Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Ad Hoc Networks 4 (2006) 225–239

Ad Hoc Networks

www.elsevier.com/locate/adhoc

A novel caching scheme for improving Internet-based mobile ad hoc networks performance ☆

Sunho Lim *, Wang-Chien Lee, Guohong Cao, Chita R. Das

Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA

Received 19 December 2003; accepted 14 April 2004
Available online 8 August 2004

Abstract

Internet-based mobile ad hoc network (IMANET) is an emerging technique that combines a wired network (e.g. Internet) and a mobile ad hoc network (MANET) for developing a ubiquitous communication infrastructure. To fulfill users' demand to access various kinds of information, however, an IMANET has several limitations such as limited accessibility to the wired Internet, insufficient wireless bandwidth, and longer message latency. In this paper, we address the issues involved in information search and access in IMANETS. An *aggregate caching* mechanism and a broadcast-based *Simple Search* (*SS*) *algorithm* are proposed for improving the information accessibility and reducing average communication latency in IMANETS. As a part of the aggregate cache, a cache admission control policy and a cache replacement policy, called *Time and Distance Sensitive* (*TDS*) *replacement*, are developed to reduce the cache miss ratio and improve the information accessibility. We evaluate the impact of caching, cache management, and the number of access points that are connected to the Internet, through extensive simulation. The simulation results indicate that the proposed aggregate caching mechanism can significantly improve an IMANET performance in terms of throughput and average number of hops to access data items.

Keywords: Aggregate cache; Cache admission control; Cache replacement algorithm; Internet-based mobile ad hoc network; Simple search algorithm

1. Introduction

Over the past decade, Internet has changed our daily life. With the recent advent in wireless technology and mobile devices, ubiquitous communication is touted to change our life further. It is envisaged that in the near future, users will be able

---

☆ This research has been supported in part by NSF grants CCR-9900701, CCR-0098149, CCR-0208734, and EIA-0202007.

* Corresponding author. Tel.: +1 814 865 2729; fax: +1 814 865 3176.

E-mail addresses: slim@cse.psu.edu (S. Lim), wlee@cse.psu.edu (W.-C. Lee), gcao@cse.psu.edu (G. Cao), das@cse.psu.edu (C.R. Das).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2004.04.013

to access the Internet services and information anytime and anywhere. To realize this vision, wireless carriers are developing state-of-the-art wireless communication infrastructures. Nevertheless, a mobile terminal (MT) [1] may still have difficulty to connect to a wired network or Internet due to limited wireless bandwidth and accessibility. Under heavy traffic, an MT has to content for bandwidth and may get blocked from a wireless base station. Moreover, in some geographically remote areas, an infrastructure may not be even available. Thus, researchers are exploring an alternative technology, called *Mobile Ad Hoc Network* (MANET), for its low cost and ease of deployment.

A significant volume of research on MANETS has appeared in the literature in the past few years [6,9,10,14–17,19]. Most of these efforts, however, have focused on developing routing protocols to increase connectivity among MTs in a constantly varying topology. Due to the users' growing interest and falling cost in accessing the wireless Internet, it has become imperative to consider the integration of MANET with the wired Internet. Thus, to put the MANET technology into the context of real life, we consider an *Internet-based* MANET, called IMANET [2], and investigate the problem of information search and access under this environment. Under IMANET, we assume that some of the MTs are connected to the Internet or wired private networks. [2] Thus, an MT may access Internet information via a direct connection or via relays from other MTs. Although there may exist many potential applications, to the best of our knowledge, none of the previous work has addressed the issues for information search and access in IMANETS. The followings are some of the applicable scenarios for an IMANET:

- *Scenario 1*: During special events such as Olympic games or World Cup Soccer, the demand from users to access the Internet and communicate among themselves are exceedingly high. While a fixed infrastructure may be in place, it is challenging to accommodate all the users due to limited wireless bandwidth. With an IMANET, users can either access the required information directly or indirectly (through relays). Moreover, they can communicate among themselves without going through a wired infrastructure.
- *Scenario 2*: A visitor in a downtown, museum, or shopping mall may need to access various type of information (e.g. exhibition info, tour info including maps, restaurants of choice, hotels, theaters, and so on). A local service provider usually provides an electronic guide such as an info-station [22] that contains the relevant information. Although a visitor may lose connection to the info-station because of mobility, he/she can still access or share the information through relays using IMANET.
- *Scenario 3*: In a battle field or emergency site, one MT may be connected to the Internet by a satellite and serve as a proxy for other MTs. The accessed information and services can be shared by the other MTs via local ad hoc communication.

An IMANET has several constraints. First, not all the MTs can access the Internet. Second, due to mobility, a set of MTs can be separated from the rest of the MTs and get disconnected from the Internet. Finally, an MT requiring multi-hop relay to access the Internet may incur a longer access latency than those which have direct access to the Internet.

To address these constraints, we propose an *aggregate caching* mechanism for IMANETS. The basic idea is that by storing data items in the local cache of the MTs, members of the IMANET can efficiently access the required information. Thus, the aggregated local cache of the MTs can be considered as an unified large cache for the IMANET. The proposed aggregate cache can alleviate the constraints of IMANETS discussed above. When an MT is blocked from direct access to the Internet, it may access the requested data items from the local cache of nearby MTs or via relays. If an MT is isolated from the Internet, it can search

---

[1] In this paper, we use the term mobile terminal (MT) to refer to a portable device (e.g. a laptop computer, a personal digital assistance (PDA), a mobile phone, a handheld computer, etc) or a person who carries it.

[2] Without loss of generality, we use Internet to refer to both of Internet and wired private network for the rest of paper.

other reachable MTs for the requested data item. Finally, if an MT is located further from the Internet, it may request the data items from other closeby MTs to reduce access latency.

Here, two issues are addressed for implementation of an aggregate caching mechanism in IMANETS:

- *Efficient search*: An efficient information search algorithm is fundamental for locating the requested data in IMANETS.
- *Cache management*: To reduce the average access latency as well as enhance the data accessibility, efficient cache admission control and replacement policies are critical. The cache admission control policy determines whether a data item should be cached, while the cache replacement policy intelligently selects a victim data item to be replaced when a cache becomes full.

Information search in an IMANET is different from the search engine-based approach used in the wired Internet. An MT needs to broadcast its request to the possible data sources (including the Internet and other MTs within the IMANET) in order to retrieve the requested data efficiently. Thus, we propose a broadcast-based approach, called *Simple Search* (*SS*) algorithm, which can be implemented on the top of existing routing protocols to locate the requested data. In addition, we propose a cache admission control policy based on the distance between MTs to reduce redundant data caching, and a cache replacement policy based on time and distance, called *Time and Distance Sensitive* (*TDS*) replacement, to reduce the cache miss ratio and increase the accessibility of the aggregate cache.

We conduct a simulation-based performance evaluation to observe the impact of caching, cache management, and access points (APs) (which are directly connected to the Internet) upon the effectiveness of IMANETS. The overall results show that the proposed methodology can relieve limitations of IMANETS and improve system performance significantly.

This paper reports our initial study of Information search and access on IMANETS. The aggregate cache idea is simple and can be used in practice to enhance the communication performance of IMANETS. Focusing on the constraints of the IMANET such as accessibility and latency, our contribution is threefold:

- A simple search algorithm is developed to facilitate information search and access in an IMANET.
- An aggregate cache for IMANETS is proposed to address the issues of accessibility and latency.
- A distance-based admission control policy and three cache replacement policies (TDS_D, TDS_T, and TDS_N) are proposed as a part of the aggregate caching scheme. These policies are capable of providing better performance than the well known LRU replacement policy.

The rest of this paper is organized as follows. Work related to the research is reviewed in Section 2. The system model and simple search algorithm, and the aggregate cache management mechanism are presented in Sections 3 and 4, respectively. Section 5 is devoted to performance evaluation and comparisons of various policies. Finally, we concludes the paper with future directions in Section 6.

2. Related work

Research on MANET has mainly focused on developing routing protocols such as Destination-Sequenced Distance Vector (DSDV) [16], Dynamic Source Routing (DSR) [10], Ad hoc On Demand Distance Vector (AODV) [17], Temporally-Ordered Routing Algorithm (TORA) [15], and their variations. These algorithms assume that a sender MT knows the location of receiver MT based on the route information, which is accumulated and analyzed by a route discovery or route maintenance algorithm. Although a route discovery operation captures the current network topology and related information, it has to be executed whenever an MT needs to transmit a data item. To avoid repetitive route discovery, the MTs can cache the previous route information. Hu et al. [9] compared the performance of two caching strategies based on the DSR routing protocol: a path cache and a link cache. In the path cache, a complete path from a source to the destination is stored. In the link cache, a group of paths, which are collected from previous route discovery or other operations, is constructed to generate a graph style data structure. In our work, instead of addressing the issue of route discovery and its caching, we emphasize on efficient information search and data caching to enhance data accessibility.

Caching is an important technique to enhance the performance of wired or wireless network. A number of studies has been conducted to reduce the Web traffic and overall network congestion by deploying various caching schemes in the Internet [3,4,21]. A cooperative caching scheme is suggested in [3], in which a couple of individual caches are treated as a unified cache and they interact among themselves to eliminate the duplicate copies, and increase cache utilization. Fan et al. [4] proposed a summary cache, where proxies share their summary of cache contents represented by bloom filters. When a proxy has a cache miss for a request, it sends the request to other proxies based on a periodically updated summary of cache contents in other proxies. A proxy cache relocation scheme is proposed based on the prediction of user's mobility to reduce delay during a handoff, a mechanism of transferring an on-going call from the current cell to the next cell to which a user moves, in a cellular network [5]. However, no such work has been conducted in a MANET, in which a network topology frequently changes.

Ren et al. [18] employed a semantic caching scheme to manage location-dependent data (e.g. weather, traffic, and hotel information), in which an MT maintains semantic description of data in a mobile environment. When an MT needs to generate a query, it processes the query, analyzes the descriptions, and finds out results (or partial results) from the appropriate cache. Based on the results, the MT tailors or reduces the query and requests the server to get the rest of results to reduce communication. In contrast to the traditional cache replacement policies, the Furthest Away Replacement (FAR) is used in this study. With this policy, a victim is selected such that it is not on the way in which the MT might move, but is located far away from the current location of the MT.

In particular in MANETS, it is important to cache frequently accessed data not only to reduce the average latency, but also to save wireless bandwidth in a mobile environment. Hara [6] proposed a replica allocation method to increase data accessibility in MANETS. In this scheme, an MT maintains a limited number of duplicated data items if they are frequently requested. Replicated date items are relocated periodically at every relocation period based on the followings: each MT's access frequency, the neighbor MTs' access frequency or overall network topology. Update of the replicated data is further considered in [7]. Since an MT cannot access data when it is isolated from others, replication is an effective means to improve data accessibility. Due to the limited size of information that an MT can maintain, however, simply replicating data items and accessing them in MANETS cannot fulfill users' requirements to access a wide variety of information, available over the Internet.

To overcome the limited information availability in MANETS, Sailhan et al. [19] proposed a cooperative caching scheme to increase data accessibility by peer-to-peer communication among MTs, when they are out of bound of a fixed infrastructure. It is implemented on top of a well-known ad hoc routing protocol, called Zone Routing Protocol (ZRP). Papadopouli et al. [14] suggested the 7DS architecture, in which a couple of protocols are defined to share and disseminate information among users. It operates either on a prefetch mode, based on the information and user's future needs or on an on-demand mode, which searches for data items in a single-hop multicast basis. Depending on the collaborative behavior, a peer-to-peer and server-to-client model are used. Unlike our approach, this strategy focuses on data dissemination, and thus, the cache management including a cache admission control and replacement policy is not well explored.

To the best of our knowledge, none of previous work has explored an aggregated caching scheme along with an efficient information search algorithm in the realm of IMANETS.

3. Information search in IMANETS

3.1. System model

In this subsection, we describe a generic system model of IMANETS. We assume that an MT can not only connect to the Internet but also can forward a message for communication with other MTs via a wireless LAN (e.g. IEEE 802.11), as used in most prior study [19,14,13]. As illustrated in Fig. 1, an IMANET consists of a set of MTs that can communicate with each other using an ad hoc communication protocols (illustrated by dashed-line). Among the MTs, some of them can directly connect to the Internet, and thus serve as *access points* [3] (AP) for the rest of MTs in the IMANET. Thus, an AP is a gateway for the Internet and is assumed to have access to any information. An MT located out of the communication bound of an AP has to access the Internet via relays through one of the access points. An MT can move in any direction and make information search and access requests from anywhere in the covered area.

When an MT is located near by an AP (e.g. within one-hop), it makes a connection to the AP directly. When an MT is located far away from an AP, however, information access has to go through several hops in the ad hoc network before reaching the AP.

3.2. Information search algorithm

As mentioned in the introduction, the main focus of this paper is to support information access in IMANETS. Unlike a routing protocol, which establishes a path between a known source and destination, any MT can be an information source in the IMANET. Thus, without knowing the destination address for any requested information, a search algorithm is needed for IMANETS as is done in the Internet. In the following, we describe the basic idea of an information search algorithm employed in the paper. This algorithm can be implemented on top of an existing routing protocol for MANETS.

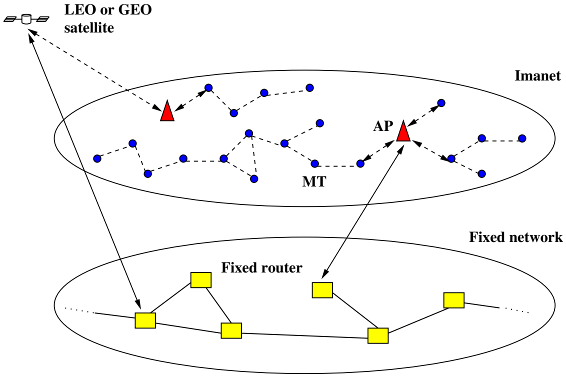


Fig. 1. A generic system model of IMANET.

Since an aggregate cache is supported in an IMANET design, requested data data items can be received from the local cache of an MT as well as via an AP connected to the Internet. When an MT needs a data item, it does not know exactly where to retrieve the data item from, so it broadcasts a request to all of the adjacent MTs. When an MT receives the request and has the data item in its local cache, it will send a reply to the requester to acknowledge that it has the data item; otherwise, it will forward the request to its neighbors. Thus, as illustrated in Fig. 2, [4] a request may be flooded in the network and eventually acknowledged by an AP and/or some MTs with cached copies of the requested data item.

Based on the idea described above, we propose an information search algorithm, called *Simple Search* (*SS*), to determine an information access path to the MTs with cached data of the request or to appropriate APs. The decision is based on the arriving order of acknowledgments from the MTs or APs. Let us assume an MT (n_i) sends a request for a data item (d) and an MT (n_k) is located along the path in which the request travels to an AP, where $k \in \{a, b, c, j\}$. The SS algorithm is described as follows:

---

[3] The AP here is a logical notation. An AP equipped with appropriate antennas can directly communicate with the Internet through wireless infrastructures including cellular base stations, and Low Earth Orbit (LEO) or geostationary (GEO) satellites.

[4] A dotted circle represents the communication range of an MT or an AP. For the sake of simplicity, we assume that both an MT and an AP have the same diameter of communication.

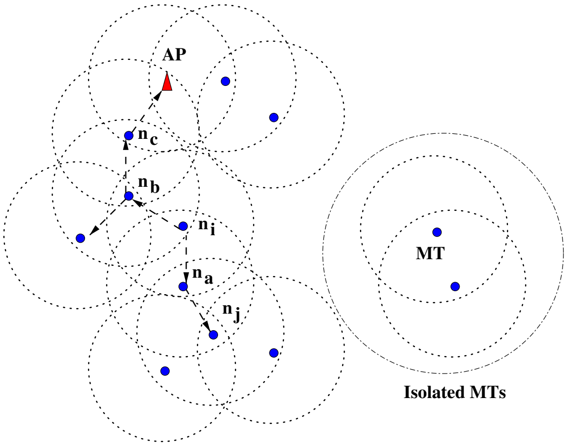


Fig. 2. An MT (n_i) broadcasts a request packet which is forwarded to the AP in the IMANET.

1. When n_i needs *d*, it first checks its local cache. If the data item is not available in the local cache and n_i cannot directly access to an AP, it broadcasts a *request* packet to the adjacent MTs (g_i). [5] The *request* packet contains the requester's id and request packet id. After n_i broadcasts the request, it waits for an acknowledgment. If n_i does not get any acknowledgment within a specified timeout period, it fails to get *d*.
2. When n_k receives a *request* packet, it forwards the packet to adjacent MTs (g_k) if it does not have *d* in its local cache. If n_k has the data *d*, it sends an *ack* packet to n_i. When an AP receives the *request* packet, it simply replies an *ack* packet. When an MT or AP forwards or sends the *ack* packet, the id of the MT or AP is appended in the packet to keep the route information. In contrast to a *request* packet, which is broadcasted, the *ack* packet is sent only along the path, which is accumulated in the *request* packet.
3. When n_i receives an *ack* packet, it sends a *confirm* packet to the *ack* packet sender, e.g. an AP or n_k. Since an *ack* packet arrives earlier from an MT or AP that is closer to n_i, n_i selects the path based on the first receipt of the *ack* packet and discards rest of the *ack* packets.
4. When n_k or an AP receives a *confirm* packet, it sends the requested data (*d*) as using the known route.

When an MT receives a *request* packet, it checks whether the packet has been processed. If the packet has been processed, then the MT does not forward it to adjacent MTs, and discards it. For an *ack*, *confirm*, or *reply* packet, the MT also checks if its id is included in the path, which is appended to the packet. Since these packets are supposed to travel only along the assigned path that is established by the *request* packet, if the MT's id is not included in the path, the packet is discarded. We use a hop limit for a *request* packet to prevent floating of packets in the network. Thus, an MT does not broadcast a *request* packet to the adjacent MTs if the number of forwarded hops of the packet exceeds the hop limit. When the MT or an AP receives a *request* packet, it does not send the data item immediately, but sends an *ack* packet because other MTs or APs, which are located closer to the sender might reply earlier. This helps in reducing network congestion

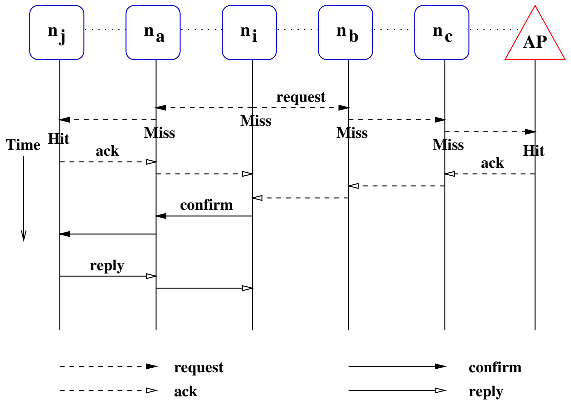


Fig. 3. A *Simple Search* algorithm in the IMANET. Let us assume that an MT (n_i) sent a *request* packet for a data item (*d*) and an MT (n_j) receives a forwarded *request* packet. n_j has the data *d* in its local cache and replies an *ack* packet to n_i. Then n_i sends a *confirm* packet to the n_j, and n_j attaches *d* to the *reply* packet. Here, dotted line between MTs or an MT and AP represents that they are located within communication range.

---

[5] For g_i, $g_i = \{n_j \mid \text{distance}(n_i,n_j) \leqslant \Upsilon\}$, where distance($n_i$,$n_j$) is calculated by $\sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$ and Υ is the diameter of communication range of the MT. The x_i and y_i are the coordinates of n_i.

and bandwidth consumption by multiple data packets.

When a set of MTs is isolated (as shown in Fig. 2) and cannot access the data of their interest because they are out of the communication range of an AP, they try to search among themselves with cached copies.

The proposed SS algorithm is illustrated in Fig. 3, where we assume n_j has the data item in its local cache that n_i requested. Once the MT receives the requested data, it triggers the cache admission control procedure to determine whether it should cache the data item. The cache management scheme is described in the next section.

4. Aggregate cache management

In this section, we present the aggregate cache management policy including a cache admission control and a cache replacement policy.

4.1. An aggregate cache

In IMANETS, caching data items in the local cache helps in reducing latency and increasing accessibility. If an MT is located along the path in which the request packet travels to an AP, and has the requested data item in its cache, then it can serve the request without forwarding it to the AP. In the absence of caching, all the requests should be forwarded to the appropriate APs. Since the local cache of the MTs virtually form an aggregate cache, a decision as to whether to cache the data item depends not only on the MT itself, but also on the neighboring MTs.

In the aggregate cache, a cache hit can be of two types: a local cache hit or a remote cache hit. A local cache hit occurs when the requested data item is available in the MT's local cache. A remote cache hit implies that the data item is available in other MTs' local cache.

4.2. Cache admission control

When an MT receives the requested data, a cache admission control is triggered to decide whether it can cache this data. In this paper, the cache admission control allows an MT to cache a data item based on the distance of other APs or MTs, which have the requested data. If the MT is located within Γ hops from them, then it does not cache the data; Otherwise it caches the data item. Since cached data can be used by closely located MTs, the same data items are cached at least Γ hops apart. Here, Γ is a system parameter.

The primary idea is that, in order to increase accessibility, we try to cache as many data items as possible, while trying to avoid too many replications. There is a tradeoff between access latency and data accessibility in data replication. If the popular data are replicated a lot, then the average access latency to average access is reduced because there is a high probability of finding those data items in another closer MT. With high duplication, however, the number of distinct data items in the aggregate cache is less. Thus, the probability of finding less popular data items from other MTs becomes low. Even though the number of copies of popular data reduces due to the cache admission control, a data is accessible from other MTs/APs with a longer delay.

Although caching popular data aggressively in closer MTs helps in reducing the latency, in this work, we give more weight to data accessibility than to access latency. A rationale behind this is that it is meaningless to reduce access latency when a set of MTs is isolated from other MTs or the AP, and they can not access any interested data items. Instead of waiting until the network topology changes, it is better for the MTs to have even higher probability of finding the requested data. Since Γ value enables more distinct data items to be distributed over the entire cache due to admission control, the overall data accessibility is increased.

4.3. Cache replacement policy

A cache replacement policy is required when an MT wants to cache a data item, but the cache is full, and thus it needs to victimize a data for replacement. Two factors are considered in selecting a victim. The first issue is the distance (δ), measured by the number of hops away from an AP or an MT, which has the requested data. Since δ is closely related to the latency, if the data item with a higher δ is selected as a victim, then the access latency would be high. Therefore, the data item with the least δ value is selected as the victim.

The second issue is the access frequency of data items. Due to mobility of the MTs, the network topology may change frequently. As the topology varies, the δ values become obsolete. Therefore, we use a parameter (τ), which captures the elapsed time of the last updated δ. The τ value is obtained by $\frac{1}{t_{\text{cur}}-t_{\text{update}}}$, where t_{cur} and t_{update} are the current time and the last updated time of δ for the data item, respectively. If τ is closer to 1, δ has recently been updated. If it is closer to 0, the updated gap is long. Thus, τ is used as an indicator of δ to select a victim.

An MT maintains the δ and t_{update} values for each data item in the local cache. The mechanism to update δ and t_{update} is described as follows (refer to Fig. 3):

1. After n_j receives the *confirm* packet, it checks the δ of the requested data item between n_i and n_j. If δ is $\geqslant \Gamma$ and is less than previously saved δ of the data item, then n_j updates the old δ with the new δ. Otherwise, n_j does not update δ, because d will not be cached in n_i based on the cache admission control. The δ value is obtained by counting the number of MTs' ids accumulated in the packet.
2. When n_i receives the data item in the *reply* packet, it checks the δ value of the data between n_i and n_j, and then chooses a victim and replaces it with d, if δ is $\geqslant \Gamma$. In addition, n_i saves δ and t_{cur}, which is t_{update} for the data item.

In this paper, we suggest a *Time and Distance Sensitive* (*TDS*) replacement based on these two parameters. Depending on the weight assigned to the two parameters, we propose three schemes below (refer to Fig. 3):

- *TDS_D*: We mainly consider the distance (δ) value to determine a victim. If there is a tie, then τ is considered the second criteria. We add the two parameters and choose the data item that has the least value of $(\delta + \tau)$. Note that δ is $\geqslant 1$, but τ is in the range of $0 \leqslant \tau \leqslant 1$.

Notations:

$t_{cur}, t_{update}, \tau, \delta$: Defined before.
C_i: A local cache in MT n_i.
d_n: A data item cached in the n^{th} slot in the local cache, where $0 \leq n < C_{size}$ (C_{size} is the cache size).
τ_n: A calculated τ value of d_n.
δ_n: A δ value of d_n.

```
(A) When $n_i$ receives a data item $d$, it calculates $\delta$. /* cache admission control is triggered. */
    if ($\delta \geq \Gamma$) {
        if (empty cache slot is available in $C_i$)
            cache $d$;
        else
            call cache_replacement_policy();
        store $\delta$ and $t_{cur}$, which is saved as $t_{update}$;
    }
    else
        do not cache $d$;

(B) Procedure cache_replacement_policy()
    calculate $\tau$ by $\frac{1}{t_{cur}-t_{update}}$;
    for $d_n \in C_i$ do {
        calculate $\tau_n$;
        find $d_n$ which has the minimum $\delta_n \times \tau_n$ value;
    }
    replace $d_n$ with $d$;
```

Fig. 4. Pseudocode of the aggregate cache management algorithm used in an MT. We use the TDS_N replacement policy. The TDS_D and TDS_T can be implemented by slightly modifying the cache_replacement_policy( ) procedure.

- *TDS_T*: A τ value is mainly considered to determine a victim. Thus, a victim is selected with the least τ value. As we mentioned before, t_{update} is updated when n_j receives the *confirm* packet and n_i receives the *reply* packet. Here, δ of the requested data item between n_i and n_j is $\geqslant \Gamma$.
- *TDS_N*: Both distance and access frequency are considered to determine a victim. We multiply the two factors and select the data item with the least $(\delta \times \tau)$ value.

The TDS_T scheme is different from the traditional *Least Recently Used* (*LRU*) cache replacement policy, which is associated with the time of reference of the data items (t_{ref}). In the LRU scheme, a requested data is cached without considering an admission control policy. Thus, whenever an MT receives the data item in the *reply* packet, one of the local data items that has the highest $(t_{\text{cur}} - t_{\text{ref}})$ value is selected as the victim. In addition, when n_j receives the *confirm* packet and n_i receives the *reply* packet, t_{ref} is updated regardless of the δ values of the requested data item between n_i and n_j.

The overall aggregate cache management algorithm is given in Fig. 4.

5. Performance evaluation

5.1. Simulation testbed

We use a wrap around network to examine the proposed idea. We assume that an AP is located in the center of an area. The MTs are randomly located in the network. The request arrival pattern follows Poisson distribution with a rate of λ. The speed (s) of the MTs is uniformly distributed in the range ($0.0 < s \leqslant 1.0$ m/s). The *random waypoint mobility* model, developed in [10], is used to simulate mobility here. With this approach, an MT travels toward a randomly selected destination in the network. After the MT arrives at the destination, it chooses a rest period (pause time) from a uniform distribution. After the rest period, the MT travels towards another randomly selected destination, repetitively. An MT does not move at all if its pause time is infinite, represented as *Inf*. If the pause time is 0, then it always moves.

Table 1
Simulation parameters

Parameter	Value
Network size (m)	3000 × 3000
Number of MTs	200
Number of data items	1000, 10 000
Cache size (items/MT)	16
Transmission range (m)	250
Number of APs	1, 4, 16
Inter request time (s)	600
Pause time (s)	0, 100, 200, 400, 800, 1600, Inf

To model the data item access pattern, we use two different distributions: Uniform and Zipf distribution [23]. The Zipf distribution is often used to model a skewed access pattern [21,8,1], where θ is the access skewness coefficient that varies from 0 to 1.0. Setting $\theta = 0$ corresponds to the uniform distribution. Here, we set θ to 0.95. We have written an event-driven simulator using CSIM [20] to conduct the performance study. The simulation results are illustrated as a function of the pause time. The other important simulation parameters are summarized in Table 1.

5.2. Simulation metric

We evaluate three performance parameters here: throughput or fraction of successful requests (Φ), average number of hops (Ω), and cache hit ratio (h) including local cache hit and remote cache hit. Throughput Φ denotes the fraction of successful requests and is used to measure the accessibility of the MTs in the IMANET. If r_{total} and r_{suc} denote the total number of requests and the number of successfully received data items, then Φ is defined as

$$\Phi = \frac{r_{\text{suc}}}{r_{\text{total}}} \times 100\%.$$

The average number of hops (Ω) represents the average hop length to the APs or MTs of successfully received data items. If Ω_r denotes the hop length for a successful request r, then Ω is expressed as

$$\Omega = \frac{\sum_{r \in r_{\text{suc}}} \Omega_r}{r_{\text{suc}}}.$$

Since the number of hops is closely related to the communication latency, we use Ω to measure average latency. Finally, the hit ratio h is used to evaluate the efficiency of the aggregate cache management. If n_{local} and n_{remote} denote the number of local hits and remote hits respectively, then h_{local}, h_{remote}, and h are expressed as

$$h_{\text{local}} = \frac{n_{\text{local}}}{n_{\text{local}} + n_{\text{remote}}} \times 100\%,$$

$$h_{\text{remote}} = \frac{n_{\text{remote}}}{n_{\text{local}} + n_{\text{remote}}} \times 100\%,$$

$$h = \frac{n_{\text{local}} + n_{\text{remote}}}{r_{\text{suc}}} \times 100\%.$$

5.3. Simulation results

In this subsection, we examine the impact of caching and cache management including admission control and replacement policy on the IMANET performance. Then we discuss the impact of number of APs. Since there are only few APs available in a given area due to limited resource environment in an IMANET, in all the discussion, we use a single AP unless otherwise stated.

5.3.1. Impact of caching

We investigate the performance implications of the aggregate cache, using two data access patterns: uniform and Zipf distributions. In Fig. 5, the TDS_D and TDS_T cache replacement policies are used for caching with data access pattern of uniform and Zipf distribution, respectively. We have simulated all other policies, but discuss only a subset of the important results. For a system without any cache, an access pattern does not make any performance difference, because a request can not be satisfied by any MT but by an AP.

In Fig. 5(a), data accessibility is greatly improved when we use the aggregate cache. Φ is increased more than twice compared to the no cache case. With caching, there is a high probability of the requested data being cached in the MT's local cache or at other MTs. Even though a set of MTs is isolated from an AP, in contrast to the no cache case, they still try to access the cached data items among themselves. Further improvement is possible depending on the access pattern. Note that almost 200% improvement is achieved compared to the no cache case, when data access pattern follows Zipf distribution.

Fig. 5(b) shows the effect of the aggregate cache on the average latency. Since a request can be satisfied by any one of the MTs located along the path in which the request is relayed to the AP, unlike to the no cache case, data items can be accessed much faster. As expected, Ω is reduced with caching by more than 50%. The results clearly demonstrate the effectiveness of the aggregate caching scheme.

5.3.2. Impact of cache management

In this subsection, we evaluate the cache management policy in terms of the impact of Γ on

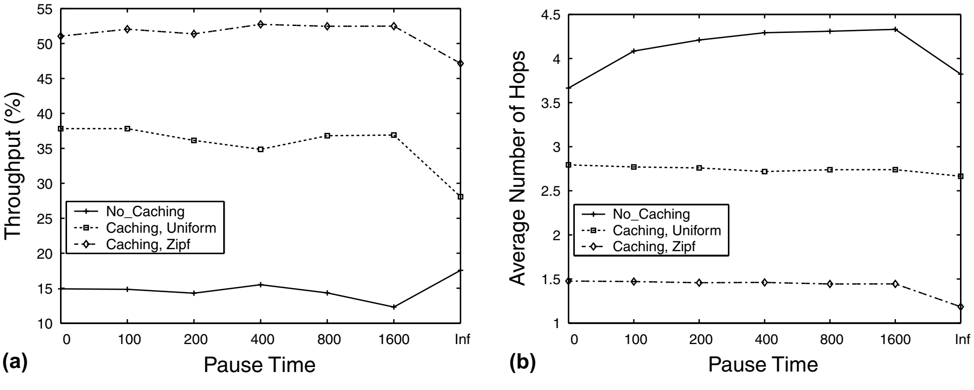


Fig. 5. (a) Throughput (Φ) and (b) latency (Ω) as a function of pause time.

admission control and impact of the cache replacement policy. We compare the performance of our TDS schemes against the *Least Recently Used* (*LRU*) policy.

Impact of Γ on Admission Control: We examine the performance effect of parameter Γ, which determines which data item can be cached. Although a high Γ value enables more data items to be distributed over the entire cache, so that more distinct data items will be cached, the average access latency will increase. In this paper, as mentioned before, data accessibility is considered more important than access latency.

In Fig. 6(a), throughput Φ degrades after $\Gamma = 5$. An MT does not cache a data item according to the admission control policy, when the data is available within five hops. Thus, performance is almost similar to the no cache case at $\Gamma = 6$, because only a few data items are cached. TDS_D has the highest Φ followed by the TDS_N and then TDS_T. Due to the uniform access pattern, δ has more effect on the performance than that of τ. Since TDS_N gives equal importance to δ and τ, it shows higher Φ than TDS_T but lower than TDS_D.

In Fig. 6(b), Φ of all schemes drops after $\Gamma = 5$ for similar reason discussed above. When the access pattern follows the Zipf distribution, however, TDS_T shows the best performance. Since t_{update} of popular data items is more frequently updated than that of less popular data items, there is a high probability of a less popular data item being selected as a victim. Also, the probability of a popular data item to be found in other MTs is high. As the result indicates, τ has more impact on throughput than that of δ. Throughput can be further enhanced by tuning the Γ value.

Impact of Cache Replacement Policy: The impact of the suggested cache replacement polices on performance is investigated with different data access patterns. Based on Fig. 6(a), we set Γ as four, five, and five for TDS_D, TDS_T, and TDS_N policies, respectively. In addition, we simulate the LRU policy for comparison.

In Fig. 7, we use uniform distribution and set the total number of data items to 1000. In Fig. 7(a), as the pause time increases, overall Φ of the TDS schemes and LRU decreases. It implies that the isolation period of a set of MTs from other MTs or the AP becomes longer due to slow movement of MTs. For instance, when an MT does not move (pause time is *Inf*) and is isolated, its data accessibility is very low for the entire simulation time. TDS_D and TDS_N have higher Φ than TDS_T in high mobility. The LRU scheme shows the lowest performance due to data access pattern.

Fig. 7(b) demonstrates the effect of the aggregate cache on the latency, where TDS_D has lower Ω than TDS_T and TDS_N. The LRU scheme shows the lowest Ω because it does not filter an accessed data item but simply caches it.

Fig. 8 shows h_{local} and h_{remote} for different pause times and the h_{remote} is almost up to 90% of h. h_{local} is quite small compared to h_{remote}, because the aggregated cache size is larger than a local cache. Our TDS schemes show higher h than the LRU policy.

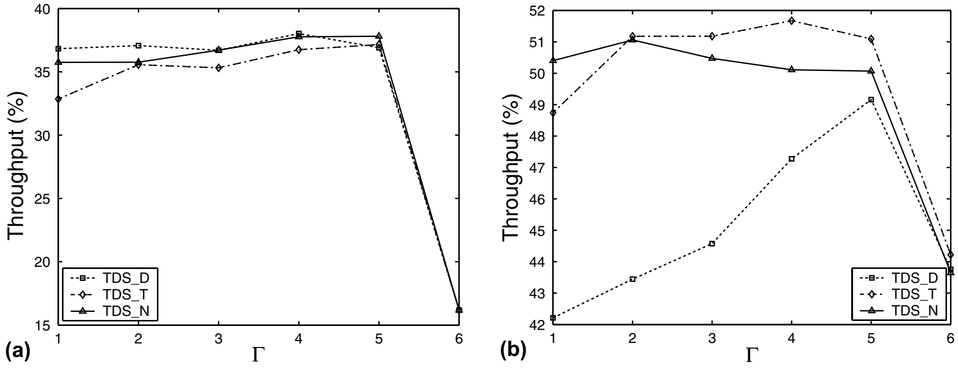


Fig. 6. Throughput (Φ) as a function of Γ: (a) uniform distribution and (b) Zipf distribution.

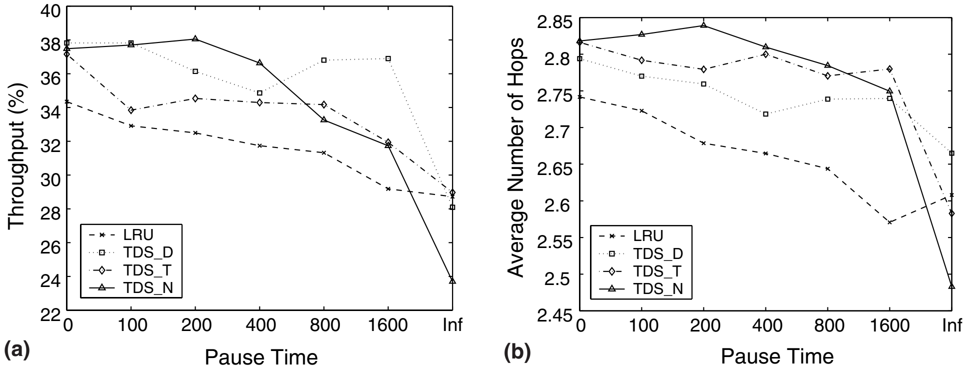


Fig. 7. (a) Throughput (Φ) and (b) latency (Ω) comparison with uniform distribution.

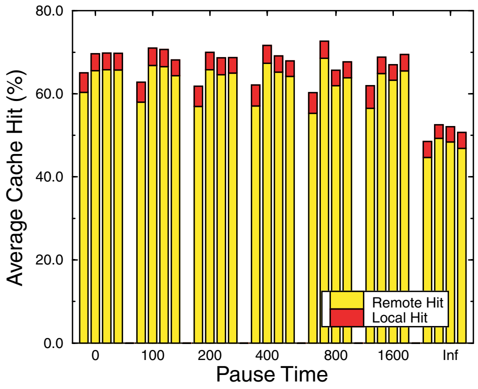


Fig. 8. Average cache hit ratio (h) comparison with uniform distribution (four stack bars for different replacement policies are shown against pause time. The LRU, TDS_D, TDS_T, and TDS_N are plotted from left to right).

Next, we examine the impact of data access pattern using Zipf distribution with 10000 data items in Fig. 9. Based on Fig. 6(b), we set Γ as five, four, and two for TDS_D, TDS_T, and TDS_N, respectively.

Fig. 9(a) demonstrates that the effect of the aggregate cache is more significant when the access pattern follows Zipf distribution. TDS_T has the best performance followed by TDS_N, TDS_D, and LRU. Since the popular data items are frequently requested, all the TDS schemes and LRU gain more benefit compared to the uniform distribution. In general, all variations of TDS replacement show better performance than the LRU scheme with the Zipf distribution. A drawback of the LRU scheme is that is has too much replication of popular data items, and thus results in lower data accessibility. However, with the TDS policies, data items are treated more fairly in the sense that the number of replications for the most popular data items is restricted due to the cache admission control.

In Fig. 9(b), all the TDS schemes have higher Ω than the LRU policy because the cache admission control allows caching only when a data item is Γ hops away. Since the LRU policy caches the requested data without using the admission control, most frequently accessed data items are stored in multiple MTs. Due to higher replication of popular data items, LRU has a smaller Ω.

Even if a data item is popular and can be received from a near by MT, it cannot be cached due to the cache admission control. Thus, the average popularity of local cache in TDS_D, TDS_T, and TDS_N is smaller compared to the LRU policy. For instance, when an MT is isolated, it can only access data items in its local cache. Because of the less popularity of data items, an MT will have less h_{local} for the TDS policies. However, in contrast to the LRU scheme, where the h_{local} is high, the TDS_D, TDS_T, and TDS_N have higher remote cache hit due to cache admission control, which prevents arbitrary data replication. This is shown in Fig. 10. Note that the TDS_T policy has slightly better hit ratio along with its high throughput.

In summary, the aggregate cache improves IMANET performance significantly and the

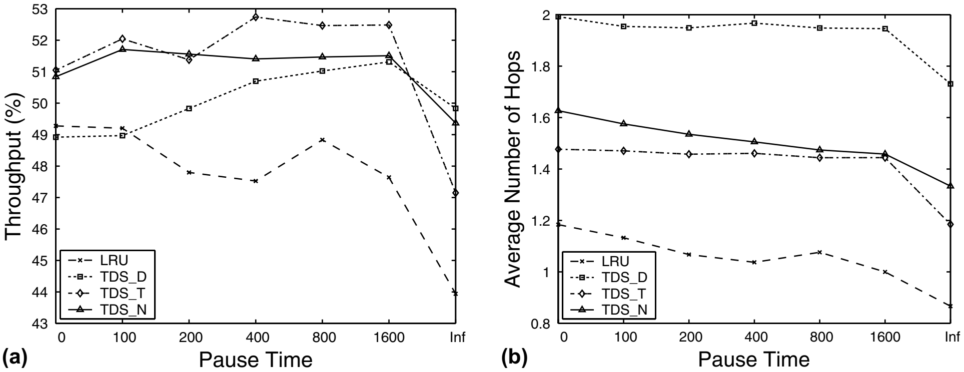


Fig. 9. (a) Throughout (Φ) and (b) latency (Ω) comparison with Zipf distribution.

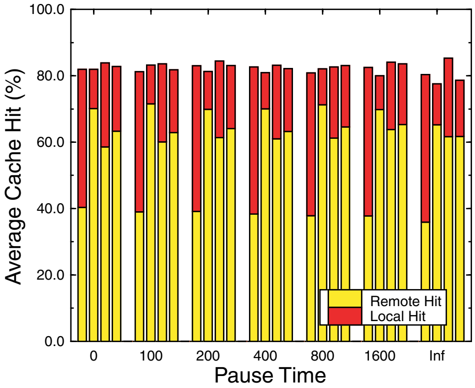


Fig. 10. Average cache hit ratio (h) comparison with Zipf distribution (Four stack bars for different replacement policies are shown against pause time. The LRU, TDS_D, TDS_T, and TDS_N are plotted from left to right).

proposed TDS scheme is a viable cache replacement policy.

5.3.3. *Impact of number of APs*

Since the number of APs can affect the performance in an IMANET, we disable the caching ability of the MTs to study the impact of number of APs. As the number of APs increases, Φ increases up to 90% (at $\text{AP} = 16$). Intuitively, if more APs are deployed in a given area, the probability of an MT being connected to an AP (either directly or indirectly) increases and thus throughput is increased.

For the effect of number of APs on the access latency, as the number of APs increases, Ω reduces as expected. This implies that the accessibility of an MT to an AP increases. These results are not included here since the focus of the paper is on caching, and the results can be found in [12].

6. Concluding remarks

In this paper, we proposed an aggregate caching scheme to improve the communication performance of an IMANET, a ubiquitous communication infrastructure consisting of both the wired Internet and wireless MANET. An IMANET is envisioned to provide access to Internet information and services from anywhere anytime. The aggregate caching concept combines the local cache of each user (MT) in forming an unified cache that can alleviate the limited data accessibility and longer access latency problems. The caching scheme includes a broadcast-based search and a cache management technique. The proposed simple search (SS) algorithm ensures that a requested data is obtained from the nearest MT or AP. The aggregate cache management scheme has two parts: a cache admission control and a cache replacement policy. The admission control prevents high data replication by enforcing a minimum distance between the same data items, while the replacement policy helps in improving the cache hit ratio and accessibility. Three variations of the replacement policy are considered in this paper by assigning different weights to the time and distance parameters of the TDS scheme.

A simulation-based performance study was conducted to examine the advantages of the proposed scheme from three different perspectives: impact of caching, impact of cache management, and impact of number of APs. The three variations of the TDS replacement policy were compared against the traditional LRU policy. It was observed that regardless of the cache replacement policies, caching in IMANETS can significantly improve communication performance in terms of throughput and average access latency compared to an infrastructure without any cache. The performance advantage of the aggregate cache was magnified for skewed access patterns. Also, performance improvement due to caching was better with even a single access point to the Internet.

There are many challenges that need further investigation to exploit the full potential of IMANETS. Currently, we are examining the following issues:

- In this paper, we assumed that data items are never updated. We would relax this assumption to incorporate data modification capability. This brings in the cache invalidation and cache update issues. In an IMANET, cache invalidation and update is challenging because of link disconnection and change of network topology. In light on this, we are currently developing various cache invalidation techniques [11] suitable for IMANETS.
- We did not consider various network topologies that may cause a network partition problem in this paper. Thus, we plan to investigate the impact of the caching scheme on communication performance under different mobility patterns including Manhattan grid and modified random waypoint.

References

[1] G. Cao, A scalable low-latency cache invalidation strategy for mobile environments, in: Proc. ACM MOBICOM, 2000, pp. 200–209.

[2] M.S. Corson, J.P. Macker, G.H. Cirincione, Internet-based mobile ad hoc networking, IEEE Internet Computing 3 (4) (1999) 63–70.

[3] S.G. Dykes, K.A. Robbins, A viability analysis of cooperative proxy caching, in: Proc. IEEE INFOCOM, 2001, pp. 1205–1214.

[4] L. Fan, P. Cao, J. Almeida, A.Z. Broder, Summary cache: a scalable wide-area Web cache sharing protocol, in: Proc. ACM SIGCOMM, 1998, pp. 254–265.

[5] S. Hadjiefthymiades, L. Merakos, Using proxy cache relocation to accelerate Web browsing in wireless/mobile communications, in: Proc. World Wide Web (WWW 10), May 2001, pp. 26–35.

[6] T. Hara, Effective replica allocation in ad hoc networks for improving data accessibility, in: Proc. IEEE INFOCOM, 2001, pp. 1568–1576.

[7] T. Hara, Replica allocation in ad hoc networks with period data update, in: Proc. 3rd International Conference on Mobile Data Management (MDM), 2002, pp. 79–86.

[8] Q. Hu, D.L. Lee, W. Lee, Performance evaluation of a wireless hierarchical data dissemination system, in: Proc. ACM MOBICOM, 1999, pp. 163–173.

[9] Y. Hu, D.B. Johnson, Caching strategies in on-demand routing protocols for wireless ad hoc networks, in: Proc. ACM MOBICOM, 2000, pp. 231–242.

[10] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer, 1996, pp. 153–181.

[11] S. Lim, W. Lee, G. Cao, C.R. Das, Cache invalidation strategies for Internet-based mobile ad hoc networks, in: Proc. 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), October 2004.

[12] S. Lim, S. Park, W. Lee, G. Cao, C.R. Das, C.L. Giles, An aggregate caching for Internet based ad hoc networks, Technical Report CSE–02–017, Deptartment of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, October 2002.

[13] H. Luo, R. Ramjee, P. Sinha, L. Li, S. Lu, UCAN: a unified cellular and ad-hoc network architecture, in: Proc. ACM MOBICOM, 2003, pp. 353–367.

[14] M. Papadopouli, H. Schulzrinne, Effects of power conservation, wireless converage and cooperation on data dissemination among mobile devices, in: Proc. MobiHoc, 2001, pp. 117–127.

[15] V.D. Park, M.S. Corson, Highly adaptive distributed routing algorithm for mobile wireless networks, in: Proc. IEEE INFOCOM, 1997, pp. 1405–1413.

[16] C. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proc. ACM SIGCOMM, 1994, pp. 234–244.

[17] C. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[18] Q. Ren, M.H. Dunham, Using semantic caching to manage location dependent data in mobile computing, in: Proc. ACM MOBICOM, 2000, pp. 210–221.

[19] F. Sailhan, V. Issarny, Cooperative caching in ad hoc networks, in: Proc. 4th International Conference on Mobile Data Management (MDM), 2003, pp. 13–28.

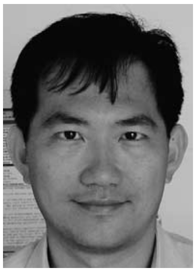
[20] H. Schewetman, CSIM User's Guide (Version 18). MCC Corp. Available from <http://www.mesquite.com>, 1998.

[21] K. Wu, P.S. Yu, J.L. Wolf, Segment-based proxy caching of multimedia streams, in: Proc. World Wide Web (WWW 10), May 2001, pp. 36–44.

[22] T. Ye, H.A. Jacobsen, R. Katz, Mobile awareness in a wide area wireless network of info-station, in: Proc. ACM MOBICOM, 1998, pp. 109–120.

[23] G.K. Zipf, Human Behavior and the Principle of Least Effort, Addison-Wesley, Cambridge, MA, 1949.

Sunho Lim is a Ph.D. candidate in the Department of Computer Science and Engineering. The Pennsylvania State University, University Park. He received his BS degree (summa cum laude) from Department of Computer Science and MS degree from Department of Computer Engineering from Hankuk Aviation University (HAU), Korea, in 1996 and 1998, respectively. His research interests are in the areas of wireless networks and mobile computing with emphasis on the cellular network, wireless LAN, Internet-based mobile ad hoc network, and mobile data management. He was a recipient of the best graduate student teaching assistant award of the Department of Computer Science and Engineering at the Pennsylvania State University in 2002. He is a student member of the IEEE.



Wang-Chien Lee is an Associate Professor of Computer Science and Engineering Department in Penn State University. Prior to joining the faculty at Penn State, he was a principal member of the technical staff at Verizon/GTE Laboratories, Inc. He received his BS from the Information Science Department, National Chiao Tung University, Taiwan, his MS from the Computer Science Department, Indiana University, and his Ph.D. from the Computer and Information Science Department, the Ohio State University. His primary research interests lie in the areas of mobile and pervasive computing, data management, wireless networks and Internet technologies. He has guest-edited special issues on mobile database related topics for several journals, including IEEE Transaction on Computer, IEEE Personal Communications Magazine, ACM WINET and ACM MONET. He was the program committee co-chair for the First International Conference on Mobile Data Access (MDA'99) and the International Workshop on Pervasive Computing (PC2000). He has also been a panelist, session chair, industry chair, and program committee members to various symposia, workshops, and conferences. He is a member of the IEEE and the Association for Computer Machinery.

Guohong Cao received his BS degree from Xian Jiaotong University, Xian, China. He received the MS degree and Ph.D. degree in computer science from the Ohio State University in 1997 and 1999 respectively. Since Fall 1999, he has been an Assistant Professor of computer science and engineering at the Pennsylvania State University. His research interests are mobile computing, wireless networks, and distributed fault-tolerant computing. He currently leads several projects on resource management and data dissemination in mobile environments. He is an editor of the IEEE Transactions on Mobile Computing and IEEE Transactions on Wireless Communications, and has served on the program committee of various conferences including ICDCS, MOBICOM, ICNP and INFOCOM. He was a recipient of the Presidential Fellowship at the Ohio State University in 1999, and a recipient of the NSF CAREER award in 2001.

Chita R. Das received the M.Sc. degree in electrical engineering from the Regional Engineering College, Rourkela, India, in 1981, and the Ph.D. degree in Computer Science from the Center for Advanced Computer Studies, University of Louisiana, Lafayette, in 1986. Since 1986, he has been with the Pennsylvania State University, where he is currently a Professor in the Department of Computer Science and Engineering. His main areas of interest are parallel and distributed computing, cluster computing, mobile computing, Internet QoS, multimedia systems, performance evaluation, and fault-tolerant computing. He is currently an Editor of the IEEE Transactions on Computers and has served on the Editorial Board of the IEEE Transactions on Parallel and Distributed Systems. He is a Fellow of the IEEE and a member of the ACM.b.