

DFLS: A New Distributed and Fault Tolerant Location Service Method for Mobile Ad hoc Networks

Nader Ebrahimpour*, Masoud Sabaei **, Manouchehr Zadahmad Jaafarloo***

* *Member of Young Researchers Club, Computer Department of Islamic Azad University Bonab Branch, Bonab Iran*

** *Faculty of Computer, Department of Computer, Industrial Amir Kabir University, Tehran Iran*

*** *Department of Computer Islamic Azad University Ilkhchi Branch, Ilkhchi Iran
naderebrahimpour@gmail.com, sabaei@aut.ac.ir*

Abstract—Location-based routing intends to send packets toward geographic direction of destination. For this, source nodes need to know about destination position via a location service. Whereas the main concept of ad hoc networking is being independent from an infrastructure, applying a distribution algorithm has been a key design principle for a location service. Destruction of single nodes shouldn't lead to disruption of location service. Moreover, scalability is a desirable characteristic for large networks. So, we offered a distributed and fault tolerant location service method (DFLS) for ad hoc networks in this paper. This method is a hierarchical and explicit Quorum-based location service method. In this method, location servers in each cluster have abstract information about other clusters. So, this method will have response time similar to flat methods. But, it is similar to hierarchical methods in terms of implementation complexity. Evaluation shows that this method (DFLS), compared to the previous methods, decreased the response time to location queries and control overheads, also it increased fault tolerance and packet delivery rate.

Keywords—Location-Based Routing, Location Service, Geographic Forwarding

I. INTRODUCTION

Routing methods in mobile ad hoc networks is generally divided into geographic and topology-based methods. By taking into account the inherent characteristics of mobile ad hoc networks, route maintenance in topology-based routing is a major problem. So, using routing algorithms based on geographic location has been welcomed [1]. This algorithm routes and sends data, by knowing the destination position. That is, every node which intends to send data to another node, should obtain position of destination node. This work is done by using location service methods. In general, geographic routing comprises two fields of study: geographic forwarding methods and location services [2]. A location service in mobile ad hoc networks is divided into two main groups: flooding based method and quorum based method.

A general classification for location service has been presented in fig.1 [3]. In the first group, each node should

maintain location of all the network nodes in a table in its local memory. So, each node should flood its location information in network to being recognized by other nodes. This work imposes high overhead to network [4, 5]. In the second group, all of the network nodes agree explicitly or implicitly on a mapping that allocates a node id to one or more nodes. Mapped nodes are network nodes that receive location information of the other nodes periodically. When a node wants to send a packet, it questions location information of destination node from location servers. In implicit quorum based location services, location servers are selected by using a Hash function. In these methods, mapping is done via id of nodes. Hash function maps the node to a geographic location and the node existing in that location is selected as server [6, 7].

There are some problems in these methods. For example, it is possible that a node may not be in a position obtained by hash function. In explicit quorum based servers, a set of nodes are considered explicitly as location servers that receive update packets of location data from other nodes and as receive location query packet from other nodes, submit location information to them. Explicit quorum based services are divided into two main groups: flat and hierarchical. In the methods of the first group, all of the location servers are symmetric in their rules. That is, they don't have hierarchical structure. Location information is maintained in data base which is the backbone of the network [8, 9, 10]. Distribution and scalability are weak in these methods. But, discovery time is fast. In the second group, i.e. hierarchical explicit quorum-based methods, location information is maintained hierarchically in server nodes. The closer are location server to nodes, the more information have about nodes. In this method, the search is done hierarchically in which as we approach to destination node, we can obtain more exact information about location [11, 12, 13]. Because overhead of flooding based location service methods is high and in implicit quorum based methods using hash function, it is possible that any node in mapping place of hash function doesn't exist, and because flat explicit quorum based methods doesn't distribute well and doesn't have desirable scalability. In this paper, we offer a hierarchical explicit quorum based method that doesn't have

this weakness. In this method, location servers in every cluster have precise information about other clusters. For this reason, this method will have response time similar to flat methods. But in terms of implementation complexity, it will be similar to hierarchical methods. This method, compared to previous methods, decreased the response time to location query and increased fault tolerance. In this paper, at first we will review the related works, then will present proposed method and at the end, we will evaluate method by simulation and will compare with similar methods.

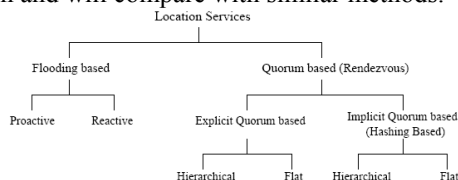


Figure 1. Fig1: classification of location service methods[3]

The rest of this paper is organized as follows. The related works is presented in Section II. Section III describes the technical details of the proposed method. The performance evaluation using simulations is presented in Section IV. Finally, Section V concludes this paper.

II. THE RELATED WORKS

As explained in introduction section, because overhead of flooding based location service is high and in implicit quorum based methods using fixed hash function, it is possible that any node doesn't exist in mapping place of hash function,[21,22](GHLS, GLS) and because flat implicit quorum based methods doesn't distribute well and doesn't have desirable scalability, the hierarchical explicit quorum based method has been studied in this paper. In this section, we study some samples of explicit quorum based location service.

A. ALS method

This method assumes that the nodes are in motion and are exchanging information and data in a virtual circle shaped environment. Location server sets are located in a circle that has identical centre with whole network circle but its diameter is smaller. Each server maintains in itself location information of one or several nodes of sector.

Updating of sector nodes begins synchronically by sending periodic packets from sector edge node, and as moves forward in a route to the centre of circle, gathers location information of other nodes, so that location information is stored in location servers. This method is quorum based and works explicitly. Destination location discovery method is done by sending a packet to the centre of network because there are a series of pointer servers, so, they can direct location query packet toward suitable location server [19].

B. Octopus method

Octopus is a method that its most important characteristic is alternation disruption tolerance of nodes in network. In

fact, it is a fault tolerant method that is very useful in ad hoc networks. Fault tolerance characteristic is obtained through a kind of redundancy, i.e. storing the location of each node in a lot of nodes of network as well as repeatedly maintenance of network status. Also, by using a technique called Synchronized aggregation, in a way that, a packet carries and updates the location of several nodes in network. The network comprises a set of wireless nodes that move in a rectangle shaped space. The set of nodes can be disrupted and connected during a time. Location information forwarding of all the nodes on a strip is done via periodic broadcasting of packets which begins by edge node. Edge node sends the packets as broadcasting in east and west directions. This packet contains location information of all the nodes on the strip.

Each node by arrival of location update packet updates its database and adds its location information and location information of its neighbour. Then finds a new goal and rebroadcasts packet. This work continues until the packet reaches edge node of other side of strip. For finding the destination, each node looks in to its local data base. If the node isn't found in that place, the node broadcasts two query packets northward and southward. This broadcasting continues until this packet reaches to a node that knows destination location information. When the packet reaches to a node that knows destination location, this node send a reply packet containing wanted destination location to query sender [9].

Methods like octopus used a lot of storage overhead, and the methods similar to ALS send their location update and location query packets toward the centre of network. Because main location servers haven't been distributed in network, they exist in network centre. So, network traffic increased in this area and it may lead to loss of some packets. The proposed methods solve these problems, and location servers are distributed in network environment.

III.3 THE PROPOSED METHOD

Whereas geographic methods have high extensibility and are show high performance in large networks, we present a distributed location service method that is highly fault tolerant and has less overhead compared to similar algorithms. This method is very successful in answering the location queries. It is assumed that nodes are moving and exchanging data packets in a virtual rectangle-shaped environment. This rectangle-shaped environment, like octopus method, is divided in to

horizontal and vertical strips. The width of each strip is a constant value a . By this partitioning, when horizontal and vertical strips reach each other, they create small squares that their sides are equal to a . The diameter of square should be r , which is transmission range of nodes, so that all the nodes in square is placed in each other transmission range. So, a should be assumed equal to $r/\sqrt{2}$, because the diameter of square is $a\sqrt{2}$. Each node can distinguish that in which horizontal and vertical is located by knowing its position i.e. (x,y).

There is also another classification in this method. Some of vertical nodes are assumed as a cluster. So, network environment where nodes are located is divided into several equal clusters, as shown in fig.2. In each cluster, a vertical strip

is appointed for location servers of that cluster. So, in this method, unlike octopus method, in which location information was located in all of the nodes, location information is stored in particular nodes. As a result, this method uses less storage overhead than octopus.

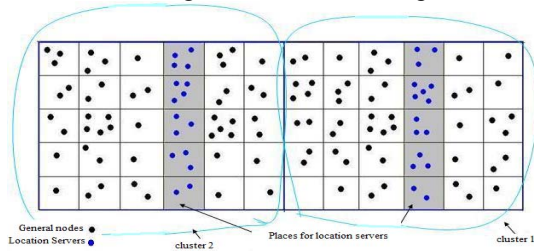


Figure 2. classification of network environment and method of placement of location and servers

The location servers distribute in network environment, so that in a particular spot of network, the traffic resulting from location discovery packet forwarding and location updating packet forwarding doesn't increase.

The proposed method has three main stages: Location updating protocol, Location discovery protocol and Forwarding protocol.

C. Location updating protocol

As mentioned above, and shown in fig.2, network environment is divided into several clusters and each cluster comprises a number of vertical strips. Because the synchronized aggregation is used for location updating packet forwarding, as the location updating packets move forward, it's resulted overhead increases. So, in this method, location servers of each cluster have placed in middle vertical strip. That is, the nodes that are in this vertical strip are location servers of that cluster. So, all the nodes placed in a particular cluster, should sent their location information to these server nodes that are located within middle vertical strip. In nodes' location updating stage, all of the nodes need not to send their location separately. Because nodes which are within a square in horizontal strip are placed in each other transmission range and neighbour nodes send their location information to neighbours via Hello protocol [14]. So, all neighbour nodes know each other's location information. Periodic Hello protocol has been used in papers [16] and [17]. In this protocol, nodes send each other Hello messages in specific time intervals. In dynamic methods like GHLS (18), a threshold distance d is assumed. When a node travels a distance equal to d or more, it sends Hello message to neighbours. A dynamic method is also used in proposed algorithm. When a node exits its residence square, sends Hello message to its neighbours. It leads to decrease of protocol overhead for low-speed nodes. A time out (T_e) is assumed for each record in neighbourhood table to limit extra extension of neighbourhood table in each node. (T_e) can be selected according to the speed of network nodes. In networks that nodes' speed is high, (T_e) should be selected smaller, because nodes information changes with high speed. It needs fast updating of nodes. And in networks that the

nodes' speed is low, (T_e) can be selected higher, because nodes' information changes with low speed. All of the nodes should send in T_e time the Hello message at least once. If any of nodes in this (T_e) time doesn't receive Hello message from its neighbours, it will delete the record of that neighbour from its neighbourhood table. So, in each horizontal strip, the node which is located in right edge or left edge sends location updating packet in which, is placed it's an its neighbours location information, via greedy Geographic forwarding towards server nodes. The next nodes that are placed in forwarding route of this packet get the packet similarly and put its and its neighbour's location information in it and send toward location servers. If so, less overhead is used for updating the location of nodes. In each square within cluster that contains servers, the right-edge or left-edge node should send the ID of nodes that their location information is located in these servers, to the other servers located in the next clusters that have the same horizontal strip. If so, location servers in each horizontal strip will know the location of nodes that are in their cluster as well as the ID of nodes that are located in location server of other clusters in the same horizontal strip. This helps us in location discovery of nodes. It is assumed in this protocol that nodes send their location information to servers, but not to all of them.

A node that is placed in updating packet route puts its location information and the location information of the neighbours in a packet and sends. Any node that is located in updating packet route toward server nodes doesn't put the location information of all the neighbours in updating packet. Because it is possible that some of the nodes may not move or move slowly, it puts the location information of the node that has been updated recently in updating packet. If so, their location has been sent to servers in previous stage, and because they haven't exited their square, it isn't necessary to update their location in the next stage. This leads to decrease of overhead for a large extent. The other issue that should be considered is number of clusters selection. Since in this location updating method, synchronized aggregation is used and the packets enlarge as they move forward in network, the clusters shouldn't be chosen so large. It causes increase of location updating overhead. The location servers of clusters exchange abstract information about nodes' location. If the number of clusters is high, the overhead resulting from the exchange of abstract information will increase. So, to prevent the increase of overhead resulting from location updating of nodes, the size of clusters should be appropriate.

There are two main issues in this stage: The reaction of algorithm when there is no node for location servers in determined place, the motion of nodes.

1) Lack of node for location servers in determined place

According to explained GLS, this issue happens rarely in network. Notwithstanding, in proposed algorithm, in each clusters, supporting location servers are located in the left horizontal strip of servers. When there is no location server in the place of location servers, updating and discovery location packets are directed toward these supporting servers and these

nodes will function as location server until servers are located in the place of location servers.

2) The motion of nodes

Because the nodes are in motion, it is possible that a node enters in or exits servers' place. If there is a node in exit time, it won't cause a problem. But, if the exiting node is the only node in that place, it should deliver its data base information to supporting nodes before exiting and when it wants to enter this place, sends a packet containing other nodes' location information query to this place. If there is node or nodes in this place, it will answer the query. Otherwise, the packet will be directed toward supporting location servers to make aware supporting location servers of its presence and receive nodes' location information.

D. 3.2. Location discovery protocol

It happens when a source node "S" wants to send a packet to a destination node "D". In Geographic protocols, it requires that node S knows the location of node D. This location information is obtained via location discovery protocol. When node S wants to be aware of the location of node D, there will be statuses below:

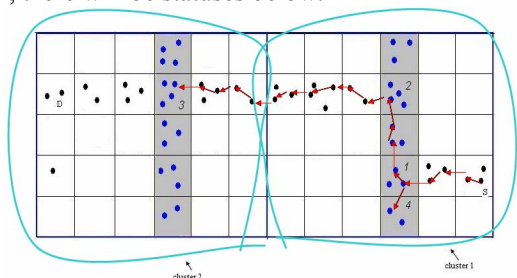


Figure 3. Method of location discovery

1. S and D are located in the same cluster and in the same strip. If so, it is sufficient that location information query is sent by S to location servers which are located in that horizontal strip (fig.3, square1). Because the destination is also in the same strip, S can obtain the location of D from that strip's location servers. It is done because location servers know the location of D and put the destination location in a location query packet and send to S. In this stage, the nodes that are in the route of packet, for decreasing overhead, can store location of D for a few moments. So that if they have a packet to D, they don't create extra overhead for asking location of D. It leads to decrease of location discovery protocol overhead and network traffic.

2. S and D are located in the same horizontal strip but their cluster is different. When server nodes receive location query from S, they don't know exact place of D. But, because the ID of D node has been sent to these servers by the servers in the same strip or in the other cluster, they send the packet to the location server of D. We can obtain the location information of D by this method.

3. S and D are in two different horizontal strips but in the same cluster. S sends location query to nodes located in its horizontal strip. Because servers' nodes don't know the place

of D and its ID, send the packet southward and northward. At this time, the packet is sent toward the location servers of that cluster. At last, it reaches to a location server which has the location of D. It sends location to S via one of the geographic methods.

4. S and D are in two different horizontal strip and cluster. S sends location query to nodes located in its horizontal strip. Because servers' nodes don't know the place of D and its ID, send the packet southward and northward. At this time, the packet is sent toward the location servers of that cluster. At last, it reaches to a location server which has the ID of D. D is in the other cluster but their horizontal strip is the same. So, it sends the packet toward server nodes of that cluster and the location information is obtained. The location is sent to S via one of the geographic methods.

E. forwarding protocol

The greedy algorithm is used for sending data or controlling packets to destination nodes.

IV. EVALUATION

General mobile information systems simulator provides a stable simulation environment for wired and wireless large communication networks. Glomosim simulates networks that have up to 1000 nodes and can be connected to each other non-monotonously. Glomosim can simulate multicasting and broadcasting of information in non symmetrical networks as well as multi-hop wireless communications in ad hoc networks [20]. For comparing and evaluating methods, we study the effect of nodes speed, density of nodes and size of network on GHLS, AODV, and DFLS. In this simulation, nodes transmission diameter is considered 250 m. each Hello packet is 250 bites that nodes every two seconds, send to each other to create one hop neighbourhood table. The time out for the records of neighbourhood table is 6 s. the movement model of nodes is random model and the simulation time is 900 s.

MAC IEEE 802.11 is used. The movement speed of nodes is between 0 and 24m/s and the time out for records in neighbourhood table is 14s. Nodes distribution model in network is monotonous. Important simulation parameters have been listed in table 1.

Table 1. simulation parameters

Simulation Time	900 s
Transmission Range	250 m
MAC Protocol	IEEE 802.11
Mobility Model	Random way-point
Maximum Speed	24m/s
Bandwidth	2mbps

F. The effect of nodes mobility

For studying the effect of nodes mobility in GHLS, AOVD and DFLS, we used 6 maximum speeds (4, 8, 12, 16, 20 and 24 m/s) and the number of nodes in this scenario is 265, that are classified to 2 clusters in DFLS method and 4 level grid in GHLS method in a 2000×2000 m2 environment.

Fig. 4(a) shows the effect of speed on these methods. When speed increases, packet delivery rate decreases for all three methods. This method is better than the other methods, because when speed increases, the routes in AODV are break and it leads to destruction of packets because of RREQ flooding. In GHLS when nodes' speed increases, the rate of location update packet forwarding increases toward servers and it increases the collision. In the proposed method, because the synchronized aggregation has been used, the overhead of updating packet isn't high. Also, speed causes the location information become out of date earlier. It affects the proposed method and decreases the rate of packets' delivery success.

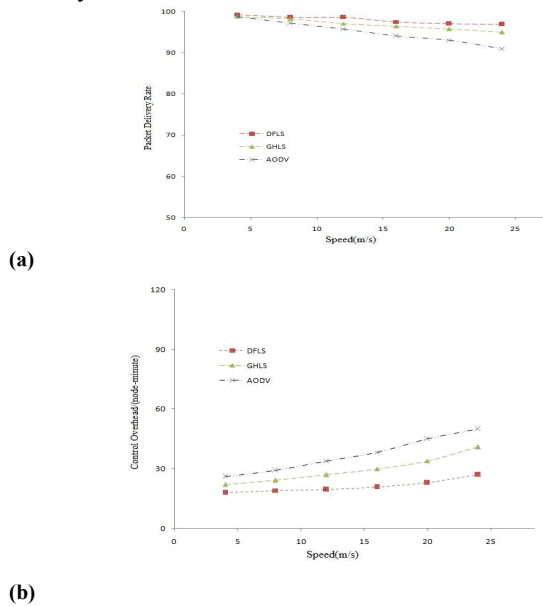


Figure 4. the effect of nodes' mobility (a. packet delivery rate, b. control overhead/node-minute)

Fig. 4(b) shows the effect of speed on these methods. When speed increases, overall controlling overhead increases too. As explained in previous part, when speed increases the routes in AODV are break. It leads to the increase of RREQ forwarding and as a result, RREP and RERR packets increase too. In GHLS method, when the speed increases, the rate of location update packet forwarding increases. But, it happens slowly in proposed method.

G. Effect of nodes' density

For studying the effect of nodes' density in GHLS, AODV and DFSL methods, we survey 5 different numbers of nodes (133, 178, 267, 356, 534) in a 2000×2000 m² environment. The nodes have been distributed with 1/30000, 1/22500, 1/15000, 1/11250 and 1/7500 densities. The maximum speed of nodes is 15 m/s.

Fig. 5(a) shows the effect of density on packet delivery rate in three methods. In AODV method, when the density of nodes is low, the routes break. For this reason, packet delivery rate is less and if the density is high, network traffic will increase and the packet will be deleted due to the traffic caused by flooding of RREQ. GHLS method has high

overhead in high density and it leads to network traffic, packet drop and packet delivery rate decrease. But, in proposed method, because synchronized aggregation is used, added overhead exist in Hello messages sending and this is less than GHLS method. So, when network density increases, packet delivery rate decreases with less rate and low packet delivery rate in low density is due to lack of node in location servers' place that leads to break of location query packets.

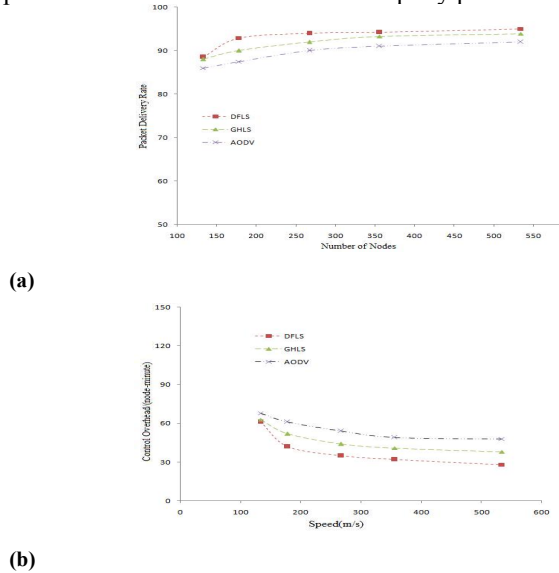


Figure 5. The effect of nodes' density (a. packet delivery rate, b. control overhead/node-minute)

Fig. 5(b) shows the effect of density on controlling overhead in three methods. In AODV method, when the density of nodes is low, the routes break and as a result, by flooding RREQ nodes, overhead increases. But, when network density is high, route break rate decreases, and as a result, controlling overhead decreases. In proposed method and in GHLS, when network density increases, location queries are done successfully. So, there is no extra overhead for resending. It leads to decrease of controlling overhead for each node in each minute. Because synchronized aggregation has been used in proposed method, controlling overhead is less than the other methods.

H. Effect of network size

For studying the effect of network size in GHLS, AODV and DFSL, we survey 6 different network sizes that have been listed in the table below. The density of nodes is constant and is equal to 1/15000 node/ m² and the maximum speed is 15 m/s.

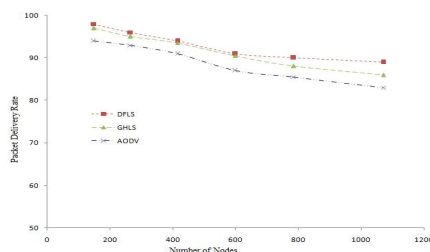
When network size increases, the average of hops a packet should cover to reach the destination increases. For AODV, when the number of hops in the route increases, route break possibility increases. So, deletion of nodes and controlling overhead increases, as shown in 7(a) and (b).

As shown in fig. 6(b), controlling overhead increases for both methods. It is due to increase of controlling packets. The proposed method has less overhead because it used synchronized aggregation. As shown in fig. 6(a), when the number of hops in packet route increases, end-to-end delay increases too. It leads to traffic and as a result, deletion of

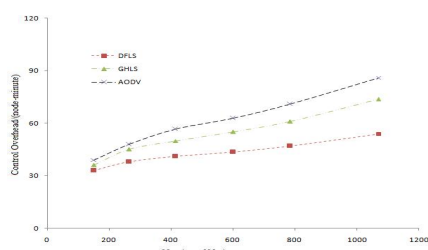
packets and packet delivery rate decreases. In this proposed method, it decreases with fewer rates, because synchronized aggregation has been used.

Table 2 the different sizes of network and the number of nodes

Network size	Number of nodes
1500*1500	150
2000*2000	265
2500*2500	415
3000*3000	600
3500*3500	785
4000*4000	1070



(a)



(b)

Figure 6. Effect of network size(a. packet delivery rate, b. control overhead/node-minute)

V. CONCLUSIONS

We offered a new location service method. We compared it with other methods in previous part and showed its better performance. In this method, because location servers have been distributed, the overhead in the place of location information servers decreases to a large extent, in comparison to other methods. And because the overall nodes in a square is considered location server, it makes the method resistant against disruptions. It can be said that this method is fault tolerant too. Because location update packet isn't sent by all of the nodes in this method, and only right-end and left-end nodes existing on horizontal strip and cluster send the packet, the created overhead will be less. Because the servers located in each square maintains only location information of nodes located in its horizontal strip, storage overhead in server nodes will decrease to a large extent. Evaluation show that this method (DFLS), compared to the previous methods, decreased the response time to location queries and control overheads, also it increased fault tolerance and packet delivery rate.

VI. REFERENCES

- [1] S. Sesay, Z. Yang and J.He, "A Survey on Mobile Ad hoc Wireless Networks", Information Technology Journal Vol.3 No.2, 2004.
- [2] I. Abraham, D.Dolev, and D.Malkhi, "LLS: A Locality Aware Location Service for Mobile Ad hoc Networks", In proc. of the Joint workshop on Foundations of Mobile computing, ACM Press, 2004.
- [3] R. Fraidman, G. Kloit, "Location Service in Wireless Ad hoc and Hybrid Networks: A Survey", Technion – Computer Science Department – Technical Report CS-2006- 10 – 2006.
- [4] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks", In Proc. of the IEEE International Conference on Communications, pages 3318–3324, 2002.
- [5] R. Friedman and G. Korland, "Timed Grid Routing (TIGR) Bites off Energy", In Proc. of 6th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc), May 2005.
- [6] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks", In Proc. of the IEEE International Conference on Communications, pages 3318–3324, 2002.
- [7] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-Hoc Networks", In Proc. of the 26th Annual IEEE Conference on Local Computer Networks (LCN), Washington, DC, USA, page 102, 2001.
- [8] S. Bhattacharya. Randomized location service in mobile ad hoc networks. In Proc. of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM) ACM Press, 2003.
- [9] R. Melamed, I. Keidar, and Y. Barel. Octopus: A Fault-Tolerant and Efficient Ad hoc Routing Protocol. In Proc. of the 24th IEEE Symposium on Reliable Distributed Systems (SRDS), 2005.
- [10] S. Dolev, S. Gilbert, N. Lynch, A. Shvartsman, and J. Welch. Geoquorums: Implementing atomic memory in mobile ad hoc networks. In Proc. of the 17th International Symposium on Distributed Computing (DISC), 2003.
- [11] I. Abraham, D. Dolev, and D. Malkhi. LLS: a Locality Aware Location Service for Mobile Ad Hoc Networks. In Proc. of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), ACM Press, 2004.
- [12] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage in SensorNets. In Proc. of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, Georgia, September 2002.
- [13] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In Proc. of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom), August 2000.
- [14] V. Ch. R. Giruka and Dr. M. Singhal, Adaptive and Efficient Position-based Protocols For Network-Layer Services in Mobile Ad-hoc Networks, PH.D. Dissertation, Department of Computer Science University of Kentucky, 2007.
- [15] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger. and R. Morris. "A scalable location service for geographic ad hoc routing," in Proc. Of ACM MobiCom, August 2000.
- [16] S. Basagni, I. Chlamtac, V. Syrotuk, and B. Woodward. A distance effect routing algorithm for mobility (DREAM). In proceedings of the 4th annual ACM/IEEE Int. Conf. on Mobile Computing and networking (MOBICOM), 1998.
- [17] Karp, B., and Kung. H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proc. 6th Annual International Conference on Mobile Computing and Networking (MOBICOM 2000).
- [18] S.M. Das, H. Pucha, and Y. C. Hu. Performance comparison of scalable location services for geographic ad hoc routing, In INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.
- [19] K. Khazaei, A Location Service Schema for routing in Ad hoc Networks, MSC Thesis, Department of Information technology Azad University of Qazvin, 2007
- [20] X. Zeng, R. Bagrodia, and M. Gerla, GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks, Proc. the 12th Workshop on Parallel and Distributed Simulations, 1998.