International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 9, Issue 10 (January 2014), PP. 55-62

Scalable and Efficient Strategies for Grid based Location Services for Bandwidth Efficient and Energy Conserving Routing in Mobile Ad hoc Network

Saba Khalid[#], Shish Ahmad [*], Dr. Mohd. Rizwan Beg
[#]*Assistant Professor, Integral University, Lucknow, India*
[*]*Associate Professor, CSE Department, Integral University, Lucknow, India*
Professor, HOD (CSE Deptt.), Integral University, Lucknow, India

Abstract:- A Mobile Ad-hoc Network (MANET) is a network topology of choice where autonomous nodes form a collaborative community. Position based routing protocols have been studied extensively over the past years, But how to efficiently provide the location information for nodes and how to control location update frequency is still a challenge. In this paper, we address the problem of maintaining location service and propose an energy aware grid location service which efficiently provide location information and conserve energy extending the lifetime of a MANET. The protocol uses hierarchical grid location service to minimize inevitable superfluous flooding by every node, and prevents location updates and queries from traversing the entire network unnecessarily, hence conserving bandwidth and transmission power.

Keywords:- Location Server(LS), Location management, Grid, Associativity, Query Success Rate(QSR)

I. INTRODUCTION

Mobile devices equipped with short range wireless communication interfaces (e.g., Wi-Fi, Bluetooth), and sometimes with a GPS receiver, are nowadays widespread and used daily by an increasing number of people. Netbooks, mobile internet devices or smartphones are some examples of such devices. Thanks to their short range wireless communication interfaces, these devices can spontaneously form a multi-hop disconnected mobile ad hoc network (DMANET). Designing a routing protocol supporting both service discovery and delivery in such kinds of networks is radically different than devising one for traditional infrastructure-based networks. Indeed due to the mobility of nodes and to the short communication range of wireless interfaces, the topology of DMANETs suffers from frequent and unpredictable changes, entailing an intermittent connectivity between nodes, therefore routing in MANET is a challenging task. Studies [1] have shown that position-based routing is a well-suited solution for the challenging task of routing in highly dynamic MANETs. As a prerequisite of position-based routing algorithms, each node in the network must be able to determine the position of the target node it wants to communicate with their own position, e.g. by means of a positioning service such as GPS. The task of locating the destination is then accomplished by a location service, a distributed service maintained directly by the participating nodes. In this paper, we present the Hierarchical Grid Location Service (HGLS). The basic operation of HLS is as follows: the area occupied by the network is divided into a hierarchy of regions. The lowest level regions are called grid or cells. Regions of one level are aggregated to form a region on the next higher level of the hierarchy. Regions on the same level of the hierarchy do not overlap. For any given node A one node in each level of the hierarchy is selected as location server for node A by means of a location server election algorithm. This location server node is called the responsible node for node A. As a node moves through the area covered by the network, it updates its responsible nodes with information about its current position. When another node B needs to determine the position of node A it broadcast the message to determine those nodes that may potentially be responsible for A. It then queries those nodes in the order of the hierarchy, starting with the lowest level region. On the first level that contains both nodes A and B the query will arrive at a responsible node of A where it can be answered. Due to its hierarchical approach and because of using the concept of responsible nodes HLS has some very desirable properties:

- it does not use flooding the number of location servers scales logarithmically with the size of the network
- it is well-suited for high node mobility
- it specifically supports non-uniform communication patterns.

This information can be used by an upper level application to provide various location dependent services. To preserve scalability, the location service must allow queries and updates to be performed using only a handful of messages. Of course, the location service itself must operate using only geographic forwarding. It should also be scalable in the following senses:

1. No node should be a bottleneck—the work of maintaining the location service should be spread evenly over the nodes.
2. The failure of a node should not affect the reachability of many other nodes.
3. Queries for the locations of nearby hosts should be satisfied with correspondingly local communication. This would also allow operation in the face of network partitions.
4. The per-node storage and communication cost of the location service should grow as a small function of the total number of nodes.

The Hierarchical Grid Location Service presented in this paper satisfies all these requirements. The remainder of this paper is structured as follows: In Section II we give an overview of related work. The HGLS algorithm is described in detail in Section III. Section IV contains the results of the simulation of HGLS with Matlab. The paper is concluded with a summary and an outlook to future work in Section V.

II. RELATED WORK

A location service consists of two algorithmic components, the location update and the location request component. The location update is responsible for distributing information about the current location of a target node T to a set of nodes called the location servers of T. If a source S wants to discover the location of T, it launches a location request which is routed through the network to one of the location servers of T. The location server can either answer the request itself or forward it to T for answering. The range of possible designs of the update and request component are limited by two extremes: flood position updates or do not send any updates at all. If updates are flooded, each node becomes location server for each other node and no requests are necessary. If no updates are sent, each node is its own and only location server, therefore requests need to be flooded in the network. A location service which uses flooding to spread position information is DREAM [2], the Distance Routing Effect Algorithm for Mobility. With DREAM, each node floods its position information in the network with varying flooding range and frequency. A location service that does not require flooding is Homezone [3]. In the Homezone location service, each node is assigned an area (the so called Homezone) in the ad-hoc network via a hash function.

Camp, Boleng and Wilcox [4] have developed and evaluated the performance of three location services: the Simple Location Service (SLS), the DREAM Location Service (DLS) and the Reactive Location Service (RLS). In all three services, the nodes maintain a table containing the location information of all the nodes in the network and update the location information in a promiscuous manner (that is, a node updates its location table, even when it overhears a reply to a location request). A node responds to a location request with a reply containing the corresponding data in its table entry. If however, the location information is not found in the local table, the location request packet is flooded in the network. On receiving a location information packet, the nodes update their tables. SLS and DLS are proactive (nodes exchange location information periodically) while RLS is reactive (location information is queried when needed). In SLS, nodes periodically transmit tables containing the location information of a few nodes in the system to their neighbours, while in DLS a node transmits its own location information to nearby nodes at a particular rate and to faraway nodes at another lower rate. In RLS, if a node does not have the location information of a required node, it first asks its neighbours and on not hearing back from the neighbours within a timeout period, floods the network with its request. Nodes that receive a location request packet and do not have the required data propagate the request. However, if the required information is available with a node, it sends a reply to the source with the required data via the reverse source route (request packet contains full route). A node updates its location table if it either receives a location information packet or if it overhears it.

Quorum based location services have been presented in[5] and [6]. In [5], uniform quorum systems are used to provide a distributed location management scheme. Node location information is maintained in location databases that form a virtual backbone. Initially, flooding is used to form the virtual backbone. The algorithm for the initiation and management of the virtual backbone is however not mentioned in [5]. The uniform quorum system is comprised of the nodes in the backbone. In [6], three different strategies are presented for selecting quorums for queries and updates, based on a node-unreachability list maintained by all nodes in the system.

III. HIERARCHICAL GRID LOCATION SERVICE

Combining geographic forwarding with a mechanism for determining the location of a node implements the traditional network layer in which any node can send packets to any other node. A trivial location service might consist of a statically positioned location server. Nodes would periodically update this server (using geographic forwarding to the server's well-known coordinates) with their current location. For a node A to contact node B, A queries the location server for B's current location before using geographic forwarding to contact B.

Using a single location server has a number of problems. The centralized server is a single point of failure; it is unlikely to scale to a large number of mobile nodes; it cannot allow multiple network partitions to each function normally in their own partition; and nodes near to each other gain no advantages—they must contact a potentially distant location server in order to communicate locally. We introduce a distributed hierarchical grid location service (HGLS) that is designed to address these problems. GLS is fault-tolerant; there is no dependence on specially designated nodes. GLS scales to large numbers of nodes; our goal is to provide a service that scales to at least the size of a large metropolitan area. Finally, GLS operates effectively even for isolated pockets of nodes. A node should be able to determine the location of any node that it can reach with geographic forwarding. That is, a location lookup should not involve nodes that are too far "out of the way" of a straight line trip from the node performing the lookup to the node being looked up.

A. System Environment

We model a mobile ad hoc network as a set of n mobile nodes that move around in a predetermined two dimensional logical grids as shown in fig.1. This is exactly the same partition method as described in [7].Each grid is a square area of size s x s. Grids are numbered (x, y) .Each host has a unique ID ranging from 0 to n-1. Each node is aware of its own location through the support of a service like GPS. Then we can easily map the location information into its grid coordinates. Each node also maintains location information of all other nodes in the system. The mobile nodes include both servers and clients; however no distinction is made between server nodes and client nodes. Thus each node acts as both a client and a server.

B. HGLS Network Partition

HGLS partitions the area containing the ad hoc network into four squares which are separated into four smaller squares and so on. The four smallest squares are called the order-1 level in the hierarchy. The only prerequisite is that a node in a given lowest-level grid must be able to send packets to all other nodes in the same grid. At each square in any level, a node selects a location server according to the proposed location server election algorithm. At startup, all nodes know the same global partitioning of the world into a hierarchy of grids with squares of increasing size, as shown in Figure 1. The smallest square is referred to as an order-1 square. Four order-1 squares make up an order-2 square, and so on. It is important that not every square made up of four order-squares is also an order square. Rather, to avoid overlap, a particular order-square is part of only one order- square, not four.This maintains an important invariant: a node is located in exactly one square of each size. This system of increasing square sizes provides a context in which a node selects fewer and fewer location servers at greater distances. In fig.2 hierarchical view of grid location service of fig.1 is present where different order levels from a hierarchy.

C. Selecting and Querying Location Servers

GLS provides for distributed location lookups by replicating the knowledge of a node's current location at a small subset of the network's nodes. This set of nodes is referred to as the node's location servers. A node A hoping to contact node B can query one of a number of other nodes that know B's location. Of course, A must be able to contact the nodes that know B's location.

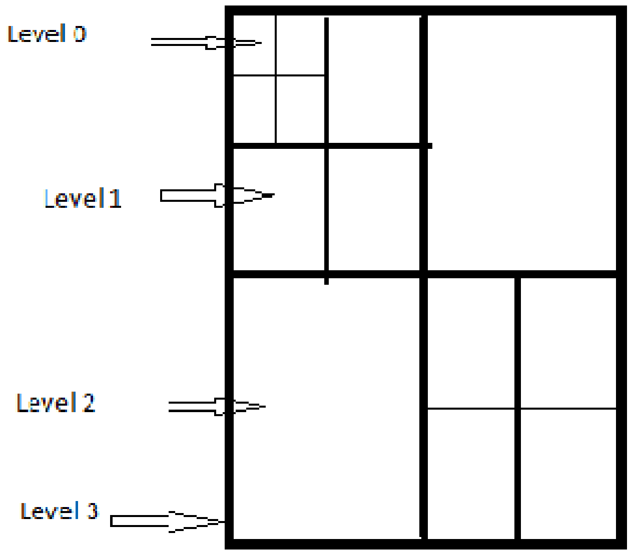

Fig. 1 Grid Location Service

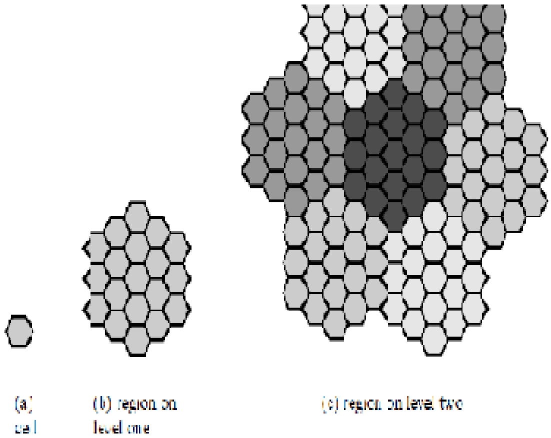

Fig. 2 Hierarchical view of Grid Location Service

This means that A's search for B's location servers and B's original recruitment of location servers ought to lead to the same servers. When B recruits location servers it uses the same information that A will have when searching for B's location servers: B's name and certain information that all nodes have at start up. The rules for selecting a location server for a node in a particular grid level are as follows:-

***Location Server Election Rules*:**

1. A node with higher level of remaining battery capacity has higher priority
2. Given several hosts with the most energy level, the one that is closest to the centre of the grid will be selected as the location server. This rule allows a host that will stay in the grid for longer be the location server. So that the update frequency of location server is reduced.
3. The host which has higher associativity of nodes connected to it will be elected as location server.

D. Location Server Election Algorithm

The location server election algorithm is executed distributedly in each level of the grid so that each node has a location server. The algorithm also triggers when the location server runs out of energy, moves out of the grid level, or turns down because of an accident.

Algorithm :

1 .Each node will periodically broadcast its HELLO message. The HELLO message contains the following five fields:-

(a) id: node ID
(b) grid: grid coordinate
(c) lsflag: location server flag(set to 1 when the node is the location server)
(d) level: battery remaining capacity level(numeric value assigned to each random node)
(e) distance: distance of node from centre of the grid.

2. After a **HELLO** period, which is predefined as the period for nodes to exchange their HELLO messages, all nodes are supposed to receive the HELLO messages from neighbouring hosts in the same grid. Then, each host will apply the location server election rules to decide which node will act as location server.

3.The node will declare itself as the location server by sending a HELLO message with the ls flag set. This node is now responsible for maintaining the host table, which is constructed from the id field of the HELLO messages.

4. All other host nodes, receiving the HELLO message from location server, will move to sleep mode if they have no packets to transmit.

As in any location management approach, whenever a node needs location information of another node,the former has to first find the location server of the latter and initiate the location discovery process. There are two mobility situations to be addressed.

1.Node moves into a new grid :Nodes will broadcast a HELLO message when they move into a new grid. The location server node in each grid will also broadcast its HELLO message when it hears the HELLO message.After the location server 's HELLO message is received, the new incoming host will decide if it should replace the existing location server.In this situation ,only the node with highest battery level, at the center position and highest associativity state than that of the original location server can replace the original gateway.This rule prevents frequent replacements of location servers.If the LS must be replaced ,then the new location server will declare itself by sending a HELLO message with the ls flag set. The original LS receiving this HELLO message, will transmit the routing and node tables to the new LS .If no LS is replaced, then the incoming node will enter sleep mode to conserve its energy. If a new node does not receive any HELLO message during periodic broadcast of HELLO message, then the new node is in an empty grid and will declare itself as the gateway.

2.Node moves out of a grid: Nodes on the edge of the grid are easier to move out of the grid than the nodes near to the center.The update frequency will increase if nodes continuously cross the edges of the grid. Consequently, we enlarge the area of grid. The area is enlarged by GR whose radius is R.The monitor area of clusterhead is also increased, and it only updates nodes that move into or out of (CR).When LS leaves the grid it must transfer its routing table to a new LS before it leaves a grid. The LS thus first sends a broadcast sequence to wake up all the hosts in the same grid. After waiting for time T,the LS will declare its departure a DEPART(grid,rt) message, where grid represents coordinate of the gateway and rt represents routing table. After receiving this message , all nodes will store the routing table,and apply the LS election algorithm to elect new LS.

IV. PERFORMANCE ANALYSIS

In this section we analyse the loction update frequency to explain the low control overhead of our scheme. The proposed Location service is evaluated using MATLAB. The specific parameters are shown in **Table 1:**

Table 1 Simulations Parameters

Parameters	Value
Number of Nodes	100
Simulation Area	100x100
Transmission Range(m)	71
Simulation Time(seconds)	300
Mobility Model	Random Waypoint Model
Energy(J)	1 Joule

A typical routing protocol GPSR is chosen to perform the simulation

A. Analysis Model

For analyzing the results we have assumed that nodes are uniformly randomly distributed in a network grid. At each simulation, several random location servers nodes are selected which fulfil the LS algorithm. We have compared our protocol with the grid location service(GLS) from the following five metrics :-

A. Location Server Stability

The location server stability is measured by determining the number of times each MN either attempts to become a LS or gives up its role as LS.

In the first figure, the scalability of the scheme is measured in terms of increasing node-count. The lower the frequency of LS changes, the more stable the grid is. As can be seen from fig.3, our scheme leads to more stable LS formation.

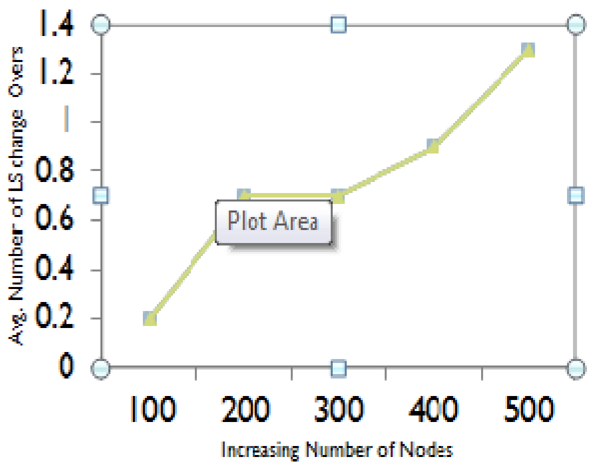

Fig. 3 Location server stability as a function of number of increasing Node Count

In figure4, we compare our scheme in term of the control cost incurred per node in kilobytes, when the number of node increases. The control cost incurred per node in GLS tends to increase with increasing number of nodes. But in our scheme this increase is very low. Messages are relayed by intermediate nodes only if they originate from the same grid(i.e., relaying is spatially limited). Furthermore, unlike in other schemes, there is a control over the number of location servers elected in our scheme, and it is proportional to the number of nodes that we have in a given area. All these desirable features prevent arbitrary improper nodes to become location servers in our scheme, and thus help to improve cluster stability. Fig. 5 shows the clustering cost incurred by a node when the node-density increases. As can be seen, when the network is sparsely-connected, the average control cost incurred by a node tends to be high in all the three schemes. However it tends to decrease as the network becomes denser. While this control cost continues to decrease in our scheme, the same is not expected in the other two schemes. In all the three schemes, the LS stability is very low when there is lower node-density. This can be due to the reason that when the node-degree is low, improper nodes are elected as LS and as a result moving nodes may create transient instability. However, when the node- degree is moderate, all the protocols try to converge quickly with the selection of proper nodes. On the other hand, when the network becomes denser, the LS election algorithms take longer time to converge due to increased control traffic, and hence this affects the LS stability. However, in our strategy, since the LS election heuristic takes associativity,energy and thus mobility, into consideration, only stable nodes are elected as LSs, and hence results in improved LS stability.

Query Success Rate (QSR)

This simulation parameter represents the ratio of the queries answered from all those sent with valid location information. As depicted in fig. 7 HGLS performs more than four times better than GLS.GLS success rate is partly effected by aggressive caching. In fig.9 we compared the query success rate with increasing node density, where our scheme outperforms GLS.

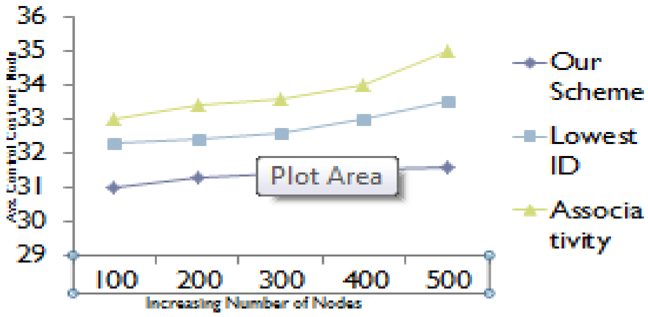

Fig. 4 Average Control Cost incurred per node as a function of increasing number of Node-Count

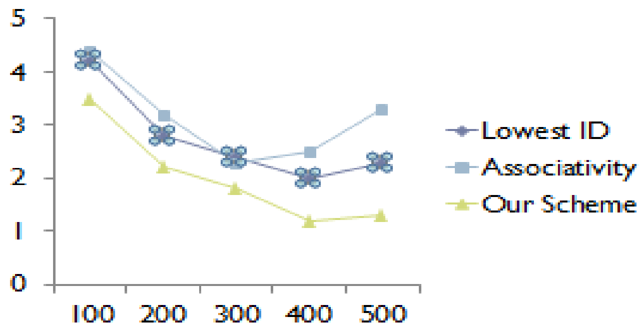

Fig. 5 Location Server stability as a function of increasing Node Density

Another parameter is Mean energy consumption by the nodes. We have calculated the energy consumed by the node in transmitting location request and in electing nodes as LS. We have calculated the remaining battery capacity(rbc) of each node by

Rrbc = Remaining Battery Capacity

Full Battery Capacity

Fig. 6 depicts the mean energy consumption by nodes.

Next parameter is Request Travel Time (RTT) it is the effective response time indicator. The location requests must be quickly sent to the targeted node using any forwarding strategies, here GPSR routing protocol is used. From fig. 8 we see that our scheme serves more request than GLS during the same interval. The QSR decreases as the number of nodes increases and requests increases. As the requests increases network overload increases and therefore drops the query success rate.

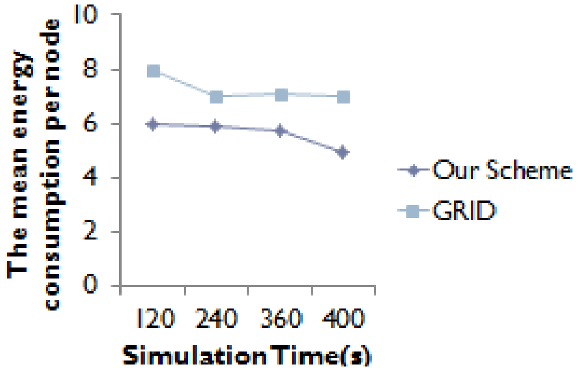

Fig. 6The mean energy consumption per node vs. the simulation time

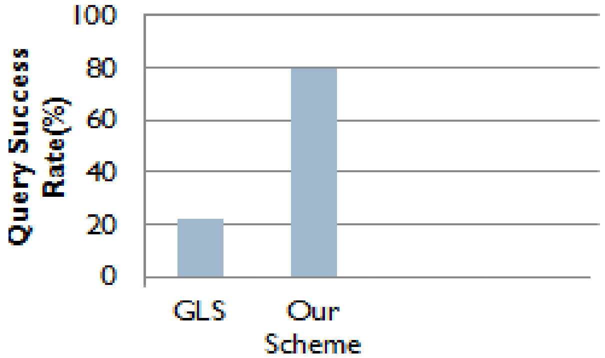

Fig. 7 Query Success Rate

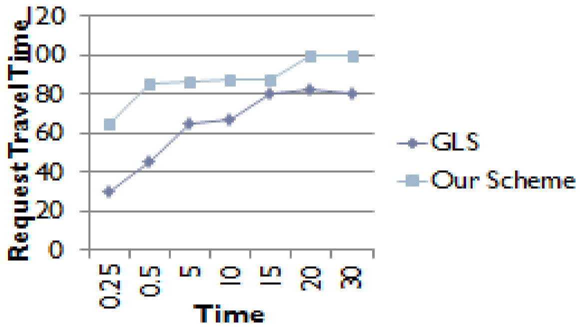

Fig. 8 Request Travel Time

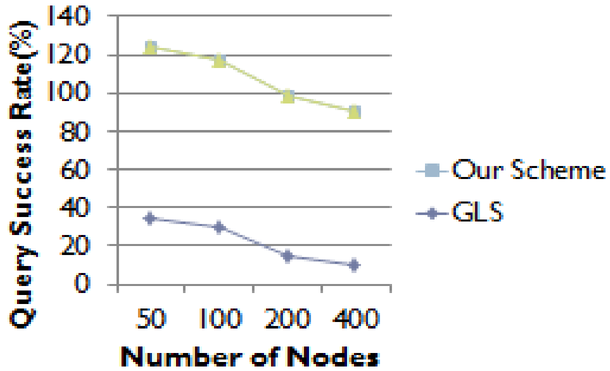

Fig. 9 Query Success Rate GLS vs. Our Scheme

V. CONCLUSION

In this paper we have proposed an efficient location service based on energy and associativity of a node to form location server. By employing the dominating nodes in a grid to perform periodic location update on behalf of other nodes, we have demonstrated that our scheme leads to less control overhead as compared to GLS. In addition our location management strategy conserves scarce resources like battery energy and wireless bandwidth by preventing the queries and responses from traversing the unnecessary parts of the d hoc network. Simulation results demonstrate that our scheme prolong the network lifetime in proportion to the host density in the whole network. We plan to further optimize the proposed scheme by comparing it with more tools and technologies in order to maintain better results.

REFERENCES

[1]. B. N. Karp and H. T. Kung, " GPSR: Greedy Perimeter Stateless Routing for Wireless Networks". In Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00), pages 243–254, Boston, Massachusetts, August 2000.

[2]. S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A.Woodward. "A Distance Routing Effect Algorithm for Mobility (DREAM)". In Proceedings of the fourth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '98), pages 76–84, Dallas, Texas, October 1998.

[3]. S. Giordano and M. Hamdi. "Mobility Management: The Virtual Home Region". Technical Report SSC/1999/037, EPFL-ICA, October 1999.

[4]. T. Camp, J. Boleng, and L. "Wilcox.Location information services in mobile ad hoc networks". In IEEE Int. Conf. Communications, pages 3318-3324,2002.

[5]. Z. Haas and B. Liang. Ad hoc mobility management with uniform quorum systems. IEEE/ACM on Networking, 7(2):228{240, April 1999.

[6]. G. Karumanchi, S. Muralidharan, and R. Prakash. "Information dissemination in partitionable mobile adhoc networks". In Proc. IEEE Symp. Reliable Distributed Systems, pages 4{13, Lausanne, Switzerland, October 1999.

[7]. J. Li, J. Jannotti, D. S. J. DeCouto, D. R. Karger, and R. Morris. "A Scalable Location Service for Geographic Ad Hoc Routing". In Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00), pages 120–130, Boston, Massachusetts, August 2000.

[8]. Chih-Min Chao,Jang-Ping Sheu and Cheng-Ta Hu, "Energy Conserving Grid Routing Protocol in Mobile Ad Hoc Network", Proceedings of ICPP , 2003.

[9]. Nicolas Le Sommer and Salma Ben Sassi, ―Location-based Service Discovery and Delivery in Opportunistic Networks‖IEEE 2011, International Conference on Networks.

[10]. Martin Mauve and Jorg Widmer, A survey on Position based routing in Mobile Adhoc Networks‖, IEEE Network,2001.

[11]. Marwane Ayaida, Hacene Fouchal, Lissan Afilal and Yacine Ghamri Doudane, A Comparison of Reactive, Grid and Hierarchical Location based Services for VANET, IEEE Network,2012.

[12]. Yongming Xie, Guojun Wang and Jie Wu, Fuzzy Location Service for Mobile Ad Hoc Networks, IEEE proc. Of CIT 2010.

[13]. Martin Mauve and Jorg Widmer, A survey on Position based routing in Mobile Adhoc Networks‖, IEEE Network,2001.

[14]. Saba Khalid, Shish Ahmad and Mohd. Rizwan Beg, "Position Based Routing Protocols in Mobile Ad hoc Network- A Critical Survey", IJARCET, ISSN-2278-1323.