_{Min}, v_{Max}]$. After arriving the destination, the mobile node will enter another pause period.

In real cases, mobile nodes may collaborate and hence move as a group instead of independently. RPGM is proposed in [6] to model this kind of team collaboration behavior. RPGM and its variations, [9] and [15], are widely used to model the group mobility in a MANET [6][10][11][14][15].

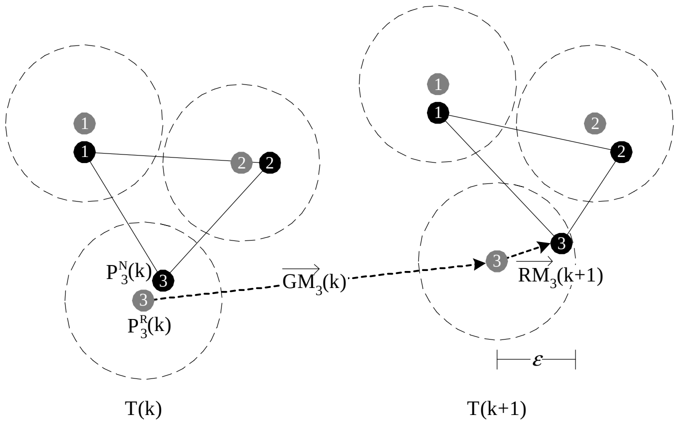


Figure 2: An example of Reference Point Group Mobility model

In RPGM, all mobile nodes are divided into several mobility groups and all mobile nodes within the same mobility group are of similar moving behavior. Suppose that the time is divided into several slots, $T(1)$, $T(2)$, $\cdots$. Each time slot is of equal time interval δ. Each node is assigned to a virtual reference node. The movement of a reference node in a time slot is called a *global motion vector*. The global motion vectors of all reference nodes in the same mobility group are the same. The location of a mobile node is uniformly distributed in the circle centered by the location of the corresponding reference node with radius ϵ. For each mobile node, the vector from the position of the corresponding reference node to the position of the mobile node is called a *random motion vector*. The characteristic of RPGM is that in addition to group motion, it allows random moving behavior for each node.

Fig. 2 shows an example movement of the mobile nodes (i.e., M_1, M_2 and M_3) of a mobility group which follows RPGM model in a motion period. In each time slot, each black node represents a mobile node and the gray node with equal number represents the corresponding reference node. The dotted circle centered by a reference node is the possible area where the mobile node will actually appear. Let $P_i^N(k)$ and $P_i^R(k)$ represent the positions of the mobile node M_i and the corresponding reference node in time $T(k)$, respectively. Consider the mobile node M_3 in Fig. 2 as an example. The positions of M_3 and its reference point in time $T(k)$ are $P_3^N(k)$ and $P_3^R(k)$, respectively. The global and random motion vectors of M_3 are $\overrightarrow{GM_3}(k)$ and $\overrightarrow{RM_3}(k)$, respectively. We then have

$$P_i^R(k+1) = P_i^R(k) + \overrightarrow{GM_i}(k)$$

and

$$P_i^N(k+1) = P_i^R(k+1) + \overrightarrow{RM_i}(k+1).$$

2.2 System Model

Suppose that there are m mobile devices, M_1, M_2, $\cdots$, M_m. Each device stores and acts as the original host of several data items. There are n data items, D_1, D_2, $\cdots$, D_n. Similar to [3], all data items are of equal size, and each data item is held by a particular mobile host as its original host. In addition to original data items, each mobile host has an extra memory space to store S_{Rep} replicas. Similar to [5], each data item D_i is updated by its original host periodically with period τ_i. The access frequency of mobile host M_i to data item D_j is assumed to be f_{ij} which is known in advance [3][4][5]. Each node is equipped with a GPS device, and hence, the position of the node is always available.

In this paper, we take RPGM as the group mobility model and the movement of each group follows a waypoint model. The speed of a group within a motion period will be a constant which follows a uniform distribution over $[v_{Min}, v_{Max}]$. We also assume that the lengths of pause and motion periods are two exponential distributions with mean μ_{Pause} and μ_{Motion} time slots, respectively. For a mobile node M_i in time $T(k)$, we have a global motion vector $\overrightarrow{GM_i}(k)$ and random motion vector $\overrightarrow{RM_i}(k)$. Let $|\cdot|$ represent the length of a vector, and then we have $v_{Min} \times \delta \leq |\overrightarrow{GM_i}(k)| \leq v_{Max} \times \delta$ in each motion period and $|\overrightarrow{GM_i}(k)| = |\overrightarrow{RM_i}(k)| = 0$ in each pause period.

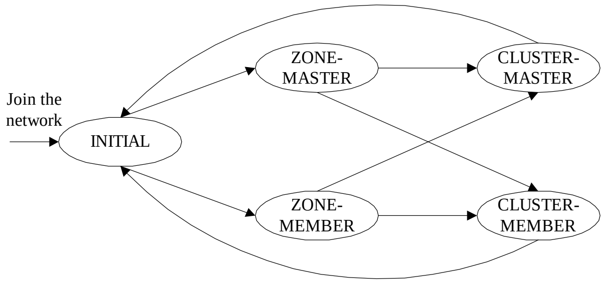


Figure 3: The state transition diagram of mobile nodes in scheme DCG

3. DESIGN OF DECENTRALIZED REPLICA ALLOCATION WITH GROUP MOBILITY

3.1 An Overview

Since the network connectivity changes frequently, scheme DRAM is executed periodically with period r time slots to adapt the replica allocation according to the network connectivity. This time period is referred to a *relocation period*. The tasks for each mobile node in a relocation period consist of two major phases: the allocation unit construction phase and the replica allocation phase.

In the allocation unit construction phase, all mobile nodes in the network are divided into several *disjoint* allocation units. The formal definition of an allocation unit is as follows.

Definition 1: An *allocation unit* is a set of mobile nodes which share their storage and do not store repeated data item unless all data items have been allocated in this allocation unit.

Then, in the following replica allocation phase, the replicas of all data items are allocated in accordance with the access frequencies of the data items and the derived allocation units of the previous allocation unit construction phase.

Clearly, the more mobile nodes each allocation unit contains, the higher data accessibility we have. However, an *unstable* allocation unit which will be separated into disconnected partitions later will not take the advantage of storage sharing which will in turn cause severe degradation of data accessibility. Although setting a shorter relocation period can reduce the degradation in data accessibility, this approach will produce more network traffic. Therefore, constructing *large* and *stable* allocation units is important to achieve high data accessibility.

In the allocation construction phase, scheme DRAM constructs the allocation units in a bottom-up manner by considering the group mobility. In addition to producing less traffic than prior schemes by avoiding blind flooding, scheme DRAM is able to construct more stable allocation units since mobile nodes in each resulting allocation unit are likely connected later. Moreover, scheme DRAM maintains the mobility groups in an adaptive manner which keeps the number of information broadcasts as small as possible and hence reduces the generated network traffic. The details of scheme DRAM are described in the following subsections.

3.2 Allocation Unit Construction Phase

Figure 3 shows the state transition diagram of mobile nodes in scheme DRAM. The tasks of mobile nodes in these states are described in the following subsections.

3.2.1 INITIAL State

Before describing the task of nodes the INITIAL state, we first define a broadcast zone as follows.

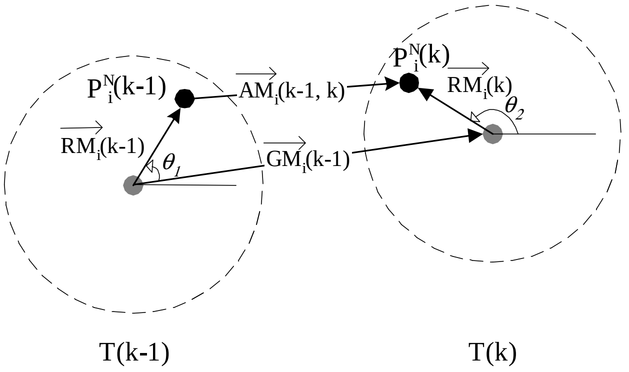
Definition 2: The *broadcast zone* of a mobile node M_i is a set of mobile nodes whose distances to M_i are smaller than or equal to a predefined TTL.

When a mobile node wants to join the MANET, it will enter the INITIAL state. In addition, a mobile node in the CLUSTER-MASTER state will enter the INITIAL state when there is no member of its cluster within its broadcast zone. A mobile node in the CLUSTER-MEMBER state will also enter the INITIAL state when its mobility group master is not within its broadcast zone. A detailed description of the CLUSTER-MASTER and CLUSTER-MEMBER states will be given in Section 3.2.3.

A mobile node in the INITIAL state will broadcast an *info* messages to all mobile nodes in its broadcast zone. Unlike scheme E-DCG, each *info* message contains a time-to-live which is a predetermined constant TTL. When a mobile host M_i receives an *info* message of M_j, M_i will accept this *info* message and forward this message to all its neighbors if the distance between M_i and M_j is smaller than or equal to TTL. Otherwise, M_i will discard the message. Therefore, only mobile nodes connected to M_j with distance smaller than or equal to TTL (i.e., those mobile nodes within the broadcast zone of M_j) will receive the *info* message of M_j. The employment of TTL can avoid blind flooding and produce less network traffic.

Let an actual motion vector $\overrightarrow{AM_i}(x,y)$ represent the motion of M_i from time $T(x)$ to $T(y)$ (i.e., the vector from $P_i^N(x)$ to $P_i^N(y)$). Suppose that $T(k-1)$ and $T(k)$ are within the same motion period, and consider the movement scenario of M_i as shown in Fig. 4. Since $0 \le |\overrightarrow{RM_i}(k)| \le \epsilon$ always holds for all mobile host M_i, we have:

$$|\overrightarrow{AM_i}(k-1,k)| \ge |\overrightarrow{GM_i}(k-1)| - 2\epsilon,$$



$P_i^N(k-1)$
$\overrightarrow{AM_i}(k-1, k)$
$P_i^N(k)$
$\overrightarrow{RM_i}(k)$
θ_2
$\overrightarrow{RM_i}(k-1)$
θ_1
$\overrightarrow{GM_i}(k-1)$
T(k-1)
T(k)

Figure 4: The locations of a mobile host M_i in $T(k-1)$ and $T(k)$

and

$$|\overrightarrow{AM_i}(k-1,k)| \le |\overrightarrow{GM_i}(k-1)| + 2\epsilon.$$

Assume that $|\overrightarrow{GM_i}(k-1)| > 2\epsilon$, and hence $|\overrightarrow{AM_i}(k-1,k)| > 0$ when $T(k-1)$ and $T(k)$ are in the same motion period. Since the speed of each mobile node is zero only in pause periods, each mobile node M_i is in the same pause period from $T(x)$ to $T(y)$ if and only if $P_i^N(x) = P_i^N(x+1) = \cdots = P_i^N(y)$. Therefore, the pause and motion periods of each mobile node can be recognized from its historical locations.

To achieve this, each mobile node M_i maintains a list of its historical locations (i.e., $P_i^N(k)$) called a position list. For each time slot, M_i will retrieve its current location, denoted as $P_i^N(k_1)$, from its GPS device and insert it into the position list. Note that the size of the position list is limited since the storage is a scare resource. Suppose that the maximal capacity of a position list is S_{List}, the current time stamp is $T(k_1)$, and the largest time stamp among all elements in the position list is $T(k_2)$, where $k_2 < k_1$. If $P_i^N(k_1) = P_i^N(k_2)$, $P_i^N(k_1)$ is discarded since $T(k_1)$ and $T(k_2)$ are in the same pause period. Otherwise, $P_i^N(k_1)$ is inserted into the position list. If the position list is full, the element with the smallest time stamp will be removed.

For each mobile node, an *info* message contains the following fields which can be obtained from its position list.

- *id*: The host id.
- ts_{Now}: The time stamp of the current relocation period.
- $P_i^N(ts_{Now})$: The location of M_i in time $T(ts_{Now})$.
- $ts_{Start}^{Pause}, ts_{End}^{Pause}$: The time stamps of the start and end time of the latest pause period.
- $ts_{Start}^{Motion}, ts_{End}^{Motion}$: The time stamps of the start and end time of the latest motion period.
- $\overrightarrow{AM_i}(ts_{Start}^{Motion}, ts_{End}^{Motion})$: The actual motion vector of the latest motion period.

3.2.2 ZONE-MASTER and ZONE-MEMBER States

After broadcasting *info* messages, the mobile nodes are classified into two groups by the lowest-id clustering algorithm proposed in [2]. Each node in the INITIAL state whose host id is the smallest one among all nodes within its broadcast zone (obtained from the id field of all received *info* messages) is selected as the master of its broadcast zone and hence enters the ZONE-MASTER state. On the other hand, the other nodes enter the ZONE-MEMBER state. A node M_i in the ZONE-MEMBER state will join the node M_j in the ZONE-MASTER state with the smallest id among the nodes within the broadcast zone of M_i by sending a *join* messages to M_j.

Each node M_i in the ZONE-MASTER state then clusters its member nodes (i.e., those node which send the *join* messages to M_i) into clusters, and all nodes within a cluster are expected to have similar motion behaviors. The employed clustering algorithm is as follows. First, all nodes are clustered by the periods of the latest pause and motion periods recorded in *info* messages since all mobile nodes in the same cluster are of equal records of latest pause and motion periods (i.e., have the same ts^{Pause}_{Start}, ts^{Pause}_{End}, ts^{Motion}_{Start} and ts^{Motion}_{End} in their *info* messages). However, several mobile nodes within different mobility groups (i.e., of different motion behaviors) may be placed into the same cluster especially when the length of a motion period is larger than the length of position lists. To achieve better results, the master node should re-cluster the resulting clusters again by considering motion vectors. With the scenario shown in Fig. 4, we have the following lemmas. The proofs of lemmas are omitted for interest of space.

Lemma 1: The expectation of $\overrightarrow{AM_i}(k-1,k)$ is equal to $\overrightarrow{GM_i}(k-1)$.

From Lemma 1, we have the following observation. If the length of position lists (i.e., S_{List}) is large enough and the length motion period is long enough, the global motion vector of each mobile node can be approximated. However, it is impractical to store too many previous positions of a mobile node. Therefore, we propose a vector clustering algorithm below. To facilitate our presentation, we denote the global motion vector of M_i from $T(k_1)$ to $T(k_2)$ (i.e., the vector from $P^R_i(k_1)$ to $P^R_i(k_2)$) as $\overrightarrow{V_i}(k_1,k_2)$. For M_i, if $T(k_1)$ and $T(k_2)$ are in the same motion period and $T(k_2) > T(k_1)$, we have $\overrightarrow{V_i}(k_1,k_2) = (k_2-k_1)\times\overrightarrow{GM_i}(k_1)$. In addition, we represent motion vectors as polar coordinates (i.e., length-angle forms) and $T(k_1)$ and $T(k_2)$ are assumed to be in the same motion period and $k_2 > k_1$.

Lemma 2: For a mobile host M_i, the maximum absolute value of the difference between the angles of $\overrightarrow{AM_i}(k_1,k_2)$ and $\overrightarrow{V_i}(k_1,k_2)$ is

$$\left|\sin^{-1}\left(\frac{\epsilon}{(k_2-k_1)\times v_{Min}\times\delta-\epsilon}\right)\right|,$$

where v_{Min}, δ and ϵ represent respectively the minimum speed of mobile nodes, the length of a time slot and the maximal length of random vectors.

Lemma 3: For a mobile host M_i, the maximal absolute value of the differences between the lengths of $\overrightarrow{AM_i}(k_1,k_2)$ and $\overrightarrow{V_i}(k_1,k_2)$ is 2ϵ.

For vectors $v_i = (r_i,\theta_i)$ and $v_j = (r_j,\theta_j)$, v_j is said to be "a neighbor in angle with maximal difference θ of v_j" if $|\theta_i-\theta_j| \le \theta$. Similarly, v_j is said to be "a neighbor in length with maximal difference *dist* of v_j" if $|r_i-r_j| \le dist$.

For each mobile node M_i, $\overrightarrow{AM_i}(k_1,k_2)$ is a neighbor in angle with maximal difference θ of $\overrightarrow{V_i}(k_1,k_2)$ where θ is equal to the result in Lemma 2. Similarly, according to Lemma 3, $\overrightarrow{AM_i}(k_1,k_2)$ is the neighbor in length with maximal difference 2ϵ of $\overrightarrow{V_i}(k_1,k_2)$.

In a mobility group, the number of the neighbors in angle with maximal difference θ of $\overrightarrow{V_i}(k_1,k_2)$ is maximal among all actual motion vectors of all group members. Similarly, the number of the neighbors in length with maximal difference 2ϵ of $\overrightarrow{V_i}(k_1,k_2)$ is maximal among all actual motion vectors of all group members. With the result of Lemma 1 and the above observations, we have the following two heuristics.

1. In a mobility group, an actual motion vector is close to the global motion vector if it has the maximal number of neighbors in angle with maximal difference θ.

2. In a mobility group, an actual motion vector is close to the global motion vector if it has the maximal number of neighbors in length with maximal difference 2ϵ.

After executing the algorithm VectorCluster, each zone master will select one cluster master for each resulting cluster. Then the selected mobile nodes will enter the CLUSTER-MASTER state, and the other nodes will enter the CLUSTER-MEMBER state.

3.2.3 *CLUSTER-MASTER and CLUSTER-MEMBER States*

The tasks of cluster masters and members consist of two steps, cluster maintenance and cluster merge, which are described below.

Cluster Maintenance: Mobile nodes which enter the CLUSTER-MASTER and CLUSTER-MEMBER states in current allocation unit construction phase will skip the cluster maintenance step and execute the cluster merge step in this allocation unit construction phase. The other mobile nodes will execute cluster maintenance step before executing cluster merge step. Since the clusters are constructed by considering group mobility, each cluster is likely to be connected in the near future. These clusters are maintained in the following adaptive fashion.

Each cluster member first sends a *status* message to its cluster master. The contents of *status* messages are similar to those of *info* messages. If the mobile node and its cluster master are disconnected, the mobile node will leave the current cluster and enter the INITIAL state. At the same time, the cluster master will remove the mobile hosts which do not send *status* messages to it from its member list.

The cluster master then checks whether the moving behaviors recorded in received *status* messages are similar to one another by the following procedure. The cluster master first clusters the motion behaviors stored in received *status* messages by the clustering algorithm proposed in Section 3.2.2. If each cluster member keeps similar motion behavior to that it used to have, the result of the execution of the clustering algorithm is one cluster which contains all cluster members. If some cluster members change their motion behaviors, the result will consist of several clusters. Let the dominating cluster of the resulting clusters be the cluster with the most nodes among all resulting clusters. Those mobile nodes not in the dominating cluster will then receive *reject* messages from the cluster master, and enter the

INITIAL state. Note that only mobile nodes with different moving behavior from the majority of the original cluster will not be in the dominating cluster. If the original cluster master is not in the dominating cluster, it will assign one node in the dominating cluster as the new cluster master. The new cluster master will then send *reject* messages to the mobile nodes which are not in the dominating cluster.

Except the first execution of scheme DRAM, most mobile nodes will be in the CLUSTER-MASTER or CLUSTER-MEMBER state since only mobile nodes disconnected with its cluster master or with different moving behavior from its cluster members will enter the INITIAL state. Therefore, the successive execution of scheme DRAM will produce less network traffic than the first execution because only the first execution of scheme DRAM requires all mobile nodes to broadcast messages. As shown in Section 4, the *adaptive* cluster maintenance is able to effectively reduce produced network traffic.

Cluster Merge: If two mobile nodes, M_1 and M_2, tend to connect to each other, replicating one data item in M_1 also makes this data item accessible for M_2. Hence, M_2 can replicate another data item which will be also accessible for M_1. As mentioned in Section 3.1, merging clusters which tend to be connected in the near future into a big allocation unit makes the mobile nodes within different clusters share their storage and hence improve the data accessibility.

In replica allocation construction, only the resulting clusters which are likely connect with one another until the next execution of scheme DRAM will be merged into the same allocation unit. Initially, each resulting cluster is assigned to be an allocation unit. Assume that the current and next executions of scheme DRAM are in $T(k)$ and $T(k+r)$, respectively. Then, we have the following definitions.

Definition 3: The *bounding rectangle* of an allocation unit is a rectangle which contains all nodes of the allocation unit. Denote the bounding rectangle of cluster C_i in $T(k)$ as $BR_i(k)$. Let $BR_i(k)$ be represented by four values: $top_i(k)$, $bottom_i(k)$, $left_i(k)$ and $right_i(k)$. Then, we have

$$\begin{aligned}
top_i(k) &= \max_{\forall M_j \text{ in } C_i} \{\text{the y coordinate of } M_j \text{ in } T(k)\} \\
bottom_i(k) &= \min_{\forall M_j \text{ in } C_i} \{\text{the y coordinate of } M_j \text{ in } T(k)\} \\
left_i(k) &= \min_{\forall M_j \text{ in } C_i} \{\text{the x coordinate of } M_j \text{ in } T(k)\} \\
right_i(k) &= \max_{\forall M_j \text{ in } C_i} \{\text{the x coordinate of } M_j \text{ in } T(k)\}
\end{aligned}$$

Definition 4: Let $BR_i^*(k+r)$ be the *estimated* bounding rectangle of the allocation unit C_i in $T(k+r)$. Assume that the latest *info* message of the master node is in $T(k)$, we have

$$\begin{aligned}
top_i^*(k+r) &= top_i(k) + r \times y \\
bottom_i^*(k+r) &= bottom_i(k) + r \times y \\
left_i^*(k+r) &= left_i(k) + r \times x \\
right_i^*(k+r) &= right_i(k) + r \times x, \qquad \text{where}
\end{aligned}$$

$$\langle x, y \rangle = \frac{\text{the average of all } \overrightarrow{AM_i}(ts^{Motion}_{Start}, ts^{Motion}_{End}) \text{ in } C_i}{ts^{Motion}_{End} - ts^{Motion}_{Start}}.$$

Definition 5: Two allocation units C_i and C_j are said to be *cluster-wise connected* in $T(k)$ if

1. $BR_i(k)$ and $BR_j(k)$ are overlapped, or
2. there exists another allocation unit C_x so that $BR_i(k)$ and $BR_j(k)$ are overlapped with $BR_x(k)$, respectively.

Definition 6: Two allocation units C_i and C_j are said to be *potentially cluster-wise connected* in $T(k+r)$ if

1. $BR_i^*(k+r)$ and $BR_j^*(k+r)$ are overlapped, or
2. there exists a C_x so that $BR_i^*(k+r)$ and $BR_j^*(k+r)$ are overlapped with $BR_x^*(k+r)$, respectively.

We observe that two allocation units C_i and C_j can be *merged* into a new allocation unit if C_i and C_j are *cluster-wise connected* in $T(k)$ and *potentially cluster-wise connected* in $T(k+r)$.

In replica allocation construction, each cluster master will broadcast a *merge* message containing the cluster master id, the current and estimated bounding rectangles. Note that the merge relation is an equivalence relation since it is reflective, symmetric and transitive. The cluster merge process can be performed by the following procedure.

3.3 Replica Allocation Phase

The objective of replica allocation phase is to identify data items to be replicated, and the locations to replicate them for each allocation unit in order to maximize the data accessibility. Similar to [5], the employed replica allocation algorithm is as follows. Let the allocation weight of data item D_j in allocation unit C_x in $T(k)$ (denoted as $w_j^x(k)$) be from now on, the expected number of data access from all mobile nodes in C_x before the next update of D_j. Suppose that the current time stamp is $T(k)$, $w_j^x(k)$ can be obtained by the following equation:

$$w_j^x(k) = f_j^x \times (U_j - k), \text{ where}$$

$$f_j^x = \sum_{\forall\ M_i \in C_x} f_{ij},$$

and U_j is the time-stamp of the next update of D_j. Since the update of each data item is periodic, U_j can be predicted in advance.

All data items are allocated in C_x according to their allocation weights in C_x in descendent order. The data item D_j will be allocated in M_j when f_{ij} is the largest among all other mobile hosts with available storage. The allocation process completes if all mobile hosts in C_x is full. Each unit master then executes the following replica allocation process as below.

```
Procedure ReplicaAllocation($\mathcal{C}$)
Input: A set of allocation units $\mathcal{C}$.
1: Calculate allocation weight (i.e., $w_j^x(k)$) for each data
   item $D_j$ in $C_x$ and then put all data items into $\mathcal{D}$.
2: while ($\mathcal{D}$ is not empty) do
3:     Let $D_j$ be the data item with largest allocation
       weight in $\mathcal{D}$. Allocate $D_j$ to $M_i$ where $f_{ij}$ is the
       largest among all mobile hosts with available storage.
4:     Remove $D_j$ from $\mathcal{D}$.
5: end while
```

Table 1: System parameters

Parameter	Value
The number of mobile hosts (m)	120
The number of data items (n)	2400
The number of groups	10
Max. no. of replicas to be stored in M_i (S_{Rep})	100
The min. speed of mobile nodes (v_{Min})	3
The max. speed of mobile nodes (v_{Max})	5
The avg. length of motion period (μ_{Motion})	5
The avg. length of pause period (μ_{Pause})	3
The max. length of random vectors (ϵ)	1
Radio Radius (R_{Radius})	5
Relocation period (r)	600
The update period for each data item (τ_i)	500
Time-to-live of *info* messages (TTL)	10

4. PERFORMANCE EVALUATION

4.1 System Model

To evaluate the performance of scheme DRAM, we implemented an event-driven simulator in C++ with SIM [1]. Scheme E-DCG is also implemented for comparison purposes. We assume that there are 120 mobile nodes in a 50×50 flatland and each node owns 20 data items. The system parameters are shown in Table 1. Similar to [3], the access frequency for each data item, f_{ij}, is determined as a positive value based on the normal distribution with mean $0.025(1 + 0.01i)$ and standard derivation 0.0025. The simulation is run for 3000 time slots.

Data accessibility is employed as the measurement of the performance of these schemes. When one mobile node M_i issues a data request, the request is successful if the required data item is stored in M_i or in another mobile node connected to M_i. Therefore, the data accessibility is defined as below:

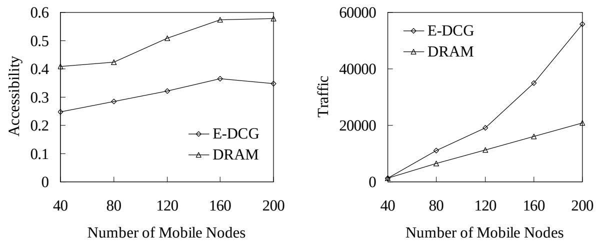
$$\text{Accessibility} = \frac{\text{Number of successful requests}}{\text{Number of issued requests}}.$$

We also use the produced network traffic to evaluate the cost of the employed schemes. Similar to [3], traffic is defined as the total hop counts of message transmission of all schemes.

4.2 The Effect of the Number of Mobile Nodes

This experiment investigates the effect of the number of mobile nodes. The data accessibility and produced traffic of both schemes with the the number of mobile nodes varied are shown in Fig. 5. The number of mobile nodes ranges from 40 to 200.

As observed in Fig. 5a, the data accessibility increases as the number of mobile nodes increases. In this experiment, the data accessibilities of scheme DRAM and E-DCG decrease from 40.88% to 57.79% and from 24.84% to 34.74%, respectively, as the number of mobile nodes increases from 40 to 200. It is because that with the same number of mobility groups, a larger number of mobile nodes groups implies that each mobility group is of more number of mobile nodes. Hence, more mobile nodes can share their storage by constructing large allocation units, and as a consequence, increase the data accessibility. It is also shown that scheme DRAM outperforms scheme E-DCG in data accessibility. It is because that scheme DRAM can take advantage of group



(a) Accessibility (b) Traffic

Figure 5: The effect of the number of mobile nodes

mobility to obtain a higher data accessibility than scheme E-DCG by constructing larger and more stable allocation units than scheme E-DCG. In this experiment, the performance gain of scheme DRAM over scheme E-DCG ranges from 48% to 60%.

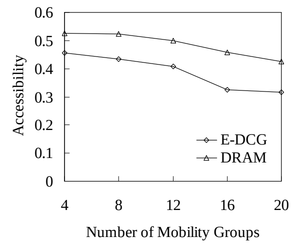
As observed in Fig. 5b, the amounts of produced traffic of all schemes increase as the number of mobile nodes increase. The network traffic reduction of scheme DRAM over scheme E-DCG increases from 5.21% to 60.74% as the number of mobile nodes increases from 40 to 200. In addition, the amount of produced traffic of scheme E-DCG increases drastically as the number of mobile nodes increases. This is due to the characteristic of blind flooding employed scheme in E-DCG that the amount of produced network traffic increases exponentially as the number of mobile nodes increases. On the other hand, the amount of produced traffic of scheme DRAM increases smoothly as the number of mobile nodes increases. The reason is that the TTL in *info* messages can effectively ease the number of exchanged messages in the INITIAL state. In addition, the adaptive cluster maintenance in scheme DRAM is able to reduce the number of broadcast messages and therefore reduce the amount of produced network traffic. This result shows the scalability of scheme DRAM over scheme E-DCG.

4.3 The Effect of the Number of Mobility Groups

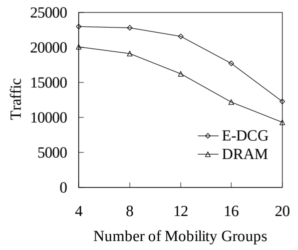
The effects in the accessibility and traffic for both schemes with the the number of mobility groups varied are shown in Fig. 6. The number of mobility groups is setting from 4 to 20.

We observe that the increment of the number of mobility groups decreases the data accessibilities of all schemes. As shown in Fig. 6a, the data accessibilities of scheme DRAM and E-DCG increases from 42.46% to 52.55% and from 31.55% to 45.7%, respectively, as the number of mobility groups decreases from 20 to 4. It is because that with the same number of mobile nodes, the smaller number of mobile groups indicates that more mobile nodes are of similar moving behavior. Hence, more mobile nodes can share their storage by constructing large allocation units, and hence, increase the data accessibility.

Although the data accessiblities of both schemes decrease as the number of mobility groups increases, the decrement of data accessibility of scheme DRAM is smoother than that of scheme E-DCG. As shown in Fig. 6a, scheme DRAM outperforms scheme E-DCG especially when the number of mobility groups is large. In Fig. 6a, the performance gain of scheme DRAM over scheme E-DCG in data accessibility increases from 15.01% to 34.58% as the number of mobility



(a) Accessibility



(b) Traffic

Figure 6: The effect of the number of mobility groups

groups increases from 4 to 20. The larger number of mobility groups indicates that the movement behavior of all mobile nodes are less regular. Therefore, the allocation units obtained by scheme E-DCG are more unstable, and hence, the data accessbility of scheme E-DCG degrades. On the other hand, scheme DRAM reduces the effect of the increment of the number of mobility groups by considering group mobility when constructing allocation units. As a result, the degradation in data accessibility of scheme DRAM is less significant than that of scheme E-DCG. This result shows the scalability of scheme DRAM over scheme E-DCG on the number of mobility groups.

As observed in Fig. 6b, the amount of traffic of all schemes increases as the number of mobility groups increases. When the number of mobility groups is large, the MANET tends to be separated into disconnected partitions since fewer mobile nodes are of similar moving behavior. Therefore, the amount of produced traffic of all schemes decreases since many mobile nodes disconnect with others when the MANET is separated into several partitions. In addition, we also observe that scheme DRAM produces less network traffic than scheme E-DCG, and the network traffic reduction of scheme DRAM over scheme E-DCG ranges from 12.64% to 24.52%.

5. CONCLUSION

We explored in this paper the problem of replica allocation in a MANET with group mobility. We first analyzed the employ group mobility model and derived several theoretical result. In light of these results, we proposed a replica allocation scheme DRAM to allocate replicas by considering group mobility. To evaluate the performance of the proposed schemes, several experiments were conducted. The experimental results showed that scheme DRAM can not only obtain higher data accessibility but also produce less traffic than prior schemes.

6. REFERENCES

[1] D. Bolier and A. Eliëns. SIM: a C++ library for Discrete Event Simulation. *http://www.cs.vu.nl/~eliens/sim/*, October 1995.

[2] M. Gerla and J. Tsai. Multicluster, Mobile, Multimedia Radio Network. *ACM Wireless Networks*, 1(3):255–265, 1995.

[3] T. Hara. Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In *Proceedings of IEEE INFOCOM Conference*, April 2001.

[4] T. Hara. Cooperative Caching by Mobile Clients in Push-based Information Systems. In *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, November 2002.

[5] T. Hara. Replica Allocation in Ad Hoc Networks with Periodic Data Update. In *Proceedings of 3rd International Conference on Mobile Data Management*, pages 79–86, January 2002.

[6] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In *Proceedings of the 2nd ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, August 1999.

[7] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt. Constrained Mirror Placement on the Internet. In *Proceedings of IEEE INFOCOM Conference*, April 2001.

[8] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 1996.

[9] B. Li. On Increasing Service Accessibility and Efficiency in Wireless Ad-hoc Networks with Group Mobility. *Wireless Personal Communications*, 21(1):105–123, April 2002.

[10] G. Pei, M. Gerla, and X. Hong. LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility. In *Proceedings of the 1st ACM Annual Workshop on Mobile Ad Hoc Networking and Computing*, August 2000.

[11] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang. A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, September 1999.

[12] W.-C. Peng and M.-S. Chen. Developing Data Allocation Schemes by Incremental Mining of User Moving Patterns in a Mobile Computing System. *IEEE Transactions on Knowledge and Data Engineering*, 15(1), February 2003.

[13] L. Qiu, V. N. Padmanabhan, and G. M. Voelker. On the Placement of Web Server Replicas. In *Proceedings of IEEE INFOCOM Conference*, April 2001.

[14] K. H. Wang and B. Li. Efficient and Guaranteed Service Coverage in Paritionable Mobile Ad-hoc Networks. In *Proceedings of IEEE INFOCOM Conference*, June 2002.

[15] K. H. Wang and B. Li. Group Mobility and Partition Prediction on Wireless Ad-Hoc Networks. In *Proceedings of IEEE ICC Conference*, pages 1017–1021, April 2002.

[16] H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke. Personal Paging Area Design Based on Mobile's Moving Behaviors. In *Proceedings of IEEE INFOCOM Conference*, April 2001.

[17] B. Xu, O. Wolfson, S. Chamberlain, and N. Rishe. Cost Based Data Dissemination in Broadcast Networks. In *Proceedings of the 8th International Conference on Database Theory*, January 2001.

[18] B. Xu, O. Wolfson, S. Chamberlain, and N. Rishe. Cost Based Data Dissemination in Satellite Networks. *ACM Mobile Networks and Applications*, 7(1), January 2002.