

# **EDGC: Efficient Dynamic Group Caching Technique for Mobile Ad hoc Networks**

**Atul Rao**

Department of Computer  
Science and Engineering

National Institute of  
Technology, Hamirpur, India

**Prashant Kumar**

Department of Computer  
Science and Engineering

National Institute of  
Technology, Hamirpur, India

**Naveen Chauhan**

Department of Computer  
Science and Engineering

National Institute of  
Technology, Hamirpur, India

## **ABSTRACT**

Mobile Ad hoc Networks are wireless networks which are composed of mobile nodes without any kind of wired infrastructure. Dynamic topology is one of the crucial ingredients. There is a problem in accessing the data items because of Ad hoc Nature. Caching can help in mobile ad hoc networks to improve the accessibility and availability of data items. In this paper, we propose a novel caching scheme called Efficient Dynamic Group Caching (EDGC) which allows grouping of mobile hosts at one hop distance. Sub Group Head and Group Master will manage the group. By using the proposed Efficient Group Caching, the caching space in nodes can be efficiently utilized and thus the redundancy of cached data is decreased and the average access latency is reduced.

## **General Terms**

Stability Factor, Group Caching,

## **Keywords**

Group Caching, MANETs, cache, query latency, Group Head, Group Master

## **1. INTRODUCTION**

With the recent growth in wireless communication technologies and mobile devices, wireless networks have become a pervasive communication infrastructure. In addition, Mobile Ad hoc Networks (MANETs) are networks that have drawn attention. A mobile Ad hoc network is a collection of wireless nodes that can dynamically be set up anywhere and anytime without using any pre-existing network infrastructure. It is an autonomous system in which mobile nodes connected by wireless links are free to move randomly and often act as routers at the same time. MANETs are actually self-organizing and adaptive networks that can be formed and deformed on-the-fly without the need of any centralized administration [4]. The two main characteristics of MANETs are mobility and multi hop communication. MANETs applications include military scenarios, sensor networks, rescue operations etc. With the recent advent in wireless technologies and mobile devices, wireless networks have become a ubiquitous communication infrastructure. Caching is widely exploited in MANETs for query processing with minimum query latency. There are many strategies available to use caching for the better performance of the network. Push and Pull are two basic cache sharing techniques in MANETs [3]. With Push-based cache sharing,

when a node acquires and caches a new data, it actively advertises the caching event to the nodes in the neighborhood. With pull based cache sharing, when a mobile node wants to access data item that is not cached locally it will broadcast a request to the nodes in its vicinity. A nearby node that has cached the data will send a copy of the data to the request originator (a pull operation) unlike pushing; pulling allows the node to utilize the latest cache contents. The original idea of caching is that the data accessed by the mobile nodes (MNs) has the properties of temporal and spatial locality [1, 6]. Higher temporal and spatial locality ensures that most accesses will go to the data that were accessed recently in the past and that resides in the cache. Due to caching frequently requested data can improve the performance of data communication. In a MANETs, if the mobile nodes cache some frequently requested data, the cached data can be served for others later. The requester node needs not to retrieve the data from the remote server or data source. The request will be fulfilled from a neighboring node who has cached the data, this eventually enhance the data accessibility.

There are some demands and issues in MANETs such as mobility of mobile nodes, power consumption of battery, and limited wireless bandwidth, which need to be addressed. [2]. Due to the movement of mobile nodes, network may be divided into many independent networks. Therefore the requestor cannot retrieve data from the data source. In existing caching techniques data is cached at every mobile node so that accessibility can be increased. These caching techniques have many problems like replication of data, power consumption and cache replacement. Therefore in this paper, we propose a novel caching scheme called Dynamic Group Caching in which data is cached at the group head and at the group master so that efficiency can be increased in terms of accessibility, minimized query latency and less power consumption at MN. Based on the proposed caching scheme the replication of data objects can be reduced. The Section 2 describes the related work, section 3 describes the Group Management, Section 4 discussed about cache management, section 5 described the Cache Access Cost Model and Section 6 concludes the paper.

## **2. RELATED WORK**

Few caching schemes for MANETs have been proposed in the literature. In this section, we provide a review of some of those that provide solutions to the caching problem. Guohong Cao et al. propose these three related caching schemes. The CacheData scheme [3] considers the cache placement policy

at intermediate nodes in the routing path between the source and the destination. The node caches a passing-by data item locally when it finds that the data item is popular, i.e., there were many requests for the data item, or it has enough free cache space. Since CacheData needs extra space to save the data, it should be used prudently. A conservative rule is proposed as follow: A node does not cache the data if all requests for the data are from the same node. However, there is no cooperative caching protocol among mobile nodes. Each mobile node independently performs the caching tasks such as placement and replacement. CachePath is also proposed for redirecting the requests to the caching node. In MANETs, the network topology changes fast and thus, the cached path may become invalid due to the movement of mobile node. The third scheme, HybridCache, is a middle solution where queries are cached by path or by data, depending on what scheme is optimal.

Yu Du et al. Presents COOP, a novel cooperative caching scheme [1] for on-demand data access applications in MANETs. The objective is to improve data availability and access efficiency by collaborating local resources of mobile nodes. The cooperation of caching nodes is twofold. First, a caching node can answer the data requests from other nodes. Second, a caching node stores the data not only on behalf of its own needs, but also based on other nodes' needs. COOP addresses two basic problems for cooperative caching in MANETs. First one is Cache Resolution which addresses how to restore a data request with minimal cost of time, energy and bandwidth. Second is Cache management we have to decide whether a mobile node decide which data item to place in its local cache. For cache resolution, COOP tries to discover a data source which induces less communication cost by utilizing historical profiles and forwarding nodes. For cache management, COOP minimizes caching duplications between neighbor nodes and allows cooperative caches to store more distinctive data items to improve the overall performance. The disadvantage of this scheme is that flooding incurs high discovery overhead and it does not consider factors such as size and consistency during replacement.

N. Chand et. al. proposes Zone Cooperative (ZC) caching scheme[2] for MANETs which is efficient . In Zone Cooperative caching scheme, one hop neighbor of a mobile client form a cooperative cache zone since the cost for communication with them is low both in terms of energy consumption and message exchange. Zone cooperative caching scheme is for data retrieval in mobile Ad hoc networks. The design rationale of ZC caching is that it is considered advantageous for a client to share cache with its neighbors lying in the zone (i.e., mobile hosts that are accessible in one hop) Mobile hosts belonging to the zone of a given host then form a cooperative cache system for this host .  
Discovery Process When a data request is initiated in a mobile node, it first looks for the data item in its own cache. If there is a local cache miss, the mobile node checks if the data item is cached in other mobile nodes within its home zone. When a mobile node receives the request and has the data item in its local cache (i.e., a zone cache hit), it will send a reply to the requester to acknowledge that it has the data item. In case of a zone cache miss, the request is forwarded to the neighbor along the routing path. If the data item is not found on the zones along the routing path (i.e., a remote cache miss), the request finally reaches the data source and the data source sends back the requested data.

Chi-Yin Chow et al. proposed a novel scheme known as GroCoCa (Group-based Cooperative caching scheme) [7].It

extended from COCA. Client mobility is a key factor to the system performance of cache management in mobile environments. In a cooperative cache environment, if an MH arbitrarily forwards its cached data items to its neighboring peers for cooperative cache replacement, there may not be much benefit to the receiving MH, since the data items may not be useful to the latter. Furthermore, it does not make good use of the cache space of the receiving MH, since it may well have moved far away from the original MH after a while. Likewise, if an MH does not cache the data item returned by neighboring peers in order to conserve its cache, thinking that the peer would still be accessible in the future, it may have regretted if the peer moves far away. In addition, when an MH forwards a data item to another peer and they possess different data access patterns, the action will reduce the peer's local cache hit ratio, and the "alien" data item will be removed very soon, as the peer is not interested in that data item. The decision of whether a data item should be cached thus depends on both factors of the access affinity on the data items and the mobility for each MH. To develop an effective cooperative caching scheme, GroCoCa is proposed which defines and makes use of TCG which is defined as a group of MHs that are geographically and operationally close, i.e., sharing common mobility and data access patterns. In GroCoCa, the common mobility pattern is discovered with an incremental clustering algorithm and the similarity of access patterns is captured by frequency-based similarity measurement.

### **3. PROPOSED SYSTEM MODEL**

#### **3.1 Motivation**

In mobile computing environment where infrastructure is not possible we come across the MANETs, potentially useful to communicate with one another for data sharing. Due to mobile Ad hoc nature there are problems like accessibility of data items, more latency, less utilization of cache space. Caching techniques provides the solutions for the above problems. Other problems that affect the performance are like the caching mobile node should be sleep mode when not in use so, battery power can be saved. In peer to peer caching cache space is not used efficiently. In MANETs on-demand routing protocol is preferred for saving the energy and bandwidth. If there is multiple paths from the same mobile node then the cache space may become full and thus degrades the performance. With the objective of considering these points we aim to design Group caching technique which will focus on Grouping of Mobile nodes and efficient caching.

#### **3.2 Network Model**

We consider the mobile Ad hoc Network as a graph  $G=(V, E)$  in which  $V$  represents the set of mobile node and  $E$  represents the set of communication links. An edge  $e = (u, v) \in E$  where  $(u, v) \in V$  exists if and only if  $u$  is in the transmission range of  $v$  and  $v$  in the transmission range of  $u$ . All the links in the  $G$  are bidirectional i.e. if  $u$  is in the transmission range of  $v$ ;  $v$  is also in the transmission range of  $u$ . we will assume the network initially not in the connected state .If network is divided; each sub network is treated as an independent network.

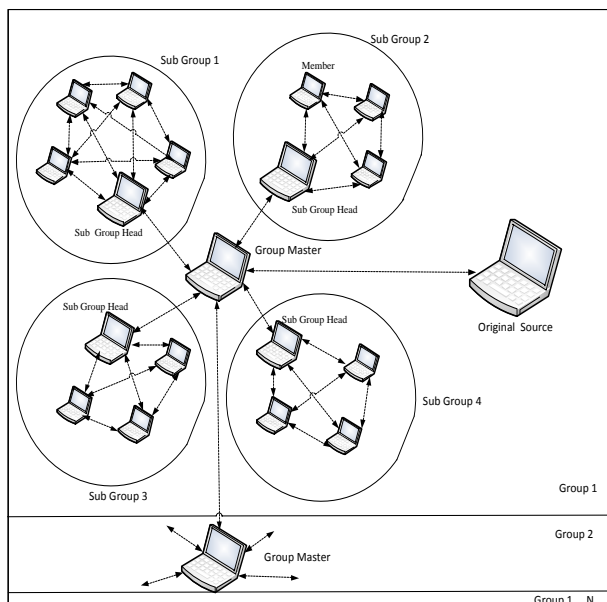
#### **3.3 Group Definition**

All the mobile nodes in the transmission range of each other are able to form a group but the condition is that they must be at one hop distance from each other .The rules for grouping of node are reported below:

1. Each Mobile Node has a unique ID which can be recognized as Node ID.
2. Each Sub Group has one Sub Group Head and each Group has one Group Master.
3. Each Sub Group contains many members' i.e. mobile nodes and the distance between each member node and the head is one hop.
4. Each Mobile Node belongs to only one Sub Group.

### 3.4 Model Description

In this we describe a generic system model of MANET. We assume that this is a heterogeneous mobile Ad hoc network where all nodes are of different capability and capacity. All nodes will communicate with each other via a wireless LAN (e.g. IEEE 802.11) as illustrated in Figure 2. MANETs consist of a set of mobile nodes that communicate with each other using the Ad hoc communication protocols shown by dashed line. In this proposed architecture, nodes are aggregated into sub groups (Sub Group 1, Sub Group 2.....Sub Group 4) according to their capability and capacity and then grouped into a group (Group1). There will so many groups in the network that may contain several subgroups. The sub group members will communicate via Sub Group head and then to the Group Master. Group Master will maintain the information about all the Sub Group Heads in a table and all the paths between the Sub Groups. Sub Group Head will maintain the information about the member nodes. Group Masters will communicate with the origin data source. It is assumed that all the nodes have cache within it. The problem came in this technique that how groups and subgroups will be created and who will be responsible for group management. There are two issues here first one is sub group creation then group creation and second is Sub Group Head selection then Group Master selection.



**Figure2. System model of MANET**

### 3.1 Group Management

Due to mobility and resource limitations of nodes group management [6, 7, 8] is important aspect for resource saving. For efficient information retrieval, search and sharing of content at various nodes group dynamics are mandatory. We distinguish three main functions in group management: Sub Group Member Discovery, Member Discovery means discovering mobile nodes that are eligible for membership

according to relevant membership attributes. Here we are considering that all nodes that are in a sub group must be one hop distance from each other. Sub Group Head Election, in this head is elected by the nodes of the network according to the capability and capacity of particular node with the help of limited flooding concept. Group Master Election, in this Master is elected by the Sub Group Head's with the help of some parameters associated with the Sub Group Head's. Group Dynamics Management means updating group membership according to the dynamics of the network's topology. Attributes related to the mobile node should be updated in the Group Head. In next section we will see how to elect Group Head and Group Master within mobile nodes according to the capability and capacity of mobile node.

### 3.2 Sub Group Head and Group Master Election

Initially there are several heterogeneous mobile nodes in mobile Ad hoc network. All are connected to one another or may not be in any topology. It is very difficult for a mobile node to get data from the source every time because it will increase the communication cost in terms of latency and resources. In this architecture we are designated a Head that will server all its members. To determine the Sub Group Head four parameters are considered. These parameters are Mobility ( $M$ ), Energy Level (Battery Life) ( $E$ ), Memory Size ( $S$ ) and Processing Power ( $P$ ). Every MN in network has mobility, energy level, memory size and processing power value. We are not bothering about how to calculate mobility, energy level etc. our aim is to consider these parameters and to calculate a value called Stability Factor ( $SF$ ). Nodes flood their capacity to the neighboring nodes at one hop distance. For head election  $SF_1$  can be calculated as in equation 1:

$$SF_1 = \frac{\langle E(x) \times S(x) \times P(x) \rangle}{\langle M(x) \rangle} \quad \text{--- (1)}$$

$SF_1$  is calculated at each node and noted down in a table and then sorted in the decreasing order of Stability Factor. The node on the first position i.e. highest  $SF_1$  will be elected as Sub Group Head as shown in algorithm 3.2.1. All nodes at one hop distance will join to this node and they will not participate in the election if it message from the other node then they deny this message because they already associated with one sub group. There is one assumption that if Sub Group Head leaves then who will take responsibility of Head. Then we re-elect the Head with in the Sub Group members according to their capability and capacity.

#### 3.2.1 Algorithm for Sub Group Head Election

1. flood (node ID,  $M, E, S, P$ )
2. for each node
3. calculate ( $SF_1$ )
4. arrange in the decreasing order of stability factor
5. highest  $SF$  node will be elected Sub Group Head

$SF_2$  Will be calculated for the selection of Group Master. To elect group Master we considered two parameters. These parameters are Bandwidth ( $B$ ) and Range ( $R$ ).  $SF_2$  Calculation is in equation 2.

$$SF_2 = \langle B(x) + R(x) \rangle \quad \text{--- (2)}$$

$SF_2$  is calculated at each Head of sub group. The Head having largest value of  $B$  and  $R$  will have the highest  $SF_2$

value. Then this head will be the group master as discussed in algorithm 3.2.2.

### 3.2.2 Algorithm for Group Master Election

1. flood (Sub Group Head ID, B, R)
2. for each Sub Group Head
3. calculate  $SF_2$
4. arrange in the decreasing order of stability factor
5. highest  $SF$  will be the Group Master

### 3.3 Group Dynamics Management

Group management over MANETs [5, 10] requires managing mobility and induced changes in group membership, in a manner that is transparent to applications. Since a new node is detected by the network layer, an event-based mechanism is installed on the Head to invite the new comer to join the group, on the condition that the group is able to accommodate more members. The battery power of the Group Master and Sub Group Head may decrease continuously but the batter power of the other members may not be. When the battery power of Group Master falls below a threshold then the node is no longer capable to perform as Group Master and similarly in case of Sub Group Head. In this situation new Group Master and Sub Group Head needs to assign for the functioning of the network. So, algorithm for re-election is described below. Re-election mechanisms are shown in the algorithm 3.3.1.

#### 3.3.1 Algorithm for Re-election

1. Check the Energy \_level of the Group Master and Sub Group Head.
2. If  $Energy\_level > Threshold\ Energy$ .
3. Then
4. Re-election of is not needed.
5. Else
6. If  $Energy\_level < threshold$ .
7. Then
8. Broadcast an  $ENERGY\_LOW$  message to all its neighbors.
9. Perform re-election
10. Calculate  $SF_1$  and  $SF_2$  value.
11. Mobile node with Highest Stability Factors value is selected as new Group Master and Subgroup Head respectively.

## 4. CACHE MANAGEMENT

The cache is focus on data caching in the system. The limited bandwidth and other problems in MANETs make the communication cost (Query Processing). Node mobility makes the network topology change frequently, meanwhile disconnection of nodes often occur. This delays the data transmission in the network. The cache at the node stores data accessed frequently by the query node or their neighbors. This decrease the query response time and improves the data accessibility when node disconnection occurs. With the changing of network topology and the updating of the data at each node, the cached data should be updated dynamically to ensure the consistency between the cached data and the original data in the database. Cache management [2] has four basic functions i.e. Cache Admission Control, Cache Resolution, Cache Consistency, Cache Replacement. All are discussed one by one below.

### 4.1 Cache Admission Control

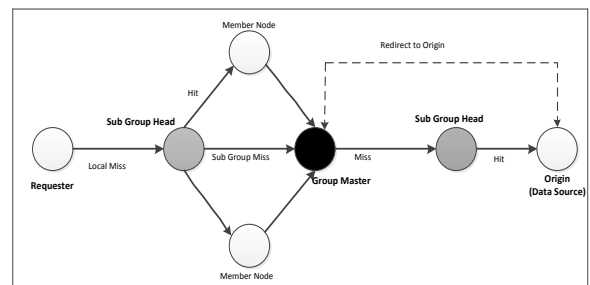
Cache Admission Control [2, 4] decides that what data items can be cached at the node so that performance of caching system can be improved. When a node received the requested data, a cache admission control is triggered to decide whether it be cache this data. In this paper according to our proposal we assume that all the nodes communicate with the Head first because Head maintains the information of all its sub group members. Head is the node that has more capability and capacity than other nodes. Head will maintain a cache Status table where all the information related to caching of data item will be available. If there is a data entry for data item  $d_i$  in the cache status table then there is no need to cache that data item again and again. Caching data at every node will takes a lot of energy, wastage of cache space and it will create unnecessary overhead. The primary idea of cache admission is to increase accessibility. If data is available at nearby terminal, why go for the access point of origin. Cache status table has following entries shown in Table 1.

**Table 1. Cache Status Table**

S.No.	Data Item	Cache Node(Node ID)	TTL Value	Access Count
1 2. 3	$d_i$	A, B, C....	3000 sec.	3

### 4.2 Cache Resolution Mechanism

Cache resolution addresses [5, 7] how to resolve a data request with minimal cost of time, energy and bandwidth. To address the problem of cache resolution we have designed the mobile adhoc network in such a way that every request is optimally resolved. Cache resolution Mechanisms is shown in terms of flow graph in Figure 3.



**Figure3. Cache Resolution Mechanism**

When there is a data request for data item  $d_i$  by a node then algorithm 4.2.1 will be followed.

#### 4.2.1 Algorithm for Cache Resolution

1. request made by a node for data item  $d_i$
2. data item  $d_i$  is checked at Sub Group Head cache
3. if a valid copy is available in the cache
4. then
5. get data item  $d_i$
6. else
7. look into its the vicinity
8. else
9. look into the Master cache
10. if a valid copy is available then
11. get data item  $d_i$

12. else
13. forward the requester to the source or to the other Sub Group with the shortest route.

### 4.3 Cache Replacement

Cache replacement is one of the most important issue in the caching of data in mobile ad hoc networks. If the cache is full then cache replacement policy [3, 4] is required. We have to decide which data should be removed from the cache. Aggarwal et al. [10, 13] classifies the existing cache replacement policies into three categories: direct-extension, key-based, and function-based [14]. In the direct-extension category, traditional policies such as LRU or FIFO are extended to handle data items of non homogeneous size. The difficulty with such policies in general is that they fail to pay sufficient attention to the data size. In the key-based policies, keys are used to prioritize some replacement factors over others; however, such prioritization may not always be ideal. Recently, function-based replacement policy has received considerable attention. The idea in function based replacement policies is to employ a function of different factors such as time since last access, entry time of the data item in the cache, transfer time cost, data item expiration time, and so on. For example, the LNC-R-W3-U algorithm, proposed by Shim et. al. [15], aims to minimize the response time in Web caching. Their cost function incorporates many system parameters such as the transfer time, the document size, and the invalidation rate. Today, function-based replacement policy has obtained priority over the traditional and key based cache replacement policies. The thought in function based replacement policies is to apply a function of different factors such as time since last access, entry time of the data item in the cache, transfer time cost, data item expiration time, and etc. For example, the VALUE based cache replacement algorithm, proposed by N.Chand et. al. [2] uses the four factors such as distance, coherency, size, TTL value. The data item with the lowest value of the VALUE function is put across for replacement. Here in this model we will use the function based cache replacement.

### 4.4 Cache Consistency

There are two schemes [3, 12] that can deal with the cache consistency problem: weak consistency and strong consistency. Under the weak consistency, a cached data object is associated with an attribute, TimeToLive (TTL). If the TTL time expires, the cached data object is removed. Under the strong consistency, if a cached data object is requested, the caching node first asks the data source to see if the cached data object is valid or not. Because of the energy concern and the concern of wireless bandwidth, we prefer using the weak consistency in mobile Ad hoc networks. We continuously checking the consistency if the cache at the Head and at the Master so that cache data will not be invalidated. This will lead to the enhancement in the performance of mobile ad hoc networks.

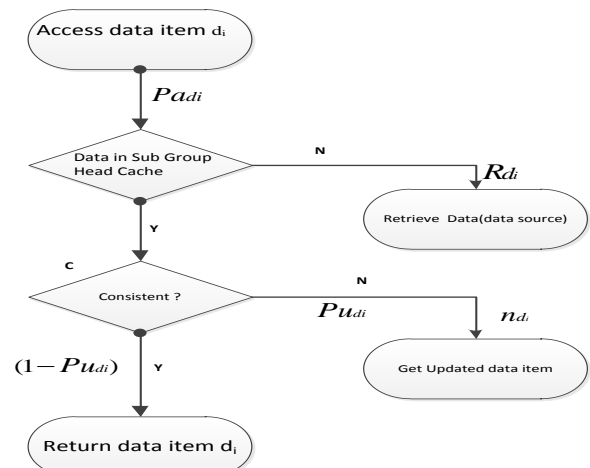
## 5. CACHE ACCESS COST MODEL

Cache Access Cost Model basically illustrated to demonstrate the maximum cost which is required to access the data item according to this proposed Group Caching Scheme. In this model the following notations are used to find the cost of accessing data item  $d_i$  [15] as discussed in Table 2.

**Table 2. Notations Used**

Notation	Description
$n$	The number of data items in the database.
$R_{d_i}$	The cost of retrieving data item $d_i$ to the cache.
$C$	The mean cost of validating the consistency of data item.
$n_{d_i}$	The cost of getting new updated copy of data item $d_i$ from the data source (origin).
$a_{d_i}$	The mean access rate to data item $d_i$ .
$u_{d_i}$	The mean update rate of data item $d_i$
$S_{d_i}$	Size of data item $d_i$ .
$Pa_{d_i}$	The probability of referencing data item $d_i$ .
$Pu_{d_i}$	The probability of invalidating cached data item $d_i$ .
$S$	The set of all the cached data items.

Basic flow diagram of cache access cost model is shown in Figure4.



**Figure4. Cache access cost model in Dynamic Group Caching**

Expression for cache access cost in Dynamic Group Caching is defined as:

$$\max_{d_i \in S} \text{cost}(d_i)$$

Where,  $\text{cost}(d_i) = Pa_{d_i}(R_{d_i} - C - Pu_{d_i}n_{d_i})$

This cost function can be explained by the cache access cost model shown in Figure3. If data item  $d_i$  is not in the cache, it will take  $R_{d_i}$  to retrieve data item in to the cache. We can say that if data item  $d_i$  is in the cache we can save the access cost by  $R_{d_i}$ . It also takes  $(C + Pu_{d_i}n_{d_i})$  to validate it and get the updated data if important. We can observe that caching the data can save the cost by  $Pa_{d_i}(R_{d_i} - C - Pu_{d_i}n_{d_i})$  per data access.

## 6. CONCLUSION

In this paper a novel caching scheme EDGC is presented which provides a solution to the caching problems so that performance of the MANETs can be increased. Communication is faster in EDGC due to the grouping of mobile nodes. Sub Group Head and Group Master is election by election mechanisms. It provides a solution in such a way that request from a node is resolved in minimum number of steps with reduced Latencies and better query processing. Cache access cost Model describes the cost of accessing data items with respect to the group caching technique which is efficient than simple caching. Future Work will introduce cache consistency and good cache replacement mechanisms in Group Caching.

## REFERENCES

- [1] Yu Du and S. Gupta, "Improving on-demand data access efficiency in MANETs with cooperative caching", Ad hoc Networks, Elsevier, pp.579-598, 2009.
- [2] N.Chand, R.C.Joshi and Manoj Misra, "Cooperative caching in mobile ad hoc networks based on data utility", Mobile Information Systems pp. 19–37,2007.
- [3] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89,2006.
- [4] MANET: Mobile Ad hoc Networks. <http://www.ietf.org/html>.
- [5] Ying-Hong Wang and Jenhui Chen, "A dynamic Caching mechanism for Mobile Ad Hoc Networks" 11<sup>th</sup> International Conference on Parallel and Distributed Computing (ICPADS '05), 2005.
- [6] Yi-Wei Ting and Yeim-Kuan Chang, "A Novel Cooperative Caching Scheme for Wireless Ad Hoc Networks: Group Caching" International Conference on Networking, Architecture, and Storage (NAS), 2007.
- [7] C.Y. Chow, H.V.Leong and A.T.S.Chan, "Group-based Cooperative Cache Management for Mobile Clients in a Mobile environment" International Conference of Parallel Processing (ICPP'04), 2004.
- [8] Matthias R. Brust "Topology-based clustered candidate selection in wireless hoc and Sensor Network." 2nd International Conference on Communication System Software and Middleware and Workshops, 2007.
- [9] Hao Yu, Patrick Martin and Hossam Hassanein "Cluster-based replication for Large Scale Mobile Ad Hoc Network" International Conference on Wireless Networks, Communications and Mobile Computing, 2005
- [10] Han Ke "Cooperative Caching Algorithm based on Grouping Nodes in Mobile Ad hoc Network" International Conference on Information and Automation, 2010.
- [11] Malika Boulkenafed "A Middleware Service for Mobile Ad hoc Data Sharing, Enhancing Data Availability" ACM International Middleware Conference, 2003.
- [12] S. Lim, W. Lee, G. Cao, and C. R. Das. "A Novel Caching Scheme for Internet based Mobile Ad Hoc Networks". In Proc. 12th International Conference on Computer Communications and Networks (ICCCN), pp 38–43, 2003.
- [13] C. Aggarwal, J. Wolf, and P. Yu, "Caching on the World Wide Web," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, Jan. / Feb. 1999.
- [14] J. Shim, P. Scheuermann, and R. Vingralek, "Proxy Cache Algorithms: Design, Implementation, and Performance," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 4, July/Aug. 1999.
- [15] L. Yin, G. Cao, and Y. Cai, "A Generalized Target-Driven Cache Replacement Policy for Mobile Environments," Proc. 2003 Int'l Symp. Applications and the Internet (SAINT), Jan. 2003.
- [16] T.Hara "Data Replication for Improving Data Accessibility in Ad Hoc Networks," IEEE Transaction on Mobile Computing, vol 5 no11, pp 1515-1532, Nov 2006.
- [17] D. Barbara and T. Imielinski "Sleepers and Workaholics: Caching Strategies for Mobile Environments. In Proc. ACM SIGMOD, pps 1–12, 1994.
- [18] H.Artaïl, H.Safa, "COACS: A Cooperative and Adaptive Caching System for MANETs," IEEE Transactions On Mobile Computing, Vol. 7, no. 8, pp 961-977 Aug 2008.
- [19] Ge-Ming Chiu, Cheng-Ru Yong "Exploiting In-Zone Broadcasts for Cache Sharing in Mobile Ad Hoc Networks" IEEE Transaction on Mobile Computing, Vol 8, No3, pp 384-397, March 2009.