# A Novel Caching Scheme for Internet based Mobile Ad Hoc Networks

Sunho Lim, Wang-Chien Lee, Guohong Cao, and Chita R. Das
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
{slim, wlee, gcao, das}@cse.psu.edu

***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. However,** IMANET **has several limitations to fulfill users' demands to access various kinds of information 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** IMANET**. A broadcast based** *Simple Search (SS) algorithm* **and an** *aggregate caching* **mechanism are proposed for improving the information accessibility and reducing average communication latency in** IMANET**. As 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 access points, which are connected to the Internet, through extensive simulation. The simulation results indicate that the proposed aggregate cache can significantly improve an** IMANET **performance in terms of throughput and average number of hops to access data. In particular, with aggregate caching, more than 200% improvement in throughput is achieved compared to the** IMANET **with no cache case, when the access pattern follows a Zipf distribution.**

***Index Terms*— Aggregate cache, Cache admission control, Cache replacement algorithm, Internet based mobile ad hoc network, Simple search algorithm**

## I. INTRODUCTION

Over the past decade, Internet has changed our daily life. With 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 to access the Internet services and information anytime and anywhere. To realize this vision, wireless carriers have been taking steps to deploy the newest 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 is not even available. Thus, researchers in the academic and industry are exploring an alternative technology, called *Mobile Ad Hoc Network* (MANET), for its low cost and ease of deployment.

A significant volume of research work on MANET has appeared in the literature in the past few years [1], [2], [3], [4], [5], [6], [7], [8]. However, research on MANET is primarily focused on developing routing protocols to increase connectivities among MTs in a constantly varying topology. Due to the users' interests in accessing the Internet, it is an important requirement to consider the integration of MANET with the Internet. Thus, to put the MANET technology into the context of real life, we consider an *Internet based* MANET, called IMANET [9], 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 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 IMANET.

However, an IMANET has several constraints. First, not all the MTs can access the Internet. Second, due to MTs' 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 IMANET. 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 caches of the MTs can be considered as an unified large cache for the IMANET. In addition, since information search in IMANET is different from the search engine based approach on the wired Internet, 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 items. As 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.

This research was supported in part by NSF grants CCR-9900701, CCR-0098149, CCR-0208734, and EIA-020 2007.

[1]In this paper, we use the term mobile terminal (MT) to refer to a portable device 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.

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

The rest of this paper is organized as follows. The related work is reviewed in Section II. The system model is introduced in Section III. The simple search algorithm and the aggregate cache management mechanism are presented in Sections IV and V, respectively. Section VI is devoted to performance evaluation and comparisons of various policies. Finally, we concludes the paper with future directions in Section VII.

## II. RELATED WORK

Research on MANET has mainly focused on developing routing protocols such as Destination-Sequenced Distance Vector (DSDV) [6], Dynamic Source Routing (DSR) [3], Ad hoc On Demand Distance Vector (AODV) [7], Temporally-Ordered Routing Algorithm (TORA) [5], and their variations. These algorithms assume that a sender MT knows the locations of receiver MTs 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. 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 [10], [11], [12]. However, no such work has been conducted in an MANET, in which a network topology frequently changes.

In particular in MANET, 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 [1] proposed a replica allocation methods to increase data accessibility in MANET. In this scheme, an MT maintains a limited number of duplicated data items if they are frequently requested. Replicated data 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. Occurrence of update of data item is further considered in [13]. Since an MT cannot access anything when it is isolated from others, replication is an effective means to improve data accessibility. Due to limited size of information that an MT can maintain, however, simply replicating data items and accessing them in MANET cannot fulfill users' requirements to access a wide variety of information databases, which is usually available over the Internet.

To overcome the limited information availability in MANET, similar approaches to IMANET have been suggested. Sailhan et al [8] proposed a cooperative caching scheme in MANET 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 [4] suggested a 7DS architecture, in which a couple of protocols are defined to share and disseminate information among users, which are not necessary to connect to the Internet. Unlike our approach, they focus on data dissemination, and thus, a cache management including a cache admission control and replacement policy in MT's local cache is not well explored.

To the best of our knowledge, none of previous work has touched an aggregated cache based caching scheme along with simple information search algorithm in the realm of IMANET.

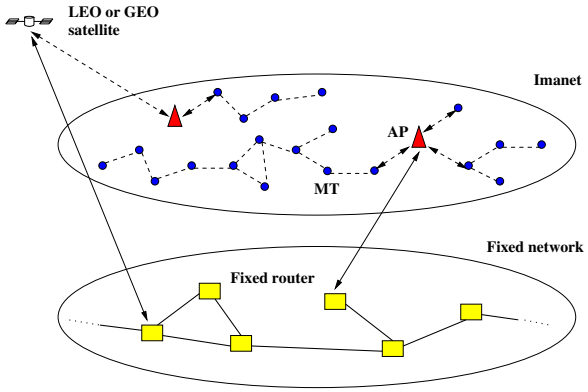



Fig. 1. A system model of IMANET.

## III. SYSTEM MODEL

In this section, we describe the system model of IMANET, which is an evolving architecture of MANET, aimed at increasing both connectivity and accessibility of MTs. We assume that an MT can not only connect to the Internet but also can forward a message for communication with other MTs via wireless LAN (e.g. IEEE 802.11), as used in most prior study [4], [8]. As illustrated in Figure 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]The access point 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.

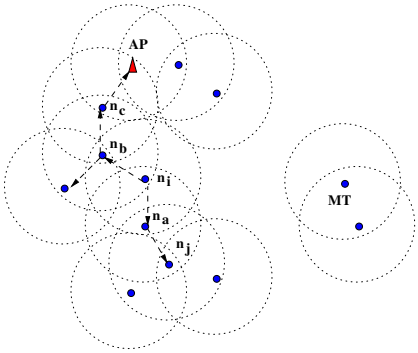



Fig. 2. An MT ($n_i$) broadcasts a request packet which is forwarded to the AP in the IMANET.

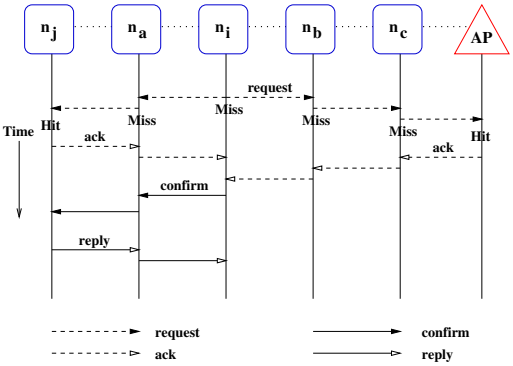



Fig. 3. A *Simple Search* algorithm in the IMANET. Let us assume that an MT ($n_i$) sent a *request* packet for a data item ($d$) and an MT ($n_j$) receives a forwarded *request* packet. $n_j$ has the data $d$ in its local cache and sends an *ack* packet to $n_i$. Then $n_i$ sends a *confirm* packet 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.

## IV. INFORMATION SEARCH IN IMANET

As for information access, the information from the Internet may be cached in some of the MTs within the IMANET. Moreover, any MT can be an information source. Without knowing the whereabout of information, a search algorithm is needed for IMANET as is done in the Internet. In the following, we describe the basic idea of an information search algorithm employed in our proposal, which can be implemented on top of an existing routing protocol for MANET.

Since the concept of an aggregate cache is supported in the IMANET, data items can be received from local caches of the MTs 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. If 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 Figure 2[4], a request may be flooded to the other connected MTs 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 the AP, where $k \in \{a, b, c, j\}$. The SS algorithm is described as follows.

1) When $n_i$ needs $d$, it first checks its local cache. If the data item is not available and $n_i$ cannot directly make a connection to an AP, then $n_i$ 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 requests, 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 replies an *ack* packet. When an AP receives the *request* packet, it simply replies an *ack* packet. When an MT or AP forwards or replies the 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 $n_k$ that has $d$ or an AP receives a *confirm* packet, it sends $d$ as the *reply* packet 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 large number of floating packets from 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 AP receives a *request* packet, it does not send the data item immediately, but sends an *ack* because other MTs or AP, which are located closer to the sender might reply earlier. This helps in reducing network congestion and bandwidth consumption by multiple replied data items.

When a set of MTs is isolated (as shown in Figure 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 them with cached copies. The proposed SS algorithm is illustrated in Figure 3, where we assume $n_j$ has the data item in its local cache what $n_i$ requested.

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

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

Once the MT receives the requested data item, 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.

## V. An Aggregate Cache Management Scheme

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

### A. An Aggregate Cache

In IMANET, 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 MTs' request should be forwarded to the appropriate APs. Since the local caches 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. Therefore, we propose a cache admission control and a cache replacement algorithm.

### B. Cache Admission Control

When an MT receives the requested data item, a cache admission control is triggered to decide whether it can cache this item. 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 item. If the MT is located within $\Gamma$ hops from them, then it does not cache the data item; Otherwise it caches the data item. The cached data items can be used by closely located MTs. Therefore, the same data items are cached at least $\Gamma$ hops apart. Here, $\Gamma$ 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 duplications. Although caching popular data items aggressively in closer MTs helps to reduce the latency, in this work, we give more weight to the 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 high probability of finding the requested data items. Since $\Gamma$ value enables more distinct data items to be distributed over the entire cache due to admission control, more data items can be accessible and thus the overall data accessibility is increased.

### C. A 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 item for replacement. Two factors are considered in selecting a victim. The first factor is the distance ($\delta$), measured by the number of hops away from the AP or MTs, which has the requested data item. Since $\delta$ is closely related to the latency, if the data item with a higher $\delta$ is selected as a victim, then the latency would be high. Therefore, the data item with the least $\delta$ value is selected as the victim.

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

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

1) After $n_j$ receives the *confirm* packet, it checks the $\delta$ of requested data item between $n_i$ and $n_j$. If $\delta$ is $\geq \Gamma$ and is less than previously saved $\delta$ of the data item, then $n_j$ updates the old $\delta$ with the new $\delta$. Otherwise, $n_j$ does not update $\delta$, because $d$ will not be cached in $n_i$ based on the cache admission control. The $\delta$ 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 $\delta$ value of the requested data item between $n_i$ and $n_j$, and then chooses a victim and replaces with $d$, if $\delta$ is $\geq \Gamma$. In addition, $n_i$ saves $\delta$ 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 factors. Depending on the weight assigned to the two factors, we discuss three schemes below. (refer to Figure 3).

- *TDS_D*: We mainly consider the distance ($\delta$) value to determine a victim. If there is a tie, then $\tau$ is considered the second criteria. We add the two factors and choose the data item which has the least value of ($\delta$ + $\tau$). Note that $\delta$ is $\geq 1$, but $\tau$ is in the range of $0 \leq \tau \leq 1$.
- *TDS_T*: A $\tau$ value is mainly considered to determine a victim. Thus, a victim is selected with the least $\tau$ value. As we mentioned before, $t_{update}$ is updated when $n_j$ receives the *confirm* packet and $n_i$ receives the *reply* packet only if $\delta$ of the requested data item between $n_i$ and $n_j$ is $\geq \Gamma$.
- *TDS_N*: Both distance and access frequency are under 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)* replacement policy, which is associated with the time of reference of the data items ($t_{ref}$). In the LRU scheme, a requested data items 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 item which has the highest ($t_{cur}$ - $t_{ref}$) 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

**Notations:**
$d_n$: data item cached in the $n^{th}$ slot in the local cache, where $0 \leq n < C$ ($C$ is the cache size).
$\tau_n$: calculated $\tau$ value of $d_n$.
$L_i$: local cache in mobile terminal $n_i$.

**(A)** When $n_i$ receives $d$, calculates $\delta$. /* cache admission control is triggered. */
**if** ($\delta \geq \Gamma$) {
  **if** (empty cache slot is available in $L_i$) **then**
    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** $d_n \in L_i$ **do** {
  calculate $\tau_n$;
  find $d_n$ which has the minimum $\delta_n \times \tau_n$ value;
}
replace $d_n$ with $d$;

Fig. 4. The pseudo code 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.

$\delta$ values of the requested data item between $n_i$ and $n_j$.

The overall aggregate cache management algorithm is listed in Figure 4.

## VI. Performance Evaluation

### A. 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 MTs' request arrival pattern follows the Poisson distribution with a rate of $\lambda$. The speed ($s$) of the MTs is uniformly distributed in the range ($0.0 < s \leq 1.0$ m/sec). The *random waypoint mobility* model, developed in [3], is used to simulate mobility here. With this approach, the MTs travel toward a randomly selected destination in the network. After they arrive at the destination, they choose a rest period (pause time) from a Uniform distribution. After the rest period, the MTs travel towards another randomly selected destination, repetitively. An MT does not move but stay where it is initially located if the pause time is infinite, represented as Inf. If the pause time is 0, then it always moves.

To model the data item access pattern, we use two different distributions: Uniform and Zipf distribution [14]. The Zipf distribution is often used to model a skewed access pattern [15], [16], [12], where $\theta$ is the access skewness coefficient that varies from 0 to 1.0. Setting $\theta$ = 0 corresponds to the Uniform distribution. Here, we set the $\theta$ to 0.95. We have written an event-driven simulator using CSIM [17] 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 I.

TABLE I
Simulation parameters

| Parameter | Value |
|---|---|
| Network size (m) | 3000 × 3000 |
| Number of MTs | 200 |
| Number of data items | 1000, 10000 |
| cache size (items/MT) | 16 |
| Transmission range (m) | 250 |
| Number of APs | 1, 4, 16 |
| Inter request time (sec) | 600 |
| Pause time (sec) | 0, 100, 200, 400, 800, 1600, Inf |

### B. Simulation Metric

We use three performance parameters: throughput or fraction of successful requests ($\Phi$), average number of hops ($\Omega$), and cache hit ratio ($h$) including local cache hit and remote cache hit. Throughput $\Phi$ 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 $\Phi$ is defined as,

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

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

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

Since the number of hops is closely related to the communication latency, we use $\Omega$ 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:

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

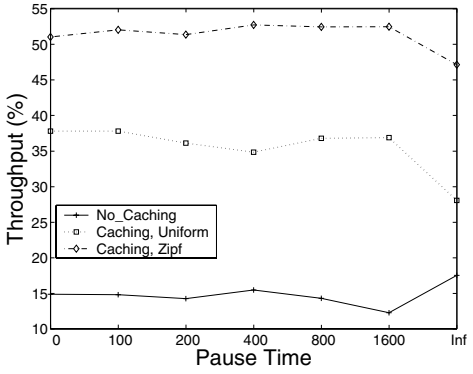
$$h_{remote} = \frac{n_{remote}}{n_{local} + n_{remote}} \cdot 100\% \ ,$$

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

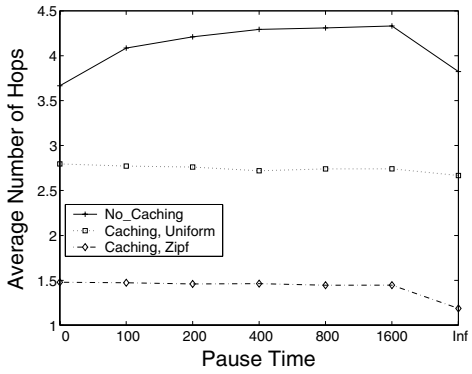
### C. Simulation Results

We have done extensive simulation in terms of the impact of caching, cache management including admission control and replacement policy, and number of APs to analyze various performance metrics. 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. Here, we include a subset of the results due to space limitation. For additional results, please refer to [18].

In Figure 5(a), data accessibility is greatly improved when we use the aggregate cache. Throughput is increased more than twice compared to the no cache case. With caching, there is a high probability of the requested data items being cached in the MT's local cache or at other MTs'. Even though a set of MTs is isolated from the AP, in contrast to the no cache case,




(a) Throughput ($\Phi$)




(b) Latency ($\Omega$)

Fig. 5. Throughput ($\Phi$) and latency ($\Omega$) as a function of pause time.

they still try to access the cached data items among them. Note that almost 200% improvement is achieved compared to the no cache case when data access pattern follows Zipf distribution. Figure 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 item can be accessed much faster. As expected, latency is reduced by more than 50% with caching. The results demonstrate the effectiveness of aggregate caching schemes.

## VII. Concluding Remarks

In this paper, we proposed an aggregate caching scheme to improve the communication performance of IMANET, a ubiquitous communication infrastructure consisting of both wired Internet and wireless MANET. 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 a unified cache that can alleviate the limited accessibility and longer message 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 item 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 duplication by enforcing a minimum distance between the same data items, while the replacement policy helps to improve 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 IMANET can significantly improve communication performance in terms of throughput and average latency compared to an infrastructure without any cache. The performance advantage of aggregate cache was magnified for skewed access pattern. Also, performance improvement due to caching was better with even a single access point to the Internet.

## References


[1] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," in *Proc. IEEE INFOCOM*, 2001, pp. 1568–1576.

[2] Y. Hu and D. B. Johnson, "Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks," in *Proc. ACM MOBICOM*, 2000, pp. 231–242.

[3] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Kluwer 1996, pp. 153–181.

[4] M. Papadopouli and H. Schulzrinne, "Effects of Power Conservation, Wireless Converage and Cooperation on Data Dissemination among Mobile Devices," in *Proc. MobiHoc*, 2001, pp. 117–127.

[5] V. D. Park and M. Corson, "Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in *Proc. IEEE INFOCOM*, 1997, pp. 1405–1413.

[6] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. ACM SIGCOMM*, 1994, pp. 234–244.

[7] C. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *2nd IEEE workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[8] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks," in *Proc. 4th International Conference on Mobile Data Management (MDM), LNCS 2574*, 2003, pp. 13–28.

[9] M. S. Corson, J. P. Macker, and G. H. Cirincione, "Internet-Based Mobile Ad Hoc Networking," in *IEEE Internet Computing*, July–August 1999, pp. 63–70.

[10] S. G. Dykes and K. A. Robbins, "A Viability Analysis of Cooperative Proxy Caching," in *Proc. IEEE INFOCOM*, 2001, pp. 1205–1214.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol," in *Proc. ACM SIGCOMM*, 1998, pp. 254–265.

[12] K. Wu, P. S. Yu, and J. L. Wolf, "Segment-Based Proxy Caching of Multimedia Streams," in *Proc. World Wide Web (WWW 10)*, May 2001, pp. 36–44.

[13] 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.

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

[15] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments," in *Proc. ACM MOBICOM*, 2000, pp. 200–209.

[16] Q. Hu, D. L. Lee, and W. Lee, "Performance Evaluation of a Wireless Hierarchical Data Dissemination System," in *proc. ACM MOBICOM*, 1999, pp. 163–173.

[17] H. Schewetman, *CSIM User's Guide (Version 18)*. MCC Corp., http://www.mesquite.com, 1998.

[18] S. Lim, S. Park, W. Lee, G. Cao, C. R. Das, and C. L. Giles, "An Aggregate Caching for Internet based Ad Hoc Networks," Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park,PA 16802, Tech. Rep. CSE–02–017, Oct 2002.