Mobility Management And Caching Of Data Packets In Heterogeneous Ad Hoc Network

Simran Choudhary
Research scholar
Department of Computer Science & Engineering
M.B.M Engineering College
J.N.V. University, Jodhpur, India
srtcgwala@yahoo.co.in

Dr. Anil Gupta
Associate Professor
Department of Computer Science & Engineering
M.B.M Engineering College,
J.N.V. University, Jodhpur, India
anilgupta@jnvu.edu.in

***Abstract*—With the evolution of mobile networks beyond 4G, multimedia streaming is becoming more and more urgent. When the three-tier system architecture is applied to mobile networks for improving the transmission quality, it must take user mobility into consideration. Multicasting consists of concurrently sending the same information from a source to a subset of all possible destinations in a network. Multicast service is becoming a key requirement of computer networks supporting 4G & multimedia streaming applications. To carry large numbers of multicast session's, a network must minimize the session's resource consumption, while meeting their quality of service requirements [13]. In this paper, we address the issues involved in information search and access in Mobile ad hoc network. A combined caching mechanism and a Broadcast-Based Search algorithm are proposed for improving the information accessibility and reducing average communication latency in MANET. As a part of the combined cache admission control policy and a cache replacement policy, called Sensitive Cache 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 wireless network, through extensive simulation. The simulation results indicate that the proposed combined caching mechanism can significantly improve a MANET performance in terms of throughput and average number of hops to access data items.**

***Keywords*—hand-off; caching; pause time; multicast**

I. Introduction

Over the past decade, the recent advent in wireless technology and mobile devices has changed our daily life. It is envisaged that in the near future, users will be able to access the network services and information anytime and anywhere. To realize this ubiquitous communication, wireless carriers are developing state-of-the-art wireless communication infrastructures. Nevertheless, a mobile client (MC) may still have difficulty to connect to a wired network or Internet due to limited wireless bandwidth and accessibility. Under heavy traffic, a MC has to struggle 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 had explored an alternative technology, called Mobile Ad Hoc Network (MANET), for its ease of deployment. A significant volume of research on MANET has appeared in the literature in the past few years [1, 2 and 14]. Most of these efforts, however, have focused on developing routing protocols to increase connectivity among MCs in a constantly varying topology. We investigate the problem of information search and access under this environment. We assume that some of the MCs are connected to the Internet or wired private networks. Thus, a MC may access Internet information via a direct connection or via relays from other MCs. Although there may exist many potential applications, to the best of my knowledge, least of the previous work has addressed the issues for information search and access.

II. RELATED WORK

The performance of wireless network can be increased by caching. A lot of research has been conducted to reduce the traffic and overall network congestion by deploying various caching schemes in the Internet [3, 11]. 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. Some of them are summarized as follows.

A. Cooperative Caching Scheme

In this scheme 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 [11, 19]. It is implemented on top of a well-known ad hoc routing protocol, called Zone Routing Protocol (ZRP).

B. Summary Cache Scheme

Here 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 [1].

C. Proxy Cache Relocation

In this scheme the prediction of user's mobility is done 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 [1]. However, no such work has been conducted in a MANET, in which a network topology frequently changes.

D. Semantic Caching Scheme

Manage location-dependent data (e.g. weather, traffic, and hotel information), in which a MC maintains semantic description of data in a mobile environment. When an MC 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 MC tailors or reduces the query and requests the server to get the rest of results to reduce communication [10, 15]. 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 MC might move, but is located far away from the current location of the MC.

E. Replica Allocation Method

This method increase data accessibility in MANET. In this scheme, a MC 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 each MC access frequency, the neighbor MC access frequency or overall network topology [2]. Since a MC cannot access data when it is isolated from others, replication is an effective means to improve data accessibility.

F. 7DS Architecture

This scheme has couple of protocols 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 search for data item in a single-hop multicast basis. Depending on the collaborative behavior, a peer-to-peer and server-to-client model is used [14]. 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 my knowledge, none of previous work has explored a combined caching scheme along with an efficient information search algorithm in the realm of MANET.

III. SYSTEM MODEL

We assume that a MC can not only connect to the wired network but also can forward a message for communication with other MCs via a wireless LAN (e.g. IEEE 802.11). As shown in Fig. 1, a MANET consists of a set of MCs that can communicate with each other using an ad hoc communication protocols (illustrated by dashed line). Among the MCs, some of them can directly connect to the network, and thus serve as access points (AP) for the rest of MCs in the MANET. A MC located out of the communication bound of an AP has to access the network via relays through one of the access points. A MC can move in any direction and make information search and access requests from anywhere in the covered area. When a MC is located nearby an AP (e.g. within one-hop), it makes a connection to the AP directly. When a MC 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.

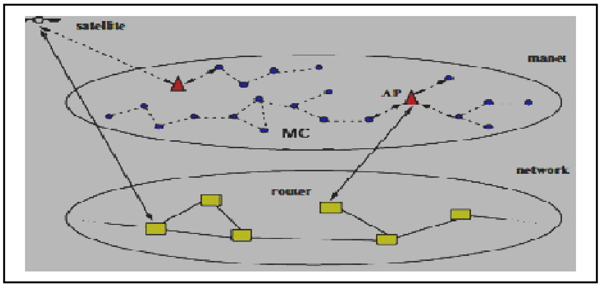

Fig. 1. A system model for manet.

IV. INFORMATION SEARCH ALGORITHM

As mentioned in the introduction, the main focus of this paper is to support information access in MANET. Unlike a routing protocol, which establishes a path between a known source and destination, any MC can act as an information source in the MANET [7, 8, 9]. Thus, without knowing the destination address for any requested information, a search algorithm is needed for MANET. This algorithm can be implemented on top of an existing routing protocol for MANET. Since a combined cache is supported in a MANET design, requested data items can be received from the local cache of a MC as well as via an AP connected to the Internet. When a MC 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 MCs. When a MC 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, a request may be flooded in the network and eventually acknowledged by an AP and/or some MCs with cached copies of the requested data item.

A. Broadcast Based Search Algorithm

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

1) When n_i needs data item 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 MCs. 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 time period, it fails to get d.

2) When n_k receives a request packet, it forwards the packet to adjacent MCs 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 a MC or AP forwards or sends the ack packet, the id of the MC 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 a MC 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 using the known route. When a MC receives a request packet, it checks whether the packet has been processed. If it is, then the MC does not forward it to adjacent MCs, and discards it. For an ack, confirm, or reply packet, MC 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 MCs id is not included in the path, the packet is discarded.
5) We use a hop limit for a request packet to prevent floating of packets in the network. Thus, a MC does not broadcast a request packet to the adjacent MCs if the number of forwarded hops of the packet exceeds the hop limit.
6) When the MC or an AP receives a request packet, it does not send the data item immediately, but sends an ack packet because other MCs or APs, which are located closer to the sender, might reply earlier. This helps in reducing network congestion and bandwidth consumption by multiple data packets.
7) When a set of MCs is isolated 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 BBS algorithm is illustrated in Fig. 2, where we assume n_j has the data item in its local cache that n_i requested. Once the MC receives the requested data, it triggers the cache admission control procedure to determine whether it should cache the data item.

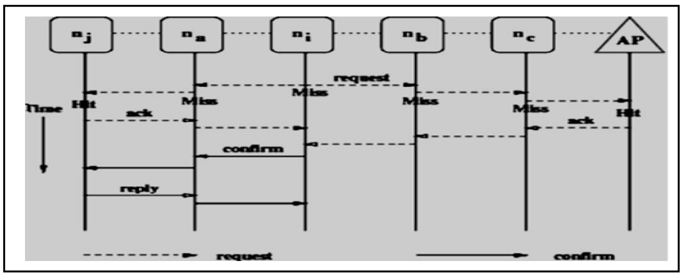

Fig. 2. Broadcast based search algorithm for manet.

V. COMBINED CACHE MECHANISM

In MANET the data items are cache in the local cache which helps in reducing latency and increasing accessibility. If a MC 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 [6]. Since the local cache of he MCs virtually form a combined cache, a decision as to whether to cache the data item depends not only on the MC itself, but also on the neighboring MCs. In the combined 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 MCs local cache. A remote cache hit implies that the data item is available in other MCs local cache.

A. *Cache Admission Control*

When a MC 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 a MC to cache a data item based on the distance of other APs or MCs, which have the requested data. If the MC 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 MCs, 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 MC. With high duplication, however, the number of distinct data items in the combined cache is less. Thus, the probability of finding less popular data items from other MCs becomes low. Even though the number of copies of popular data reduces due to the cache admission control, a data is accessible from other MCs/APs with a longer delay. Although caching popular data aggressively in closer MCs 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 MCs is isolated from other MCs or the AP, and they cannot access any interested data items. Instead of waiting until the network topology changes, it is better for the MCs 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.

B. *Cache Replacement Policy*

A cache replacement policy is required when a MC 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.

1) Distance (δ): The first issue is the distance (δ), measured by the number of hops away from an AP or a MC, 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.

2) Access frequency (τ): The second issue is the access frequency of data items. Due to mobility of the MCs, 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 $1/t_{cur}-t_{update}$, 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. A MC 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 Figure. 2):

a) 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 ≥ Γ 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 δ will not be cached in n_i based on the cache admission control. The δ value is obtained by counting the number of MCs ids accumulated in the packet.

b) 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 δ, if δ is ≥ Γ. In addition, n_i saves δ and tcur, which is t_{update} for the data item.

C. Sensitive Cache Replacement

In this paper, we suggest a Sensitive Cache Replacement (SCR) based on Distance and Time parameters. Depending on the weight assigned to the two parameters, we propose three schemes below:

1) SCR_D: In this scheme 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 (δ + τ). Note that δ is ≥ 1, but τ is in the range of 0 ≤ τ ≤ 1.

2) SCR_T: In this scheme τ value is considered to determine a victim. Thus, a victim is selected with the least τ value. As we mentioned before, tupdate 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 ≥ Γ.

3) SCR_N: Both distance and access frequencies are considered to determine a victim. We multiply the two factors and select the data item with the least (δ×τ) value.

The SCR_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 a MC receives the data item in the reply packet, one of the local data items that 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 δ values of the requested data item between n_i and n_j.

D. Combined Cache Management Algorithm

The overall pseudo code of the combined cache management algorithm used in a MC is as follows.

NOTATIONS:

δ: Distance between two MCs or between an AP and MC, which has the requested data
τ : Elapsed time of last updated δ
t_{cur} : Current time of δ for data item
t_{update} : Last updated time of δ for data item
C_i: A local cache in MC(n_i)
d_i: 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.

```
1) When MC(n_i) receives a data item d, it calculates δ. /* Cache admission control is triggered */
        if(δ ≥ τ) {
        if (empty cache slot is available in C_i)
        cache d;
        else
        call cache_replacement_ policy();
        Store δ & t_cur, which is saved as t_update;
        }
        else
        do not cache d;
2) Procedure cache_replacement_policy()
        calculate τ by 1/t_cur-t_update ;
        for d_n ∈ C_i do {
        calculate τ_n;
        find dn which has the minimum δ_n×τ_n value;
        }replace d_n with d;
```

We use the SCR_N replacement policy. The SCR_D and SCR_T can be implemented by slightly modifying the cache_replacement_policy() procedure.

3) Complexity Analysis: For each data item d_n received by MC (n_i) a calculation of τ_n will be performed and a d_n which has the minimum $\delta_n \times \tau_n$ value will be selected. For that n linear calculation will be performed. Other steps are taking constant unit time. Thus, the time complexity of algorithm is θ(n).

For each data item d_n received by MC (n_i) a unit space will be required. Thus for execution of this algorithm for n items require θ(n) space.

VI. EXPERIMENTAL RESULTS

We assume that an AP is located in the center of an area. The MCs are randomly located in the network. The request arrival pattern follows Poisson distribution with a rate of λ. The speed (s) of the MCs is uniformly distributed in the range ($0.0 < s \leq 1.0$ m/s). The random waypoint mobility model is used to simulate mobility. With this approach, a MC travels toward a randomly selected destination in the network. After MC arrives at the destination, it chooses a rest period (pause time) from a uniform distribution. After the rest period, MC travels towards another randomly selected destination, repetitively. A MC does not move at all if its 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. The zipf distribution is often used to model a skewed access pattern, where h is the access skewness coefficient that varies from 0 to 1.0. Setting h = 0 corresponds to the uniform distribution. Here, we set h to 0.95. We e used Qualnet simulator to conduct the performance study. The other simulation parameters are summarized in Table I.

TABLE I. SIMULATION PARAMETERS

Parameter	*Value*
Network size (m)	3000 x 3000
Number of MCs	200
Number of data items	1000, 10000
Cache size (items/MC)	16
Transmission range (m)	250
Number of AP	1, 4, 16
Inter request time (s)	600
Pause time (s)	0, 100, 200, 400, 800, 1600, Inf

A. *Simulation Metric*

We evaluate three performance metrics 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 fraction of successful requests and is used to measure the accessibility of MCs in the MANET. 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

$$\phi = r_{suc}/r_{total} \times 100\%$$

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

$$\Omega = \Sigma_{r \in rsuc} \ \Omega_r / r_{suc}$$

Since the number of hops is closely related to communication latency, we use Ω to measure average latency. Finally, the hit ratio h is used to evaluate the efficiency of the combined 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

$$h_{local} = n_{local} / n_{local} + n_{remote} \ x \ 100\%$$

$$h_{remote} = n_{remote} / n_{local} + n_{remote} \ x \ 100\%$$

$$h = n_{local} + n_{remote} / r_{suc} \ x \ 100\%$$

B. *Simulation Results*

Since there are only few APs available in a given area due to limited resource environment in an MANET, in all the cases, we use a single AP unless otherwise stated.

Impact of caching

We investigate the performance inference of the combined cache, using two data access patterns: uniform and zipf distributions. In Figure.3, Figure.4, the SCR_D and SCR_T cache replacement policies are used for caching with data access pattern of uniform and zipf distribution, respectively. In Figure.3, data accessibility is greatly improved when we use the combined cache. Throughput 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 MCs local cache or at other MCs. Even though a set of MCs is isolated from an AP, in contrast to the no cache case, they still try to access the cached data items among themselves. Note that almost 200% improvement is achieved compared to the no cache case, when data access pattern follows zipf distribution. Figure.4 shows the effect of the combined cache on the average latency. Since a request can be satisfied by any one of the MCs 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, latency Ω is reduced with caching by more than 50%. The results clearly demonstrate the effectiveness of the aggregate caching scheme.

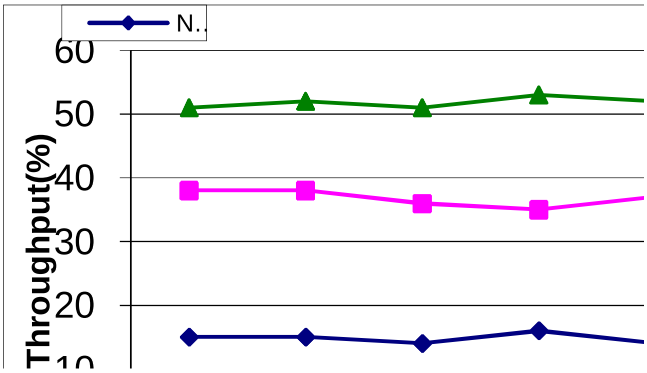

Fig. 3. Throughput (ϕ) as a function of pause time.

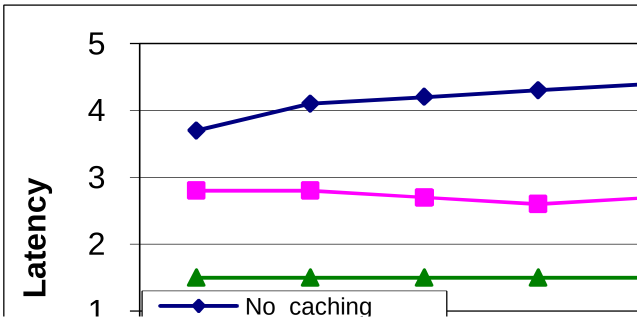

Fig. 4. Latency (Ω) as a function of pause time.

Impact of cache management

We evaluate the cache management policy in terms of the impact of Γ on admission control and impact of the cache replacement policy. We compare the performance of our SCR schemes against the Least Recently Used (LRU) policy.

Impact of Γ on Admission Control

We examine the performance effect of system parameter Γ(atleast no. of hop), 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 Figure. 5, throughput ϕ degrades after Γ = 5. A MC 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 Γ = 6, because only a few data items are cached. SCR_D has the highest ϕ followed by the SCR_N and then SCR_T. Due to the uniform access pattern, δ has more effect on the performance than that of τ. Since SCR_N gives equal importance to δ and τ, it shows higher ϕ than SCR_T but lower than SCR_D. In Figure. 6, ϕ of all schemes drops after Γ = 5 for similar reason discussed above. When the access pattern follows the zipf distribution, however, SCR_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 MCs 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.

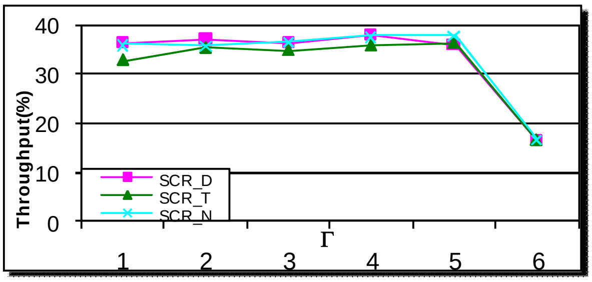

Fig. 5. Throughput (ϕ) as a function of Γ: uniform distribution.

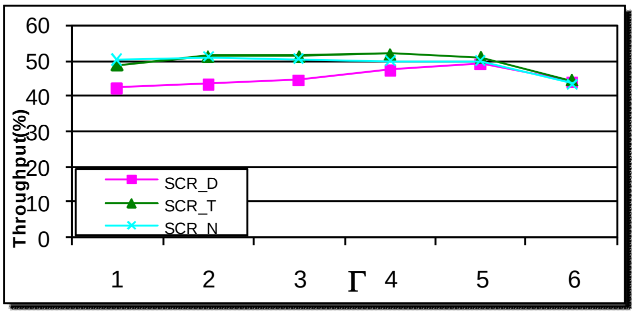

Fig. 6. Throughput (ϕ) as a function of Γ: Zipf distribution.

Impact of Cache Replacement Policy

The impact of the suggested cache replacement polices on performance is investigated with different data access patterns. Based on Figure. 6, we set Γ as four, five, and six for SCR_D, SCR_T, and SCR_N policies, respectively. In addition, we simulate the LRU policy for comparison. In Figure. 7, Figure.8, we use uniform distribution and set the total number of data items to 1000. In Figure. 7, as the pause time increases, overall ϕ of the SCR schemes and LRU decreases. It implies that the isolation period of a set of MCs from other MCs or the AP becomes longer due to slow movement of MCs. For instance, when a MC does not move (pause time is Inf) and is isolated, its data accessibility is very low for the entire simulation time. SCR_D and SCR_N have higher ϕ than SCR_T in high mobility. The LRU scheme shows the lowest performance due to data access pattern.

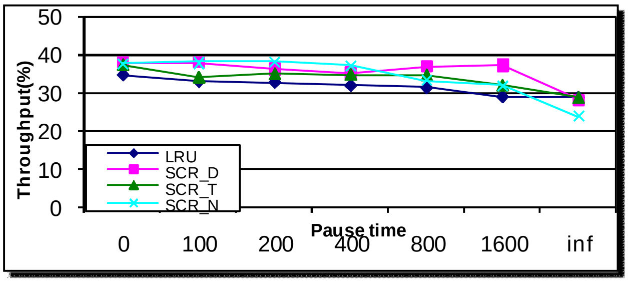

Fig. 7. Throughput (ϕ) as a function of pause time: uniform distribution.

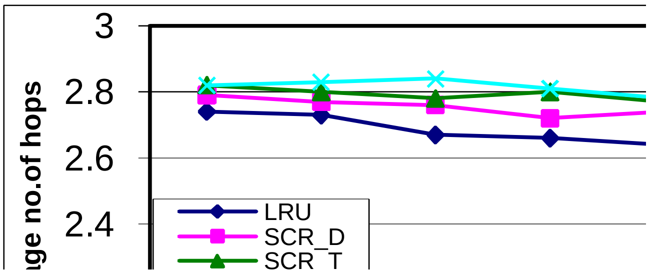

Fig. 8. Latency (Ω) as a function of pause time: uniform distribution.

Figure. 8 demonstrate the effect of the combined cache on the latency, where SCR_D has lower Ω than SCR_T and SCR_N. The LRU scheme shows the lowest Ω because it does not filter an accessed data item but simply caches it. A drawback of the LRU scheme is that it has too much replication of popular data items, and thus results in lower data accessibility. However, with the SCR 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. Since the LRU policy caches the requested data without using the admission control, most frequently accessed data items are stored in multiple MCs. 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 nearby MC, it cannot be cached due to the cache admission control. Thus, the average popularity of local cache in SCR_D, SCR_T, and SCR_N is smaller compared to the LRU policy. For instance, when an MC is isolated, it can only access data items in its local cache. Because of the less popularity of data items, an MC will have less h_{local} for the SCR policies. However, in contrast to the LRU scheme, where the h_{local} is high, the SCR_D, SCR_T, and SCR_N have higher remote cache hit due to cache admission control, which prevents arbitrary data replication.

Impact of number of AP

Since the number of APs can affect the performance in a MANET, we disable the caching ability of the MCs to study the impact of number of APs. As the number of APs increases, ϕ increases up to 90% (at AP = 16). Intuitively, if more APs are deployed in a given area, the probability of a MC 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 a MC to an AP increases.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a combined caching scheme to improve the communication performance of a MANET. The combined caching concept combines the local cache of each user (MC) in forming 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 Broadcast Based search (BBS) algorithm ensures that a requested data is obtained from the nearest MC or AP. The combined 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 distance and time parameters of the SCR 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 AP. The three variations of the SCR replacement policy were compared against the traditional LRU policy. It was observed that regardless of the cache replacement policies, caching in MANET could significantly improve communication performance in terms of throughput and average access latency compared to an infrastructure without any cache. The performance advantage of the combined cache was magnified for skewed access patterns.

There are many challenges that need further investigation to exploit the full potential of MANETS. 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 MANET, cache invalidation and update is challenging because of link disconnection and change of network topology. We did not consider various network topologies that may cause a network partition problem. Thus, there is a need to investigate the impact of the caching scheme on communication performance under different mobility patterns including modified random waypoint.

REFERENCES

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

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

[3] Y. Hu, D.B. Johnson, “Caching strategies in on-demand routing protocols for wireless ad hoc networks”, in Proceedings of ACM MOBICOM, 2000, pp. 231–242.

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

[5] S. Lim, W. Lee, G. Cao, C.R. Das, “Cache invalidation strategies for mobile ad hoc networks”, in Proceedings of 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), October, 2004.

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

[7] V.D. Park, M.S. Corson, “Highly adaptive distributed routing algorithm for mobile wireless networks”, in Proceedings of IEEE INFOCOM, 1997, pp. 1405–1413.

[8] C. Perkins, P. Bhagwat, “Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers”, in Proceedings of ACM SIGCOMM, 1994, pp. 234–244.

[9] 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.

[10] Q. Ren, M.H. Dunham, “Using semantic caching to manage location dependent data in mobile computing”, in Proceedings of ACM MOBICOM, 2000, pp.210–221.

[11] F. Sailhan, V. Issarny, “Cooperative caching in ad hoc networks”, in Proceedings of 4th International Conference on Mobile Data Management (MDM), 2003, pp. 13–28.

[12] S.A. Hosseini, R. Budiarto, T. Wan, “Survey and new approach in service discovery and advertisement for mobile ad hoc networks”, IJCSNS 7 (2), 2007, pp.275–284.

[13] L. Chunlin, L. Layuan, “A QoS multicast routing protocol for clustering mobile ad hoc networks”, Computer Communication 30 (7) 1641–1654, (2007).

[14] J. Xu, Q. Hu, W. C. Lee, and D. L. Lee, “Performance Evaluation of an Optimal Cache Replacement Policy for Wireless Data Dissemination

under Cache Consistency”, IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.1, 2004.

[15] Q. Ren, M. H. Dunham, “Using Semantic Caching to Manage Location Dependent Data in Mobile Computing”, in Proceedings Sixth Ann. ACM/IEEE International Conference, Mobile Computing and Networking (MobiCom 2000), 2000, pp. 85-90.

[16] K. Lai, Z. Tari, and P. Bertok, “Mobility Aware Cache Replacement for Location Dependent Information Services”, Technical Report TR-04-04 (RMIT School of CS & IT), Tampa, Florida, USA, November 16-November 18, 2004, pp. 50-58.

[17] B. Zheng, J. Xu, and D. K.Lee, “Cache Invalidation and Replacement Strategies for Location-Dependent data in Mobile Environments”, IEEE Transactions on Computers, 51(10): 1141-1153, October 2002.

[18] G. Anandharaj, Dr. R. Anitha “A Power-Aware Low-Latency Cache Management Architecture for Mobile Computing Environments,” International Journal of Computer Science and Network Security, VOL.8 No.10, October 2008

[19] Narottam Chand, R.C. Joshi and Manoj Misra, “Cooperative Caching in Mobile Ad Hoc Networks Based on Data Utility,” International Journal of Mobile Information Systems, 3(1): 19-37, 2007.

[20] L. Filipponi , A. Vitaletti , G. Landi , V. Memeo , G. Laura and P. Pucci "Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors", *Proc. 4th IEEE Int. Conf. Sensor Technol. Appl.*, pp.281 -286 2010

[21] G. Cardone , A. Corradi and L. Foschini "Cross-network opportunistic collection of urgent data in wireless sensor networks", *Comput. J.*, vol. 54, no. 12, pp.1949 -1962 2011.

[22] Z. Jun , D. Simplot-Ryl , C. Bisdikian and H. T. Mouftah "The internet of things", *IEEE Commun. Mag.*, vol. 49, no. 11, pp.30 -31 2011.